Reino Virrankoski

Open Source Platform Development in Wireless Automation under IEEE 802.15.4 Standard

ACTA WASAENSIA 428

Vaasan yliopisto
UNIVERSITY OF VAASA

ACADEMIC DISSERTATION

To be presented, with the permission of the Board of the School of Technology and Innovations of the University of Vaasa, for public examination in Auditorium Wolff (B201) on the 25th of September, 2019, at noon.

Reviewers

Professor Jari Iinatti
University of Oulu
Department of Communications Engineering
P.O.Box 4500
FI-90014 University of Oulu
Finland

Associate Professor Edith Ngai
Uppsala University
Department of Information Technology
Uppsala University Box 337
SE-751 05 Uppsala
Sweden

Julkaisija	**Julkaisupäivämäärä**	
Vaasan yliopisto	Syyskuu 2019	
Tekijä(t) Reino Virrankoski	**Julkaisun tyyppi** Artikkeliväitöskirja	
	Julkaisusarjan nimi, osan numero Acta Wasaensia, 428	
Yhteystiedot Vaasan yliopisto Tekniikan ja innovaatiojohtamisen yksikkö PL 700 FI-65101 VAASA	**ISBN** 978-952-476-876-4 (painettu) 978-952-476-877-1 (verkkoaineisto)	
	URN:ISBN:978-952-476-877-1	
	ISSN 0355-2667 (Acta Wasaensia 428, painettu) 2323-9123 (Acta Wasaensia 428, verkkoaineisto)	
	Sivumäärä 261	**Kieli** englanti
Julkaisun nimike Avoin alustakehitys IEEE 802.15.4 -standardin mukaisessa langattomassa automaatiossa		

Tiivistelmä

Tämä väitöskirja käsittelee avointa alustakehitystä IEEE 802.15.4 -standardin mukaisessa langattomassa automaatiossa. Tutkimusmenetelmä on empiirinen.

Työssä sovelletaan alustaperustaista suunnittelutapaa, joka tähtää yleiskäyttöisen avoimen anturialustan kehittämiseen. Suunnittelun tavoitteita tarkennettiin haastattelemalla alan asiantuntijoita teollisuudesta ja yliopistomaailmasta. Tuloksena suunniteltiin ja toteutettiin anturialusta, the UWASA Node.

Implementointituloksista voidaan vetää johtopäätös, että anturialustan tavoiteltu yleiskäyttöisyystaso saavutettiin. Toisaalta saavutettu yleiskäyttöisyystaso lisäsi alustan ohjelmistoarkkitehtuurin monimutkaisuutta.

Kaupallisten IEEE 802.15.4 -standardia tukevien anturialustojen tulo markkinoille vähentää avointen anturialustojen käyttöä, mutta ne eivät ole katoamassa. Kaupalliset ohjelmistot ovat tyypillisesti suljettuja ja sidoksissa tiettyyn alustaan, mikä tekee niistä sopimattomia tutkimus- ja tuotekehityskäyttöön. Vaikka nykyään on saatavilla useita kaupallisia langattomia anturi- ja toimilaiteverkkoja, vaaditaan vielä paljon työtä ennen kun kaikki esineiden Internetiin (Internet of Things) liittyvät visiot voidaan toteuttaa. Tämä koskee erityisesti langattomassa anturi- ja toimilaiteverkossa hajautetusti tai paikallisesti toteutettavia toimintoja. Säätötekniikan näkökulmasta keskeinen kysymys on, miten tunnettuja säätömenetelmiä tulee soveltaa langattomassa automaatiossa, jossa langaton anturi- ja toimilaiteverkko on osa automaatiojärjestelmää. Avoimet anturialustat ovat tärkeä työkalu sen selvittämisessä.

Asiasanat

Langattomat anturi- ja toimilaiteverkot, langaton automaatio, esineiden Internet

Publisher	**Date of publication**	
Vaasan yliopisto	September 2019	
Author(s) Reino Virrankoski	**Type of publication** Doctoral thesis by publication	
	Name and number of series Acta Wasaensia, 428	
Contact information University of Vaasa School of Technology and Innovations P.O. Box 700 FI-65101 Vaasa Finland	**ISBN** 978-952-476-876-4 (print) 978-952-476-877-1 (online)	
	URN:ISBN978-952-476-877-1	
	ISSN 0355-2667 (Acta Wasaensia 428, print) 2323-9123 (Acta Wasaensia 428, online)	
	Number of pages 261	**Language** English

Title of publication
Open Source Platform Development in Wireless Automation under IEEE 802.15.4 Standard

Abstract
This doctoral dissertation focuses on open source platform development in wireless automation under IEEE 802.15.4 standard. Research method is empirical.
A platform based approach, which targets to the design of a generic open source sensor platform, was selected as a design method. The design targets were further focused by interviewing the experts from the academia and industry. Generic and modular sensor platform, the UWASA Node, was developed as an outcome of this process.
Based on the implementation results, a wireless sensor and actuator network based on the UWASA Node was a feasible solution for many types of wireless automation applications. It was also possible to interface it with the other parts of the system. The targeted level of sensor platform genericity was achieved. However, it was also observed that the achieved level of genericity increased the software complexity. The development of commercial sensor platforms, which support IEEE 802.15.4 sensor networking, has narrowed down the role of open source sensor platforms, but they are not disappearing. Commercial software is usually closed and connected to a specified platform, which makes it unsuitable for research and development work. Even though there exits many commercial WSN solutions and the market expectations in this area are high, there is still a lot of work to do before the visions about Internet of Things (IoT) are fulfilled, especially in the context of distributed and locally centralized operations in the network. In terms of control engineering, one of the main research issues is to figure out how the well-known control techniques may be applied in wireless automation where WSN is part of the automation system. Open source platforms offer an important tool in this research and development work.

Keywords
Wireless Sensor Networks, Wireless Automation, Internet of Things

ACKNOWLEDGEMENT

This doctoral dissertation wraps up my work with wireless sensor and actuator networks over a time span of more than ten years from the emerging level of these technologies to the current state of the art.

I express my deepest thanks to my supervisors, Professor Erkki Antila and Professor Heikki Koivo. Thanks to their expertise, advising and patience, this work is now completed.

I also want to thank Professor Mohammed Elmusrati and Professor Timo Mantere from the School of Technology and Innovation of the University of Vaasa and Professor Riku Jäntti from the Department of Communications and Networking of Aalto University for a good cooperation with many valuable discussions, advices and joint research activities.

I thank Professor Mani Srivastava for hosting my staying as a Visiting Researcher in the Center for Embedded Networked Sensing in UCLA in 2003, Professor Andreas Savvides for hosting my staying as a Visiting Assistant Researcher in Yale University in 2004-2005, Professor Dhadesugoor Vaman for a good long-term cooperation with the Center of Excellence in Battlefield Communications in Prairie View A&M University and Professor Lijun Qian for a good cooperation with the Center of Excellence in Research and Education in Big Military Data Intelligence in Prairie View A&M University.

I thank all students and workmates with who I have had an opportunity to work with, related to the topics discussed in this dissertation, in the Control Engineering Laboratory of the Helsinki University of Technology, Department of Computer Science in the University of Vaasa and Department of Communications and Networking in Aalto University.

I express my sincere thanks to the following foundations that have supported me with grants during this work: Nokia Foundation, Eemil Aaltonen Foundation, Finnish Society of Automation, Neles Oy 30 Years Foundation, KAUTE Foundation, the Foundation for Economic Education, and Southern Ostrobothnian Student Nation of the University of Helsinki (EPO).

My warmest thanks belong to my family. My wife, children and my parents have shown supernatural patience and endless support during this work.

Contents

Figures

SYMBOLS

α	A pre-defined speech signal filtering parameter
$\alpha(t)$	Oscillator phase deviation
$\alpha^i(t)$	A phase deviation of clock i
γ_{ij}	Best estimate for distance between nodes i and j
δ_f	Clock skew
δ_f^i	A clock skew of clock i
ΔC_s	First order temporal derivatives of C_s
$\Delta\Delta C_s$	Second order temporal derivatives of C_s
ΔC_{cep}	First order derivatives of the mel-cepstral coefficients
$\Delta\Delta C_{cep}$	Second order derivatives of the mel-cepstral coefficients
ε_{ij}	Independent zero-mean random variable with bounded variance
ε_j	Additive white Gaussian jitter
$\epsilon(t)$	Random deviations in the oscillator output model
μ_{id}	The average of the distances between the feature matrix of the measured signal and the feature matrices of the voice samples in the database
$\hat{o}(N)$	Time offset estimate based on N samples
σ_ε	The variance of the distance measurement noise
σ_{id}	The standard deviation of the distances between the feature matrix of the measured signal and the feature matrices of the voice samples in the database
$\sigma_{P_{Rx}}^i$	Standard deviation of the received signal strength in the measurement point i

$\sigma^*_{P_{Rx}}$	Computed optimal threshold value for the standard deviation of the received signal strength
σ_S	The variance of the elements of S
$\varphi(t)$	Time error part in the oscillator output model
φ_0	Time offset in the oscillator output model
φ_0^i	A time offset of clock i
a	A clock skew ratio, which defines how the periods of the oscillators are related to each other
$\hat{a}(N)$	The optimum value of a, which minimizes $L(a)$
$\hat{a}(N+1)$	A recursive estimate of a for $N+1$ measurements
A	A loss constant in the power loss model by Chipcon
A^*	Computed optimal value of the loss constant A
$\hat{b}(N+1)$	A recursive estimate of a for $N+1$ measurements by using numerically more stable modification of the estimator $\hat{a}(N+1)$
B_f	A filterbank matrix of triangular filters to enhance the frequencies which are located in the area of human speech
c	Empirical oscillator constant
C	A centered mel-cepstral matrix
C_{cep}	Mel-cepstral coefficients computed by taking the row-wise average of C_{sp}
$C^i(t)$	A time report of clock i at time moment t
$\hat{C}_j^i$	Estimated time report of clock i at time moment j
$\bar{C}_N$	Average vector, where each element is the average of the respective column in C_s
C_p	A matrix of mel-cepstral coefficients
C_s	Smoothened mel-cepstral matrix $C_s = \bar{M}C$

C_{sp}	A matrix which contains such columns of C_s, which stand for the speech portions of the signal
C_x	Covariance matrix of the measured acoustic data computed from the centered observation matrix $\tilde{X}(k)$
d	A distance between transmitter and receiver
d_0	A reference distance in the path loss model
d_{ij}	Distance between nodes i and j
$\hat{d}_l$	Estimate of the distance between transmitter and receiver
d_{max}	Maximum distance between two nodes in the network
D	Distance matrix
$\widetilde{D}$	Matrix of noisy squared distances
D_c	Centered square distance matrix
D_{C2}	Best rank 2 approximation of D_c
D_e	Diagonal matrix having the eigenvalues of C_x in its diagonal in a decreasing order
D_f	Frequency drift
D_r	Density reachability parameter
D_s	Matrix of squared distances
$\widetilde{D}_{sij}$	Noisy squared distance between nodes i and j in D_s
$\bar{e}$	A time error vector with components which are jointly Gaussian
$\hat{e}$	Noise vector in reference-triggered increment time model
e_j	Oscillator time error in the jth transition
f_{CF}	Cost function for the difference between actual and estimated distances
f_s	Sampling frequency

f_0	Nominal clock frequency in the oscillator output model
F_s	Feature matrix in the speaker identification
G	Diagonal matrix in SVD
G_2	A matrix carrying the first two eigenvalues of G
$\hat{h}_j$	A ratio of count increments between successive recording instants at time moment j
K_{max}	An empirical parameter which is used to check the data consistency in a broadcast based time synchronization to reduce the error caused by the time-varying transmission delays
l_{ij}	Length of the Euclidean path between nodes i and j
L	The length of the sample
$L(a)$	A -log likelihood function of the clock skew ratio a
L_w	A Hamming window for signal windowing
m	A pre-defined parameter to fit the value of the threshold T_{th}
M	Mixing matrix
$\bar{M}$	A smoothening vector, which is used to smooth C
n	Number of nodes in WSN
n_p	Path loss exponent in the path loss model
n_p^*	Computed optimal value of the path loss exponent
N_{bins}	Number of bins used in the Discrete Fourier Transform
N_{cep}	Number of cepstral coefficients considered in the discrete cosine transform
N_{lost}^*	Computed optimal threshold value for the number of lost packets
N_{lost}^i	Number of lost packets in the measurement point i
N_{min}	Required minimum number of nodes in a cluster

N_w	Number of elements in the Hamming window
p_ε	Small (infinitesimal) positive constant
p_{ij}	Probability that distance d_{ij} can be measured
p_1	A probability that a node can detect its neighboring node within its communication range
p_2	A probability that a node fails to detect its neighboring node within its communication range
P_{dB}	A power spectrum matrix which is computed by converting P_s to decibels
P_{loss}	Log-distance path loss
P_{Rx}	Received signal strength
P_s	A power spectrum matrix which is computed from P_w by multiplying it with the filterbank matrix $P_s = P_w B_f$
P_{Tx}	Transmitted signal strength
P_w	A power spectrum matrix of the acoustic signal
P_0	Received power at the reference distance d_0
r_i	Density range: a radius of the smallest node i centered disk that covers $D_r - 1$ other nodes in the vicinity of node i
R	Sensor node communication radius
s	A matrix of source signals
s_i	Source signal i
S	An estimator matrix that estimates D_s
S_{ij}	Distance ij in S
S_4	The best rank 4 approximation of S
t	Actual time in the context of time synchronization
t_{ref}	Reference time

t_j^{err}	Time synchronization error with a broadcast j
t_w	The time length of the Hamming window L_w
T	The length of the oscillator period
T_{th}	A pre-defined threshold for the distance between the feature matrices of the measured voice signal and the samples in the database
T_0	A nominal length of the oscillator period
T_0^i	A nominal period of oscillator i
U	Left singular vectors in SVD
U_2	$n \times 2$ matrix of top left singular values of U
V	Right singular vectors in SVD
w_+	Increment of the node weight
W	Whitening matrix of the measured acoustic data
x	A matrix of observed signals
$\bar{x}$	A speech signal vector
$\bar{x}_p$	Filtered speech signal vector
x_i	Observation (measurement) of sensor i
x_j	Deterministic delay
X	Node coordinate matrix
$X(k)$	Observation matrix of k measurements
$\tilde{X}(k)$	Centered observation matrix of k measurements
X_σ	A zero-mean Gaussian random variable with a standard deviation σ, represents flat fading
X_2	Node relative coordinates in two dimensions
$\bar{1}$	A vector of ones

ABBREVIATIONS

AC	Alternating current
BPSK	Binary Phase Shift Keying
BSS	Blind signal separation
COP	Common operational picture
CO2	Carbon dioxide
CPS	Cyber-Physical Systems
CPU	Central Processing Unit
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
DFL	Device-free localization
DFT	Discrete Fourier transform
DLL	Data Link Layer
DSSS	Direct Sequence Spread Spectrum
FFD	Full-function Device
FPGA	Field-programmable gate array
FRAM	Ferroelectric Random Access Memory
ICA	Independent component analysis
IIC	Industrial Internet Consortium
IoT	Internet of Things
IPv6	Internet Protocol version 6
ISA	International Society of Automation
ISM	Industrial, Scientific and Medical

Li-Fi	Light Fidelity
LSE	Least square estimate
MAC	Medium Access Control
MDS	Multidimensional Scaling
ML	Maximum likelihood
MLE	Maximum likelihood estimate
NB-IoT	Narrowband Internet of Things
OS	Operating System
OSI	Open Systems Interconnection
PCA	Principal component analysis
PCB	Printed Circuit Board
QPSK	Quadrature Phase Shift Keying
RF	Radio Frequency
RFD	Reduced Function Device
RIT	Reference-triggered increment time relation model
RPT	Reference-triggered progressive time relation model
RSS	Received signal strength
RSSI	Received Signal Strength Indicator
SLAM	Simultaneous localization and mapping
SVD	Singular value decomposition
TETRA	Trans-European Trunked Radio
UDP	User Datagram Protocol
USB	Universal Serial Bus

VPN	Virtual private networking
2D	Two dimensions
3D	Three dimensions
5G	5th generation cellular mobile communications
6LoWPAN	IPv6 over Low power Wireless Personal Area Networks
WPAN	Wireless Personal Area Network
WSN	Wireless Sensor Networks

LIST OF PUBLICATIONS

Publication 1: Drineas, P., Javed, A., Magdon-Ismail, M., Pandurangan, G., Virrankoski, R. and Savvides, A. (2006). Distance Matrix Reconstruction from Incomplete Distance Information for Sensor Network Localization. *In the Proceedings of the Third Annual IEEE Communications Society Conference on Sensor, Mesh, and Ad Hoc Communications and Networks (SECON'06)*, September 25-28, 2006, Reston, VA, USA.

Publication 2: Ahmed, F., Virrankoski, R. and Elmusrati, M. (2010). Improving RSSI Based Distance Estimation for IEEE 802.15.4 Wireless Sensor Networks. *In the Proceedings of IEEE ICWIT 2010*, Honolulu, Hawaii, August 28th - September 3rd, 2010.

Publication 3: Virrankoski, R. and Savvides, A. (2005). TASC: Topology Adaptive Spatial Clustering for Sensor Networks. *In the proceedings of the 2nd IEEE International Conference on Mobile Ad Hoc and Sensor Systems (MASS'05)*, November 7-10, 2005, Washington DC, USA.

Publication 4: Yigitler, H., Mahmood, A., Virrankoski, R. and Jäntti, R. (2012). Recursive Clock Skew Estimation for Wireless Sensor Networks using Reference Broadcasts. *IET Wireless Sensor Systems, Volume 2, issue 4, December 2012, pp. 338-350.*

Publication 5: Eltahawy, B. and Virrankoski, R. (2016). Unified Information Privacy Preserving Model. *In the proceedings of the International Conference on Communications, Computer Science and Information Technology (ICCCSIT)*, Dubai, United Arab Emirates, 12-14 March, 2016.

Publication 6: Virrankoski, R. and Keskinen, S. (2009). GENSEN: A Novel Combination of Product, Application and Technology Platform Development in the Context of Wireless Automation. *In the Proceedings of 14th International Conference on Productivity & Quality Research (ICPQR 2009)*, October 19-23, Alexandria, Egypt.

Publication 7: Yigitler, H., Virrankoski, R. and Elmusrati, M. S. (2010). Stackable Wirless Sensor and Actuator Network Platform for Wireless Automation: the UWASA Node. *Aalto University Workshop on Wireless Sensor Systems*, November 19th, 2010, Espoo, Finland.

Publication 8: Ahonen, T., Virrankoski, R. and Elmusrati, M. (2008). Greenhouse Monitoring with Wireless Sensor Network. *In the proceedings of 2008 IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications*, October 12-15, 2008, Beijing, China.

Publication 9: Björkbom, M., Timonen, J., Yigitler, H., Kaltiokallio, O., Vallet, J., Myrsky, M., Saarinen, J., Korkalainen, M., Cuhac, C., Koivo, H., Jäntti, R., Virrankoski, R. and Vankka, J. (2013). Localization Services for Online Common Operational Picture and Situation Awareness. *IEEE Access, Vol. 1, no. 1, pp.742-757, 2013*.

Publication 10: Bocca, M., Virrankoski, R. and Koivo, H. N. (2008). Text and Language Independent Speaker Identification by Using Short-Time Low Quality Signals. *Workshop on Wireless Communication and Applications (WoWCA 2008)*, April 2, 2008, Vaasa, Finland.

Publication 11: Bocca, M., Galperti, C., Virrankoski, R. and Koivo, H. N. (2006). Estimating the Number of Persons in an Unknown Indoor Environment by Applying Wireless Acoustic Sensors and Blind Signal Separation. *In the proceedings of the Mobile Computing and Wireless Communications International Conference (MCWC 2006)*, September 17-20, 2006, Amman, Jordan.

Publication 12: Virrankoski, R., Wulayinjiang, M. and Linh L. M. (2016). Frequency Converter Integration to Wireless Sensor Network. *In the proceedings of the International Conference in Industrial Informatics and Computer Systems (CIICS 2016)*, Dubai, United Arab Emirates, 13-15 March, 2016.

Publication 13: Höglund, T., Virrankoski, R. and Mantere, T. (2016). Solar Energy Harvesting Solution for the Wireless Sensor Platform The UWASA Node. *In the proceedings of 5th International Conference on Sensor Networks (SENSORNETS 2016)*, Italy, Rome, February 17-19 , 2016.

AUTHOR'S CONTRIBUTION

In Publication 1: The author participated in the formulation of the presented algorithm for distance matrix reconstruction in the presence of noise, incomplete distance information and sensor node failures. He implemented part of the code that was used in the simulations and contributed to the writing.

In Publication 2: The author participated in the design of the way how the standard deviation of the RSSI and the packet loss are utilized to improve the RSSI based distance estimation. He instructed Ahmed Faheem's work and commented on the article manuscript.

In Publication 3: The author developed the clustering algorithm, wrote code for the simulations, performed the simulations, analyzed the results and wrote the article. The work was done during author's stay as a visiting assistant researcher in Yale University and it was supervised by Prof. Andreas Savvides.

In Publication 4: The author participated in the algorithm development at discussion and brainstorming level. He also commented on the article manuscript.

In Publication 5: The author participated in the privacy preserving model design and instructed Bahaa Eltahawy's work. He also commented on the article manuscript.

In Publication 6: The author developed the way how the platform approach should be applied to sensor platform design jointly with Simo Keskinen. He also listed the mentioned general targets and requirements for sensor networks in industrial automation. The author wrote most of the article. Simo Keskinen provided input information and commented on the manuscript.

In Publication 7: The author gave the starting point guidelines and conditions for the presented sensor platform design. He instructed Huseyin Yigitler's work and participated in the planning, design, experiments and evaluation of the UWASA Node sensor platform. He also commented on the article manuscript.

In Publication 8: The author participated in the planning and building up the experimental sensor network to Marten's Research

Associations Greenhouse. He participated in the paper writing and instructed Teemu Ahonen's work.

In Publication 9: This article wraps up the main results of Wireless Sensor Systems in Indoor Situation Modeling and Wireless Sensor Systems in Indoor Situation Modeling II projects. The Author has been one of the main planners of these research projects, the coordinator of the whole project entity and the principal investigator of the University of Vaasa's parts of these projects. He has participated in the planning of the presented integrated system and test scenarios, and in the execution of the test scenarios. He has also commented on the article manuscript.

In Publication 10: The author participated in the speaker identification algorithm implementation with Matlab with Maurizio Bocca. Then he participated in the definition of the simulation scenarios and in the analysis of the simulation results. He also participated to the paper writing. The work was supervised by Prof. Heikki Koivo.

In Publication 11: The author participated in the data analysis and in the writing of the article manuscript. Maurizio Bocca implemented the code and collected the data. Prof. Heikki Koivo supervised the work.

In Publication 12: The author participated in the definition of the application requirements with the industrial partners. Then he instructed Maiwulan Wulayinjiang's and Le Manh Linh's work, when they built up the application and performed the test scenarios at the test site of the industrial partner. The author wrote the whole article.

In Publication 13: The author participated in the definition of the energy harvesting scenario and instructed Thomas Höglund's work. He also participated in the writing of the article.

1 INTRODUCTION

1.1 Background

The advances in communication systems, computer systems and other electronics have enabled a rapid development of wireless sensor and actuator networks during the latest decade. An increasing amount of computation and processing power can be built into small space with a decreased amount of energy required for these operations.

Originally the highest interest to develop wireless sensor networks (WSN) came from the military side, but once the technology became more mature, also other paths of development started to exist (Pister 2001). In terms of business potential and technology impact, one of the most interesting areas is wireless automation. Small, low-cost wireless devices can provide access to such places, which cannot be connected by cables. These places can be either moving or rotating parts of the machines, or locations in harsh conditions, where cabling is not an option. In power systems, the wireless devices can be utilized to reduce a risk of sparks in explosive atmospheres and to eliminate latent currents induced to the wired connections by the electromagnetic field. Compared to the completely cabled network, the use of low-cost wireless devices allows us to collect more complete and redundant data, which can be utilized in advanced control and monitoring systems. (Shen 2004), (Flammini 2007), (Flammini 2009), (Paavola & Leiviskä 2010)

What it comes to system architecture itself, WSNs enable us to perform computation in the network in a distributed or locally centralized manner. This changes the traditional way of designing automation systems. Many operations can be performed locally without swapping information back and forth between the actuators and the centralized network control. Operations can also be event based so that instead of transmitting continuously measurements which are made by using a constant sample rate, the system can make decisions based on the measured data, and transmit it only when needed. The sample rate can also be increased or decreased based on the measured data so that more information is collected when it is needed, but lower sample rate is applied when the measurements indicate that the targeted system performance is achieved.

This dissertation work focuses on the design and use of open source platforms in wireless automation under IEEE 802.15.4 standard. IEEE 802.15.4 is the most

commonly applied standard in the context of wireless sensor and actuator networks. It uses three license-free industrial, science and medical (ISM) bands: 2400–2483.5 MHz (worldwide), 902–928 MHz (North America) and 868.0–868.6 MHz (Europe). Most commonly used is the first one, since it is applicable worldwide. (IEEE802.15.4, 2018)

1.2 Objectives and Research Method

IEEE 802.15.4 standard is targeted to low-cost and low-speed communication within a relatively short range. Communication devices have typically limited energy resources, which limits their performance in terms of measurement frequency, computation and transmission. Low-power communication in 2.4 GHz band does not penetrate different materials that well either. On the other hand, the standard enables the use of low-cost devices so that single unit limitations can be compensated by the redundancy provided be the number of communication devices, and the communication range can be extended by using so-called multi-hop communication over several radio links from the starting point to the end point. (IEEE802.15.4, 2018)

This kind of architecture presents challenges for the automation system design. Compared to the computer systems plugged to the electric grid and communicating over cabled transmission medium, the computation and data transmission capacity in WSNs is much lower. The communication reliability is also weaker, because some data packets can get lost in wireless communication and some of them can get corrupted during measurement or transmission, and carry then erroneous or misleading data. Typically the communication in WSN can also suffer from time variant delays. These shortages can be compensated by applying distributed computation, energy-efficient algorithms and data fusion and data compression methods. Typically the WSN is not alone, but forms a part of the communication and computation system in automation architecture. As a consequence, interfacing between different types of networks is also important to get the system operate reliably. (Eriksson 2008), (Björkbom 2010), (Koivo & Elmusrati 2010)

Wireless sensor network consists of wireless sensor platforms called sensor nodes or sometimes sensor motes. One sensor node contains at least a microcontroller or a microprocessor, a radio, some memory, a power source and one or several sensors. Different types of devices, which are equipped with a microprocessor and a radio, can also operate as actuators in wireless sensor network. Then the entity of sensor nodes and actuators can be called wireless

sensor and actuator network. There exists a huge amount of industrial sensors, which can be used in wireless sensor nodes. During the early years of the WSN development, many sensor nodes and communication protocols were tailored based on the specific need of a certain application. The problem with these designs was that once the applied technology in the network or in the application system expires, the whole WSN application expires and must be re-designed. Moreover, automation applications rely on standardization, and these sorts of case specific network designs did not support standardization work for wireless automation that well. Everything advised us towards a need of a generic solution: how to develop such a software and hardware platform, that once we know the measurement needs of the particular application, we can select suitable industrial sensors and plug them to the platforms so that the WSN is ready to operate with as minimal software and hardware modifications as possible? A further question is how to interface the WSN with the rest of the automation system in a robust and reliable manner.

A further challenge that follows from the targeted generic solution is the system design. Developed WSN must automatically initialize itself for use, adapt to changes in the network architecture and control its operation. The existence of WSN must be also taken into account in the automation system design. A crucial question is how the control design must be done, when there exist limitations in the data transmission and computation capacity, time-variant communication delays, missing and erroneous measurements in the data, time synchronization errors etc.? On the other hand, also the amount of data can be much bigger than before, and the data redundancy can be utilized.

Selected research method is empirical. First an overview of the existing research field and the existing standards is presented. Then the industrial requirements are mapped by expert interviews, and platform approach is selected for the generic sensor platform design. In the results part, the selected publications first present some algorithms for WSN initialization and control, and also a brief discussion about security is presented. Then the developed sensor platform, the UWSA Node, is presented and its performance is evaluated trough a set of applications. After going over the results, a discussion based on them is presented, some conclusions made about the current state of the art and some directions pointed about the expected development in the nearby future.

1.3 Contributions

1.3.1 Network Initialization and Control

Related to WSN initialization and control, contributions related to localization, clustering and time synchronization are presented in this work.

Publication 1 and Publication 2 discuss about localization by using distance estimation, which is based on radio signal fading indicated by the Received Signal Strength Indicator (RSSI). Publication 1 presents a method to use singular value decomposition (SVD) for distance matrix reconstruction in the case of noisy and incomplete data. Relative sensor node locations can then be computed by applying multi-dimensional scaling (MDS) to the reconstructed distance matrix. Publication 2 presents a method to improve RSSI-based distance estimation by using the standard deviation of the RSSI and the packet loss rate as reliability measures to weight the distance estimates.

Publication 3 presents a distributed clustering algorithm that partitions WSN into a set of isotropic non-overlapping clusters and selects one cluster head for each cluster. The number of clusters depends on the network topology.

Publication 4 presents a recursive clock skew estimation method for WSN in the case the time synchronization is done by using reference broadcasts.

1.3.2 Security

Publication 5 presents a discussion about privacy issues in data networks. Based on this discussion, a unified privacy preserving model is presented. Then a set of recommendations for the network architecture is given based on the privacy preserving model.

1.3.3 Platform

There are two contributions related to the sensor platform. Publication 6 presents a way how the platform approach should be applied to the planning and design of wireless sensor networks for wireless automation. The main target is to bring the technical genericity and performance of the application platform up to such a level that it enables a fast production of different kinds of applications. This can be further utilized to make rapidly different kinds of products from the applications through productization. A method called combined platform

development is suggested to utilize the requirements of the different applications to find the highest possible genericity level for the application platform.

Based on the analyzed requirements for wireless sensor and actuator networks for wireless automation under IEEE 802.15.4 standard, a wireless sensor platform, the UWASA Node, is made. First description of its software and hardware architecture is presented in Publication 7. Platform design follows the principles presented in this dissertation and its performance is evaluated through different applications.

1.3.4 Applications

Publication 8 presents a WSN for greenhouse monitoring. The size of the modern greenhouses can be several hectares, and in the Nordic climate they require heating and artificial lighting for remarkable part of the growing season. Extra carbon dioxide (CO2) is also used in the greenhouses to improve the growth. As a consequence, there is a need to monitor the different layers of the microclimate inside the greenhouse for accurate climate control. For this purpose, a WSN to measure temperature, humidity, light intensity and CO2 content was developed. The network consisted of Sensinode Microseries sensor nodes (Sensinode 2007) equipped with the pre-mentioned sensors. An experimental setup to evaluate the network performance was done at Martens Research Association's greenhouse in Närpiö, Finland.

Indoor situational awareness focuses on building interior monitoring. In police, rescue and military operations it is important to know where the people are inside the building, how many are there and what are they doing. In the pre-surveillance related to police and military operations, it is also important to perform the monitoring in an unnoticeable way as part of the preparation before sending your own troops to the building. Publication 9 wraps up the main results of two indoor situational awareness projects and presents the developed integrated system, that was used to compute and share the real-time common operational picture (COP). The UWASA Node was used as part of the system, especially in device-free people detection and tracking.

Publication 10 presents a text and language independent speaker identification method, which is based on cepstral analysis. Speech features are characterized by the cepstral coefficients and their first and second order derivatives. Then the feature matrix, which is computed from the measured acoustic (speech) signal, and the feature matrices, which are computed from the known voice samples in the database, are compared to each other and the speaker identification is done

based on their similarity. Since the wireless sensor nodes have just limited resources, method suitability for WSN is evaluated by using noisy low quality signals and by varying the sampling frequency, the length of the sample, the number of bins used in the discrete Fourier transform (DFT) and the number of cepstral coefficients used in the computation.

In Publication 11, blind signal separation (BSS) is applied to the acoustic data collected by WSN to estimate the number of persons who are talking. Independent component analysis (ICA) is used for the blind signal separation and the voice samples are collected by using Mica2 sensor nodes (Mica2, 2003).

Publication 12 presents a joint use of WSN and frequency converters. In the described application, the UWASA Node is interfaced to communicate with frequency converter so that one can transmit data between the sensor nodes and actuators and the frequency converter over the WSN. Developed system performance is then evaluated in the experimental setup at the industrial partner's test site. There a machine equipped with six frequency converters is operating, and data is collected from the frequency converters during the operation. In addition to data collecting capability, communication reliability is tested and evaluated.

Publication 13 presents a solar energy harvesting prototype and its evaluation for the UWASA Node. First the energy harvesting prototype design is described and then its performance with the UWASA Node is evaluated through experiments. In the experimental analysis, particular attention is paid for the performance level the node can reach with the energy harvested by the presented solution.

1.4 Structure of the Thesis

The rest of this dissertation is organized as follows: Chapter 2 presents a general introductory discussion about WSNs and wireless automation. Existing wireless automation standards for WSN, WirelessHART and ISA 100.11a are discussed and compared to each other in Chapter 3. Then Chapter 4 presents the evolution of the existing open source systems. The results of the attached publications are summarized in Chapter 5. Discussion about published results and about the current state of the art is presented in Chapter 6. Some directions for the future development are also pointed out in the same chapter. Finally, Chapter 7 concludes the dissertation.

2 WIRELESS SENSOR NETWORKS AND WIRELESS AUTOMATION

2.1 The Concept of Wireless Sensor Networks

Wireless sensor systems have developed rapidly since the beginning of 2000's. The early ideas were focusing mainly on wireless ad hoc networks for monitoring and communication in military systems. These networks could be deployed rapidly by using miniaturized wireless devices called sensor nodes. In many early scenarios, the size of the wireless devices was supposed to be so small that they could be deployed like dust and operate in an unnoticeable manner. (Pister 2001)

Wireless sensor nodes are platforms, which are equipped at least with a microcontroller or processor, memory, a radio, one or several sensors and power source. There are two conflicting main interests in the sensor node development: The node size, price and power consumption should be minimized, but at the same time the node must be as efficient as possible in terms of sample rate, data transmission capability and computation power. WSN can operate without fixed base stations or fixed number of nodes, and the nodes can enter or leave the network. The nodes can communicate with each other either directly or by using multi-hop path, which consists of several radio links between the nodes. To enable this performance, the networking protocols must operate in a distributed manner. This operation can be either fully distributed or locally centralized, if the network contains some nodes which act as cluster heads and have more resources. Distributed networking enables distributed computation so that remarkable amount of data can be processed in the network and only the requested information will be submitted through the gateway from the WSN to the upper levels of the communication system. (Dargie & Poellabauer 2010), (Sohraby 2007), (Tynan 2005)

Once the WSN technology has developed from its early levels, it has also diverged. Some developers have set their main focus on the minimization of node size and energy consumption. These nodes are often used for such applications, where the main purpose of the WSN is to collect measurements which are then analyzed in a centralized manner outside the wireless network. Some developers emphasize also the idea of distributed network operation, which requires distributed algorithms and more efficient nodes. As a consequence, node size and energy consumption are compromises between the minimization and performance requirements. This is typical for sensor nodes, which are developed

for wireless automation. However, since the microprocessors, power sources and other electronic components are still developing rapidly, it is probable to achieve a higher performance with a smaller device size and lower energy consumption in the nearby future.

2.2 Sensor Networking under IEEE 802.15.4 Standard

2.2.1 Physical Layer Specifications

IEEE 802.15.4 is a standard for low-power and low data rate communication within short distance. The standard enables the networking of very low-cost devices without any underlying infrastructure. Originally it was targeted for wireless personal area networks (WPAN), but it is also used in industrial automation to cover similar areas as local area networks. In the standard development, a kind of basic IEEE 802.15.4 network was assumed to have a communication range of 10 meters and a data transfer rate of 250 kbit/s. Communication range can be extended by increasing the transmission power, which leads to higher energy consumption. Respectively, power consumption can be reduced by applying lower transmission power, which decreases the communication range. (IEEE802.15.4, 2018)

In terms of Open Systems Interconnection (OSI) model, the standard defines only physical layer and medium access control (MAC) layer, which is the lower part of the data link layer in the OSI model. In the physical layer, IEEE 802.15.4 devices can use three frequency bands: 868.0-868.6 MHz (center frequency 868 MHz) in ITU Region 1, 902-928 MHz (915 MHz) in ITU Region 2 and 2400-2483.5 MHz (2.45 GHz) worldwide. Respectively, the number of available channels in each band is 1 for 868 MHz, 13 for 915 MHz and 16 for 2.45 GHz. Two of these bands, 915 MHz and 2.45 GHz, are located on license-free industrial, scientific and medical (ISM) bands. Since the 2.45 GHz band is one of the ISM bands and available worldwide, it is most commonly used in WSNs that operate under IEEE 802.15.4 standard. (IEEE802.15.4, 2018)

Originally the IEEE 802.15.4 standard specified two physical layers; one to support 20 kbit/s transmission speed for 868 MHz band and 40 kbit/s transmission speed for 915 MHz band, and another one to support 250 kbit/s transmission speed for 2.45 GHz band. These specifications are based on direct sequence spread spectrum (DSSS) modulation technique. Later the maximum data rates of the two lower bands were improved to reach 250 kbit/s as well. Four alternative physical layers are defined so that three of them use the combination

of DSSS and binary phase shift keying (BPSK) or DSSS and quadrature phase shift keying (QPSK). The latter one is used in the 2.45 GHz band. There is also one physical layer defined for 868 and 915 MHz bands so that it uses a combination of binary keying and amplitude shift keying. (IEEE802.15.4, 2018)

In addition to these three bands and their specifications by IEEE 802.15.4 standard, there are further variations defined by standardization groups IEEE 802.15.4a – IEEE 802.14.4e. However, those are out of the focus of this dissertation work, which focuses on sensor networking under IEEE 802.15.4 protocol.

2.2.2 Network and Data Transmission

IEEE 802.15.4 standard defines two types of network nodes: full-function device (FFD) and reduced function device (RFD). FFD can communicate with every other device in the network. It can also relay messages between other devices and operate as a network coordinator. In addition to coordinating the entire network, the FFD can also act as a cluster head and coordinate the network cluster, if such architecture is applied. RFD can only communicate with FFD, and it cannot act as any kind of coordinator. The RFDs are typically simple devices with scarce resources, and they are used only for simple tasks. FFDs can be different types of sensor platforms and actuators, which operate in the network.

Every network must have a PAN coordinator, which works as a coordinator of the whole network. As a consequence, every network must have at least one FFD. Every device in the network has its own 64-bit identifier. In some cases shorter 16-bit identifiers can be used in a restricted environment. Two network types are defined: a star network and a peer-to-peer network. (IEEE802.15.4, 2018)

In a star network, there is a central device which has a direct radio link with the rest of the devices. The central device must be a FFD, since it operates also as a network coordinator. The rest of the devices can be RFDs or FFDs. This network architecture fits best for relatively simple networks, which are used to collect data, which is then processed in a centralized manner.

In peer-to-peer network, the network topology can form arbitrary patterns, which are only limited by the node locations and communication range between the nodes. A multi-hop paths consisting of several radio links between the nodes can be applied to transmit messages between such nodes, which are not connected to each other by a direct radio link. There can be both FFDs and RFDs in the network. It can be further structured so that the FFDs form the trunk and the

RFDs the leaves of the network. This enables us to create local structures, such as clusters with the cluster member nodes and cluster head node, to the network. Clustered networks with a local coordinator (cluster head) in each cluster are called mesh networks. Peer-to-peer architecture forms the basis for ad hoc networks, which are capable to perform self-organization and self-management operations. Computation and other operations can be performed in a fully distributed manner in each node, in a locally centralized manner in clusters and cluster heads, or in a fully centralized manner in a network coordinator node after collecting the data from the network.

There are four fundamental frame types defined for data transmission: data, acknowledgement, beacon and MAC command frames. In addition, a superframe structure, which consists of sixteen equal length slots, can be applied. It is typically used with such applications, which require synchronization and short response times. Data transfers between nodes can be coordinated by beacon messages and by the carrier sense multiple access with collision avoidance (CSMA/CA). Point-to-point networks can use either unslotted CSMA/CA or other synchronization mechanisms. If beacon messages are not used, the CSMA/CA with random backoff can be applied. Acknowledge messages to ensure the reception of the transmitted packet can be applied in data critical applications, but their use is optional. (IEEE802.15.4, 2018)

2.3 Sensor Networks as an Enabler of New Type of Automation Applications

2.3.1 Benefits Provided by Wireless Network

So far the size of the wireless sensor nodes in IEEE 802.15.4 networks varies typically from some square centimeters to the size of average cellphones. This size range makes it possible to use standard electronic components in the nodes. It is also small enough to make it possible to attach the nodes to many kinds of mobile systems or system parts. They can be added to many existing systems without a need to modify the system itself. The sensor nodes can be used for space or areal monitoring in such spaces, where cabled connections cannot be used to cover it. They can also be mounted to such places, where the harsh conditions, such as dust, dirt, temperature, vibrations etc., make it difficult or impossible to use cabled sensors.

Compared to the cabled systems, the WSN provides also savings, flexibility and more data. Even the obvious fact that one gets rid of the cables in a wireless deployment means remarkable savings, because the cabling costs can cause the

major part of the total costs of the automation system. The unit price of the sensor nodes is typically less than the price of the cabled sensor system in respective use.

WSN automatically configures itself if new nodes are entering or existing nodes leaving the network. This gives a lot of flexibility compared to the cabled automation system, because new devices can be added and removed from the network on fly without any manual installation or configuration work. For example, mobile machines can join and leave the network as needed and sensor-equipped cargo units can be followed in real time once they move in different parts of the logistic chain. Cheaper unit costs and flexibility in the WSN architecture and its installation makes it possible to collect more measurements than before from the observed system. This can be further utilized, as discussed in the following subsections. (Dargie & Poellabauer 2010), (Frotzscher 2014), (Sakthidharan & Punitha 2014), (Sohraby 2007)

2.3.2 The Utilization of Data and Distributed Architecture

Compared to the older systems, WSN architecture enables to install more measurement points and collect more measurement data. This data redundancy can be utilized in many ways. It provides a more complete model about the phenomenon which is observed by the measurements, and makes it possible to improve the accuracy and robustness of the control algorithms. It is also easier to apply advanced control methods, such as adaptive, predictive and self-tuning control, once there is more diverse data available. Moreover, the distributed architecture of WSN enables distributed data processing, which makes it possible to apply local control loops in the network without swapping the measurements and the control command data back and forth between the centralized network control and the location where the measurements are made and the control is applied. This makes it possible to shift from hierarchical control to the distributed or locally centralized control.

The distributed network architecture also provides remote access to the different parts of the system, and this access can be utilized in monitoring and control. Several system parts and individual devices can be remotely accessed, if needed. Measurement data can also be remotely collected from several sites for further use. This can be beneficial, for example, in remote service and in condition monitoring.

Possible wear, breach and other system malfunctioning can be detected from the collected data. Service operations and immediate need for serviceman intervention can be based on the detected problems. As a consequence, the service can be scheduled based on the actual need of the monitored system instead of using just usage hour based service schedule. This would be more efficient and economical for both the system user and the service provider. The

automatic condition monitoring will improve the productivity and extend the system lifetime, since the problems can be detected and fixed at their early levels before they become bigger and more difficult to fix. Remote access provide also other types of benefits, since a vendor can monitor his products from one location, make software updates to several machines in several locations from one place etc. The data collected by WSN can also be utilized to make more precise error models for system fault diagnostics. (Dargie & Poellabauer 2010), (Frotzscher 2014), (Sakthidharan & Punitha 2017), (Sohraby 2007)

2.4 The Challenge to Fill the Automation Requirements

2.4.1 Performance

Many automation applications have strict requirements rising from the nature of the system and from the applied standards. These requirements can be related to the speed, response time, robustness and reliability. WSN must fill these requirements to be feasible for automation.

A sampling rate producing data fast enough, so that the control can follow the process and react to the changes, is required. Required sampling rate depends a lot on the system. In process automation it can easily be in the magnitude of several kHz, but in some simple monitoring and control applications few times in an hour can be enough. In addition to being limited by the sensor node processor speed, the sampling rate is also limited by many hardware issues, such as the sensor saturation time, the time that it takes to write the measured data to the sensor node memory, bus speed in the printed circuit board (PCB) etc.

In addition to sampling rate, the WSN data processing capacity is also limited by the computation speed and transmission speed. Some data can be processed locally in the node so that it is not necessary to transmit everything which is measured, but a certain amount of data must be transmitted to the upper levels of the network. IEEE 802.15.4 standard relies on the transmission speed of 250 kbit/s (IEEE802.15.4, 2018). In some cases it is enough to satisfy the application requirements but in some cases it can become a bottleneck. (Frotzscher 2014), (Sakthidharan & Punitha 2017)

2.4.2 Reliability

Since the sensor nodes are cheap and easy to deploy, a lot of data can be collected by the WSN. However, compared to the cabled system, there are also more challenges related to the data. Some data packets can get lost or corrupted during the processing and transmission. The content of the data packet can also get corrupted in the measurement because of sensor malfunctioning. Wireless

communication does not preserve the order, which implies that the packets are not necessarily received in the same order as they are transmitted. To summarize, the received data can be incomplete, misleading and disordered in terms of time.

If a sensor node gives index to each packet it transmits, the correct order of the received packets can be restored and also the locations of the missing data packets in the data flow can be found. If the sampling rate of the node is known and constant, the time series of the received measurement data can also be reconstructed correctly. However, this is the case with one node only. Since there are several nodes in the network, the indexing of the measurement data is not enough to reconstruct the time series of the received data correctly. Time stamps are required instead. To be able to place the data measured by different nodes to the same time series, each node and actuator in the network must have same time, which requires time synchronization in the network. Several time synchronization protocols are developed to set and keep up the time synchronization in WSN during its operation.

There exist different types of handshake protocols to ensure that every transmitted packet is arrived. In general, the way how they work is that every transmission is agreed and then every transmitted packet acknowledged. If the acknowledgement is not received in a pre-defined time, the receiver sends a retransmission request to the transmitting node. There are two main problems in the handshake protocols: the increase of time-variant delays and the communication load in the network. That makes them unfeasible for many automation applications.

In automation and real-time applications, different data fusion methods are a good way to improve the received data. For example, filters can be used to estimate the missing data and to detect the erroneous measurements and to replace them with the estimates of the correct values. Simultaneously they will also reduce the effect of noise, which always exists in the measurements. Since the data fusion can be executed in a centralized manner in the network coordinator or in the cluster head, it does not require any additional messaging between the nodes. (Frotzscher 2014), (Sakthidharan & Punitha 2017)

2.4.3 Power Supply

In the case of automation systems, the WSN energy-efficiency and low energy operation requirements must be relaxed that much that the system performance requirements can be filled. The minimization of the energy consumption is still important, but it must be done by taking into account the boundary conditions that follow from the requirements of the automation system.

The power supply of the sensor nodes is a challenge. If they are powered by batteries, the battery lifetime must be as long as the service period of the rest of

the system. It is not realistic to think that the batteries would be manually changed more often. The battery should not remarkably increase the overall size, unit prize and weight of the sensor node, which sets further challenges for the battery use. Alternatively the energy can be harvested from the environment. Many methods to harvest from different sources, such as light, vibrations, temperature difference, electromagnetic field, air or liquid flow etc., are developed and a combination of them, or a combination of harvesting methods and battery, can also be applied. Even though the energy harvesting solves the sensor node power supply issue in some cases, it has also many limitations. In most of the cases, the energy collected by harvesting is not enough for very high performance in node operation. The combination of different harvesting methods makes the hardware design more complex and also increases the unit price of the node. If a combination of energy harvesting and battery is applied, one must still change the battery according to some service schedule. (Chen 2014)

Technical development of batteries, energy harvesting components and some other electronic components like supercapacitors, which can be utilized in energy harvesting, is also rapid. This will improve the energy harvesting opportunities in the future. However, the power supply of the wireless sensor nodes still remains a critical issue that limits the applicability of WSNs in wireless automation.

2.5 Business Potential

WSN technology is currently in a transition process from emerging technology to a well defined part of the automation systems. During the first decade of 2000's, a remarkable part of WSN related research activities were used to solve the basic low level issues, such as data transmission protocols, energy efficiency and localization. These issues are still indeed important, but currently there exists feasible solutions for them. The system development is proceeding so that the relation and interfacing between WSN and the other parts of the automation system is also considered. There exist solutions like 6LoWPAN (IPv6 over Low power Wireless Personal Area Networks), which enable IPv6 (Internet Protocol version 6) packets to be sent over IEEE 802.15.4 networks (6LoWPAN 2018). Wireless sensor nodes and actuators have also that much computation and memory resources, that the required data structure conversions to communicate between the WSN and the devices into which the sensor nodes are connected, can be made locally in the nodes. A sensor node can also act as a gateway between WSN and other type of network. Similar interfacing and compatibility issues have also been considered in cabled communication systems. This has led to convergence, which is currently discussed under the concepts of Digitalization, Industrial Internet, Internet of Things (IoT), Cyber-Physical Systems (CPS) and Big Data. (Alippi 2014), (Khaitan & McCalley 2015), (Sakthidharan & Punitha 2017)

By definition, Internet of Things is a network of physical objects or "things" embedded with electronics, software, sensors, and network connectivity, which enables these objects to collect and exchange data. IoT allows the objects to be controlled and sensed remotely in the existing network infrastructure, which enables closer integration between the physical world and computer systems. These networks can be cabled, wireless or a combination of them (Acharjya & Geetha 2017). If the degree of distribution is very high or the role of sensors and actuators is dominating, the system is usually classified to more general class, which is called the Cyber-Physical System (CPS). By definition, in CPS a system of collaborating computational elements is controlling the physical entities. The main difference between traditional embedded systems and CPS is that instead of standalone devices, the CPS is typically designed as a network of interacting elements with physical input and output (Khaitan & McCalley 2015).

Digitalization and Industrial Internet are more general terms, which are often used in strategic and marketing purposes when talking about the applications of IoT and CPS. Industrial Internet has also been popularized by Industrial Internet Consortium (IIC) (Industrial 2018).

The expectations about the technological and commercial impact of IoT and CPS are very high. According to Cisco IBSG, the number of connected devices exceeded the world population somewhere between 2008 and 2009. Moreover, the number of devices connected to the Internet was estimated to be around 25 billion on 2015 and predicted to double around 50 billion until 2020 (Evans 2011). In the USA, the National Science Foundation (NSF) has defined CPS to be one of the key areas of research.

Wireless automation and WSNs are important sub-areas of IoT and CPS. As a consequence, the increase of the amount of their industrial applications will follow the increase of IoT and CPS. According to ON World prediction, the global WSN shipset shipments will increase from 100 million on 2011 around 500 million on 2015 and at least one billion until 2017. This prediction covers all radio technologies applied to WSNs, but they estimate that during the predicted period, IEEE 802.15.4 with ZigBee will make the largest market share though the WiFi and Bluetooth low power variants will grow faster. (Hatler 2018)

As a summary, the business potential and the commercial expectations are currently very high in the area of wireless automation. However, to be able to successfully commercialize the wireless automation systems under IEEE 802.15.4 standard, offering the technical novelties is not enough. Customer interests and customer satisfaction must also be considered. Two crucial questions are how to show the additional value the wireless automation provides and how to guarantee the reliability of the new kind of technology. To be able to answer properly, one must connect the customer needs and technical opportunities. It can be done by pointing out shortages in the existing automation systems as well as completely new kind of automation solutions, which can be provided by wireless automation. These solutions must be

compatible with the rest of the automation system, and the interfacing must be provided as a part of the wireless solution.

3. EXISTING WIRELESS AUTOMATION STANDARDS WIRELESS HART AND ISA 100.11A

3.1 WirelessHART

Highway Addressable Remote Transducer Protocol (HART) is one of the most common industrial automation protocols designed especially for process automation. It has been in the markets since 1980s, and it supports both digital and analog communication in 4-20 mA wiring. WirelessHART is a HART compatible wireless sensor networking technology, which is developed by HART Communication Foundation and governed by the FieldComm Group, which also governs HART. It uses IEEE 802.15.4 standard with Direct Sequence Spread Spectrum (DSSS) radio in 2.4 GHz ISM band. WirelessHART was introduced to the markets in September 2009, and approved by International Electrotechnical Comission as IEC 62591 standard in October 2010. (Fieldcomm 2018)

There are eight different device types defined in WirelessHART: router, field device, adapter, handheld device, access point, gateway, network manager and security manager. Each device is required to have a capability to act as a router, which can route packets in a mesh type of wireless network. Field devices are equipped with one or several sensors or with actuating capabilities (such as valves etc.), and they perform different sensing and actuating tasks in the network. Adapters provide a connection between cabled network that operates under wired HART and wireless mesh network that operates under WirelessHART. The handheld devices can be used to access the WirelessHART network directly, for example, in configuration or diagnostic purposes. Access Point connects the wireless network to the gateway between wireless network and the cabled automation network. There can be one or several Access Points in single network architecture. Gateway acts as a bridge between the wireless network and the host applications that are located in the cabled automation network. Network Manager and Security Manager can either be located on the gateway or be separate from it. Gateway and Network Manager can also be redundant, if preferred. (Nixon 2012), (Fieldcomm 2018)

There are strict key requirements to ensure system reliability and compatibility in WirelessHART. Every device must be compatible with every other WirelessHART device, and support routing. Time synchronization is always required, and communication is divided into 10 ms time slots. Communication security functions must always be in use; it is not possible to implement WirelessHART

without using them. This property is made intentionally to make the WirelessHART eligible for industrial automation, because the communication security is a mandatory feature in the automation system. (Lennvall 2008), (Nixon 2012), (Fieldcomm 2018)

The WirelessHART Network Manager computes and keeps up the overall routing information in the mesh network. It is based on the graphs formed by devices and wireless links between them, and made redundant whenever possible. Routing information computation is continuously repeated to adapt to the changes in the network topology. The Network Manager computes a communication schedule for each device based on the available channels, routing information, time synchronization and the particular communication needs of that device. The communication schedule is then transmitted from the Network Manager to each device in the form of transmit and receive slots in which that particular device can transmit and receive data. (Nixon 2012), (Fieldcomm 2018)

WirelessHART can be scaled up by using several Gateways that connect the automation network to HART-over-IP backbone. Then one or several Access Points can be used to connect between WirelessHART and HART-over-IP. This is well suitable for distributed control systems, since the architecture enables them to connect through multiple Gateways. If multiple Access Points are used, the centralized WirelessHART Network Manager will manage the network resources. It can avoid the use of same channels in the overlapping WirelessHART networks to avoid interference, and re-use the channels in non-overlapping networks to maximize the use of the network resources in terms of number of devices and system throughput. Since there can be multiple HART-over-IP backbone Access Points, the throughput to backbone can be remarkably higher than a throughput of a single WirelessHART network, and the backbone can connect multiple WirelessHART networks that are located to different parts of the factory. (Nixon 2012), (Fieldcomm 2018)

Compared to seven layers OSI-model, there are five layers in WirelessHART protocol stack: physical, data link, network, transport, and application layer. In addition to these layers, there is the centralized Network Manager, which is responsible for routing, scheduling and overall network resource management. (Nixon 2012), (Fieldcomm 2018)

WirelessHART uses 2.4 GHz IEEE 802.15.4 DSSS physical layer. It supports channels 11-26 in 2400-2483.5 MHz band with a 5 MHz gap between adjacent channels. Supported data rate is up to 250 kbits/s. (Nixon 2012), (Fieldcomm 2018)

WirelessHART data link layer is based on IEEE 802.15.4.-2006 MAC and fully compatible with it. It is extended further by adapting fixed 10 ms timeslots with synchronized frequency hopping and time division multiple access (TDMA) to enable collision free reliable communication. Timeslots are managed by a superframe, which presents a sequence of consecutive time slots. The total length of the slots in a superframe forms a period, and each superframe repeats itself according to its period. Every WirelessHART device supports multiple superframes with a different number of timeslots. This enables a mixing of fast, slow, cyclic and acyclic network traffic. Communication reliability is improved further by enabling channel blacklisting. Channels which are observed to suffer from continuous interference or other transmission disturbances can be put on the blacklist, and the Network Manager can completely disable the use of those channels. Time synchronization messages are added to the transmission acknowledgement packets to continuously repeat the time synchronization to keep it up, as required by TDMA. (Nixon 2012), (Fieldcomm 2018)

WirelessHART Network Layer is responsible of routing and security. Security Manager operates there, and the layer takes care of data end-to-end integrity and privacy in the wireless network. Route tables for packet routing along graphs are kept up in a Network Layer as well as time tables to allocate communication bandwidth for specific services. (Nixon 2012), (Fieldcomm 2018)

WirelessHART Transport Layer provides a reliable, connectionless communication to the Application Layer. Received packets are acknowledged by the end device so that in the case of a lost packet, the source device can retransmit it. (Nixon 2012), (Fieldcomm 2018)

The Application Layer in WirelessHART is HART. This makes WirelessHART compatible with HART applications and enables the wireless extension of the existing HART-based industrial automation systems. (Nixon 2012), (Fieldcomm 2018)

3.2 ISA100.11a

International Society of Automation (ISA) has developed the standard ISA100.11a under the title “Wireless Systems for Industrial Automation: Process Control and Related Applications”. It was approved by ISA100 standards committee as ISA100.11a on 2009, and by IEC as IEC 62734 on 2014. Certification and testing is sustained by the ISA100 Wireless Compliance Institute (ISA100 2018).

The ISA100.11a targets to provide an easy compatibility with the Internet and IP based networking, and simultaneously also flexibility and a variety of different architectural options. Like WirelessHART, it also uses IEEE 802.15.4 standard with DSSS radio in 2.4 GHz ISM band. ISA100.11a defines a protocol stack, system management and security options. Eight different roles for the devices are defined, and each device can have one or several roles. Defined roles are I/O device (input-output), router, provisioning device, Backbone Router, Gateway, System Manager, Security Manager and System Time Source. Every device does not need to support routing or provisioning. As a consequence, some devices are capable for only star topology networking and only devices equipped with the provisioning capability, can support other devices to join or leave the network. (ISA100 2018), (Nixon 2012)

Compared to the OSI seven layer model, ISA100.11a protocol stack has five layers: physical layer, data link layer, network layer, transport layer and application layer. In addition to these five layers, a centralized System Manager is defined by a detailed description and a set of services. The physical layer is based on 2.4 GHz IEEE 802.15.4 DSS. Data link layer uses its own MAC protocol, which is a modified, non-compliant version of IEEE 802.15.4-2006 MAC. The data link layer includes the usual OSI model functionalities such as establishment of the packet structure, framing and error detection. In ISA100.11a, the data link layer is extended to include also the following functions (Nixon 2012):

- link local addressing
- message forwarding
- physical layer management
- adaptive channel hopping
- clock synchronization
- detection and recovery of message loss and message addressing
- timing and integrity checks

In ISA100.11a, the length of the time slot is configurable. Three channel hopping sequences and five hopping patterns are defined. When a device joints to the network, the System Manager will configure the time slot and hopping patterns according to network configurations. The data link layer applies graph routing, where each device can be associated with several graphs. The one which is used in a certain type of communication depends on the type of that communication.

This routing scheme enables a simultaneous communication of different types of data in the network because different packet types may be routed via different paths based on the graphs they are associated. (Nixon 2012)

The routing graphs are calculated by the System Manager. It uses the input from the network configuration and the information the network devices provide about the radio frequency (RF) environment. Different traffic types, amount of data, requirements for frequency, real-time requirements (allowed latency), the performance of different channels and the energy resources of each device are all taken into account in the routing graph computation. The System Manager assigns contract IDs for the graphs it computes and notifies the network devices about the routing graphs and the respective contract IDs. Then the devices can use the contract IDs in their messages to notify the nodes on the route about the routing graph that must be applied for that packet. These contract IDs are carried within the data link layer header. (Nixon 2012)

In the network layer, the ISA100.11a devices can be connected directly to the Internet. ISA100.11a network edge backbone routers that support 6LoWPAN will fragment and reassemble the IPv6 packets so that they can be transmitted over the IEEE 802.15.4 based ISA100.11a network. In the ISA100.11a network, the packet routing will happen by using the DLL headers. Even though the IPv6 packet fragmentation and reassembly is supported on the network level, there is a bottleneck in the data transmission capacity: ISA100.11a does not define how multiple backbone routers could coordinate the reassembly of the fragmented packets. As a consequence, a fragmented data flow from one source must be all routed through the same backbone router, to be able to reassemble it properly. This limits the IPv6 data transmission capacity from a single source to the capacity of one backbone router. (Nixon 2012)

In the transport layer, ISA100.11a supports connectionless service based on User Datagram Protocol (UDP) with an enhanced message integrity check and end-to-end security. (Nixon 2012)

The only application, which is defined in ISA100.11a application layer, is System Manager. Otherwise only a set of services for user applications is specified. (Nixon 2012)

3.3 Comparison

WirelessHART (IEC 65291) and ISA100.11a (IEC 62734) both use IEEE 802.15.4 communication standard in 2.4 GHz ISM band. WirelessHART uses IEEE

802.15.4-2006 MAC as it is. ISA100.11a uses a modified version of it. Both technologies are feasible for wireless automation and there are several manufacturers that offer products for them. They both support wireless, low-power mesh communication in wireless sensor and actuator networks. ISA100.11a can support HART, but WirelessHART which is especially designed for that purpose, is more efficient and straightforward for that purpose. (Lennvall 2008), (Nixon 2012)

WirelessHART is designed to be a wireless extension of HART protocol and fully compatible with it. It can be connected via access points to HART over IP backbone and by using that backbone to the other parts of the automation system. All WirelessHART devices must be compatible with each other and capable for routing to be able to operate as a node in a mesh type of network. FieldComm Group (former HART Communication Foundation) coordinates the development and performs the interoperability testing between WirelessHART devices. Security functions must always be in use, and the application layer of WirelessHART is HART. (Lennvall 2008), (Nixon 2012)

ISA100.11a is designed to support a wide range of applications including the industrial automation. Since it is not directly connected to any communication protocol in cabled automation systems, it leaves many options to be defined for the user. As a consequence, starting to use ISA100.11a is not as straightforward for the user as staring to use WirelessHART. On the other hand, since ISA100.11a is not tied to one existing protocol, there are many options how to build the network depending on the particular needs. The application layer only defines a set of services for user applications and the System Manager. Security functions are available, but the use of them is not mandatory. Different network devices can have one or several different roles, and all of them do not need to support routing. (Lennvall 2008), (Nixon 2012)

WirelessHART devices produced by different manufacturers are guaranteed to be compatible with each other (requirement in performance testing), they all support mesh networking and the communication security functions must be in use. Also the compatibility with HART field devices exists. In ISA100.11a, these properties are all available, but they must be taken into account and configured in the network design. This makes WirelessHART straightforward to use in the wireless extensions and updates of HART-based automation systems. On the other hand, ISA100.11a offers an in-build way for IP based networking and direct connections to the Internet or IP based intranet, since the network layer supports IPv6 addressing by using 6LoWPAN. The only shortage is that only one backbone router can be used for the communication of one particular data flow, because

the backbone routers do not coordinate the packet fragmentation and reassembly with each other. (Lennvall 2008), (Nixon 2012)

ISA100.a enables also a wider variation of network architectures, since there can be devices that support routing and devices that are only capable for star topology networking. All devices do not need to support other nodes joining or leaving to the network either. The drawback is that more application specific tailoring is required. (Lennvall 2008), (Nixon 2012)

WirelessHART works well in certain type of automation applications and provides a straightforward way to include wireless communication in the existing HART applications. This is a benefit but also a shortage, since the standard strictly limits the architectural options that can be used in WirelessHART networks. ISA100.11a enables more architectural options and leaves more space to include application specific features, but also requires more case specific configuration. Both standards are mainly targeted for such automation architectures, where data is first collected over the wireless network to the backbone network and then over the backbone to the automation controller, where the control computation is executed. Then the control commands are sent over the backbone and the wireless network to the devices. Development of generic solutions, which include distributed or locally centralized computation in the network and seamless interfacing to other parts of the automation system leaves space for open platform development. However, the support of these standards would be a benefit for the open platform as well. (Lennvall 2008), (Low 2013), (Nixon 2012)

4. EVOLUTION OF THE EXISTING OPEN SOURCE SYSTEMS

4.1 General Trends

Originally the main target in open source development was in tiny, dust kind of sensors with very low power consumption and low cost (Pister 2001). This target implied also low data sample and transmission rate and short communication range. Once several different types of WSN application areas were considered, the research and development targets became more diverge. Micro and nanosize platforms are still one area of interest, but the development trend that became particularly important for wireless automation was to allow a bit bigger platform size and power consumption to be able to achieve higher performance. This performance covers data sampling, data transmission and computation capacity and speed, since they are all critical to fill the automation system requirements.

Since the beginning of 2000's, the size (external measures) of the wireless platforms targeted for wireless automation has been varying roughly between the size of the USB flash drive and the size of a smartphone. Increased CPU processing power, main memory size, dynamic multiple operational modes, secure and reliable communication, energy efficiency and reduced overall costs have been among the targeted key features in the sensor node development all the time. Some of these target features, such as increased processing power and energy efficiency, are conflicting with each other. As a consequence, compromises and tradeoffs between the conflicting targets are also required in the development. At the same time the technology development has effected the platform development. Microprocessors and microcontrollers have become more efficient. The memory capacity has been increasing and its price reducing at the same time. This has made it possible to add more program and data memory in the sensor nodes. Big commercial producers are also bringing an increasing amount of system-on-chip solutions to the markets.

The general development of electronics has produced new components and improved the existing ones such as supercapacitors, solar cells, other energy harvesting components, GPS chips etc. Many of them are more efficient but also cheaper, smaller and less energy consuming than before. This has an impact on the sensor node development.

Some of the biggest steps are taken in the software development. In WSNs, small devices with scarce memory, computation and communication resources require

their own type of software solutions. The operating system must be light-weight in terms of memory and energy consumption, and it must enable distributed operation. It must support IEEE 802.15.4 standard and the most important standards like WirelessHART and ISA100.11a. Many important solutions related to the optimization of energy consumption and the communication and operation reliability, are done in the software. For example, there has been a huge amount of work in energy-efficient MAC protocol development during the latest 17 years. Different operational modes, like time synchronized, non-synchronized and event based modes are also all defined by the software. Hardware independency, backward compatibility and separation between the OS and the applications are some of the general targets in the software development. Recently, after the breakthrough of IoT and Industrial Internet, increasing effort in the WSN software development have also been paid to the compatibility and interfacing between the WSN and other parts of the automation system.

4.2 Examples

One of the most well-known examples from the early years of the WSNs is the family of Mica sensor nodes (called motes in marketing) by the University of California at Berkeley, Crossbow and MEMSIC. The family consists of Mica (Hill & Culler 2002), Mica 2 (Mica2 2003), Mica2dot (Mica2dot 2003) and Micaz (Micaz 2004) sensor nodes. These nodes were introduced on 2002-2004. All of them are equipped with Atmel AT mega 128L processor and use ISM bands. Three first mentioned have four frequency band options; 868/916 MHz, 433 MHz and 315 MHz, Micaz uses 2.4 GHz band (precisely from 2.4 to 2.48 GHz). The data rate of Mica2 and Mica2Dot is stated to be 38.4 Kbaud. Micaz has IEEE 802.15.4 compatible radio transceiver and it supports a data rate of 250 kbps.

All Mica nodes use TinyOS, which is an open source operating system for embedded systems such as WSNs. It is developed originally by the alliance of University of California at Berkeley, Intel Research and Crossbow Technology. TinyOS is written by nesC, which is such a variation of C programming language, which is targeted for the limited memory resources of the sensor nodes. TinyOS programs consist of software components and tasks, which are connected to each other by using interfaces. The operating system developer community provides interfaces, components and tasks for common abstractions such as communication, routing, sensing with different types of sensors and data processing. The memory allocation in TinyOS is static and there is no clear differentiation between the operating system and the application. Once the code

for particular application is compiled, it produces a single executable entity (TinyOS 2018), (Dargie & Poellabauer 2010).

XYZ node was introduced by Embedded Networks and Applications Laboratory at Yale University on 2005. It is equipped with OKI ML67Q500x ARM THUMB microprocessor and Chipcon CC2420 radio, which supports IEEE 802.15.4 communication in 2.4 GHz ISM band. System clock rate can be controlled between 1.8 and 57.6 MHz. This enables different operational modes depending on the computation and communication needs. Low clock frequency modes can be used to save energy. Higher clock rate modes with 32 kB of RAM, 4 kB of boot ROM and 256 kB of flash provide resources for more intensive computation. The memory can be further extended with 2 Mbits external RAM (Lymberopoulos & Savvides 2005).

XYZ node uses SOS operating system, which is developed at University of California, Los Angeles. Like TinyOS, it is also targeted to memory limited embedded devices like sensor nodes. It consists of modules, messages and kernel. Memory allocation and the reprogramming of the modules are dynamic (Han 2005). Like TinyOS, also SOS does not have a clear distinction between the operating system and the application (Dargie & Poellabauer 2010).

Sensinode Microseries sensor platform uses Chipcon's CC2420 radio and supports both TinyOS and FreeRTOS operating systems. The radio operates in 2.4 GHz band, and its maximum data rate is 250 kbps within 100 m transmission range. However, the existence of obstacles can easily reduce the maximum communication range. Node microcontroller MSP430 operates at 8 MHz and has 10 kB of RAM and 256 kB of flash memory. CC2420 transceiver has its own 4 Mbit serial data flash memory (Sensinode 2007).

Sensinode Nanostack operates on top of FreeRTOS operating system. It was one of the first protocol stacks to enable IP-based networking in WSN, since it included 6LoWPAN with IPv6 and UDP implementations (Ahonen 2008).

FreeRTOS operating system is designed for embedded devices, and it follows the principles to be simple and light weight in terms of small memory footprint, low overhead and fast execution. It is one of the most common operating systems for microcontrollers and small microprocessors. FreeRTOS is written mostly in C with a couple of assembly functions included. The kernel consists of only three C files. There is a host program called a thread tick method that enables the run of multiple threads. It switches tasks based on priority and utilizes round-robin scheduling scheme. There is also a tick-less operation mode for low-power applications (FreeRTOS 2018).

FreeRTOS is shared under MIT License, which is one of the free software licenses. The latest release is published in December 2017 (FreeRTOS 2018).

Contiki operating system (Contiki 2018) is originally created by Adam Dunkels from Swedish Institute of Computer Science and then further developed by a strong alliance that consists of industrial and university partners. Contiki was first time released on 2003, and the latest release came out in August 2015. It is targeted for power and memory constrained wireless embedded devices having its special focus on IP based networking and IoT (Dunkels 2004), (Contiki 2018). There are three networking stacks available in Contiki; the uIPv6 stack, which supports IPv6 networking (Durvy 2008), the uIP TCP/IP stack, which supports IPv4 networking (Dunkels 2003) and the Rime stack. Last one is a light weight communication stack for low-power wireless networks, for which the overhead of IP based networking is too high (Dunkels 2007).

Since Contiki is targeted for low power embedded systems, special attention has been paid for the energy efficiency and small power consumption. There are several mechanisms to reduce the power consumption of the system which is run by Contiki, and a low power radio protocol called ContikiMAC (Dunkels 2011), (Contiki 2018).

Currently there are many sensor platforms available by the commercial producers. Remarkable technical achievements during the latest years include the system on chip solutions, improvements in energy efficiency, increasingly utilized energy harvesting solutions and increased memory capacity. Power sources are also developing and becoming more long-lasting. Applied radio technologies are diversifying as mentioned in chapters 6-7. A typical sensor node has its use voltage around 1.8-3.6 V, sleep mode power consumption in the magnitude of microamperes, active mode in the magnitude of milliamperes and price varying from a couple of US dollars up to 125 USD. CPU clock frequency is usually within a range of 8-32 MHz, though there are some up to 500 MHz. Real-time clock is around 32 kHZ. Memory capacity in sensor platforms has been remarkably increasing from kilobytes to gigabytes, and new innovative memory solutions, such as ferroelectric random access memory (FRAM), have been adopted (Nodes 2018), (Srinivasa 2013).

A powerful option for the sensor platform is offered by the single-board computers, like the Rasberry Pi (Rasberry 2018) and Beagle Boards (Beagleboard 2018). These boards can run the usual PC operating systems and they have similar computation resources, which makes them much more powerful than low-resource sensor nodes. On the other hand, their power consumption is higher compared to sensor nodes and separate modules for the WSN transceiver

and sensors are required. One architectural option is to use the single-board computers as cluster heads or gateways in WSN, and otherwise more resource-limited nodes.

A low-power but efficient hardware extension for sensor nodes can be offered by a field-programmable gate array (FPGA). There have been some experiments about a joint use of sensor node and FPGA and the area is under intensive research. Many signal processing applications such as filtering and Fourier analysis can be performed much more efficiently with FPGA instead of just using the node microcontroller and memory. This can enable more advanced data processing in the network in future (Das 2011), (Goh 2012), (Piedra 2012),(Liao 2013), (Kumar 2013), (Engel 2014).

5. RESULTS

5.1 Network Initialization and Control

Related to network initialization and control, publications 1 and 2 are focusing on localization, Publication 3 on clustering and Publication 4 on time synchronization in WSN.

5.1.1 Localization

If the sensor nodes are equipped with localization capability such as GPS or if they are manually placed in static locations, there is no localization problem. However, there are many cases where these assumptions do not hold. Satellite positioning is not available inside buildings, and it might be too expensive to include the satellite positioning capability in every sensor node outside. Placing the nodes and writing down their positions manually is doable with relatively small node deployments, but this method is either inefficient or completely infeasible with deployments consisting of large number of sensor nodes.

In network-based localization, sensor node locations are computed based on the measured distances between them. Nodes can be equipped with distance measurement devices such as ultrasonic transmitter-receiver pairs (Savvides 2003), or distances between them can be estimated by using radio signal fading and routing information. In addition to distances, it is also possible to measure angles between them (Niculescu & Nath 2004), though it is more uncommon. Multidimensional Scaling (MDS) is a localization method, which uses the distance information and returns the relative coordinates of the nodes (Shang 2003), (Shang & Ruml 2004), (Xi & Zha 2004). The term relative coordinates is used here for the coordinates in two or three dimensions (2D or 3D), which are correct up to rotation reflection and translation. If there are three nodes in 2D or four nodes in 3D available in the preferred reference coordinate system, the final translation from relative to reference coordinates can be made.

Sensor node location computation by using MDS is based on distance matrix singular value decomposition (SVD). Assume D is a matrix of measured or estimated distances between sensor nodes so that each element d_{ij} in D is a distance between nodes i and j. For simplicity we can assume that $d_{ij} = d_{ji}$ for the same pair of nodes. In practice it means that the average of two

measurements or the more reliable one of them is used. For a sensor network that consists of n nodes, D becomes $n \times n$ matrix with symmetric lower and upper triangles and zeros in the diagonal (node distance to itself). D is converted to centered square distance matrix D_c by conversion

$$D_c = -\frac{1}{2}\left(I_n - \frac{1}{n}E_n\right) D^2 \left(I_n - \frac{1}{n}E_n\right), \quad (5.1)$$

where I_n is $n \times n$ identity matrix and E_n is $n \times n$ matrix that consists entirely of ones. When centered in this way, D_c becomes a symmetric $n \times n$ square matrix so that

$$D_c = D_c^T. \quad (5.2)$$

By definition, the singular value decomposition (SVD) of D_c is

$$D_c = UGV^T, \quad (5.3)$$

where G is a diagonal matrix having the singular values of D_c in its diagonal in decreasing order. These singular values are the nonnegative square roots of the eigenvalues of $D_c D_c^T$. Since D_c is a square symmetric matrix, it follows that the SVD in (5.3) becomes

$$D_c = UGV^T = UGU^T. \quad (5.4)$$

In (5.4), the left singular vectors of D_c are carried by the columns of U and the right singular vectors by the columns of U^T. Since G presents the singular values of D_c in decreasing order, it presents them in their order of importance related to the original data of D. As a consequence, the first three columns of matrix X in

$$X = UG^{\frac{1}{2}} \quad (5.5)$$

are the relative coordinates of the sensor nodes in 3D. Respectively, first two columns of X provide the coordinates in 2D. More formally, relative coordinates in 2D can be solved by using the best rank 2 approximation of D_c :

$$D_{c2} = U_2 G_2 U_2^T, \quad (5.6)$$

where U_2 is an $n \times 2$ matrix of the top left singular values of U. Then the relative node coordinates in 2D are given by

$$X_2 = U_2 G_2^{\frac{1}{2}}. \quad (5.7)$$

Since the sensor node coordinates are computed from the distance matrix when MDS is applied, the accuracy of the result depends on the accuracy of the distance measurements. In practice the distance measurements are always noisy. If the distance is computed based on the radio signal fading, which is indicated by the measured RSSI, the nature of the radio channel introduces some uncertainty to the distance estimate. Possible failures in sensor node operation and communication can also act as a source of error in the distance estimation. Moreover, since every node is usually not directly connected to every other node in the network, some pairwise distances cannot be directly measured. Those distances which exceed the node communication range must be approximated. One common and straightforward solution is the utilization of routing information. Once the communication path between two nodes which are not directly connected is known, the pairwise distance is estimated by computing the Euclidean length of the path. By doing so the distance estimate becomes always bigger or equal than the real distance. The use of Euclidean paths provides quite good results with random uniform node deployments, but if there are more complicated network shapes or obstacles that cannot be penetrated by the communication, the localization error caused by the distance estimation error with the Euclidean paths becomes remarkable.

Publication 1 presents distance matrix reconstruction for sensor node localization in the case of noisy and incomplete distance measurements. It also shows that when the method is applied, the resulting distance matrix reconstruction error is bounded and decreasing inversely proportional to $\sqrt{n}$, where n is the number of nodes in the region of deployment.

In square distance matrix D_s, each element is a square of the distance between nodes i and j. In other words

$$d_{s_{ij}} = d_{ij}^2, \qquad i,j \in [1 \dots n] \tag{5.8}$$

It is shown in Publication 1 that in d dimensions

$$rank(D_s) \leq d + 2. \tag{5.9}$$

It follows from (5.9) that in 2D, the rank of the square distance matrix D_s is at most 4. Thus, even though D_s has n^2 entries, it has only 4 linearly independent columns. As a consequence, carefully chosen $8n$ entries are enough for square distance matrix reconstruction. In practice it is not possible to make such an ideal chose of the right entries for reconstruction, but in Publication 1 the reconstruction is done by sampling the entries of D_s. In sampling, probabilities p_{ij} for respective distances d_{ij} are defined such that noisy distances

$$\widetilde{D}_{s_{ij}} = \begin{cases} d_{ij}^2 + \varepsilon_{ij} & with\ probability\ p_{ij} \\ unknown & with\ probability\ 1 - p_{ij} \end{cases}. \quad (5.10)$$

In (5.10) ε_{ij} are independent zero-mean random variables with bounded variance. In the idealized case one can assume that $p_{ij} = 1$ if $d_{ij} \leq R$ and $p_{ij} = 0$ if $d_{ij} > R$, where R is the sensor node communication radius. In a realistic case, it follows from the random nature of the radio path and from other possible sources of error, that there is always some uncertainty in R and it varies in different moments and directions. In Publication 1, two assumptions are made to satisfy this: 1) all p_{ij}'s are known and 2) $p_{ij} \geq p_\varepsilon > 0$ for all $i, j \in [1 \dots n]$, where p_ε is some small positive constant.

In Publication 1, an estimator matrix S to estimate the square distance matrix D_s is constructed with following entries:

$$S_{ij} = \begin{cases} \frac{d_{ij}^2 + \varepsilon_{ij} + \gamma_{ij}(1 - p_{ij})}{p_{ij}} & if\ d_{ij}\ was\ measured \\ \gamma_{ij} & otherwise\ (1 - p_{ij}). \end{cases} \quad (5.11)$$

In (5.11), γ_{ij} are representing the best estimates for the distances between nodes i and j. Based on (5.9), the rank of S in 2D is at most 4. The localization algorithm *SVD-Reconstruct* presented in Publication 1 proceeds as follows:

1) Given noisy measurements $\widetilde{D}$ as presented in (5.10) and estimates γ_{ij} for those pairwise distances which are not measured, construct S as presented in (5.11).

2) Construct S_4, a best rank 4 approximation of S by using the SVD of S.

3) Compute the relative coordinates of the sensor nodes in 2D by using S_4 instead of D^2 as an input matrix in (5.1)-(5.7).

It holds for all entries of S, that the expectation value

$$E[S_{ij}] = d_{ij}^2 \quad \forall\ i, j\ \in [1 \dots n]\,, \quad (5.12)$$

because it was assumed that $E[\varepsilon_{ij}] = 0$ for all $i, j \in [1 \dots n]$. Since p_{ij} are bounded (probabilities), the variances of the entries of S are also bounded. Based on these properties and the bounds presented in (Achlioptas & McSherry 2001) it is proven in Publication 1 that

$$\|D^2 - S_4\|_F \leq O\left(n d_{max}^4 + n^{\frac{3}{2}} d_{max}^3\right). \quad (5.13)$$

It follows from (5.13) that the average square error per entry in S_4 is

(5.14) $O\left(\frac{d_{max}^4}{n} + \frac{d_{max}^3}{\sqrt{n}}\right)$.

Assuming that d_{max} is independent on n, (5.13)-(5.14) verifies the Publication 1 result that when SVD-Reconstruct algorithm is used, the node distance reconstruction error decreases inversely proportional to $\sqrt{n}$, where n is the number of nodes in the network.

In MDS, the idealized model $p_{ij} = 1$ if $d_{ij} \leq R$ and $p_{ij} = 0$ if $d_{ij} > R$ is usually assumed for the distance measurements (Shang 2003), (Shang & Ruml 2004), (Xi & Zha 2004). Under that assumption, (5.11) gets a form

(5.15) $S_{ij} = \begin{cases} d_{ij}^2 + \varepsilon_{ij} & if\ d_{ij} \leq R \\ \gamma_{ij} & if\ d_{ij} > R. \end{cases}$

In (5.15), the square of the shortest Euclidean path between nodes *i* and *j* is usually used as γ_{ij}. The form (5.11) presented in Publication 1 includes (5.15) but is more general. In (5.15), any knowledge about the distance measurement probabilities, radio environment and sensor node properties can be utilized to define the distance measurement probabilities p_{ij} and the estimates γ_{ij} for those distances that cannot be directly measured.

Figure 1 presents Publication 1 simulation results in the case of random uniform and random corridor type of node deployments. The WSN deployment area was 1×1 and the node communication radius was set to $R = 0.165$. The number of nodes was varied between 100 and 500. Since the deployment area and communication radius were kept constant, this variation of the number of nodes made the node connectivity (direct connections to other nodes) vary between 5 and 42. The left plots in Figure 1 are noiseless. In the right plots, the distance measurements were corrupted with noise. Applied noise had zero mean and uniform distribution which is 63% of the actual measurement.

In the simulations illustrated in Figure 1, a probability that node *i* can measure its distance to node *j* was selected so that

(5.16) $p_{ij} = \begin{cases} 1 & if\ R \leq 0.165 \\ \frac{1}{100} & if\ R > 0.165. \end{cases}$

Selection (5.16) fills the assumption that there is a small, nonzero probability also for nodes located further than *R* from each other to measure (or estimate) their mutual distance. Three options were used to approximate those distances that cannot be measured: $\gamma_{ij} = 0$, $\gamma_{ij} = R^2$ and $\gamma_{ij} = (shortest\ graph\ path)_{ij}^2$. In this expression, shortest graph path is the same as shortest Euclidean path between the nodes i and j. Executed simulations are presented in Figure 1. In each

simulation scenario, the number of nodes was varied from 100 to 500 and 10 simulations were run for each size. Plots in Figure 1 present the average square distance reconstruction error in each case. In addition to four different choices of γ_{ij}, also theroretical error (5.13)-(5.14) is plotted.

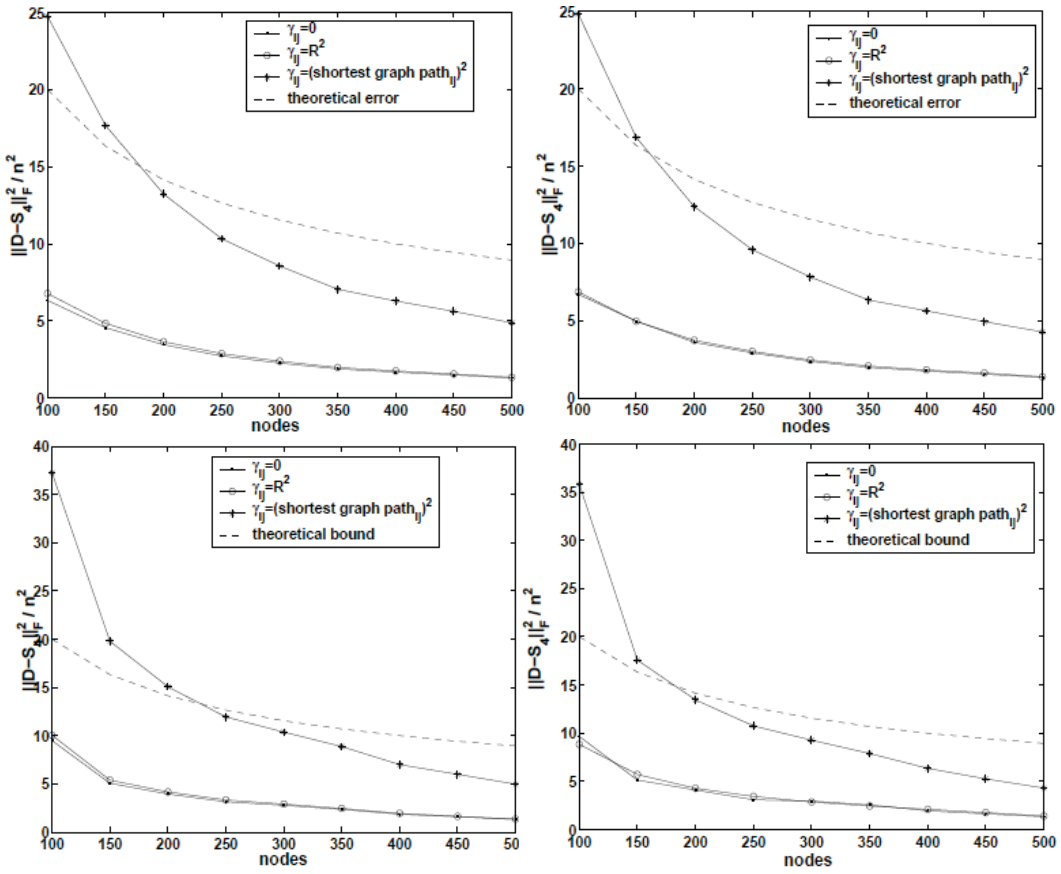


Figure 1. Simulation results for SVD-Reconstruct performance in Publication 1. Up left: random uniform deployment without noise, up right: random uniform deployment with uniformly distributed random noise which is 63% of the actual measurement. Down left: corridor type of deployment (see Figure 2) without noise. Down right: corridor type of deployment with similar noise as uniform case.

Simulations verify the result (5.13)-(5.14), which indicate that the distance reconstruction error is inversely proportional to $\sqrt{n}$, where n is the number of nodes in WSN. The results also indicate that the SVD-Reconstruct algorithm is noise tolerant. It was expected, because it was proven in the Publication 1 that σ_S, the variance of the entries of S, is bounded by

(5.17) $\sigma_S^2 \leq \max_{i,j} \frac{2}{p_{ij}} \left((d_{ij}^2 - \gamma_{ij})^2 + \sigma_\varepsilon^2 \right).$

In (5.17), the bound is dominated by the first term on the left side, and the effect of the noise variance σ_ε is smaller. The bound (5.17) also indicates that the closer to d_{ij}^2 the $\gamma_{ij}$is the better the reconstruction.

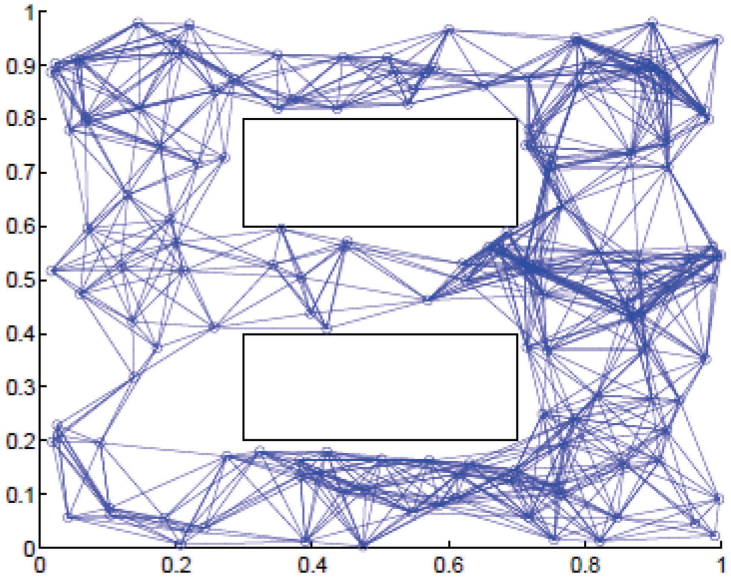


Figure 2. The corridor type of node deployment used in simulations in Publication 1.

On the second set of simulations in Publication 1, the performance of SVD-Reconstruct was evaluated and compared with MDS-MAP (Shang 2003) in such cases, when nodes have certain probability to fail to detect each other and measure their pairwise distance even if the distance is less than R. Two types of sensor nodes were considered: type 1 fails to detect a neighboring node within its communication rage with probability p_1 and type 2 fails with probability p_2. WSN deployment area was scaled to 1×1. Probability that sensor nodes detect each other if their distance is more than R was kept in 0.01. For the missing pairwise distances, estimates $\gamma_{ij} = 0$ and $\gamma_{ij} = R^2$ were used for SVD-Reconstruct and the squares of the shortest Euclidean paths for MDS-MAP. In first two experiments, node communication radius was set to $R = 0.1$. In the first case the probabilities to fail were $p_1 = \frac{2}{3}$ and $p_2 = \frac{3}{4}$ and in the second case $p_1 = \frac{1}{2}$ and $p_2 = \frac{3}{4}$. Results are presented in Figure 3.

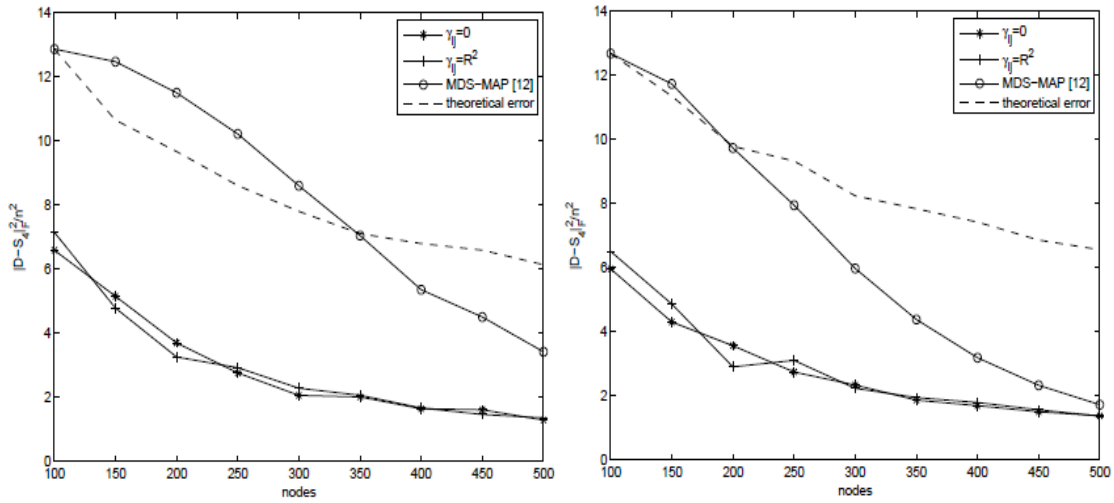


Figure 3. Comparison between SVD-Reconstruct and MDS-MAP in Publication 1. Probabilities $p_1 = \frac{2}{3}$ and $p_2 = \frac{3}{4}$ are used in the left plot for sensor types 1 and 2 to fail to detect neighboring node within their communication radius. On the right plot, probabilities $p_1 = \frac{1}{2}$ and $p_2 = \frac{3}{4}$ are used respectively. Node communication radius was $R = 0.1$ for all nodes.

In third experiment, probabilities were kept at $p_1 = \frac{1}{2}$ and $p_2 = \frac{3}{4}$, and the R was increased from 0.1 to 0.165. Simulation result is presented in Figure 4.

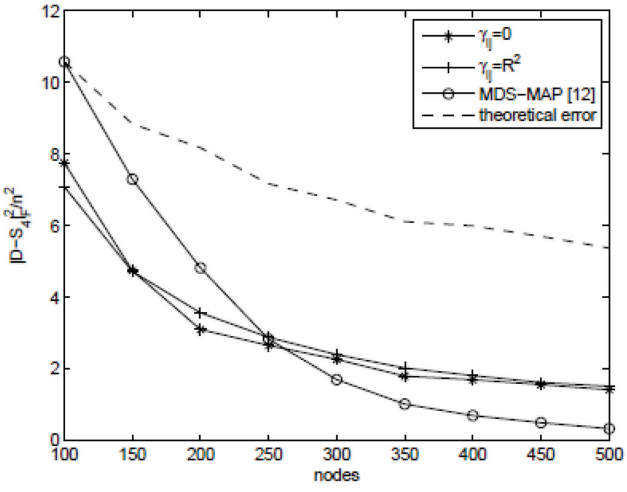


Figure 4. Comparison between SVD-Reconstruction and MDS-MAP in Publication 1. Probabilities $p_1 = \frac{1}{2}$ and $p_2 = \frac{3}{4}$ are used, but compared to Figure 3, here R is increased from 0.1 to 0.165.

The second set of simulations illustrated in figures 11-12 shows that the SVD-Reconstruct performs better than MDS-MAP with sparse sensor node deployments, but when the node deployment density increases, MDS-MAP performs better in the case of random uniform deployment.

One of the most common methods to compute the distance between two nodes which have direct radio link between them is based on the measured radio signal power fading. Once the power of the transmitted signal and the power of the received signal are known, one can first compute the power loss and then the distance based on the power loss. However, the properties of the radio channel such as path loss, shadowing and multipath fading, make the radio signal fading more complicated, since in addition to distance, it also depends on many other parameters and can vary over time (Bertoni 1999). Publication 2 presents a simple method to improve the power loss based distance estimation. The log-distance path loss model is

$$(5.17)\quad P_{loss} = P_{Tx} - P_{Rx} = P_0 + 10n_p log\left(\frac{d}{d_0}\right) + X_\sigma,$$

where P_{Tx} is the transmitted signal strength, P_{Rx} is the received signal strength, d_0 is a reference distance, P_0 is the received power at the reference distance and d is the distance between transmitter and receiver. X_σ is a zero-mean Gaussian random variable with a standard deviation σ, and it represents the flat fading. In Chipcon specification used for the power loss in Publication 2 (Aamodt 2008), it is assumed that the reference distance is one meter and the effect of flat fading is negligible. Thus, the power loss in distance d becomes

$$(5.18)\quad P_{loss} = -10n_p log_{10}(d) + A.$$

Compared to (5.17), in (5.18) the sign of the path loss exponent n_p is changed and a loss constant A is used instead of P_0. The initial assumption for A is $A = P_0$. Once the applied transmission power is known and the power of the received signal can be measured, P_{loss} can be computed as their difference $P_{loss} = P_{Tx} - P_{Rx}$. Then the distance can be solved from (5.18):

$$(5.19)\quad d = 10^{\left(\frac{P_{loss}-A}{-10n_p}\right)}.$$

However, (5.19) is a simplified model which holds well only in electromagnetically isolated vacuum. It does not provide good results in realistic radio environments in the area of IEEE 802.15.4, because of the effects of path loss, shadowing and multipath fading.

In Publication 2, the distance estimates (5.19) are improved by using a bounded optimization, which optimizes the parameter values to compute the distance estimates based on received signal strength measurements. The standard deviation of the measurements and the number of lost packets in each set of measurements are used as criteria to ignore the most unreliable measurements in the optimization computation.

The experiments in Publication 2 are made by using Sensinode Nanoseries, which was equipped with Radiocraft RC2301AT radio module and IEEE 802.15.4 supporting Texas Instruments CC2431 system on chip RF transceiver (Sensinode 2008). The longest distance was first limited to 30 m and later to 28 m, since it was observed that the communication with RC2301AT in-built ceramic antennas became highly unreliable on longer distances. Measurements started in 1 m distance, and the distance between transmitting and receiving nodes was increased in one meter steps from 1 up to 28 meters. In every distance, as many packets was transmitted as was required to receive 50 packets. A handshake protocol between two nodes was applied so that the receiving node was able to write down the measured power of each received packet and the total number of packets that was transmitted to get 50 packets through. Transmission power was kept constant. Based on these measurements, one was able to compute the average power loss, power loss standard deviation, and the number of lost packets in each distance from 1 to 28 meters in one meter steps.

For every distance i, $(i \in [1, \dots 28])$ the distance estimates $\widehat{d}_i$ were counted by using the equation (5.19). Then the sum of the absolute values of the difference between actual distances and estimated distances was used as a cost function:

(5.20) $f_{CF} = \sum_{i=1}^{28} \left| d_i - \widehat{d}_i \right|$.

The standard deviation of the received signal strength $\sigma_{P_{Rx}}^{i}$ and the number of lost packets N_{lost}^{i} were counted from a set of 50 measurements in each measurement point. They indicated the communication reliability on that particular distance. Since the bigger values indicated weaker reliability in communication, they indicated also a weaker reliability of that particular distance estimate. Based on this observation, the applied optimization routine computed optimal threshold values $\sigma_{P_{Rx}}^{*}$ and N_{lost}^{*}. If they were exceeded in some measurement point, that point was ignored in the optimization computation. Two other optimized parameters were path loss exponent n_p and loss constant A. MATLAB optimization routine fminsearch (Matlab 2018) was used to solve the optimal values $\sigma_{P_{Rx}}^{*}$, N_{lost}^{*}, n_p^{*} and A^{*}.

Another set of experimental data, which was not used in the optimization computation, was then used to analyze the effect of the optimization. Left plot in Figure 5 presents the true and estimated distances when (5.19) was directly applied. Each dot is the average computed from 50 measurements and the respective standard deviation is presented by the error bars. It is easy to observe that for distances longer than 10 meters, the estimates became weak. Right plot in Figure 5 presents the real and estimated distances when the estimates were computed by using the optimized parameter values for σ_{Rx}, N_{loss}, n_p and A. Red circle around the dot indicate such measurement points, which were included to the computation. Points without red circle were excluded, because either σ_{Rx} or N_{loss}, or both of them, exceeded the values of σ_{Rx}^* and N_{loss}^* in these points.

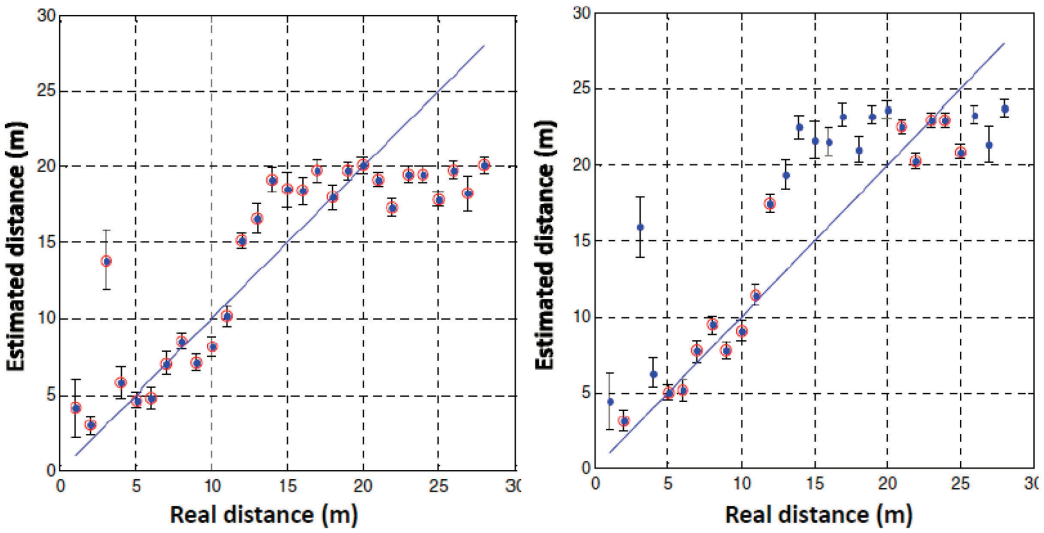


Figure 5. Distance estimation with and without optimization in Publication 2. Real and estimated distances, which are computed without optimized parameter values (left) and the same results which are computed by using the optimized parameter values (right). Red circle around the dot indicate that the particular point is used in the optimization computation.

Results indicate that the optimization method presented in Publication 2 clearly improves the distance estimates, which are based in radio signal power fading between transmitter and receiver. It was also relevant to use the standard deviation of the received signal strength and the number of lost packets in a certain measurement points as criteria to evaluate the reliability of the distance estimate on each particular distance.

6.1.2 Clustering

In large WSNs that consist of big number of sensor nodes and actuators, it is beneficial to organize the network into clusters. Clustered architecture helps the network management in general, and the spatial grouping of nodes can be utilized in data aggregation, data compression and transmission power control. Since the correlation of the measured data is higher between sensors located in close proximity to each other, the data compression rate can be higher between the nodes that are located into the same cluster. In data aggregation, the WSN communication load and sensor node resource consumption can be balanced by collecting data from only part of the nodes from each cluster in time. Clustering information can be used in WSN power control based on the fact that it usually requires less power to reach the nodes in the same cluster compared to the communication power which is needed to reach the nodes which are located to other clusters in the same network.

Publication 3 presents a distributed algorithm that partitions the WSN into a set of locally isotropic non-overlapping clusters. The algorithm can trace down locally isotropic subsets of nodes in globally non-isotropic node deployment, and cluster the network with respect to local density attributes. The size and number of the clusters depend on the node deployment, and they are not set in advance. There is no need to know the node locations either, but it is assumed that each node can measure distances to its one hop neighbors and exchange information about its two hop neighborhood. The only parameters which must be given in advance are the minimum number of nodes per cluster N_{min} and density reachability parameter D_r.

There are k-means clustering (McQueen 1967) methods that are applied to WSN clustering (Ghiasi 2002), (Kanugo 2002), (Klein 2002). Also such distributed WSN clustering algorithms are proposed, where clusterhead selection is based on node connectivity (Basagni 1999), (Amis 2000), randomness (Bandyobadhyay & Coyle 2003) and received signal strength (Younis & Fahmy 2004). What makes a difference between them and the TASC clustering algorithm presented in Publication 3, is the way how TASC clusters a non-uniform node deployment with respect to WSN topology so that it targets to minimize the density variation in each cluster.

It is assumed in Publication 3 that each node can compute the distance to its one-hop neighbors and then share this information so that every node becomes aware of connections and distances in its 2-hop neighborhood. Based on this information, every node computes the shortest Euclidean paths between every pair of nodes in its 2-hop neighborhood by using Floyd-Warshall algorithm

(Floyd 1962), (Warshall 1962). Information about distances and shortest paths are then used to compute weight for every node. It follows from the principle of optimality (Bertsekas 2000), that the shortest Euclidean path between two nodes includes all shortest paths between all pairs of nodes that are located in that path. It is proven in Publication 3 that if the node weight is incremented by one every time a shortest path between two nodes passes it, the node that tends to be the midmost related to all shortest communication paths, gets the biggest weight. If some of the paths have even number of nodes, there can be several nodes with equal biggest weights in the middle. The weight computation is further improved by taking the computed Euclidean distances into account as follows: Assume nodes a, b and c are all located in a shortest Euclidean path between nodes i and j. Once a node b is located between nodes a and c in a path between nodes i and j, its weight is incremented by w_+, which is

$$w_+ = \frac{l_{ab}+l_{bc}}{l_{ij}}. \quad (5.21)$$

In (5.21), l_{ab} is the distance between nodes a and b, l_{bc} distance between nodes b and c, and l_{ij} the length of the whole Euclidean path from node i to node j. In other words, the weight of node b is incremented as much as the Euclidean length of the edges connected directly into it contributes to the length of the whole path between nodes i and j. Once the node weights are computed in this way, the midmost nodes are better indicated by the weights also in such a case where there is variation in the Euclidean distances between one hop neighbors.

In addition to weights, each node applies the density reachability criteria to further limit its 2-hop neighborhood into such subset of its 2-hop neighbors, where node density variations are similar or higher. This mechanism pulls the areas of high node densities towards the center of the cluster, as illustrated in Figure 6.

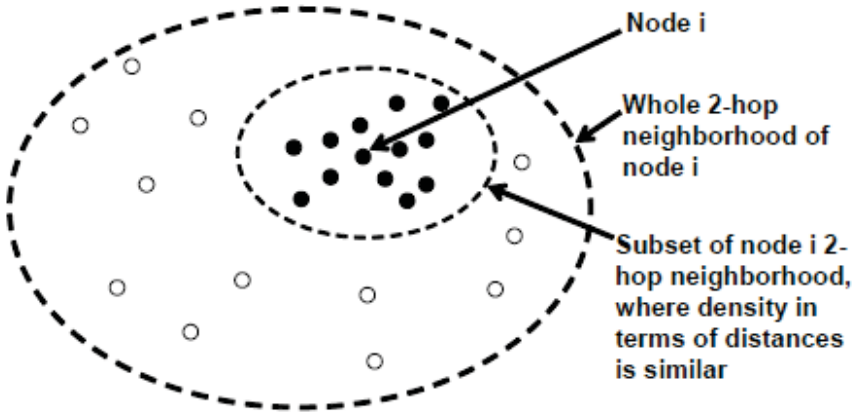


Figure 6. The effect of density reachability as illustrated in Publication 3: node i figures out such a subset of its 2-hop neighborhood, where density in terms of distances is similar or higher.

Density reachability is originally applied to cluster spatial data in the presence of obstacles (Ester 1996), (Zaane & Lee 2002). In Publication 3, each node defines its density range by using the density reachability parameter D_r such that for each node i, the density range r_i is the radius of the smallest node i centered disk, that covers $D_r - 1$ other nodes in the vicinity of node i. The density reachability is then defined so that node j is density reachable from node i, if it is located in node i 2-hop neighborhood and there exists an Euclidean path from i to j, where the length of every hop l satisfies the constraint $l \leq r_i$. By definition, D_r is a constant number of nodes given a priori. When the value of D_r is increased, more nodes and bigger density variations are included in the density reachable subset. The upper bound for the density reachable subset is the whole 2-hop neighborhood of node i. The effect of density reachability pulls the cluster heads towards most dense groups in the cluster, but the selection among density reachable nodes is still based on weights. An example of the density reachable subset selection is presented in Figure 7.

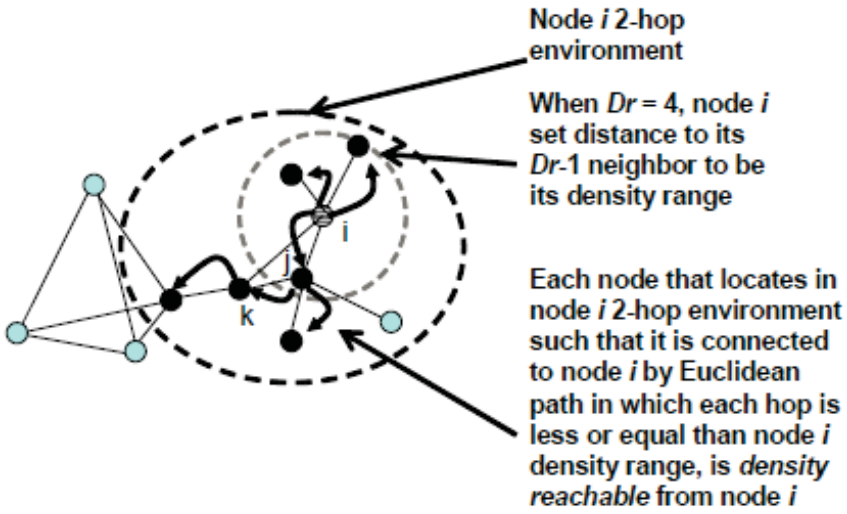


Figure 7. An example of the selection of density reachable subset of node 2-hop neighborhood in Publication 3.

The Topology Adaptive Spatial Clustering (TASC) algorithm presented in Publication 3 assumes that each node knows the shortest Euclidean paths between nodes in its two-hop neighborhood, required minimum cluster size in terms of number of nodes, and density reachability parameter D_r. Then the algorithm proceeds as follows:

1) *Each node computes its own weight based on the shortest Euclidean paths in its 2-hop neighborhood.*

2) *Each node broadcasts its weight to its 2-hop neighborhood and receives the weights of its 2-hop neighbors.*

3) *Each node nominates the node having biggest weight in the density reachable subset of its 2-hop neighbors and broadcasts the nominee to its 2-hop neighborhood.*

4) *Each node receives all nominees in its 2-hop neighborhood and elects the closest nominee to its leader.*

5) *Each node that ends up to the cluster where the total number of nodes is smaller than pre-specified minimum cluster size joins to the closest cluster, where the number of nodes exceeds the required minimum cluster size.*

To be able to compute its weight, each node must send two messages to its two-hop neighborhood and receive two messages from each of its two hop neighbors.

Cluster evaluation metrics in Publication 3 are based on Delaunay triangulation. It triangulates a finite set of points in a plane such that the standard deviations of the triangle angles are minimized by using 60 degrees as a mean. Thus, Delaunay triangulation gives an optimal planar subdivision in terms of spatial uniformity (De Loera 2010). Cluster area is defined as a sum of cluster Delaunay triangle areas. It is equal to the area of the polygon, which is defined by the outermost Delaunay triangle edges. Cluster density ($nodes/m^2$) is defined as a number of nodes in the cluster divided by the cluster area. Since the node density variation is indicated by the variation of Delaunay triangle edge lengths, relative node density variation is defined as Delaunay triangle edge length standard deviation in a cluster divided by average Delaunay triangle edge length in that same cluster. Cluster shape is characterized by a distance ratio, which is defined as minimum distance from polygon center point to the node in a polygon vertex divided by maximum distance from the polygon center point to the node in a polygon vertex. Three examples of cluster Delaunay triangulation and the use of the evaluation metrics are presented in Figure 8.

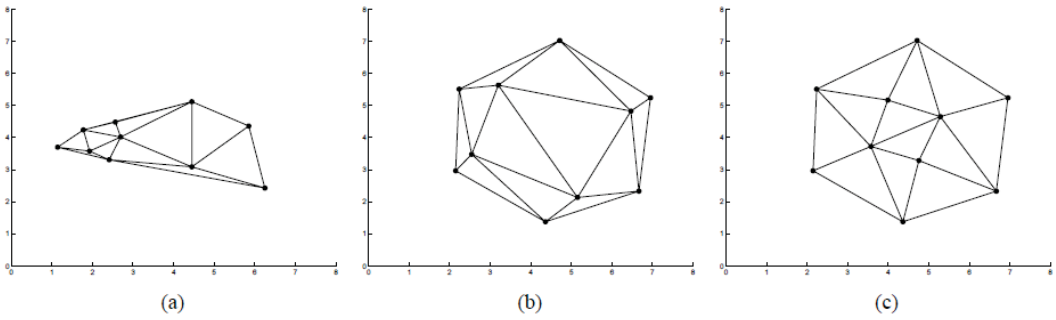


Figure 8. Three examples of cluster evaluation in Publication 3. Relative standard deviation of Delaunay triangle edge lengths is a) 0.559, b) 0.385 and c) 0.248. Cluster distance ratio is a) 0.462, b) 0.912 and c) 0.912. Since the node locations in a convex hull of clusters b) and c) are exactly the same, the value of the cluster distance ratio is also same in both of the clusters, but the smaller value of the relative standard deviation of Delaunay triangle edge lengths indicate that compared to b), nodes are more evenly spaced in cluster c).

TASC clustering algorithm properties are evaluated in Publication 3 by running a set of simulations, where 100 different node deployments in the area of 1000 x 1000 were considered. Node communication range was varied in steps of 50 from 200 to 400. Respective average node connectivity (number of neighbors per node) was 10.31, 15.35, 21.09, 27.32 and 33.80. Node distance measurement range was assumed to be equal with the communication range. Density reachability parameter was set to $D_r = 4$, that was also the required minimum cluster size.

Figure 9 presents the relative node density variation in clusters and its comparison to the relative node density variation in the respective network scenarios. The average relative node density variation computed from 6697 clusters was 0.5211 (52.11 %), which was clearly less than the average relative node density variation computed from the network scenarios that was 0.688 (68.8%). When studying the individual network scenarios, the relative node density variation in clusters was clearly less than the relative node density variation in the whole network in each case. These simulation results verified the main result that TASC can cluster a non-uniform WSN according to network topology into locally uniform clusters.

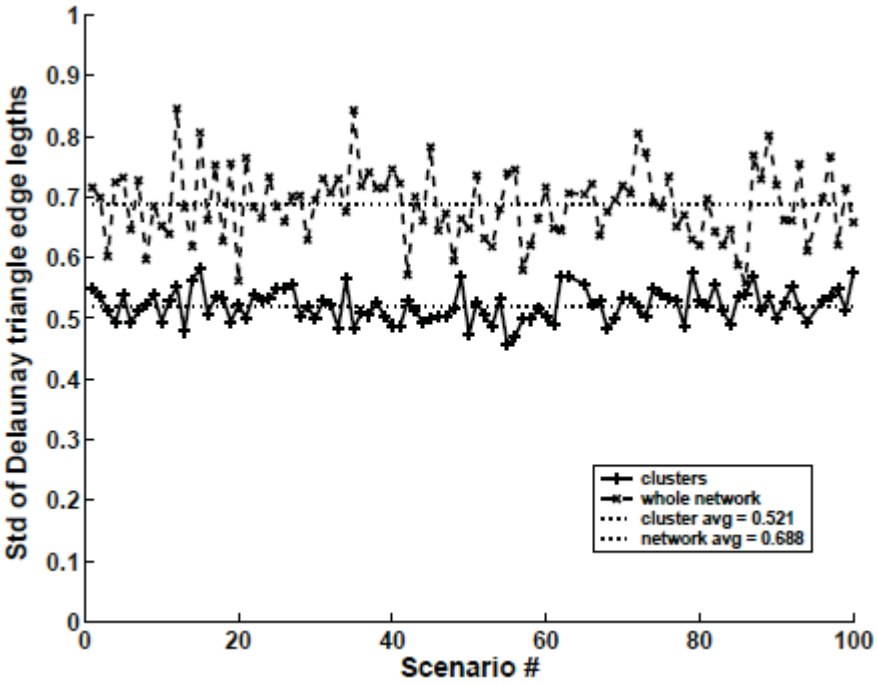


Figure 9. Relative node density variation in clusters and its comparison to the relative node density variation in each respective network scenario in Publication 3. Clusters are plotted with solid and the network scenarios with dashed line. Compared to the network scenarios, the

relative node density variation is smaller in the clusters computed by TASC in each case.

Figure 10 shows that the clustering outcome remained consistent when the node communication range was increased in the steps of 50 from 200 to 400 and the average number of neighbors per node was increasing respectively. The outcome consistency was sustained by the density reachability: since the density reachability parameter remained constant in $D_r = 4$, the density reachable subsets did not change much even though the node 2-hop neighborhoods increased.

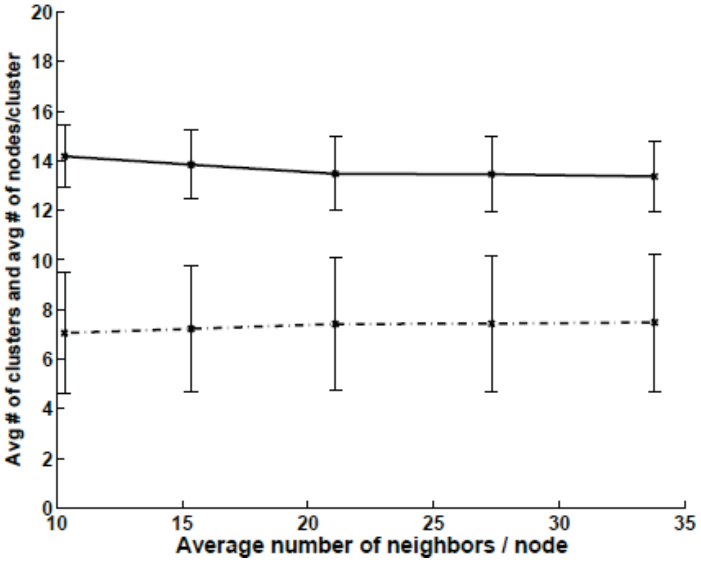


Figure 10. Average number of clusters (upper solid line) and average number of nodes per cluster (lower dashed line) when the node communication radius was increased in Publication 3 simulations. Standard deviation is indicated by the errorbars.

TASC clusters the WSN so that node density variations in each cluster are smaller than the density variations in the whole network. As a consequence, the cluster size can be smaller in dense deployment areas but become bigger in sparse deployment areas. This assumption is also verified by Publication 3 simulation results, as illustrated in Figure 11.

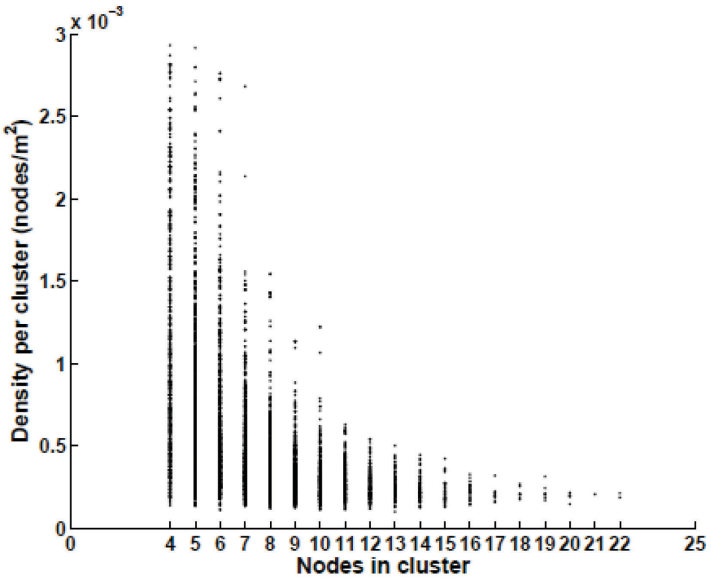


Figure 11. Since TASC clusters the network with respect to local density variations, the cluster size can be smaller in dense deployment areas but become bigger in sparse deployment areas. This is also indicated by Publication 3 simulation results.

In practice, the distance measurements (or estimates) are always noisy. TASC algorithm noise tolerance was evaluated in the simulations by corrupting the distance measurements by additive noise. Applied noise was white, Gaussian distributed and its standard deviation was entered as a percentage of the actual distance. Simulation results indicated that TASC was able to keep consistent clusters up to such a level, where noise standard deviation was 30% of the actual measured distance. When the noise level was increased further from 30%, the clustering result was weakening fast and at 50% noise level TASC failed to produce separate clusters in 10% of the simulated network scenarios. The effect of the measurement noise on the cluster size and cluster uniformity is presented in Figure 12.

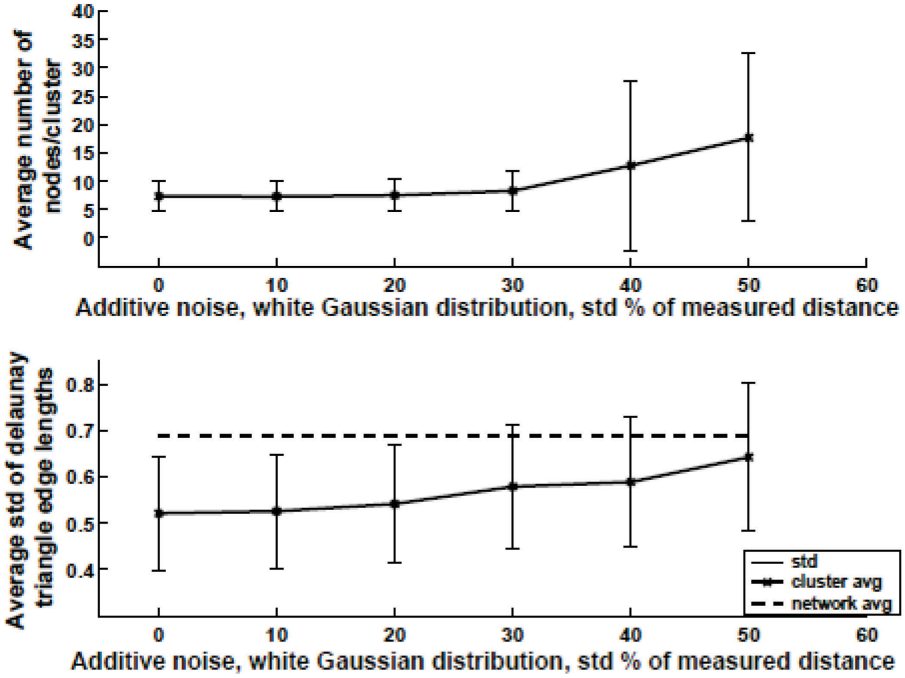


Figure 12. Noise effect to TASC clustering outcome in Publication 3. Upper plot presents the average number of nodes per cluster and lower average relative node density variation in the clusters. As indicated by the results, TASC tolerates noise well up to level where additive white Gaussian noise standard deviation is 30% of the actual measured distance.

The overall cluster distance ratio computed from 6697 clusters in Publication 3 simulation outcome was 0.4966 and the respective standard deviation 0.1619. These values are enough to show that TASC does not end up to flat node chain type of clusters.

6.1.3 Time Synchronization

Most of the WSN applications require common synchronized time. It is required for the synchronous execution of the different tasks in a distributed system, and for the correct chronological ordering of the measured information. Time synchronization is also a basic requirement in many communication and control protocols which are developed for WSNs. In practice, the synchronized time system is always synchronous up to certain accuracy. Every sensor node and actuator in the network is equipped with a microprocessor or microcontroller, and has a clock, which is a counter driven by a crystal oscillator. The time report

of such a clock differs from the ground truth time because of the error that consists of time offset, clock skew (frequency offset), frequency drift and wideband noise (Allan 1987), (Serpedin & Chaudhari 2009).

A network wide time synchronization requires the estimation of time offset and clock skew, and a messaging protocol to exchange the time reports between the sensor nodes (Sundararaman 2005), (Serpedin & Chaudhari 2009). The messaging protocols can be further divided to broadcast (Elson 2002), (Maróti 2004) and handshake based (Mills 2006), (Ganeriwal 2003), (Noh 2007). In theory, it would be possible to keep the time synchronization between different clocks in the required accuracy limits by changing the time reports frequently enough and compensating the time error based on that information. In real applications it would be very inefficient and energy consuming. As indicated by Pottie and Kaiser, the energy which is required to transmit one kilobit over 100 m is equivalent to the energy required to execute 3 million instructions (Pottie & Kaiser 2000). As a consequence, broadcast based time synchronization with computationally more complex algorithms suits better for WSNs.

Publication 4 presents a recursive clock skew estimation for WSNs using reference broadcasts. Developed algorithm takes into account the correlation between the time measurements and the communication uncertainty, which in WSN is mainly caused by lost packets and time varying transmission delays (Wu 2011). Developed model and algorithm is first analytically compared with the existing ones and then evaluated through experiments.

In the oscillator model used in Publication 4, the oscillator is modelled as nonlinear autonomous system $\dot{x}(t) = f(x(t))$, which has a non-trivial solution $x_s(t) = x_s(t+T)$, where T is the period. Consider perturbation

(5.22) $\dot{x}(t) = f\big(x(t)\big) + b(t).$

It has a solution

(5.23) $x_p(t) = x_s\big(t + \alpha(t)\big),$

where $\alpha(t)$ is defined as phase deviation. If (5.22) is a linear combination of uncorrelated white noise sources, $\alpha(t)$ is Gaussian so that

(5.24) $\alpha(t) \sim N(\mu, \sigma^2(t)).$

The mean of (5.24) is constant but the variance grows linearly in time (Demir 2000):

(5.25) $\sigma^2(t) = ct.$

In (5.25), c is the empirical oscillator constant. The phase deviation at different time moments is correlated, but jointly Gaussian. Its covariance is (Demir 2000)

(5.26) $E\{\alpha(t)\alpha(t+\tau)\} = \mu^2 + c \cdot min\{t, t+\tau\}.$

In (Allan 1987), the oscillator output is presented using the equation of sinusoidial voltage

(5.27) $V(t) = V_0 sin\left(2\pi f_0\left(t + \varphi_0 + \delta_f t + \frac{1}{2}D_f t^2 + \epsilon(t)\right)\right),$

where $f_0 = \frac{1}{T_0}$ is the nominal clock frequency, φ_0 is the time offset, δ_f is the clock skew, D_f is the frequency drift and $\epsilon(t)$ represents random deviations. The time error part $\varphi(t) = \varphi_0 + \delta_f(t) + \frac{1}{2}D_f t^2$ consists of short-term and long-term error factors. For short-term analysis such as time synchronization, it can be assumed that the frequency drift, which represents the long-term effects, such as crystal ageing, is $D_f = 0$. (Allan 1987)

In Publication 4, the phase deviation and the time error are related to each other by using (5.27) as a solution for (5.22). The solution has a period of

(5.28) $T = \frac{T_0}{1+\delta_f},$

where T is the period of the oscillator, which has a nominal period of T_0. Since $\alpha(t)$ is Gaussian, also

(5.29) $\epsilon(t) = \left(1 + \delta_f\right)\alpha(t) - \varphi_0$

is Gaussian so that

(5.29) $\epsilon(t) \sim N\left((1+\delta_f)\mu - \varphi_0, \left(1+\delta_f\right)^2\sigma^2(t)\right).$

A counter driven by the crystal oscillator is incremented once in a period. It can be assumed that in the beginning, the counter is set to zero. In jth transition, the time error is

(5.30) $e_j = t - jT.$

The timing error has a zero mean and its variance is increasing as indicated by (5.25):

(5.31) $e_j \sim N(0, cjT).$

The covariance of the time error between different transitions becomes

(5.32) $E\{e_i e_j\} = cTmin(i,j),\ i \neq j.$

These properties indicate that e_j is a Wiener process (Durrett 2000).

Assume oscillators 1 and 2, with nominal periods T_0^1 and T_0^2. These oscillators are both initialized to zero at $t = 0$. Oscillator 1 acts as a reference clock counter and counts exactly m low to high transitions of its oscillator output for each time record. If the transition counts at the first sampling instant are m and k respectively, the relation between count values according to (5.30) is

(5.33) $e_m^1 + mT^1 = e_k^2 + kT^2$

If (5.33) is solved with respect to m and by utilizing the result (5.28), it becomes

(5.34) $m = \frac{\left(1+\delta_f^1\right)T^2}{T_0^1}k + \frac{1}{T^1}\left(e_k^2 - e_m^1\right) = \frac{\left(1+\delta_f^1\right)T^2}{T_0^1}k + e_{m,k},$

where $e_{m,k}$ is the total error $e_{m,k} = \frac{1}{T^1}\left(e_k^2 - e_m^1\right)$.

Since e^1 and e^2 are independent Gaussian variables, the distribution of total error $e_{m,k}$ is

(5.35) $e_{m,k} \sim N\left(0, \frac{m(c^1+c^2)}{T^1}\right),$

where c^i is an empirical oscillator constant for oscillator i. In Publication 4, the periods of the oscillators are related to each other by defining a scalar parameter a, which is called the clock skew ratio:

(5.36) $a \triangleq \frac{T^2}{T_0^1}\left(1 + \delta_f^1\right).$

Let $\bar{m}$ be a vector of N reports of oscillator 1 and $\bar{k}$ be the vector consisting of the corresponding count values of oscillator 2. In Publication 4, the relation between progressive time records $\bar{m}$ and $\bar{k}$ is defined as reference-triggered progressive time relation model (RPT):

(5.37) $\bar{m} = [m, 2m, \dots, Nm]^{tr}$

$\bar{k} = [k_1, k_2, \dots, k_N]^{tr}$

$m = a\bar{k} + \bar{e}.$

In (5.37), $\bar{e}$ is the time error vector with components $e_i = e_{im,k_i}$, which are jointly Gaussian $\bar{e} \sim N(\bar{0}, S)$. It can be written so that

(5.38) $S = E\{ee^{tr}\} = \frac{cm}{T^1} UU^{tr}$,

where c is an experimental oscillator constant, and U is a lower triangular matrix

(5.39) $U = \begin{bmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{bmatrix}$.

In Publication 4, the unknown clock is estimated by using the (RPT) model (5.37). According to the model, the –log likelihood function of a becomes

(5.40) $L(a) = K + \frac{1}{2}(a\bar{k} - \bar{m})^{tr} S^{-1}(a\bar{k} - \bar{m})$,

where K is a constant which is independent of a. The optimum a which minimizes the –log likelihood function (5.40) is

(5.41) $\hat{a}_{ml}(N) = \frac{m\sum_{i=1}^{N-1}(k_{i+1} - k_i)}{k_1^2 + \sum_{i=1}^{N-1}(k_{i+1} - k_i)^2}$.

It can be seen from (5.41) that the ML estimate for the clock skew ratio is a function of the recorded successive time reports of the clock counters instead of the progressive output of them. With backward difference counters, the counter vectors become

(5.42) $\hat{k} = \left[k_1, \hat{k}_2, \ldots, \hat{k}_N\right]^{tr} = U^{-1}\bar{k}$

$\hat{m} = m[1, 1, \ldots, 1]^{tr} = m\bar{1} = U^{-1}\bar{m}$,

where $\hat{k}_j = k_j - k_{j-1}$. The model (5.37) can be presented equivalently by the reference-triggered increment time model (RIT):

(5.43) $\hat{m} = a\hat{k} + \hat{e}$,

where $\hat{e}$ is the noise vector consisting of elements $\hat{e}_j = e_j - e_{j-1}$, which are independent identically distributed random variables:

(5.44) $\hat{e}_j \sim N\left(0, \frac{cm}{T^1}\right)$.

In Publication 4, the associated ML estimate for clock skew ratio by using the RIT model (5.43), abbreviated as MLE-RIT, is presented as

(5.45) $\hat{a}_{ml}(N) = \frac{\hat{k}^{tr}\hat{m}}{\hat{k}^{tr}\hat{k}}$.

In RPT and RIT models it was assumed that each increment of the oscillator 1 reference counter 1 will perfectly trigger a simultaneous event to record the count of oscillator 2. In practice it is better to assume an external recording trigger, which triggers a simultaneous time recording in two counters. It is also assumed that the external trigger is phase-locked to the low-to-high transitions of oscillator 1. The backward difference vector of this model oscillator 1 output records become

(5.46) $\hat{m} = [m_1, \hat{m}_2, \dots, \hat{m}_N]^{tr}$.

The ratio of count increments between successive recording instants can be written as

(5.47) $\hat{h}_j = \frac{\hat{k}_j}{\hat{m}_j}$.

With these assumptions and model (5.43), an external-triggered increment time relation model (EIT) is defined in Publication 4:

(5.48) $\bar{1} = a\hat{h} + \hat{e}_m$.

In (5.48), $\hat{e}_m$ is Gaussian, but its components are not identically distributed:

(5.49) $\hat{e}_m \sim N\left(\bar{0}, \frac{c}{T^1} diag\left(\frac{\bar{1}}{\hat{m}_1}, \frac{\bar{1}}{\hat{m}_2}, \dots, \frac{\bar{1}}{\hat{m}_N}\right)\right)$.

The -log likelihood function of a according to EIT (5.48) is

(5.49) $L(a) = K + \frac{1}{2}\sum_{j=1}^{N} \frac{\hat{m}_j(1-a\hat{h}_j)^2}{c/T^1}$.

Similarly as in (5.40)-(5.41), the optimum a that minimizes (5.49) is

(5.50) $\hat{a}(N) = \frac{\sum_{i=1}^{N} \hat{k}_i}{\sum_{i=1}^{N} \frac{\hat{k}_i^2}{\hat{m}_i}} = \frac{\bar{1}^{tr}\hat{k}}{\hat{k}^{tr}\hat{h}}$.

In Publication 4, (5.50) is defined as minimum least square estimate for external-triggered increment time (MLE-EIT). Result (5.50) is further manipulated so that

(5.51) $\bar{1}^{tr}\hat{k} = \hat{a}(N)\left(\hat{k}^{tr}\hat{h}\right) = \hat{a}(N)\left(\hat{k}^{tr}\hat{h} + \hat{k}_{N+1}\hat{h}_{N+1} - \hat{k}_{N+1}\hat{h}_{N+1}\right)$.

Substituting (5.51) to the estimate of $N + 1$ measurements leads to recursive estimator:

(5.52) $\hat{a}(N+1) = \frac{\bar{1}^{tr}\hat{k} + \hat{k}_{N+1}}{\hat{k}^{tr}\hat{h} + \hat{k}_{N+1}\hat{h}_{N+1}} = \frac{\hat{a}(N)\left(\hat{k}^{tr}\hat{h} + \hat{k}_{N+1}\hat{h}_{N+1} - \hat{k}_{N+1}\hat{h}_{N+1}\right) + \hat{k}_{N+1}}{\hat{k}^{tr}\hat{h} + \hat{k}_{N+1}\hat{h}_{N+1}}$

$$\hat{a}(N+1) = \hat{a}(N) + \frac{\hat{k}_{N+1}}{\hat{k}^{tr}\hat{h}+\hat{k}_{N+1}\hat{h}_{N+1}}\left(\bar{1} - \hat{h}_{N+1}\hat{a}(N)\right).$$

Result (5.52) is one of the main results in Publication 4: recursive clock skew estimator for MLE-EIT.

Presented estimator (MLE-EIT) is analytically compared with three other types of clock skew estimators; MLE-RIT, constant estimate for RIT (CE-RIT) and least square estimate for RPT (LSE-RPT). This comparison was done by analyzing the average and the variance of the clock skew estimation error. The bias of the estimate is given by the average deviation of the actual parameter (Sorenson 1980). For unbiased estimators, the estimator efficiency is defined as the ratio of the Cramér-Rao lower bound (Cramér 1946), (Rao 1945) to the estimation error variance. If the variance is equal to the lower bound, the estimator is efficient (Sorenson 1980). Third criteria that was set for the estimator was consistency. If the estimate converges to the actual parameter, the estimator is consistent.

The quality analysis in Publication 4 shows that MLE-EIT and MLE-RIT both are unbiased, efficient and consistent. CE-RIT is biased and LSE-RPT is not efficient. Since MLE-EIT is closer to realistic time synchronization model for WSN, it is the best among the ones considered in the analysis.

In Publication 4, the result (5.50)-(5.52) is first derived without taking into account the deterministic time-varying transmission delay, which is caused by the wireless communication. Then the delay is included to the model by considering two independent clocks. The time reports of these at arbitrary time instant are

$$C^1(t) = \left(\bar{1} + \delta_f^1\right)(t + \alpha^1(t)) + \varphi_0^1 \tag{5.53}$$

$$C^2(t) = \left(\bar{1} + \delta_f^2\right)(t + \alpha^2(t)) + \varphi_0^2.$$

In (5.53) δ_f^1 and δ_f^2 are the clock skews, φ_0^1 and φ_0^2 time offsets, and $\alpha^1(t)$ and $\alpha^2(t)$ the phase deviations of clocks C^1 and C^2 respectively. If t is the ground truth time, the time reports of the clocks are related to each other such that

$$C^1(t) = \frac{\bar{1}+\delta_f^1}{\bar{1}+\delta_f^2}C^2(t) + \left(\bar{1} + \delta_f^1\right)\alpha^{1,2}(t) + \varphi_0^{1,2}\,, \tag{5.54}$$

where

$$\alpha^{1,2}(t) = \alpha^1(t) - \alpha^2(t) \tag{5.55}$$

$$\varphi_0^{1,2} = \varphi_0^1 - \frac{1+\delta_f^1}{1+\delta_f^2}\varphi_0^2.$$

If the clocks are driven by oscillators with nominal periods T_0^1 and T_0^2 respectively (5.54) can be written as

$$C^1(t) = a\frac{T_0^1}{T_0^2}C^2(t) + \frac{T_0^1}{T^1}\alpha^{1,2}(t) + \varphi_0^{1,2}. \quad (5.56)$$

In Publication 4 delay model, x_j is assumed to be a deterministic delay and ε_j additive white Gaussian jitter. Since the phase deviation has uncorrelated disjoint increments, it implies that $\hat{\alpha}(\tau) = \alpha^{1,2}(t+\tau) - \alpha^{1,2}(t)$ is uncorrelated with $\alpha^{1,2}(t)$ (Demir 2000). As a consequence, since the communication jitter is additive white Gaussian noise, C_j^i does not correlate with $\hat{C}_{j+1}^i$, and the incremental time model can be defined as

$$\hat{C}_j^1 = a\frac{T_0^1}{T_0^2}\hat{C}_j^2 + \frac{T_0^1}{T^1}\hat{\alpha}_j^{1,2} - \hat{\varepsilon}_j - \hat{x}_j. \quad (5.57)$$

If the deterministic communication delay is assumed to be constant between the successive broadcasts, the backward difference $\hat{x}_j = 0$. Then the incremental time relation model can be simplified as

$$1 \approx a\frac{T_0^1\hat{C}_j^2}{T_0^2\hat{C}_j^1} + \hat{\beta}_j, \quad (5.58)$$

where $\hat{\beta}_j$ has a normal distribution and

$$\hat{\beta}_j = \frac{1}{\hat{C}_j^1}\left(\frac{T_0^1}{T^1}\hat{\alpha}_j^{1,2} - \hat{\varepsilon}_j\right). \quad (5.59)$$

If all reference time broadcasts would be received without delays, the reference-triggered increment time model would be the best to use. In practice the time-varying deterministic transmission delays in wireless communication will change the spacing between received reference time reports. Then the external triggered time relation model models better the reference broadcasts based time synchronization in WSN. Similarly as in (5.48)-(5.52), the N sample MLE for this model is

$$\hat{a}(N) = \frac{T_0^1}{T_0^2}\cdot\frac{\sum_{j=1}^N \hat{C}_j^2}{\sum_{j=1}^N \hat{C}_j^2\left(\hat{C}_j^2/\hat{C}_j^1\right)}. \quad (5.60)$$

Model (5.60) can be updated recursively such that

$$\hat{a}(N+1) = \hat{a}(N) + \frac{T_0^2}{T_0^1}\cdot\frac{\hat{C}_{N+1}^2}{\sum_{j=1}^{N+1}\hat{C}_j^2\left(\hat{C}_j^2/\hat{C}_j^1\right)}\left(1 - \hat{a}(N)\frac{T_0^1\hat{C}_{N+1}^2}{T_0^2\hat{C}_{N+1}^1}\right). \quad (5.61)$$

The local time of any node can be translated to the reference time at any arbitrary time moment by utilizing the estimated clock skew ratio. For example, at an

arbitrary moment of time $t \geq t_{N-1}$, the time report of node C_2 can be translated into reference time t_{ref} by

(5.62) $t_{ref} = C_{N-1}^1 + \hat{a}(N-1)T_0^1\hat{k}_c.$

In (5.62), $\hat{k}_c$ is the current time moment t count by node C_2 counter, that was initialized at time moment $t = t_{N-1}$. Even though the time offset and the deterministic communication delay are not distinguishable, the total time offset can be estimated by using equation

(5.63) $C_j^1 - C_{j-1}^1 - \hat{a}(j-1)T_0^1\hat{k}_j = \varphi_0^{1,2} + x_j + \varepsilon_j.$

Since the time offset estimate focuses on narrow time window, the phase deviation increment can be ignored. Then the time offset becomes $o = \varphi_0^{1,2} + x_j$. In Publication 4, the random communication delay is assumed to be independent, identically distributed Gaussian sequence and the time offset estimate is

(5.64) $\hat{o}(N) = \frac{1}{N-1}\sum_{j=2}^{N}\left(C_j^1 - C_{j-1}^1 - \hat{a}(j-1)T_0^1\hat{k}_j\right).$

Two more features that are added to clock skew estimation model in Publication 4 to reduce estimator numerical sensitivity and to improve its consistency. Because the clock skew ratio is usually very close to T_0^1/T_0^2, the finite word processor numerical error sensitivity for the estimator (5.52) is high. Less sensitive estimator is presented as a modification of (5.52) such that

(5.65) $\hat{b}(N+1) = \hat{b}(N) + \frac{\hat{k}_{N+1}}{\hat{k}^T\hat{h}+\hat{k}_{N+1}\hat{h}_{N+1}}\left(r_{N+1} - \hat{h}_{N+1}\hat{b}(N)\right),$

where

(5.66) $\hat{b}(N) = \frac{\bar{1}}{T_0^2}(\hat{a}(N) - \bar{1})$

$r_N = \frac{\bar{1}}{T_0^2}\left(\bar{1} - \hat{h}_N\right).$

The deterministic transmission delays in wireless communication are time-varying (Maroti 2004). Because of the communication uncertainty, the packets can be received in different order than they are transmitted, and some packets might get lost. In broadcast based time synchronization this may cause a situation, where the received transmission carries a reference time, which is behind the local clock time. If this time difference is big enough, it may lead to incorrect update of the clock skew estimate, and an inconsistent estimator

behavior. To prevent the clock skew estimator for doing this, the received reference time data is checked for consistency by using criteria

(5.67) $\left|\hat{C}_j^1 - a\frac{T_0^1}{T_0^2}\hat{C}_j^2\right| \leq K_{max}.$

Received reference time data is used if the inequality (5.67) holds, but discarded if it does not hold. K_{max} is an empirical parameter which depends on the reference time broadcast period and sensor platform specific parameters. It must be adjusted according to time synchronization accuracy requirements and software implementation.

In Publication 4, the developed clock skew estimator for WSN time synchronization is evaluated by experimental implementation to Sensinode Nanoseries platform (Sensinode 2008), (FreeRTOS 2018). The local clock in the nodes was implemented as a 16-bit counter, that was configured to increment in every 4 μs, which was the period of one tick in the experiments. The communication stack was timestamping incoming and outgoing beacon frames in the same way, as described in (Maróti 2004). Further details of the experimental setup used for the experiments of Publication 4, are presented in (Mahmood & Jäntti 2009).

In the experiment setup, the counter tick of 4 μs was used in both nodes. Node 1 was sending a synchronization message to Node 2 once in second. The time error in synchronization was the time difference between the reference time in Node 1 and the local time in Node 2. It was corrected by using the estimated clock skew. Thus, the time error with the broadcast index j was

(5.68) $t_j^{err} = C_j^1 - \hat{a}C_j^2,$

where $\hat{a}$ was the clock skew estimate associated with the jth broadcast. It was found in the experiments that without clock skew compensation ($\hat{a} \equiv 1$) the time synchronization error was growing 2,4 $\mu s/s$. Three types of clock skew estimation methods were compared; MLE-EIT, LSE-RPT and LS regression estimation with a table length of 8. Synchronization time errors with these tree methods are presented in Figure 13.

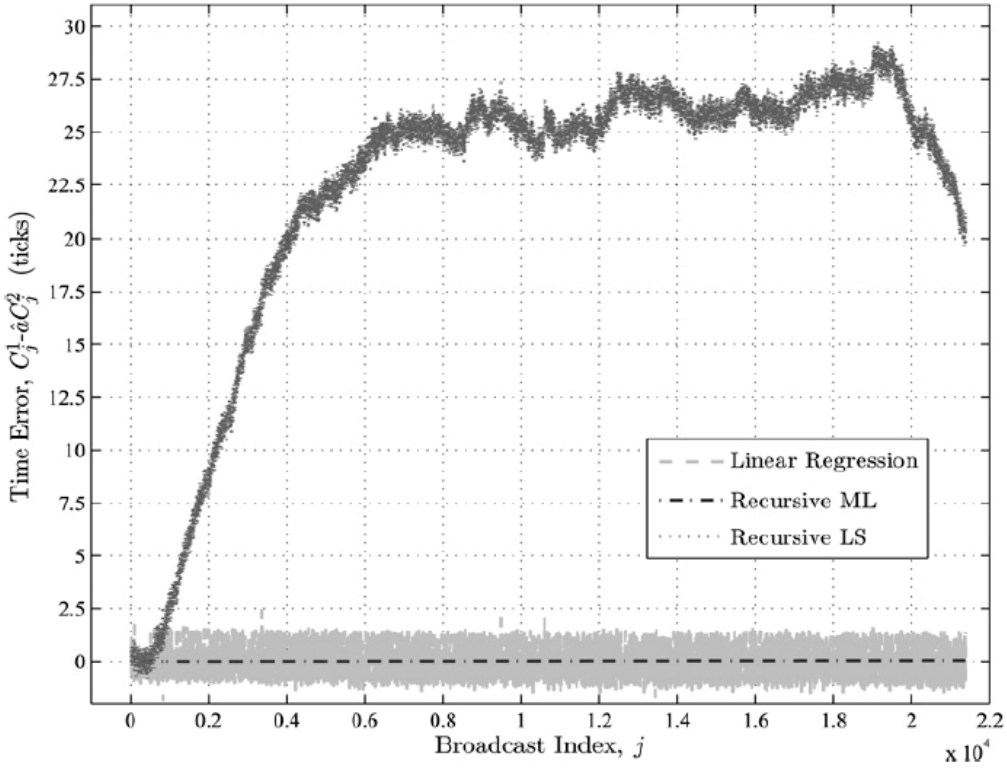


Figure 13. Publication 4 comparison of the time synchronization error, when the clock skew is estimated by using MLE-EIT (Recursive ML), LSE-RPT (Linear Regression) and LS regression (Recursive LS) methods. Estimation error is presented in ticks where one tick equals 4 μs.

The results in Figure 13 show the best performance of the developed MLE-EIT recursive clock skew estimator. The lower performance of the LS linear regression is mainly caused by high response time and estimation error variance.

The time error variation with different values of K_{max} (5.67) is presented in Figure 14. The clock skew ratio was estimated using recursive MLE-EIT and the error is presented in ticks in the same way as in Figure 13.

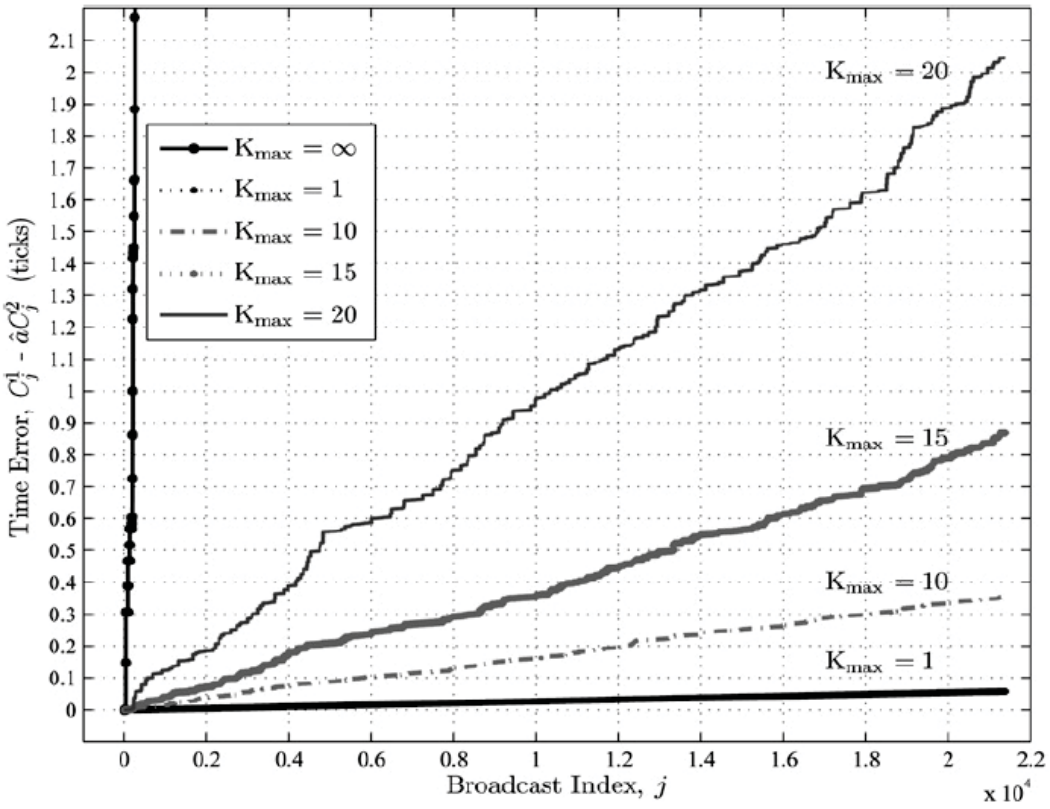


Figure 14. Time synchronization error in different values of K_{max} in Publication 4 experiments. Value $K_{max} = \infty$ equals the case when K_{max} is not applied at all.

Figure 14 indicates the importance of the careful adjustment of K_{max}. The higher the value, the greater the time synchronization error. The error increased remarkably faster, if the data consistency criteria (5.67) was not applied at all, which equals the case $K_{max} = \infty$ in Figure 14.

The numerical sensitivity of recursive MLE-EIT was evaluated by implementing the algorithm for 64-bit double precision and 32-bit single (float) precision. For 32-bit single precision it was also implemented as a numerically more stable modification (5.65)-(5.66). Results are presented in Figure 15. The time synchronization error increases remarkably, when the accuracy was changed from 64-bit double to 32-bit single precision, but the error with 32-bit single precision was staying closer to 64-bit precision, when the numerically more stable estimator modification was applied.

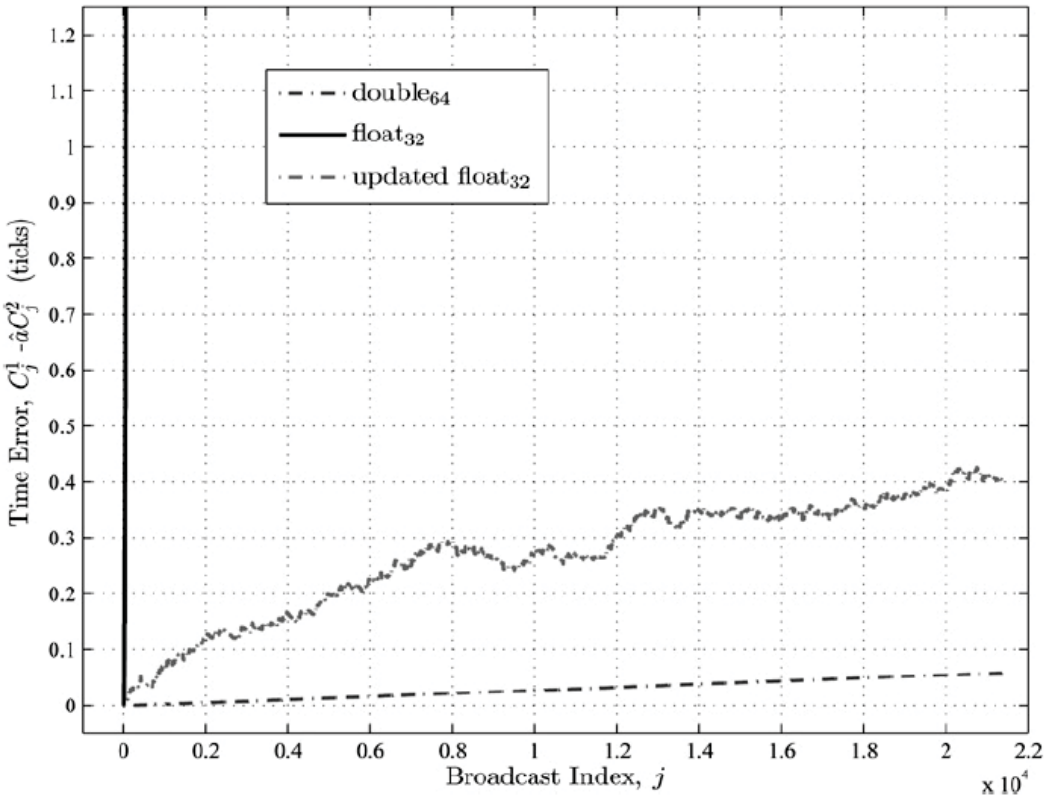


Figure 15. Experiment of numerical sensitivity of recursive MLE-EIT clock skew ratio estimator in Publication 4. The estimator is implemented for 64-bit double and 32-bit single (float) accuracy. 32-bit implementation was also done with numerically more stable modification of recursive MLE-EIT (5.65)-(5.66) labelled as updated float in the plot.

In the last experiment of Publication 4 the time synchronization error with different synchronization message broadcast periods was evaluated. When the broadcast period was varying between 60-300 s, average synchronization error varied between 0.176-0.961 ticks (one tick equal to 4 μs) and its standard deviation respectively between 0.961-3.954.

5.2 Security

Security must be taken into account in the early levels of the WSN system design far before the implementation. It is the way to ensure that the critical security factors are included to the system architecture. Then the overall security level is sufficient and the security functions can later be updated without a need to

partially or completely rebuild the whole system. As discussed in Chapter 4, it is mandatory to include the security functions to the system architecture in WirelessHART. In ISA100.11a it is optional.

Publication 5 presents a disassembly of privacy into its main components and points out security shortages in the current network deployments. The main outcome of that publication is a unified privacy preserving model. General architecture is presented in Figure 16. This model can be utilized to identify the main security factors that must be considered, and a set of practical implementation recommendations based on them.

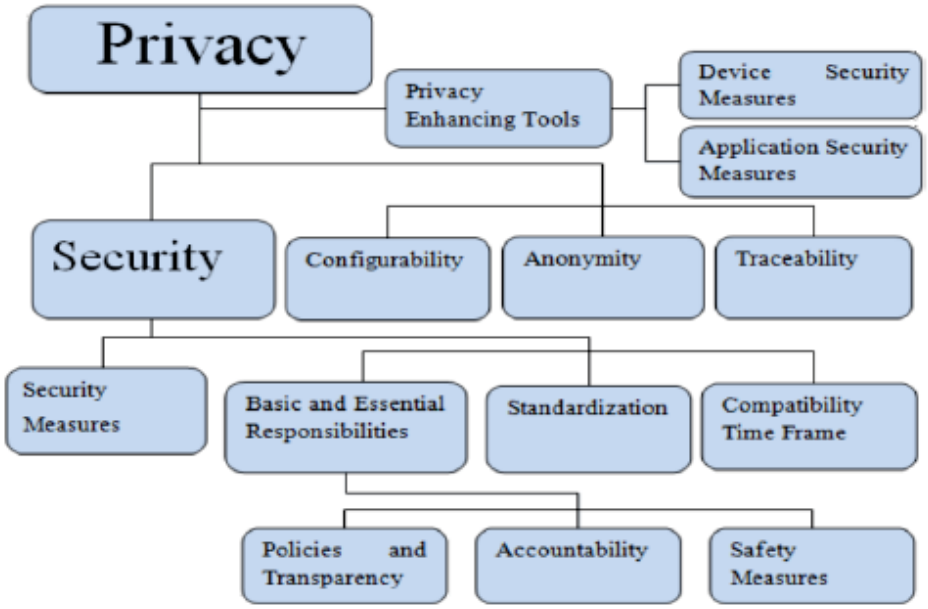


Figure 16. A general architecture of the unified information privacy preserving model presented in Publication 5.

5.3 Platform

Wireless sensor platform requirements and its planning and design process are discussed in Publication 6 and referred in report (Virrankoski 2012). After that, design of a wireless sensor platform based on the discussed requirements is presented in Publication 7.

5.3.1 Platform Planning and Design Process

Typically automation system consists of operatively independent units called modules. Hardware, software and mechanical architecture must all be taken into account in the modular structure. Production structures are based on modular (or platform) architecture in which the modules can also be called platforms. In modular production architecture one can freely combine different types of platforms. This property can be exploited in several production sectors where the physical compatibility is the only requirement.

The situation is much more challenging in intelligent embedded systems, such as wireless sensor networks. The associated challenges can also be seen as an opportunity, if one can systematically foresee the current and close future requirements and direct the product development accordingly. Several requirements must be fulfilled to make the modules compatible with certain generation products and to enable transition from current to next generation. These requirements can be fulfilled by applying platform structuring and platform-based planning.

In the context of technology, the term platform was earlier used for physical construction elements such as bridges, skeletons, metal plates etc. In 1990s platform became common term in electronics. In the beginning it was used to characterize software architectures and later other application areas of electronics. Software architecture platform is illustrated in Figure 17 (Jakobsson 1993).

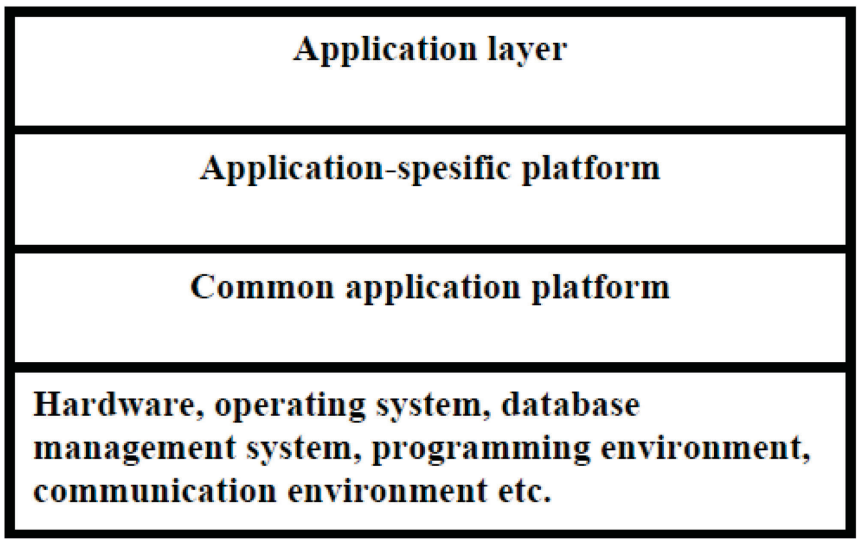


Figure 17. Software architecture platform (Jakobsson 1993).

There exists several platform definitions and characterizations (Meyer & Lehnerd 1997), (McGarth 1995), (Sanchez 1996). A product platform can be defined to be a common part in certain product group, family or generation. Different kinds of modifications as well as practices to improve process and information management can be built on top of the product platform. It can also be seen as a level up to which the platform is already brought to enable the production of customer-specific products as fast and efficiently as possible, as it is illustrated in Figure 18. Developing a platform up to product platform level is more expensive and time consuming than customization. Once the product platform exists, customization processes can be done fast, efficiently and parallel to different customers based on their particular needs (Saaranen & Keskinen 1998).

From company and company network point of view, a platform concept has a rich applicability. Some of the main advantages are as follows:

- Platform concept enables product management and provides tools for product lifetime support.
- By applying the platform concept, one can take advantage of both mass production and customization benefits. In other words the platform concept enables mass customization.
- In addition to products, the platform concept can also be applied to processes, services and to the whole business concepts.

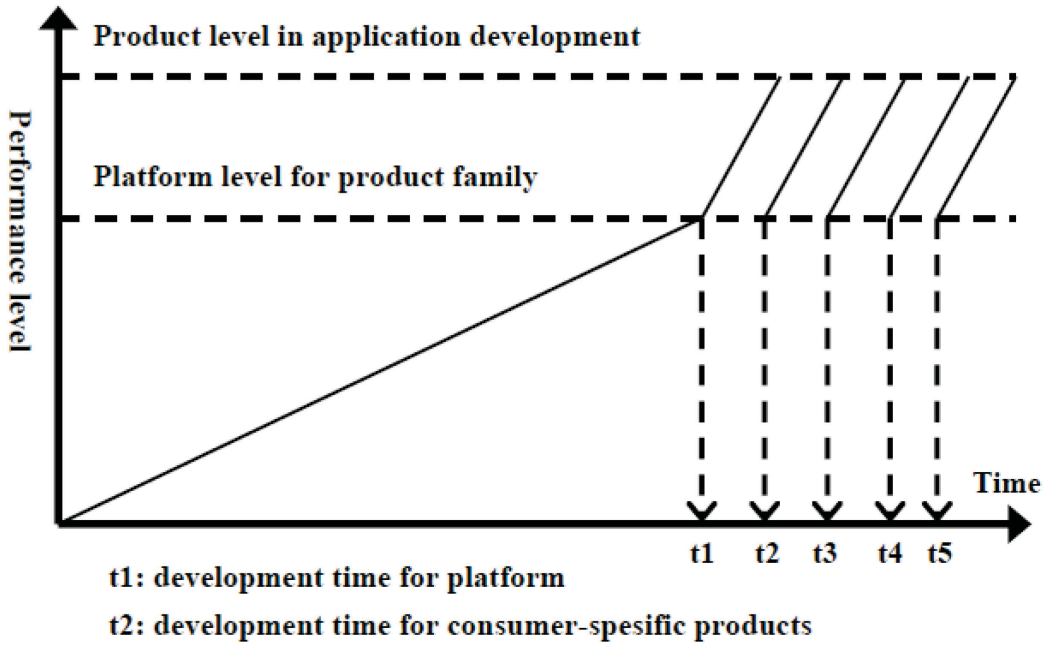


Figure 18. Product platform and customization processes. Developing a platform up to a product platform level is more expensive and time consuming than customization. Once the product platform exists, customization processes can be done fast, efficiently and parallel to different customers based on their particular needs (Saaranen & Keskinen 1998).

In the context of WSNs, Hill et al. (Hill 2004) presented a classification to four different platform classes. Even though their hierarchical classification provides one way to divide heterogeneous WSN to different classes, it does not meet the requirements of wireless automation. Less attention is paid to the data processing in a sensor network during data aggregation and too much is relied on TinyOS operating system (TinyOS 2018). TinyOS does not offer such a generic level that it would enable fast and easy modifications depending on the particular application needs. It is also suffering about weak documentation and uncoordinated application development. It does not offer compatibility with other embedded systems typically used as part of the automation system. The role of the IP-based networking is also underestimated. IP based networking with IPv4 or IPv6 is playing an important role when wireless and cabled parts of automation systems are integrated together. In wireless sensor and actuator networks, this support can be offered, for example, by 6LoWPAN compliant protocol stack (6 LoWPAN 2018).

Publication 6 presents a combination of product, application and technology platform development in the context of wireless automation. A technology platform consists of the available hardware components, communication standards (here in particular IEEE 802.15.4), radio technology, power sources, control algorithms and software. An application platform is made by utilizing the available technology platform and by developing its properties further and combining them in a novel way. Then applications can be made by utilizing the properties of the application platform and making the software and hardware configuration, which is needed in each particular application. Once the application platform is complete, a product platform will be designed. In that process the entire design of the application platform will be evaluated in terms of commercial and production aspects, such as component costs and quality, IPR rights, contractors etc. Some software or hardware components may be replaced by similar ones, which are better in terms of these criteria. Application platform and product platform development can happen in parallel, and in the best case the application platform can serve as a product platform without any major changes. A combined platform development process is presented in Figure 19.

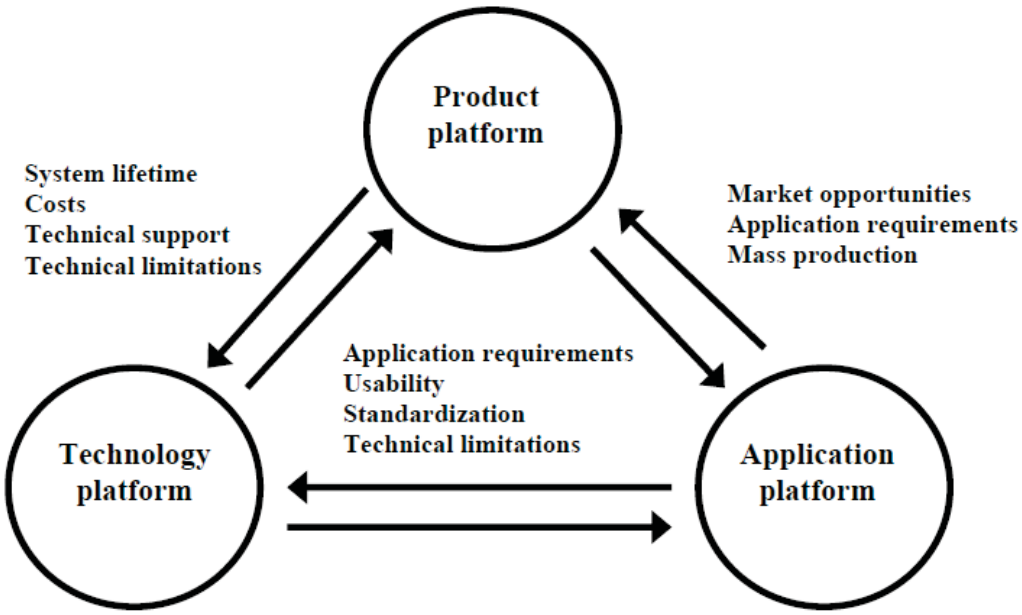


Figure 19. A combined platform development process as presented in Publication 6.

The target of the development is such an application platform (and then product platform), which allows a fast production of applications (and then products) as illustrated in Figure 18. In the case of a platform development for wireless

automation under IEEE 802.15.4 standard, the main objective is to develop a generic wireless sensor network architecture, which scales up for different numbers of sensor nodes, enables advanced networking and data processing and is easy to equip with different types of existing commercial sensors depending on the particular application needs. Advanced networking properties such as multi-hop support, automatic network configuration, re-configuration, time synchronization and advanced data processing should be available in the application platform so that they do not need to be tailored separately for different applications. In Publication 6 it was pointed out that the design of the generic wireless sensor network architecture requires research and development in the following main tracks:

1) Technology platform development
    - Integration and definition of generic, compatible hardware components (nodes, sensors, radios)
    - Software and driver libraries for the hardware (including protocol stacks)
    - Configuration tool to integrate the application design
2) Protocol development
    - A flexible communication and networking protocol stack for the generic sensor network platform
    - Implementing the useful and energy-efficient parts of the time synchronized mesh protocol used, for example, in WirelessHART
    - Better support for asynchronous sampling, data compression, data fusion and control algorithms as required in the technical requirements of the application platforms
3) System validation and testing
    - Development of five applications (Virrankoski 2012)
    - Interaction between application platform and technology platform development
4) Evaluation of the commercialization capabilities

- Marketing opportunities of the developed technology platform and application platforms
- Technical requirements set by the commercialization process

Collecting information about the requirements of several different types of applications provide data for the application platform and application development. Comparing the collected requirements of different applications with each other, one can find a subset of such requirements, which are common to most, ideally all, applications. That subset defines the level up to which the common application and product platform can be developed (see Figure 18). All the development after that point will be application specific. Once the technology development produces updates for the technology platform, this process needs to be repeated to produce a new update of the application and product platforms.

Publication 6 is related to GENSEN (Generic Sensor Network Architecture for Wireless Automation) research project, which is documented in (Virrankoski 2012). The sensor network and sensor platform design process was the starting point of the project, and some further aspects are also discussed in the report.

The co-design of software and hardware components is a mandatory approach in the case of embedded systems such as WSNs. Every organization can have their unique features in the design process, but in general the processes follow the pattern presented in Figure 20 (Virrankoski 2012).

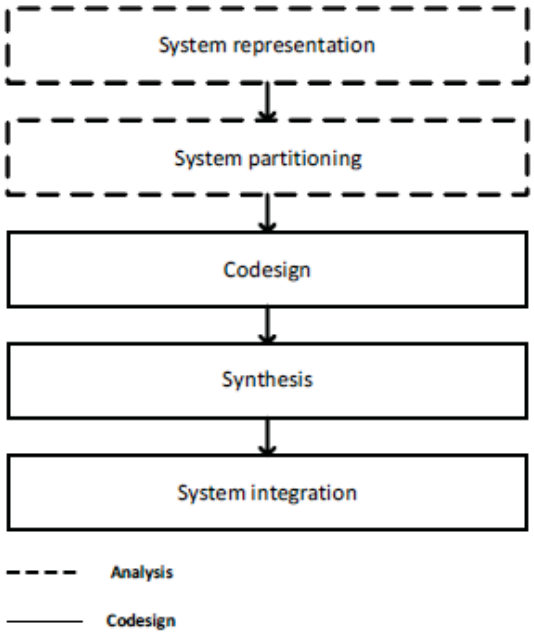


Figure 20. General pattern of an embedded system design process (Virrankoski 2012).

More specific model describing the design process of WSNs is presented in Figure 21. It can be applied to both case specific design projects and also to a platform project as a part of industrial product development process (Virrankoski 2012).

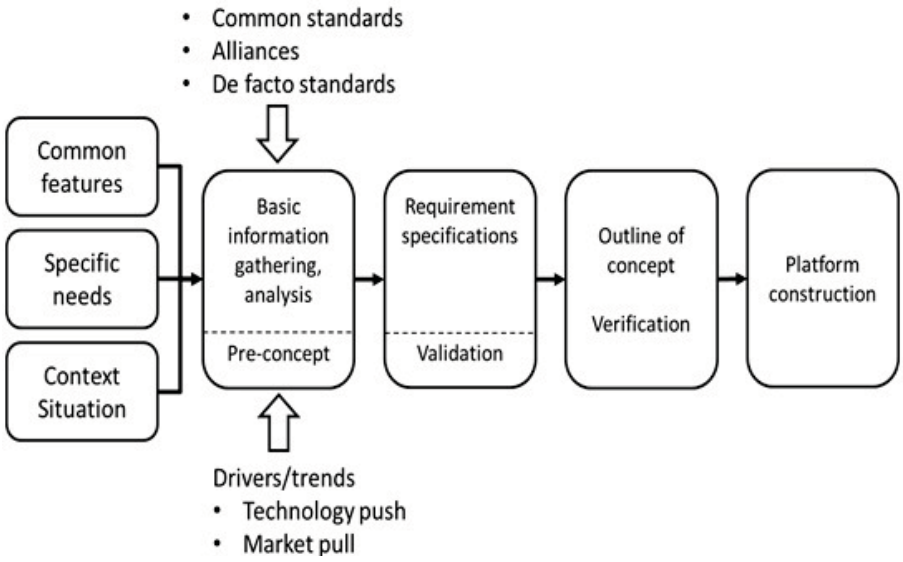


Figure 21. The design process of WSN application platform (Virrankoski 2012).

Common target features of industrial WSNs completed with the particular application needs in the targeted customer area provide a starting point for the WSN development process. In the industrial automation, typical critical requirements are security and communication reliability, system ability to fill the real-time requirements, system performance in terms of sampling rate and data transmission capability, sensor node power supply and the length of the reliable communication range. In addition to pre-mentioned requirements, a feature which is increasingly interesting is the amount of intelligence that can be added to the network. Since the amount of the measurement data is continuously increasing, it would be reasonable to pre-process and compress the data in the network so that only that part of information, which is needed for monitoring and control, is transmitted to the upper levels. Local intelligence in the wireless actuators and sensor nodes makes it possible to create local control loops and local network responses. These functionalities enable a higher degree of distribution in automation systems as before. New features can be applied only if the basic performance and reliability requirements are still fulfilled. (Virrankoski 2012)

As part of the GENSEN project, a set of interviews for the market and commercialization analysis was made in Spring and Summer 2011. Altogether 18 experts from 12 different companies, universities and universities of applied sciences were interviewed. Figure 22, which presents the way how the business potential consists of different sub-entities, was shown to each interviewed person as a part of the preparation material to direct the discussion to the main issues (Virrankoski 2012).

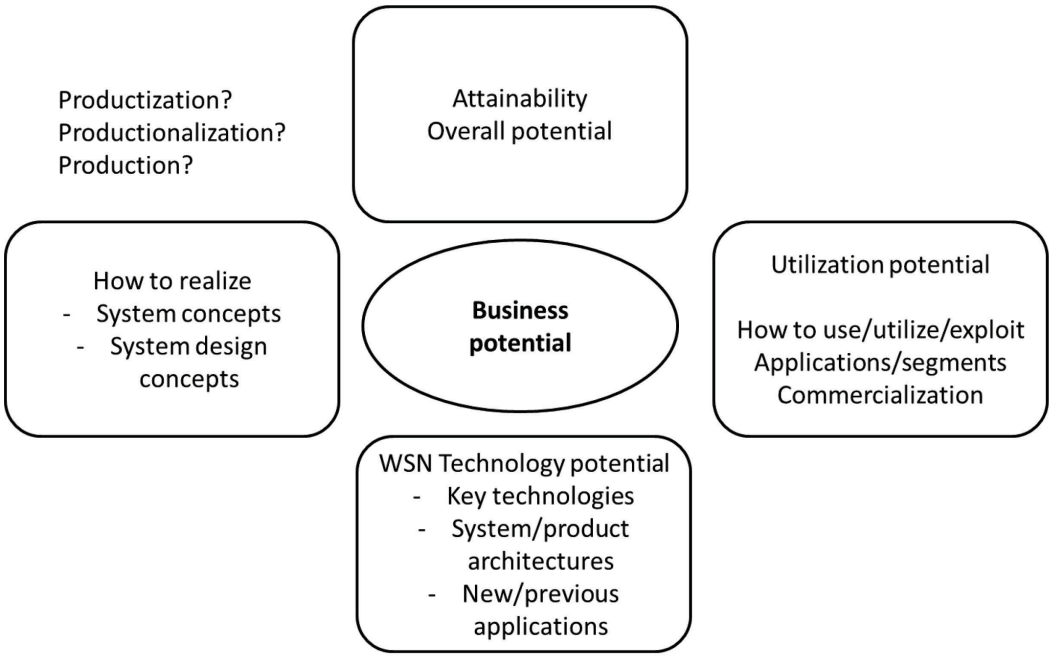


Figure 22. The sub-entities of WSN business potential. This figure was used as a starting point in the interviews on 2011 (Virrankoski 2012).

As indicated in the uppermost block of Figure 22, the results of the analysis and synthesis will provide the answers to the strategic questions regarding how much can be achieved and managed and what is the total business potential. Left and right blocks indicate how the total potential can be utilized. Left block stands for the product selection and system design of the producing and marketing company. The process is divided into several phases starting from application description to system planning and design and finally ending up in the commercialization. Right block stands for the recognition of the utilization potential. In big shipments, the services and support is today a remarkable part of the business. Its share of the annual exchange can be around 30-40% or even more (Virrankoski 2012).

In each interview, the current state of the art, the factors with remarkable impact and views about the future were all discussed. Common view was that the general development phase was in the area of rapid growth in the hype curve. That area includes also some extra enthusiasm with some unrealistic expectations. According to a well-known presentation by Geoffrey Moore et al. (Moore 1998), the area of wireless automation is currently experiencing a rapid growth, which also includes some turbulence. In such a phase some old companies are typically dying and some new ones are born. The ones which can keep their business

running in the competition will produce such a growth of the sales which will lead to breakthroughs of the new technology. One common view indicated by several interviews was that in terms of technological maturity, the RFID technology is several years ahead of WSN technology. Their hybrid solutions were expected to exist in the nearby future (Virrankoski 2012).

In the interviews, it was also expected that the standardization will play the most remarkable role in the development of the whole WSN technology sector. ISA and IEEE standards were considered equally strong in challenging real-time industrial applications. The selection between them, and possibly some other options, was expected to be defined by each particular application environment with its customer requirements. Some traditional solutions were also expected to be replaced by wireless field buses. It was assumed that most probably different coalitions and licensing strategies will solve the competition between de facto standards (Virrankoski 2012).

Additional factors mentioned in the interviews were the availability of the key components, algorithm research and development work, and in some cases also the easiness of the network components production and installation process. It was notified that different business opportunities were not clear yet, but the increase of subcontracting based product development and services was expected. More companies focusing on system integration was also expected to exist in the nearby future (Virrankoski 2012).

Crucial requirements for a WSN specification model in the area of wireless automation pointed out in the interviews (Virrankoski 2012) are the following ones:

- Application area and operation environment
- Network mobility type: static (non-mobile) or mobile
- System performance requirements: sample rate, data transmission capability, real-time performance capability, data transmission reliability
- Sensor node power supply and power consumption
- What are the other functions and characteristics that need to be considered?
- Management of product portfolio: reusability, scalability and configurability

- Life cycle management, maintenance and expendability
- Overall system safety and reliability
- How to interface WSN with other parts of the automation system?
- Installation
- Is there existing design tools or need to develop new ones?
- What are the requirements rising from the production and distribution logistics?
- How to make a product complete for marketing?

The interviewed assumed that certain application areas of WSNs could be a potential niche-area for some companies in Finland. So far the recognition and utilization of such relatively narrow market slots have created some success stories. However, during the time of the interviews the wider views of the future of wireless automation were still shadowed by technological shortsightedness (Virrankoski 2012).

5.3.2 Developed Sensor Platform

Based on the generic sensor platform idea presented in Publication 6 and the information about industry interests collected as a part of the GENSEN project (Virrankoski 2012), a generic sensor platform, the UWASA Node, was designed and presented first time in Publication 7.

Both at software and hardware level, the UWASA Node has a generic architecture, which responds to the design targets presented in Publication 6 and illustrated in Figure 18. Modular and stackable architecture enables the node to operate in many different roles from low power consumption message relaying node to an actuator, which is equipped with several sensors, large memory and high computation capacity. Node hardware architecture with different configuration options is presented in Figure 23.

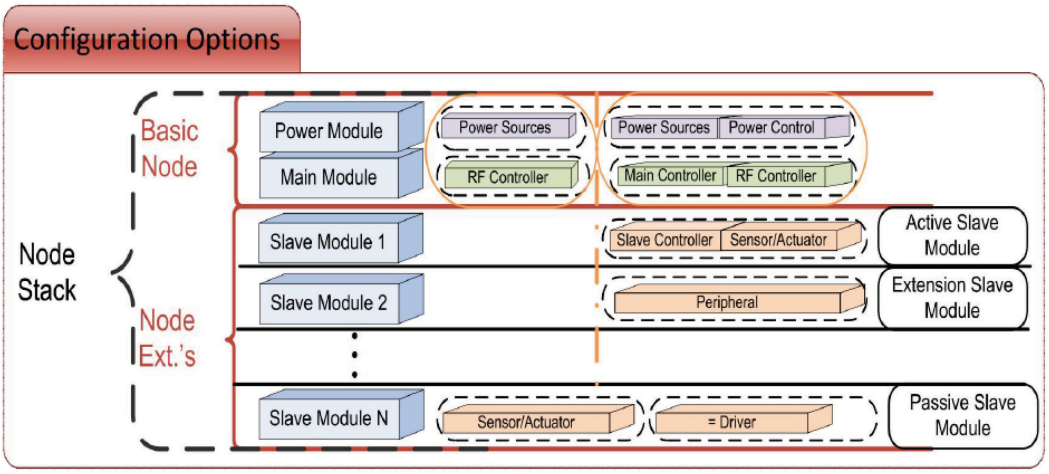


Figure 23. The UWASA Node hardware architecture and its configuration options presented in Publication 7.

The basic part of the node consists of the Main Module and Power Module. The Main Module has node's main controller, RF controller and wireless transceiver (radio). The Power Module has power sources and power regulation features. It is responsible to provide power related interfaces to all modules of the node. As many slave modules as needed can be added. There are three slave module types: Active Slave Module, Passive Slave Module and Extension Slave Module. Active Slave Module is equipped with its own microcontroller, Passive Slave Module is not. Both of them can carry one or several sensors. Extension Slave Module operates as an extension of the Main Module. It can be equipped, for example, with extension memory or additional radio. The application dependent hardware development can be done by using suitable slave boards. A picture of the UWASA Node is presented in Figure 24.

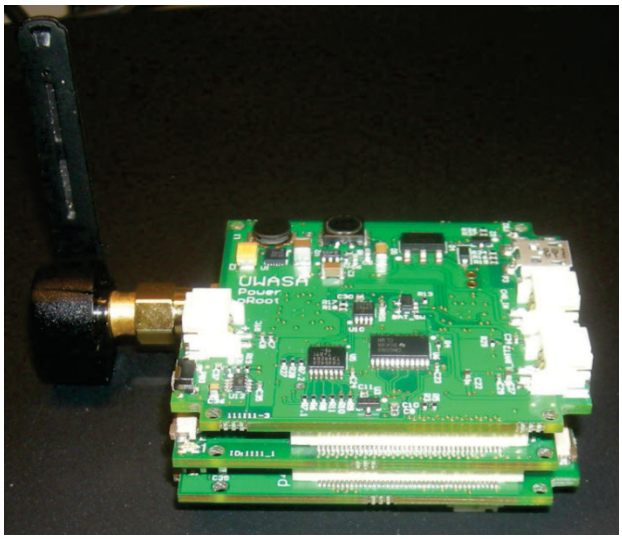


Figure 24. The UWASA Node.

There are four comprehensive properties that are considered in the UWASA Node design to make it suitable for wireless automation applications: 1) power control, 2) time synchronization, 3) complicated software demand and 4) application development. For power control purposes, the main controller can be turned off in the low-power mode, and the RF controller used for the mandatory operations. Moreover, the Power Module provides control interfaces and the unified software architecture enables the power state control of each processor in the Main Module and Slave Modules. The time synchronization must be managed both between different components in the node and between different nodes in the network. Time between all processors in the node can be synchronized by using a heart-beat signal, which is provided either by the main controller or by the RF controller. Synchronization between different nodes in the network is done by using the network time synchronization protocol, as discussed in Publication 4. The complicated software demand rises from the issue that multiple different processors require different software to be developed. The problem is solved by using a hardware abstraction middleware, which provides a level where most of the application software development can be done over the provided application interfaces. Finally, to enable a flexible and hardware-less software development environment, a development board was designed for the UWASA Node.

The design of the UWASA Node enables to power up the node by using a rechargeable battery or an external power source. The external power source can be either power harvested from the environment or connection to external power

supply such as electric grid. The Power Module is equipped with a dynamic power path management, which can efficiently utilize simultaneously two suppliers. There are 10 independent voltage regulators, from which two have fixed outputs to supply the Main Module and eight have adjustable voltage levels for the support of the Slave Modules. The Power Module contains also a battery fuel gauge. It is accessible by the main controller and can be used to monitor the battery level and the power demand of different applications.

The Main Module is responsible of wireless communication, network level time synchronization, performing data processing and decision making, managing in-node data exchange and in-node power mode. The main features of the UWASA Node Main Module, as summarized in Publication 7, are the following ones:

1. Main controller and RF controller are both programmable microcontrollers; RF controller has 8051 based, 8-bit processor that runs at either 16 MHz or 32 MHz, whereas main controller has ARM7TDMI-S 32-bit processor that can run at up to 72 MHz.

2. RF Controller has integrated IEEE802.15.4 MAC and integrated positioning engine. Main controller has integrated Ethernet MAC and USB2.0 Full Speed device as well as many standard serial interfaces.

3. RF controller can switch off the power of the main controller, when excessive process power is not needed. The main controller can control enable states and voltage level of power sources through the interfaces provided by the power module.

4. Individual peripherals of main controller and RF front end of RF controller can be disabled.

5. Operating frequencies of both RF controller and main controller can be adjusted.

6. Intra-node time synchronization is solved by the means of heart-beat signaling provided by either RF controller or main controller. Network level time synchronization is ensured by the RF controller.

The main features of the UWASA Node Power Module, as summarized in Publication 7, are the following ones:

1. The UWASA Node supports LiIon-LiPo battery connection. The Power Module is designed to operate with the safety limits of LiIon-LiPo batteries.

2. The Power Module contains integrated LiIon-LiPo battery charger, which allows online charging of the battery, when the other power source is connected to power supply.

3. Two independent supply options are provided through Power Module; the whole node supports dynamic power path management when the node is supplied via battery and other source (e.g. solar panel, vibration based power generator, USB connection, etc.).

4. Instantaneous (power drain)/charge status of the battery can be monitored.

5. The voltage level of the main power line of the node can be monitored.

6. 10 independent power regulators are provided on the Power Module. 8 of them with adjustable output voltage levels; where 4 of them are switching regulators and other 4 being linear regulators. Those regulators should be used to supply slave modules, to allow main controller to turn on/off the whole slave module hardware to minimize quiescent currents.

5.4 Applications

5.4.1 Greenhouse Monitoring

In the past, there was just one cabled box equipped with sensors in the middle of a greenhouse to measure the environmental parameters. Greenhouse heating, lighting and ventilation systems were also so simple, that it was enough to set them manually or by using a simple control system. In modern greenhouses the automation system requirements are different. The size of the greenhouse is typically so big that there can be remarkable variations in the greenhouse climate. Heating, lighting, ventilation and other greenhouse condition control systems are also built in such a way that advanced control operations and a separate adjustment of different parts of the greenhouse are possible. However, to be able to do it in proper way, more data is also needed.

Light, temperature, humidity and carbon dioxide (CO_2) are all important for the plants, and different plant species have different needs. In the northern countries the daylight period is so short during half of the year and temperature so low, that heating and artificial lighting are needed in the greenhouses. Since the CO_2 is important for the plants in the photosynthesis process, the growing conditions can be further improved by using some extra CO_2. The need of extra CO_2 enables

us to use CO2, which is collected from some industrial process and tie it to the biomass in the greenhouse instead of just emitting it to the atmosphere (Timmerman & Kamp 2003).

Publication 8 presents a design and experimental deployment of WSN for greenhouse monitoring. Applied sensor platform is Sensinode Microseries (Sensinode 2007), which communication protocol supports 6LoWPAN (6LoWPAN 2018). Software and hardware implementation is made for three different sensors: temperature & humidity, luminosity and CO2 level. The hardware implementation required two different sensor boards to be attached to the WSN platform, because the CO2 sensor had higher use voltage and higher saturation time than the other sensors. Sensor platform and a sensor board equipped with temperature & humidity sensor and luminosity sensor are presented in Figure 25.

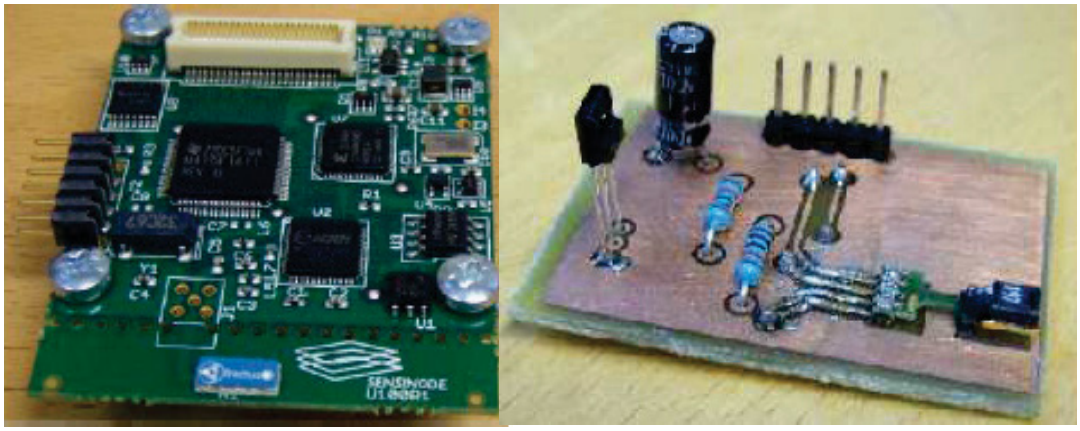


Figure 25. Sensinode Micro.2420 U 100 platform (left) and a sensor board equipped with SHT75 relative temperature and humidity sensor and TAOS TSL262R luminosity sensor.

Experimental deployment was done in Marten's Greenhouse Research Foundation's tomato greenhouse in Western Finland. The size of the greenhouse was 18 x 80 meters and the height of the tomato foliage was around four meters. The WSN deployment consisted of five nodes, four of them acting as a measuring nodes and the fifth one as a gateway to the PC. In the preliminary experiments it was observed that the tomatoes and the tomato foliage was attenuating the radio signal so that the reliable communication range in the greenhouse was only one third of the observed reliable range in the open space. As a consequence, the maximum distance between the nodes was limited to 15 meters. Communication architecture was a simple star topology to make the transmission scheduling easy

and to avoid collisions. The gateway node was acting as a coordinator and polled periodically the measuring nodes, which were only able to transmit when polled. Periodical sleep and wake periods were applied so that each node turned on its radio for 15 seconds and then turned off its radio and went back to sleep mode for 255 seconds (4 min 15 s). Figure 26 presents two of the nodes deployed to the tomato foliage and protected from the humidity.

Figure 26. Sensor nodes (inside red squares) deployed to the Martens tomato greenhouse during the three hours experiment described in Publication 8.

During the experiment, each node measured temperature, humidity and luminosity once in four minutes continuously on three hours. During that time the coordinator node sent 200 transmission requests and each measuring node responded 50 times. Observed packet loss was 5%, which was tolerable level in this particular application. The fact that CO2 sensor had high power consumption and long saturation time made it tricky to use it simultaneously with the other sensors. As a consequence, a separate set of experiments was made with it. Experimental results indicated that the WSN is a feasible solution for greenhouse monitoring. Existing microclimate layers and other condition differences within the greenhouse can be followed by using a WSN that covers the different parts of it. Measured data can be utilized in the greenhouse condition control to adjust the growth conditions as close to optimal as possible. Temperature readings during the three hour experiment are presented in Figure 27.

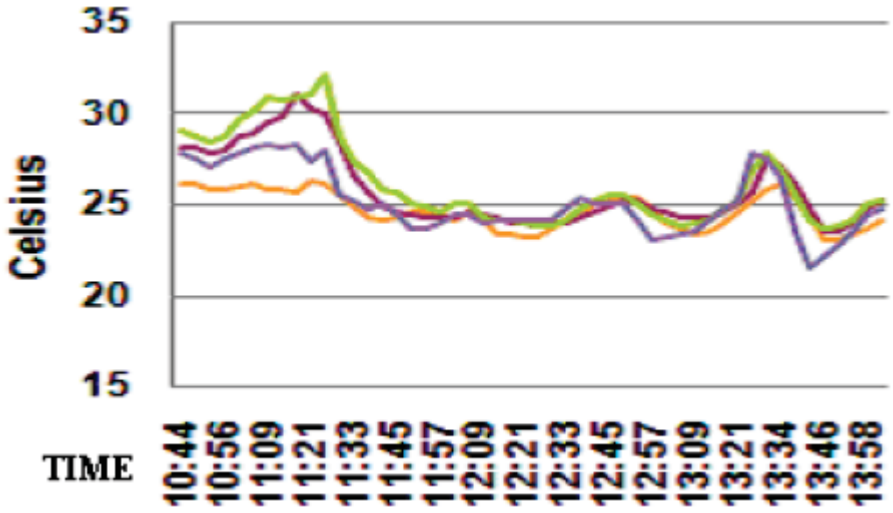


Figure 27. Temperature readings of four sensor nodes in Martens tomato greenhouse during the three hours experiment described in Publication 8.

5.4.2 Situational Awareness

Situational awareness has been one of the key interest areas in WSN development as long as the technology has existed. A lot of data from different types of environments can be extracted by various types of sensors. Since the sensor nodes have small size and they are relatively cheap, huge amounts of them can be rapidly deployed to cover the area of interest. Some of the sensors can also be carried by the people operating in the area or by the vehicles or autonomously moving systems, such as robots and drones. Measured sensor data can be combined with other sources of information to compute and visualize a model of the complete situation in the observed area. This model must be updated continuously based on new observations to make it a real-time situation model.

Publications 9-11 focus on situation awareness in urban and indoor environment. Building interior is a challenging environment since it is divided to rooms and corridors. Walls block the line of sight and may either attenuate or completely block the radio signal. Satellite positioning is not available inside the building despite coincidental signal near the windows. There are many cases, where the situation awareness about the building interior is a critical factor for a successful mission. These missions can be rescue operations in the case of fire or earthquake, or reconnaissance and combat operations performed by police or military forces. In each of these cases it is crucial to know how many people are in

the building and where they are. Also mapping of the building interior and any further information about the activities of the people in the building are important to know.

Wireless sensor systems enable a reconnaissance of the situation in the building before own forces are sent there. Reconnaissance information can be utilized to plan a successful operation and to save human lives. As a part of the reconnaissance, some persons in the building can also be identified based on the sensor data.

Publication 9 presents an indoor situation modeling system, which consists of several subsystems and produces a real-time situation model from the building interior based on the sensor data and other sources of information. The overall system architecture is presented in Figure 28.

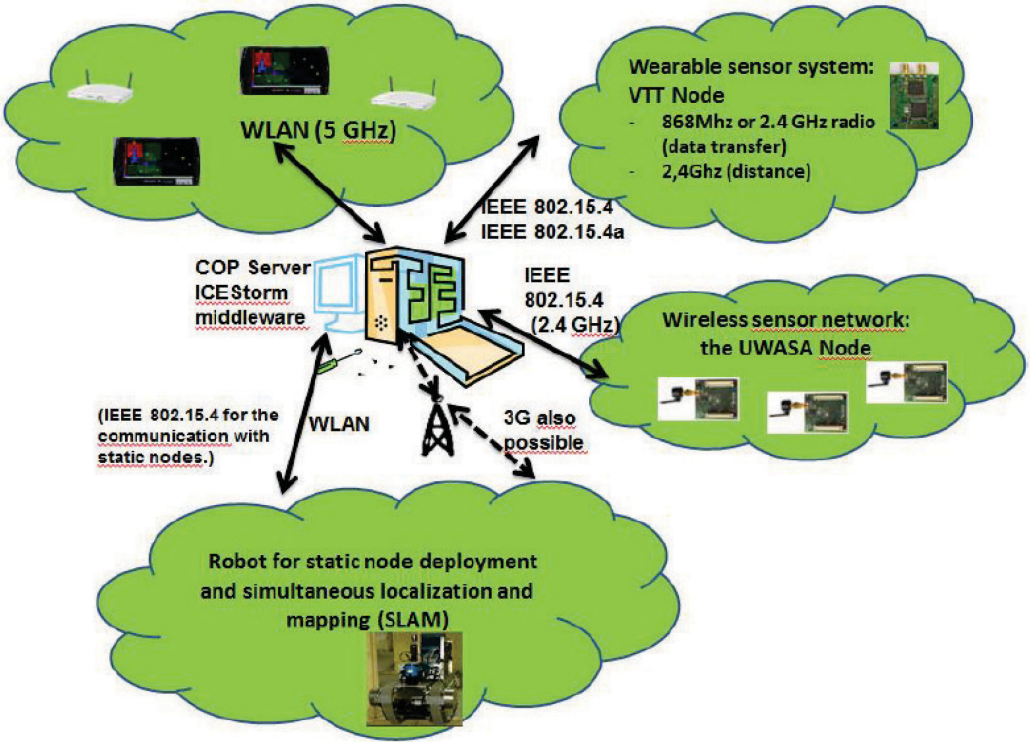


Figure 28. The overall system architecture of the indoor situation modeling system presented in Publication 9 and (Virrankoski 2013).

There are three sensor-equipped subsystems that measure data in the developed situation awareness system: wearable sensor system, deployable wireless sensor network and mobile robot. Measured data is transmitted to the COP server, where the indoor situation model, called the common operational picture (COP),

is computed by combining the measured data and other sources of information. Then the COP server sends the COP to the operating forces and other relevant parts of the command chain.

The wearable sensor system is carried by own forces who enter the building. It tracks their locations in the building and passes the location information to the COP server. In the server the location information is combined with the available maps and building interior templates. That information is then utilized in the troop command. The system also helps to navigate inside the building and warns automatically, if a group member is left behind the others in a dangerous or otherwise unplanned way. The wearable sensor system consists of VTT Node, inertial navigation unit and GPS chip, which is used if the satellite positioning signal is available. It uses Nanotron 2.4 GHz, IEEE 802.15.4a radio for radio based ranging and distance measurements between the group members. Data transmission and other communication with the COP server is done by using RC232, 868 MHz RC1180HP radio (Korkalainen 2012).

Deployable WSN consisted of UWASA Nodes, which are described in detail in Publication 7 (see 5.3.2). In the developed system, these nodes are used for device-free localization (DFL). This method is based on the fact that the human presence in the network area causes radio signal fading. RSS-based DFL is based on radio tomographic imaging, in which the changes in the propagation field of the monitored area are estimated, and then an image of this field is computed based on these changes (Patwari & Agrawal 2008), (Wilson & Patwari 2010). There are two major advantages in DFL. First, the sensor node does not need any additional sensors for this monitoring since it is based on RSS. Second, the monitored person does not need to carry any device with him, and it is easy to do the monitoring in an unnoticeable way. A network monitoring and management framework developed for DFL in the context of the developed indoor situation modeling system, is presented in detail in (Yigitler 2013). An example of DFL is shown in Figure 29.

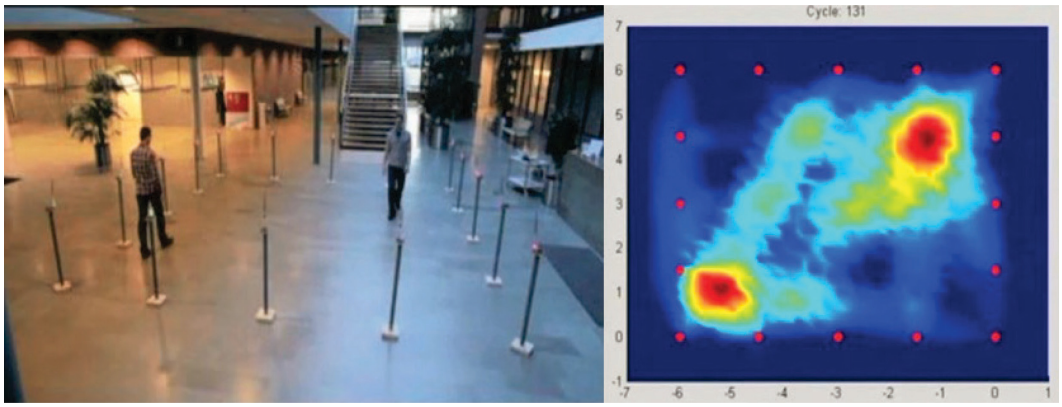


Figure 29. An example of the DFL developed for the situation awareness system discussed in Publication 9: Two persons in the WSN area (left) and the respective radio tomographic image (right).

A teleoperated mobile robot controlled from the command post, is used for simultaneous mapping and tracking. The robot is equipped with a laser range finder and a dead reckoning system for the mapping. There is also a camera with a pan-tilt unit to provide the robot view to the robot operator. The weight of the robot is about 100 kg and its battery capacity 100 Ah (Matusiak 2008). In the indoor situation modeling system presented in Publication 9, the robot was also equipped with a sensor node that acts as a gateway for other subsystems, a sensor node deployment device to carry and deploy sensor nodes on its route and a communication unit to communicate with the COP server and the command post teleoperator. The simultaneous mapping and tracking performed by the mobile robot differs from the complete simultaneous localization and mapping (SLAM) approach because the loop closing mechanism is not employed (Dissanayake 2001), (Durrant-Whyte & Bailey 2006), (Bailey & Durrant-Whyte 2006), (Myrsky 2011), (Saarinen 2012).

In addition to producing a map and a video stream from an unknown building interior, the mobile robot can deploy static nodes on its route and location stamp them, when they are deployed. Moreover, the robot can continuously estimate its distance to the sensor nodes, which communicate directly with it by utilizing the measured RSSI. Since the robot knows its location in its own coordinate system, it can act as a mobile beacon for the deployed and wearable sensor nodes so that they can be localized in the same coordinate system. The whole system can then be transformed from the robot coordinates to some other preferred reference coordinate system, such as external map coordinates, once at least four points

(three in 2D) are known in both coordinate systems. The mobile robot and its simultaneous mapping and tracking result example are presented in Figure 30.

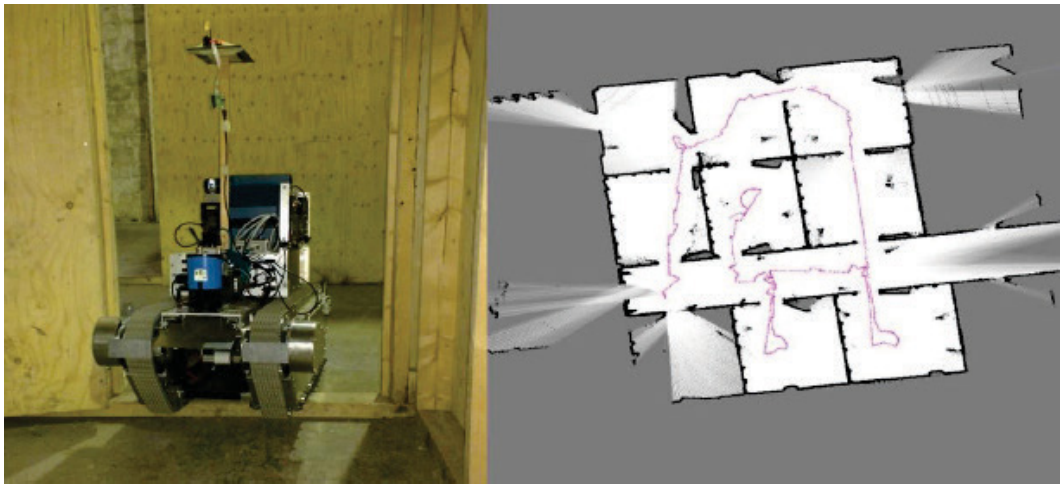


Figure 30. A mobile robot described in Publication 9 (left) and an example of its simultaneous mapping and tracking result (right).

The medium access of different subsystems must be either synchronized or non-overlapping communication bands must be used to avoid interference and packet collisions between different transmissions. In the developed situation awareness system described in Publication 9 and (Virrankoski 2013), the synchronization is used within subsystems. The selection of non-overlapping bands is used in the system integration. Deployable WSN uses IEEE 802.15.4 communication in 2.4 GHz band. System internal communication is synchronized to be able to perform DFL. Wearable sensor system uses IEEE 802.15.4a communication in 2.4 GHz band for ranging and IEEE 802.15.4 communication in 868 MHz band for data communication. The mobile robot uses multi-interface routers between it and the operator at COP server. These routers support a wide range of communication protocols including 3G HSPA, CDMA 450/2000, WiMAX, Wi-Fi, LTE, Flash-OFDM and Trans-European Trunked Radio (TETRA). They also support virtual private networking (VPN) to enable secure connection. In case of the developed situation awareness system, the robot was using WLAN (IEEE 802.11) in 2.4 GHz band and alternatively 3G for data communication between it and the COP server. A sensor node was connected with the robot to act as a gateway between deployed and wearable sensor systems and the robot.

COP server receives data from different subsystems. It computes a common operational picture by using that information and other relevant sources of information, such as digital maps of the environment. COP server is also

responsible about sharing the common operational picture with own troops operating in the building and with other echelons of the operation command. The developed indoor situation modeling system consists of distributed objects, and the interactions between them are enabled by using object-oriented middleware Internet Communication Engine (ICE) (Henning 2004). In the system described in Publication 9, the ICE is augmented by several services, such as publisher-subscriber topic based event distribution system IceStorm. The structure of the COP server is presented in Figure 31.

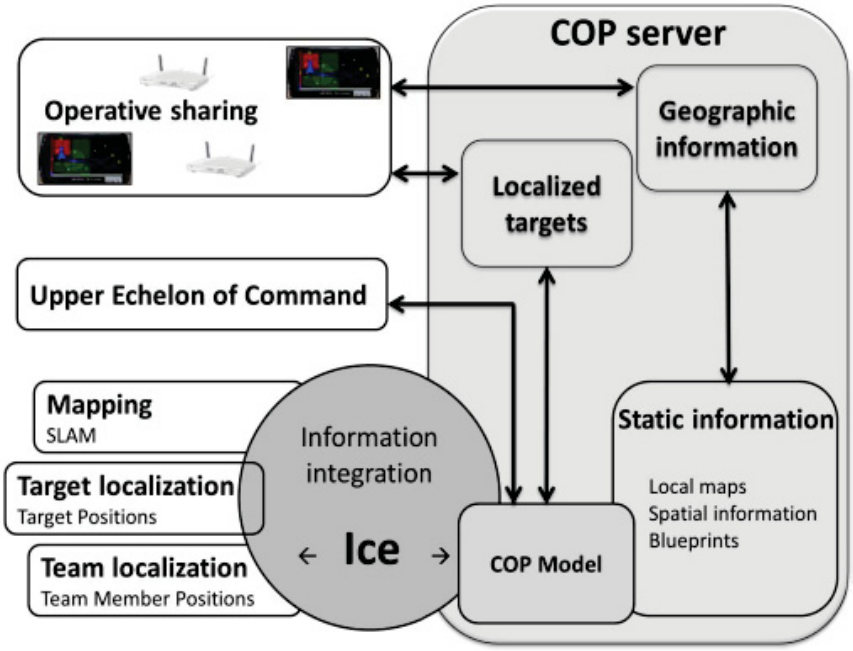


Figure 31. COP server architecture. Figure from Publication 9 and (Virrankoski 2013).

Common operational picture is shared with own troops operating in the building by using IEEE 802.11a WLAN communication, which uses 5 GHz frequency band. Troops are using Samsung Galaxy phones as portable computers to communicate with the COP server. COP is presented on the phone screens as illustrated in Figure 32.

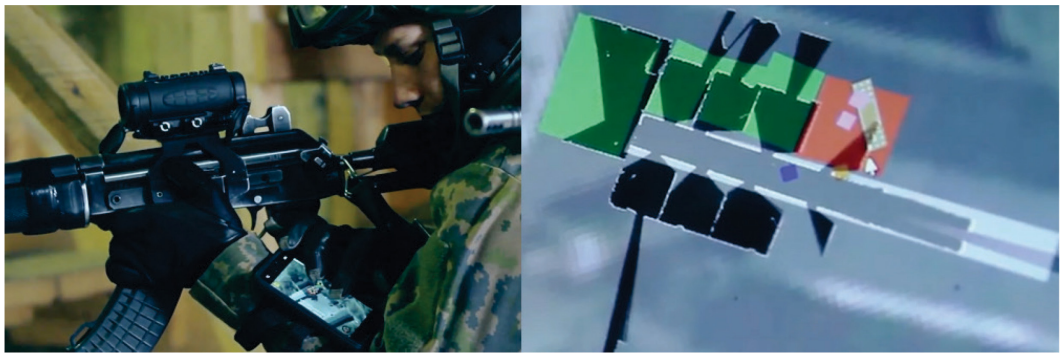


Figure 32. Common operational picture is computed by the COP server and shared with the own troops by using IEEE 802.11a 5GHz WLAN. Information provided by the sensor systems is combined with map and observations are shown by different colors and symbols. Portable device (left) view is zoomed on the right.

Once the targeted performance was first achieved in laboratory experiments, the integrated indoor situational awareness system was tested in a field environment. This environment is used for the urban warfare training by Finnish Army at Santahamina military training facility, and the test was carried out in November 2012. Applied test case was a hostage situation, where enemy forces and their hostages were inside an unknown building. In the test scenario, a platoon of soldiers was divided so that part of them served as own forces entering the building and part of them as enemy forces (targets) and their hostages. All the subsystems presented in Figure 28 were in use. During the scenario, a WLAN network of 300×600 meters covering the building and its surrounding environment was established by using four portable access points. The mobile robot advanced first to the building, performed simultaneous mapping and tracking, deployed sensor nodes within its route and used two different 3G connections to ensure the connection during the operation. In the COP model the mapped rooms were colored red if some enemy or hostile elements were found, and green once they were cleared out of danger. The locations of individual soldiers and other localized objects were also shown in the COP. Twenty UWASA Nodes were used for device-free localization. Three wearable sensor systems were used to localize own forces who advanced to the building. Own forces received the COP to their portable devices (see Figure 32) and they were able to expand the WLAN, when needed. The real-time COP was utilized and the own forces were able to figure out the location of the hostages and enemy forces and took over the building.

In addition to DFL, the deployable WSN can be equipped with other sensors to perform other measurement tasks for situational awareness. In the context of indoor situation modeling projects, also cameras and acoustic sensing were considered. A camera implementation for the UWASA Node with CMUcam3 (Cmu 2018) can produce low-resolution images that can be processed in the node, transmitted over the sensor network and then utilized in the computation of the situation model (Cuhac 2011), (Virrankoski 2013). Speaker identification is discussed in Publication 10 and the estimation of the number of persons by using the acoustic signal in Publication 11.

Every speaker can be identified based on the individual features of his speech. These features are text and language independent, and depend only on the personal features of his voice and the way how he speaks. Characterized speaker features can be matched against the features of the recorded voice samples to see, if the observed person is the same as the recorded one. Public authorities, such as police and military, can match the voice samples against their databases to see if someone of the persons they have already classified and saved in their records is present in the observed target. WSNs equipped with acoustic sensors are well suited to collect these voice samples in an unnoticeable way during the target reconnaissance. However, because of the scarce resources of the sensor nodes, the system performance in terms of sampling frequency, sample length, computation load, data transmission capacity etc. cannot be at the same level as it is in cabled high-capacity acoustic systems.

Publication 10 presents a text and language independent speaker identification, which uses short-time low quality signals. The method is based on mel-cepstral analysis (Bimbot 2004), (Furui 1981) and its performance in the case of short-time low quality signals is investigated to figure out its feasibility for WSN implementation. In the presented speaker identification method, a speech signal of N samples is first collected to vector

(5.69) $\bar{x} = [x(1) \quad \cdots \quad x(N)]$.

Then the speech signal is filtered to enhance high frequencies which are reduced in human speech production. Filtered signal $\bar{x}_p$ is computed elementwise

(5.70) $x_p(i) = x(i) - \alpha x(i-1), \qquad i = 2, \ldots N.$

In (5.70), α is a pre-defined parameter, which usually belongs to range [0.95, 0.98] (Bimbot 2004). Filtered signal is then Hamming windowed with a Hamming window

(5.71) $L_w = t_w f_s,$

where t_w is the time length of the window and f_s the sampling frequency. In Publication 10, applied Hamming window length was $t_w = 0.030\ s$ and the shift between two windows was set to 2/3 of the window length. Then a discrete Fourier transform (DFT) is applied to each window of the signal. Transformed results are collected to matrix T, where each column represents one Fourier transformed window and has N_{bins} elements, where N_{bins} is the number of bins used in DFT. Since the DFT provides a symmetric spectrum, it is enough to use the first half of each column of the matrix T, and collect them to matrix F, which has the first $N_{bins}/2$ rows of T. A power spectrum, that indicates how big portion of the signal power is carried by its different frequencies, is calculated from F by squaring the norm of each element:

(5.72) $P_w = [|F(i,j)|^2]; \qquad i = 1, \dots, \frac{N_{bins}}{2};\ j = 1, \dots, N_w.$

The power spectrum matrix is multiplied by filterbank matrix B_f to enhance the frequencies which are located in the area of human speech such that

(5.73) $P_s = P_w B_f.$

In (5.73), B_f is a filterbank of triangular filters whose central frequencies are located at regular intervals in mel-scale. The use of mel-frequency filterbank reduces the random variation in the area of high frequencies by increasing the bandwidth of the mel-filters (Stevens 1937). Then P_s is converted to decibels P_{dB}, and the mel-frequency cepstral coefficients (MFCC) are computed by applying the discrete cosine transform (DCT) such that the elements of the mel-cepstral matrix C_p are

(5.74) $C_p(k,l) = a(k) \sum_{i=1}^{N_{bins}/2} \left\{ P_{dB}(i,l) \cos\left(\frac{\pi(2i-1)(k-1)}{N_{bins}}\right)\right\},$

where $1 \le k \le N_{cep}$, $1 \le l \le N_w$ and

(5.75) $a(k) = \begin{cases} \sqrt{\frac{N_{bins}}{2}} & , k = 1 \\ \sqrt{\frac{4}{N_{bins}}} & , 2 \le k \le N_{cep} \le \frac{N_{bins}}{2} \end{cases}.$

The main advantage of DCT is that it converts statistically dependent spectral coefficients to statistically independent cepstral coefficients (Bogert 1963), (Oppenheim & Schafer 1968), (Oppenheim & Schafer 1989). In (5.74)-(5.75) N_{cep} is the number of cepstral coefficients considered in the transform. It is upper

limited by $N_{bins}/2$, the number of elements in each column of P_{dB}. It depends on the number of bins (N_{bins}) used in the DFT.

A centered mel-cepstral matrix C is computed by removing the first row of C_p, which carries the first coefficients of each window, and by centering the remaining columns by subtracting the mean of each column from it. The first row of C_p is ignored because it represents only the overall average energy contained in the spectrum. C is then smoothened by multiplying its each column by a smoothening vector $\bar{M}$, which is computed such that for $i = 1, \dots, N_{cep} - 1$:

$$M(i) = 1 + \frac{N_{cep}-1}{2} \sin\left(\frac{\pi i}{N_{cep}-1}\right). \quad (5.76)$$

This smoothening de-emphasizes the lowest and highest order coefficients of C (Juan 1987) and the smoothened mel-cepstral matrix becomes $C_s = \bar{M}C$. In C_s, each column carries the mel-cepstral coefficients of one window. Some of the windows are related to the speech portions of the signal, and some of them are related to the background noise or silence. In Publication 10, a criterion which is based on the averages is used to select the speech related columns. First, an average vector $\bar{C}_N = [C_N(1) \cdots C_N(N_w)]$, where each element $C_N(i)$ is the average of the respective column in C_s, is normalized to range [0, 1]. Then, the matrix C_{sp}, containing those columns of C_s, which stand for the speech portions of the signal, is constructed by using the selection criterion

$$C_{sp} = [C_s(j) | C_N(j) \geq \mu\{\bar{C}_N\}], \quad j = 1, \dots, N_w. \quad (5.77)$$

In (5.77), $C_s(j)$ refers to the j:th column of C_s, and $\mu\{\bar{C}_N\}$ is the overall average of $\bar{C}_N$. The final mel-cepstral coefficients are computed by taking the row-wise average of C_{sp}:

$$C_{cep} = \begin{bmatrix} \mu\{C_{sp}(1,1) \cdots C_{sp}(1,n)\} \\ \vdots \\ \mu\{C_{sp}(N_{cep}-1,1) \cdots C_{sp}(N_{cep}-1,n)\} \end{bmatrix}, \quad (5.78)$$

where n is the number of mel-cepstral vectors selected in (5.77), $n \leq N_w$. The information is extended to capture the dynamics of the speech by computing the first and second order temporal derivatives of C_s. The first order temporal derivatives are computed according to

$$\Delta C_s(i,j) = \frac{\sum_{k=-\theta}^{\theta} k C_s(i,j+k)}{\sum_{k=-\theta}^{\theta} k^2}. \quad (5.79)$$

In (5.79), $1 + \theta \leq j + k \leq N_w - \theta$ and $1 \leq i \leq N_{cep} - 1$. The second order derivative $\Delta\Delta C_s$ is computed by computing the first order derivative of ΔC_s

according to (5.79) (Bimbot 2004), (Rabiner & Juang 1993). Finally, the first and second order derivatives of the mel-cepstral coefficients are computed by selecting the columns from ΔC_s and $\Delta\Delta C_s$ in the same way as in (5.77) and by computing the row-wise averages in the same way as in (5.78). Then the results are collected into feature matrix (called feature vector in Publication 10) F_s:

(5.80) $F_s = \left[C_{cep}^T \quad \Delta C_{cep}^T \quad \Delta\Delta C_{cep}^T\right]^T$.

Feature matrix F_s has $3 \times \left(N_{cep} - 1\right)$ elements and it characterizes the speaker.

As a part of the simulations in Publication 10, a voice sample database was collected. The database included 15 languages and 60 individuals, 15 women and 45 men. To guarantee the text and language independency, at least two samples and more than one language, if possible, was recorded from each person. Finally there were 190 voice samples with a length varying between 8-10 seconds in the database. The recording was done in a normal office environment with a commercially available wired microphone. Because of that setup, also the usual background noise was included to the recorded voice samples. Collected database was divided into two parts; the first 140 samples were used to study the algorithm accuracy to identify the correct speaker, and the remaining 50 samples were used to study the algorithm accuracy to indicate if the observed speaker is not found among the recorded voice samples. Euclidean distance between the feature matrix of the recorded sample and the feature matrices of the samples in the database was used as a similarity measure.

In the first experiment, the length of the recorded voice sample was set to 8 s and the sampling frequency to 8 kHz. Then the N_{bins} used in the DFT was varied from 128 to 2048 and the N_{cep} considered in the computation from 10 to 1024 ($N_{cep} \leq N_{bins}/2$). The results are presented in Figure 33. Maximum 78% accuracy in the speaker identification was achieved with $N_{bins} = 512$ and $N_{cep} = 100$. It was also observed in the simulations that the N_{cep} has much bigger impact to the algorithm performance than N_{bins}. As indicated by the plot in Figure 33, the identification accuracy was best with $N_{bins} = 100$. For smaller values of N_{bins}, the result was rapidly weakening. For bigger values, it was slightly weakening because of the signal overlearning.

In the second set of simulations the number of bins in DFT and the number of cepstral coefficients were kept in the best found configuration $N_{bins} = 512$, $N_{cep} = 100$. The length of the sample (L) was varied from 2 s to 8 s and the sampling frequency (f_s) from 200 Hz to 8 kHz. These ranges were considered similar as the ones that can be applied when measuring acoustic signals with wireless sensor nodes. The combined effect of f_s and L is shown in Figure 34.

With 8 s long samples and f_s between 7 and 8 kHz, the identification accuracy was between 70 and 80%. Then it weakened moderately when f_s was reduced from 8 kHz to 2 kHz but collapsed rapidly once it was reduced further from 2 kHz.

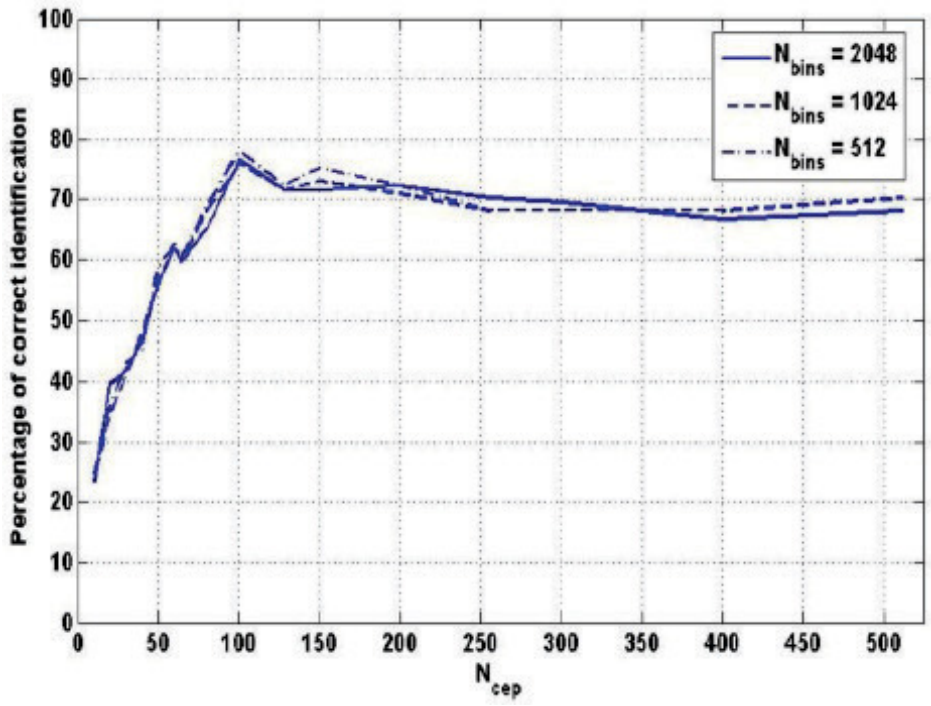


Figure 33. The effect of N_{bins} and N_{cep} for the speaker identification accuracy in Publication 10 simulations, when the length of the sample was kept in 8 s and sampling frequency in 8 kHz.

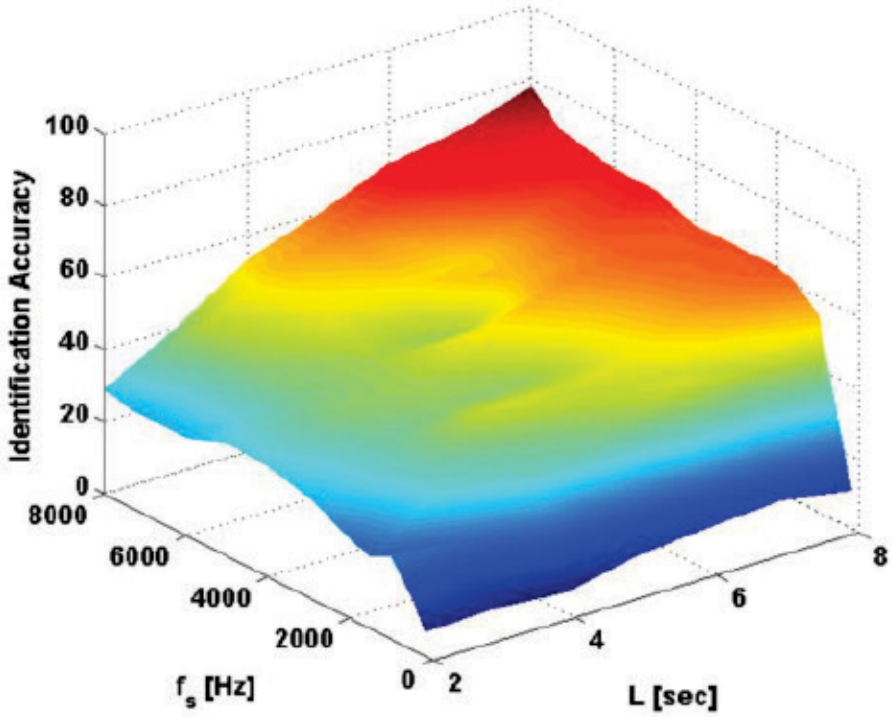


Figure 34. The combined effect of f_s and L on the speaker identification accuracy, as observed in Publication 10 simulations. $N_{bins} = 512$ and $N_{cep} = 100$.

Third set of simulations in Publication 10 was focusing on the accuracy to indicate if the person whose voice is measured is not represented by the voice samples in the database. A light-weight criterion

(5.81) $T_{th} = \mu_{id} + m\sigma_{id}$

was applied. In (5.81), μ_{id} is the average and σ_{id} the standard deviation of the distances between the feature matrices of the measured signal and the database samples in the case of correct identification in a first and second set of simulations. Parameter m is pre-defined to fit the value of the threshold T_{th}. If the minimum distance between the feature matrix of the measured voice signal and the feature matrices of the database voice samples was bigger than T_{th}, a conclusion that the observed person was not represented by the database voice samples, was made. Based on the simulations, the most correct classification with criterion (5.81) was reached when parameter m was within a range $m \in [0.5; 1]$. Then the maximum accuracy to detect non-presence correctly was 65-70%.

The setup in Publication 10 focuses on such situation, where one person speaks in time and the speaker identification is done based on that measured particular signal. In practice the situation may often be such that several people are talking simultaneously. In the case of reconnaissance operation, the number of persons present is not necessarily known but must be figured out. Publication 11 presents a way to apply independent component analysis (ICA) for that purpose.

In principle, ICA is widely applied in signal processing (Back & Weigend 1997), (Hyvärinen 1999), (Kiviluoto & Oja 1998), (Ristaniemi & Joutsensalo 1999), (Vigario 1997), (Vigario 1998). It targets to find a linear representation of non-Gaussian data so that the components are statistically independent. If we assume that there exists n sources transmitting simultaneously and h observing sensors so that $n \leq h$, each observation of the sensor i can be written as

$$x_i = a_{11}s_1 + a_{12}s_2 + \cdots + a_{1n}s_n, \tag{5.82}$$

where source signals are noted by s_i. If we further assume that there exist k observations over a discrete time, all observations can be presented as

$$x = Ms, \tag{5.83}$$

where x is $h \times k$ observation matrix, M is the mixing matrix and s is $h \times k$ matrix of measured signals. In ICA model (5.83), the statistically independent source signals s_i are latent variables that cannot be directly measured. Once the mixing matrix M is estimated, the independent source signals can be solved from (5.83):

$$s = M^{-1}x. \tag{5.84}$$

Even though the source signals are assumed independent, it does not matter if they have an overlapping spectrum. ICA works also in blind signal separation (BSS) scenario, where we need to find the original individual signals from the mixed signal without prior knowledge about the number of signal sources. In the standard ICA, the measured data is preprocessed before the signal separation computation. In the preprocessing the observation matrix X is whitened so that the new observation matrix $\tilde{X}$ has uncorrelated components and their variances are equal to unity. Centered measurement vectors $\tilde{x}_i$ are whitened by using a transform

$$v_i = W\tilde{x}_i, \tag{5.85}$$

where W is the whitening matrix. Once the covariance matrix (C_x) of the measured acoustic data is known, the whitening matrix can be solved by applying principal component analysis (PCA) such that

(5.86) $W = D_e^{-1/2} E^T$.

In (5.86), D_e is a diagonal matrix having the eigenvalues of C_x in its diagonal in a decreasing order, and E is a matrix of associated principal eigenvectors. In the case of h sensor nodes and k measurements by each of them, the observation matrix becomes

(5.87) $X(k) = \begin{bmatrix} x_{11} & \cdots & x_{1k} \\ \vdots & \ddots & \vdots \\ x_{h1} & \cdots & x_{hk} \end{bmatrix}$.

The centered observation matrix $\tilde{X}(k)$ is computed from (5.87) by subtracting the mean of each measurement vector (each row) from it. Then the covariance matrix becomes

(5.88) $C_x = Cov\{\tilde{X}(k)\} = E\{\tilde{X}(k)\tilde{X}(k)^T\}$

$$= \begin{bmatrix} E\{\tilde{x}_1(k)\tilde{x}_1(k)^T\} & \cdots & E\{\tilde{x}_1(k)\tilde{x}_h(k)^T\} \\ \vdots & \ddots & \vdots \\ E\{\tilde{x}_h(k)\tilde{x}_1(k)^T\} & \cdots & E\{\tilde{x}_h(k)\tilde{x}_h(k)^T\} \end{bmatrix}$$

$$= \frac{1}{k} \begin{bmatrix} \tilde{x}_{11}^2 + \cdots + \tilde{x}_{1k}^2 & \cdots & \tilde{x}_{11}\tilde{x}_{h1} + \cdots + \tilde{x}_{1k}\tilde{x}_{hk} \\ \vdots & \ddots & \vdots \\ \tilde{x}_{h1}\tilde{x}_{11} + \cdots + \tilde{x}_{hk}\tilde{x}_{1k} & \cdots & \tilde{x}_{h1}^2 + \cdots + \tilde{x}_{hk}^2 \end{bmatrix} = \frac{1}{k}\tilde{X}(k)\tilde{X}(k)^T.$$

If the number or source signals $q \leq h$ and signal to noise ratio in the measured mixed signals is large enough, there exists a clear difference or magnitude in the C_x eigenvalues so that first q eigenvalues correspond to the source signals and the remaining $h - q$ eigenvalues correspond to the noise (Ham 2000). In the case of mixed speech signals, q gives an estimate for the number of (speaking) persons who are present in the monitored space.

In the first experiment in Publication 11, four mixed signals were recorded in noisy environment by using a cabled microphone with 8 kHz sampling frequency and 8 bits per sample. Once C_x was computed as presented in (5.87)-(5.88), its eigenvalues were

(5.89) $diag\{D\} = \begin{bmatrix} 0.01640 \\ 0.01435 \\ 0.01411 \\ 0.01370 \\ 0.00080 \\ 0.00079 \\ 0.00074 \\ 0.00072 \end{bmatrix}$.

There is a clearly recognizable difference of magnitude between fourth and fifth eigenvalue in (5.89), which indicates that the number of speakers in the original mixed acoustic signal was 4. The behavior of the covariance matrix eigenvalues during the whole 5.5 s sampling period is presented in Figure 35. When the 8 kHz sampling rate was applied, the difference of magnitude between 4^{th} and 5^{th} eigenvalue of C_x became visible during the first second of the sampling.

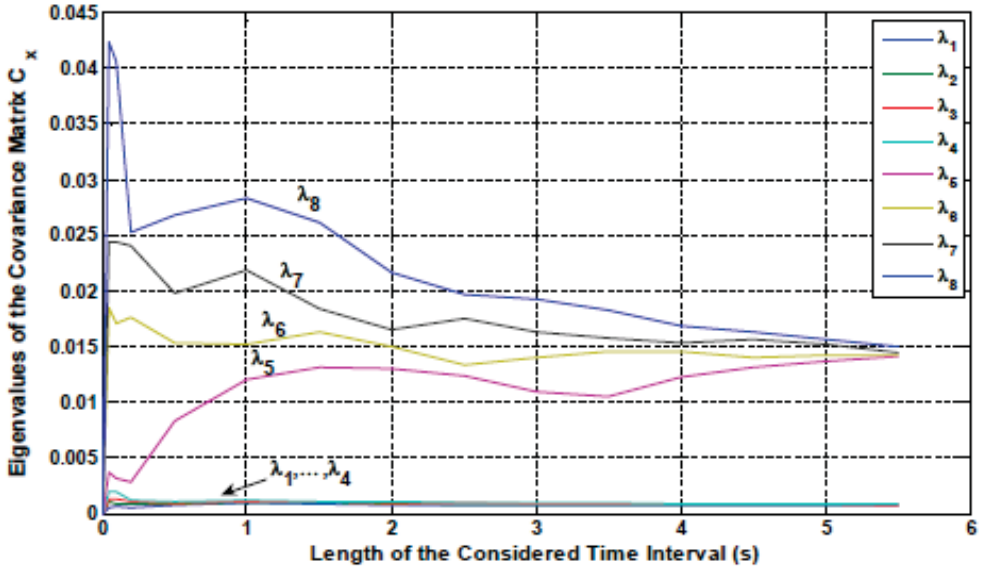


Figure 35. The behavior of the eigenvalues of observation matrix covariance matrix (C_x) in the first experiment in Publication 11. The difference of magnitude between λ_4 and λ_5 indicates the existence of four sources (speakers). The difference of magnitude becomes detectable during the first second. Sampling rate in this experiment was 8 kHz and 8 bits per sample was applied.

Second experiment in Publication 11 was done by using three Mica2 sensor nodes (Mica2 2003) equipped with low-power microphones. Applied sampling rate was 3 kHz. Same analysis as in (5.87)-(5.88) was computed and the resulting eigenvalues of the covariance matrix C_x were

$$(5.90)\quad diag\{C_x\} = \begin{bmatrix} 0.04426 \\ 0.02415 \\ 0.00952 \end{bmatrix}.$$

The difference between second and third eigenvalue is still the biggest suggesting two speakers, but there was no similar clear difference of magnitude as in (5.89). The result was expectable since the applied sample rate was less than half of the

one used in the first experiment. Another factor that was weakening the result was the accuracy of the time synchronization. Since the same mixed signal is measured simultaneously by several sensor nodes, the correctness of the BSS outcome highly depends on the accurate time synchronization between the nodes. In further experiments in Publication 11, the time synchronization error between the sensor nodes was increased from 0 to 15 milliseconds. The covariance matrix eigenvalues did not produce any useful result for BSS after the error exceeded 3 ms.

The results of Publication 11 indicated that it is possible to estimate the number of persons from the mixed simultaneous speech signal by using ICA based BSS with short-time low-quality samples. With 8 kHz sampling frequency the difference of magnitude in the observation matrix covariance matrix eigenvalues was easily recognizable, but with 3 kHz sampling frequency the result was highly dependent on the time synchronization accuracy between the sensor nodes.

5.4.3 WSN with Frequency Converters

In general, frequency converter is used to modify the frequency of the alternating current (AC). Based on this property, frequency converters are used in electricity distribution systems and in the control of electric motors, because the motor speed depends on the AC frequency. The control of the motors driving the system or process is an essential part of its automation. If the automation system collects part of the measurements it utilizes by using WSN, it would be beneficial to have a direct communication link between the frequency converters and WSN. Otherwise, the WSN data must be transmitted first to separate gateway, and then from the separate gateway to the frequency converter by using another cabled network. This will introduce additional communication delay and make the communication system architecture more complex. Even though there is often a cabled connection to the frequency converter having its own IP address, this is not always the case. In some cases the frequency converter location can be mobile, difficultly accessible or harsh (dust, dirt, vibrations, temperature, chemicals etc.) in such a way that cables can be easily damaged or cabling is not possible at all. In these cases the wireless connection between the frequency converter and the rest of the automation system will also enable wireless control of the frequency converter and wireless collection of the frequency converter data. The frequency converter data can be utilized to extract information of the current state of the electric motor and system or process, which is run by the frequency converter.

Publication 12 presents software and hardware integration between frequency converter and WSN. Used frequency converter is produced by Vacon, which is nowadays a registered trademark of Danfoss Group (Vacon 2018). Applied sensor platform is the UWASA Node, which is presented in Publication 7. Three different types of communication links are required: communication between the UWASA Node and Vacon frequency converter, communication between UWASA Nodes and communication between gateway node and data logging PC.

Figure 36. The type of the Vacon frequency converter which was used in the implementation in Publication 12.

The type of the Vacon frequency converter which was used in the implementation is presented in Figure 36. Removable control panel is connected to the frequency converter via RS-232 serial port. The same port is used to connect the UWASA Node with the frequency converter. The converter has its own Vacon Human Machine Interface (VHMI) protocol for interfacing, and it is implemented to the connected UWASA Node. Wireless communication between UWASA Nodes uses IEEE 802.15.4 based MAC protocol. Conversion software, which interfaces between VHMI and wireless communication, is implemented to those UWASA Nodes, which are connected to the frequency converters. The software converts VHMI packets arriving from the frequency converter to MAC format so that the data can be sent over the wireless channel. Respectively, it converts the MAC packets that arrive over the wireless channel to VHMI format so that the data can be sent from the node to the frequency converter. At the other end, the gateway

node is connected to the data logging PC via USB port. Conversion software, which converts between MAC and USB packets, is implemented to the gateway node.

In the experiments of Publication 12, a sophisticated mobile machine that was operated by several electric motors was monitored. Data was collected over WSN from six frequency converters, which were controlling the electric motors that run the machine during the operation. Ten parameters were collected from each frequency converter in a sampling interval of $200\ ms$, which equals five times per second from each frequency converter. System setup is presented in Figure 37. A gateway node and the data logging PC were staying in one location, but the frequency converters with the connected nodes were moving with the machine so that the distance between them and the gateway node varied between 3 and 30 meters.

Application software for the communication was designed so that the WSN formed a star topology where each node communicated directly with the gateway node. Communication followed the master-slave principle such that the nodes which were connected to frequency converters were allowed to transmit data only when polled by the gateway node. A network control program was run in the data logging PC and it sent the commands and requests to other nodes through the gateway node. It was observed during the experiments that it took $50 - 90\ ms$ to read the data from the frequency converter to the connected sensor node. Based on that observation, a $100\ ms$ timer was set between reading and data transmission requests to give node enough time to read the data and convert it from VHMI format to wireless transmission format. The reading and transmission requests were done in sequence to avoid collisions between data packets transmitted by different nodes.

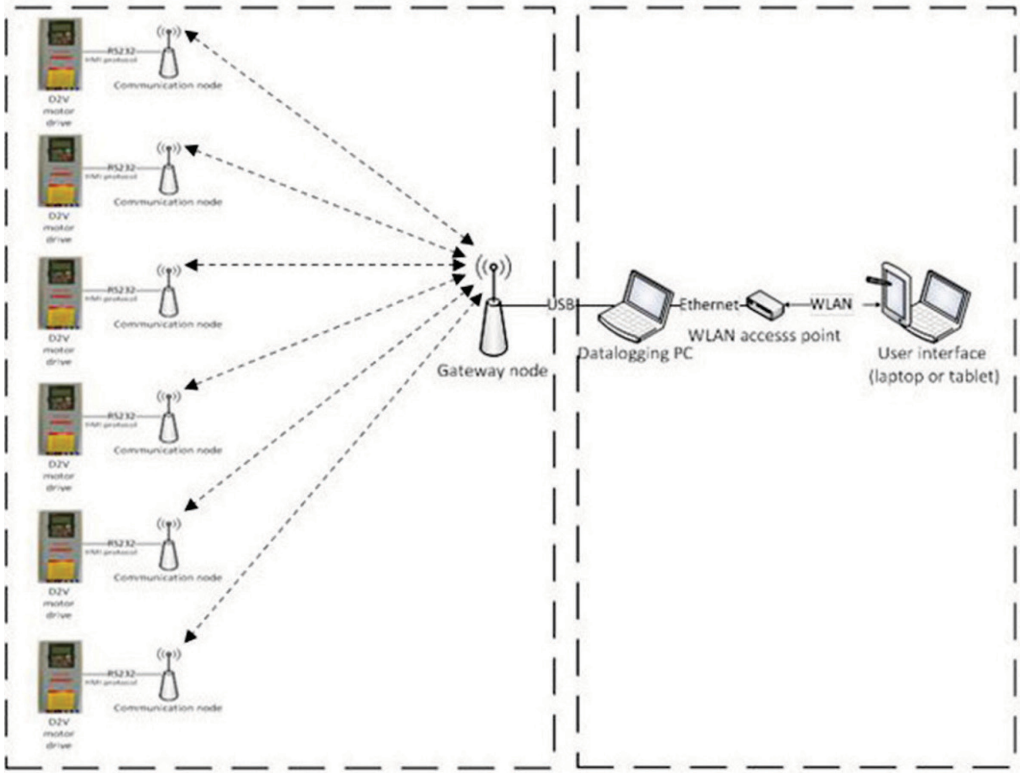


Figure 37. A setup of the experiment presented in Publication 12. Ten parameters were wirelessly collected from six frequency converters in a sampling interval of 200 *ms*. Frequency converters were moving with the machine and distance between them and the gateway node varied between 3 and 30 meters during the operation.

In the first set of experiments the system was able to collect data from the frequency converters as expected, but the observed packet loss varied between 9-11% of the transmitted packets having an average of 9.92%. Such a level is not acceptable in terms of communication reliability. To improve it, a resend request mechanism, which is presented in Figure 38, was implemented. In the communication application the gateway node first sends read command to all six nodes and then after 100 *ms* starts to send transmission requests to all nodes one by one. After sending transmission request to one node, the gateway node waits 50 *ms* if it can receive a response semaphore indicating that the response from the node is received. If the response is not received in that timeframe, the gateway node will resend the transmission request to that node and wait 40 *ms* more. If the response was still not received, the packet was counted as lost and the gateway node was moving to request the next node.

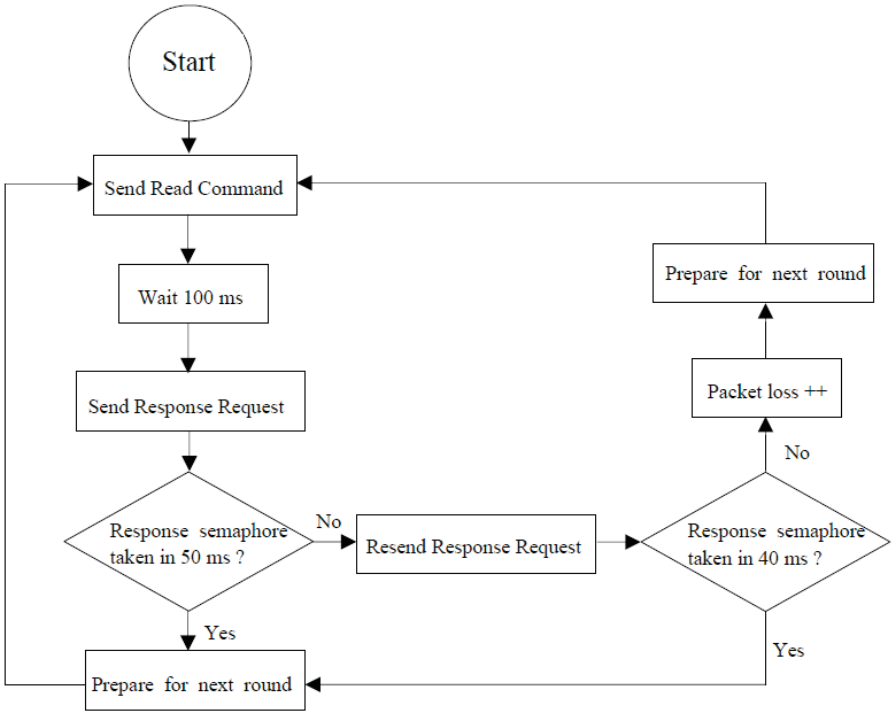


Figure 38. A software flowchart of the resend mechanism implemented in Publication 12.

Once the experiment was repeated with the implemented resend request mechanism, the packet loss was between 0.5 – 2% of the transmitted packets having an overall average of 1.22%. This was a remarkable improvement compared to the earlier 9.92%. Measured packet loss without resend request mechanism is presented in Figure 39 and with resend request mechanism in Figure 40. This improvement was achieved by adding just one resend request, as indicated by the flowchart in Figure 38. More resend requests can be added if even higher communication reliability is still needed. However, a bigger number of resend requests will also introduce longer communication delays.

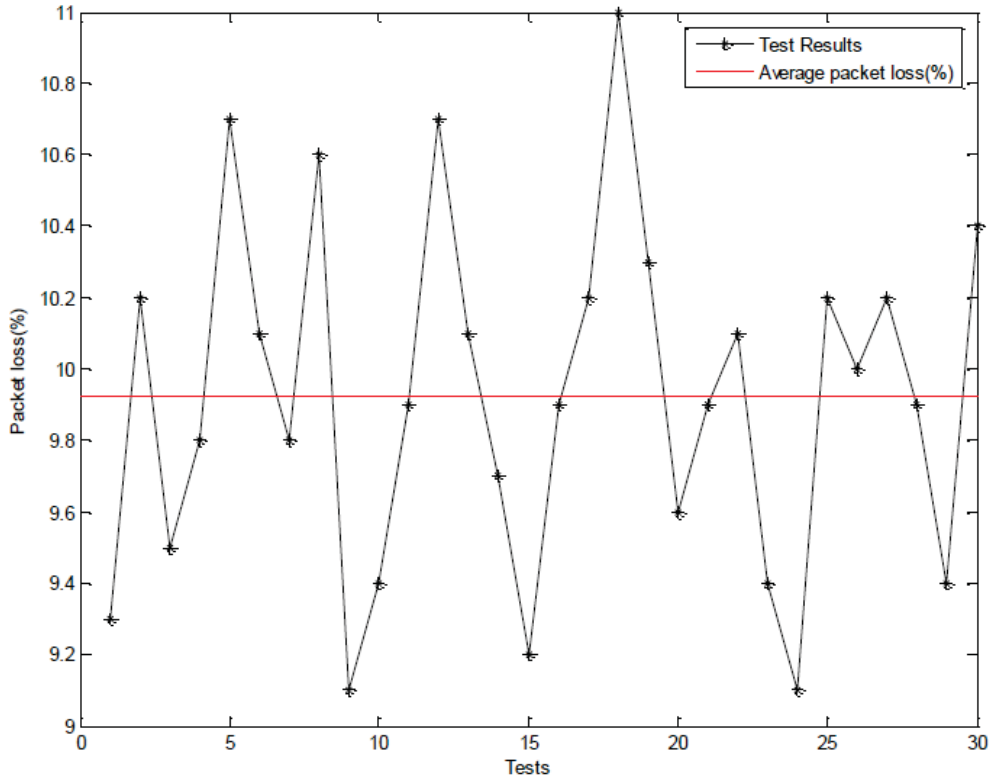


Figure 39. Packet loss rate as percentage of transmitted packets over 30 experiments in Publication 12 without resent request mechanism. Packet loss rate varied between 9-11% and the overall average packet loss indicated by red line in the plot was 9.92% of the transmitted packets.

In the experiments the system was able to collect data from six frequency converters that controlled the electric motors of the mobile machine in industrial environment. Ten parameters were collected once in 200 ms from each of the frequency converters. Achieved data transmission rate at the gateway node was 16.8 $kbits/s$ and the average percentage of lost packets with the resend request mechanism 1.22%. The distance between the gateway node and the nodes which were connected to the frequency converters varied between 3 and 30 meters during the operation. Even though the main focus in the experiments was to collect data from the frequency converters, the implementation enables bidirectional communication so that in addition to data transmission requests, also other type of commands can be sent to the frequency converter over the WSN by using the implemented application. In addition to star topology architecture, a preliminary implementation with some experiments was also done for a multi-hop setup.

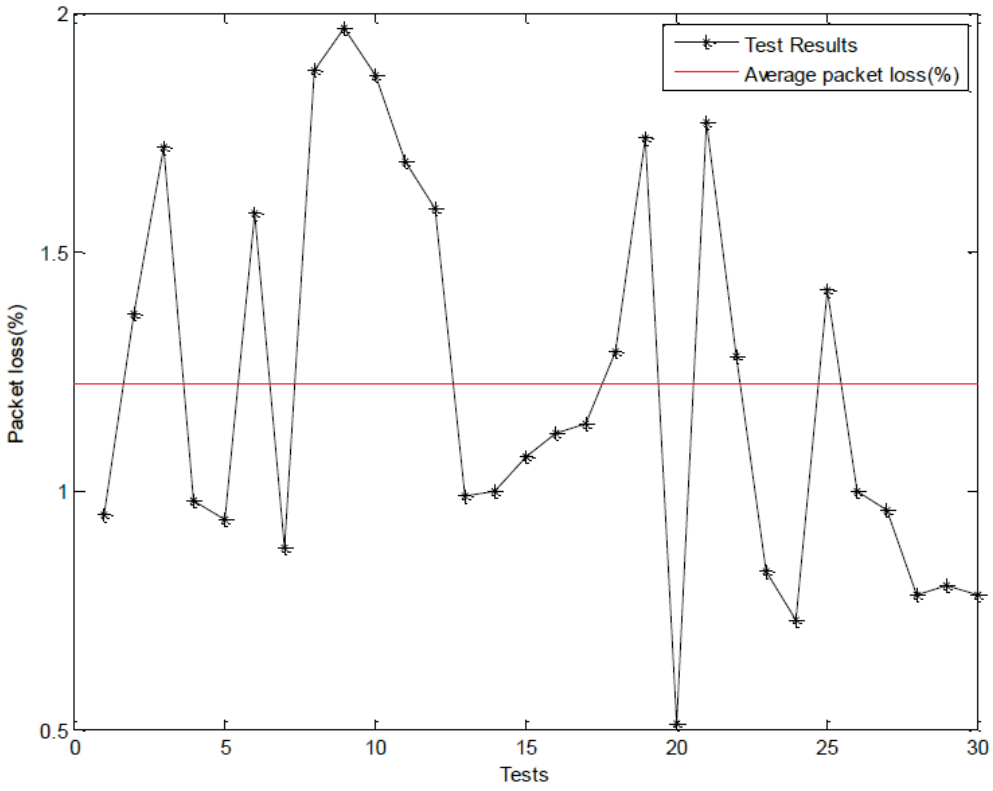


Figure 40. Packet loss rate as percentage of transmitted packets once the resend request was implemented and the experiments repeated with it in Publication 12. The packet loss rate varied between 0.5 and 2% and the overall average packet loss was 1.22%.

5.4.4 Energy Harvesting

Wireless sensor systems enable us to increase the number of measurement points and to get access to such parts of the system, which are not accessible by cabled systems, such as moving and rotating machine parts etc. Wireless connection can also be useful in harsh conditions, where temperature, dirt, dust, vibrations etc. can easily damage the cables. However, the power supply of the sensor nodes is a challenge. Continuous power supply is needed during the operation to keep up the system and to guarantee the system robustness and reliability. There are many applications where cabled power supply is not an option. The sensor node batteries can be manually charged and changed in experimental setups, but in the industrial environments it is not the case. To be feasible to use, the service period of the WSN should be the same as the service period of the whole automation system. Since the sensor nodes require relatively low operation power, there are

many applications where the required power can be harvested from the surrounding environment. Light (solar) and wind energy, continuous mechanical movement like rotation or vibration, and temperature and pressure differences are examples about such sources of energy, which can be utilized in energy harvesting for WSN.

Publication 13 presents a description of the developed solar energy harvesting and energy management solution for the UWASA Node and its performance evaluation through experiments. Developed solution is based on the AmbiMax system (Park & Chou 2006). Its architecture is presented in Figure 41. AmbiMax is a modular energy harvester, which can simultaneously utilize one or several energy sources. A reservoir capacitor array consisting of separate supercapacitor for each energy harvester is adopted. These supercapacitors are then used to power a common voltage rail, which is used to feed power for the wireless sensor node. The power is fed either directly for the components of the operating node, or for the node battery charger. Since the available harvested energy and node power consumption both continuously vary over time, the battery is necessary to balance and guarantee the power supply. If less energy is harvested than needed by the node, the missing part is extracted from the battery. If more is harvested than needed, the remaining part will be charged to the battery.

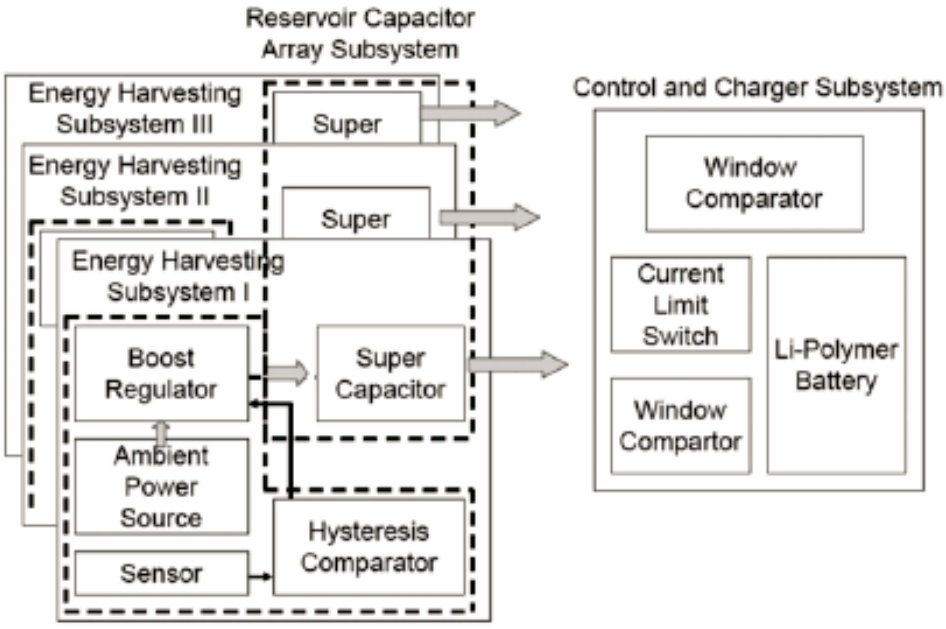


Figure 41. The architecture of the AmbiMax platform (Park & Chou 2006), which is used as a reference design in Publication 13.

In the design presented in Publication 13, similar but slightly more efficient components as the ones in AmbiMax are used. Moreover, a low-power undervoltage lock-out circuit and a real-time clock controlled latch switch are added to disconnect the voltage rail from the sensor node. This disconnecting takes place if the rail voltage drops below a pre-defined threshold or if the sensor node signals the voltage rail to shut down for a length of a sleeping period. The UWASA Node Power Module (see Publication 7) is designed for one-cell lithium ion batteries with a nominal voltage of 3.7 V and it accepts voltages between 1.8 – 4.2 V. In the implementation presented in Publication 13, light is used as an energy source. A solar cell with a size of 92×61 mm and a nominal power of 0.45 W is used with the LTC3105 energy harvesting integrated circuit. The low start-up voltage of LTC3105 allows it to harvest energy also during the low illuminance periods, when the photovoltaic cell output voltage is low. The energy harvester is designed to support 4.2 V, which is a maximum voltage for one-cell lithium-ion battery. Developed prototype is presented in Figure 42.

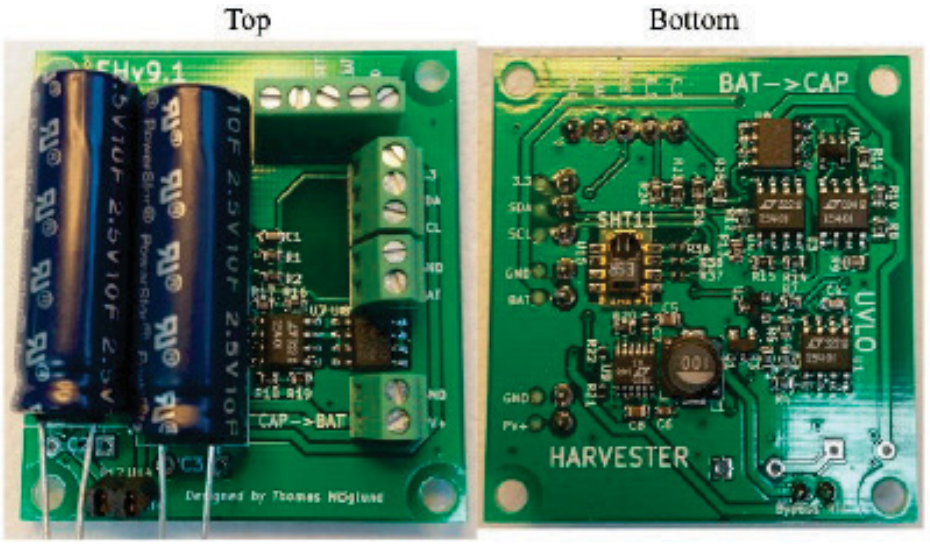


Figure 42. Publication 13 energy harvester prototype developed by Thomas Höglund.

Developed energy harvester was located on the roof with a line of sight to the sky and used continuously there during six days. The experiment was done in Finland during winter time, when the daylight time was short and the temperature some degrees below the freezing point. Connected data logger was used to measure time, illuminance, and the voltages of the battery, solar cell, supercapacitor and MPCC pin of the LTC3105. Observed battery and solar cell voltage, and illuminance are presented in Figure 43.

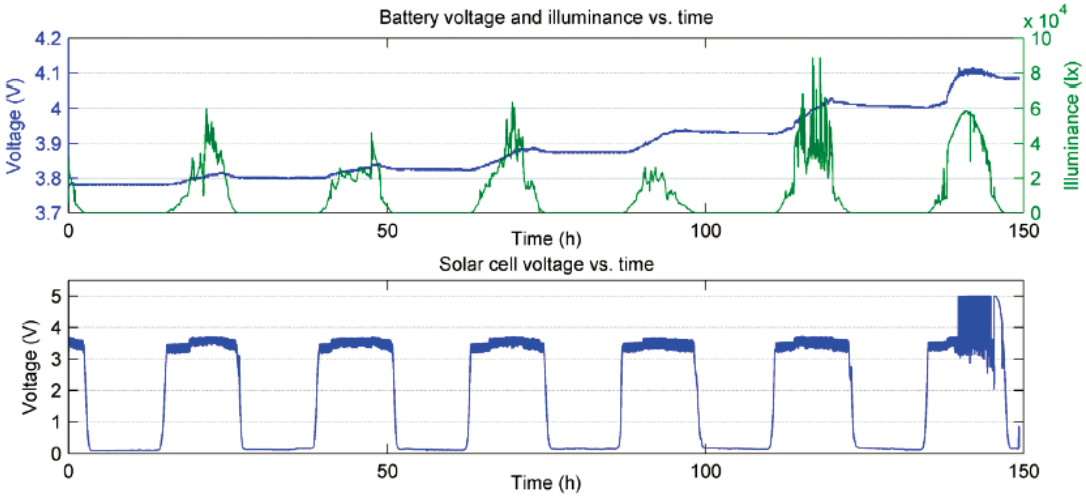


Figure 43. The performance of the developed energy harvester in Publication 13. Upper plot presents luminance (green curve) and battery voltage (blue curve) and lower plot solar cell voltage during six days experiment.

On average, the energy harvester was active 9.0 h and harvested at 35.6 mW efficiency per day. In terms of energy, on average 1.16 kJ was harvested per day and 2.14 J per minute when the harvester was active. The total energy harvested during six days was 6.9 kJ, which is equal to 51% of the 1000 mAh and 3.7 V LiPo battery capacity. It was observed in the experiments that the startup and the initialization of few sensors and wireless communication in the UWASA Node consumed between 0.7 and 1.5 J. Measuring 3-axis accelerometer in 10 bits in 500 Hz sample rate for two seconds consumed 1.82 J. Thus, for the UWASA Node, reading several sensors with such sample rate and transmitting thousands of bytes would consume $3-10\,J$ for measurements and $50-100\,J$ for transmission. If the transmitted amount of data is kept relatively low and carefully scheduled sleep and wake periods are used to achieve energy savings, the UWASA Node can operate at an interval of 3-4 minutes using the presented energy harvesting solution.

One solar cell was used in the experiments in publication 13, but the energy harvester is designed so that more solar cells or other types of energy harvesting sources can be added and used simultaneously, if more energy is needed.

6. DISCUSSION

This doctoral dissertation focuses on open source platform development under IEEE 802.15.4 standard. Presented work in this area consists of three main entities: algorithms related to network initialization and control, platform planning and design process, and applications to evaluate the performance of the developed algorithms and developed platform.

6.1 Published Results

6.1.1 Algorithms

Presented algorithms are related to localization (Publication 1, Publication 2), clustering (Publication 3) and time synchronization (Publication 4). Distance matrix reconstruction method presented in Publication 1. It improves the performance of MDS, which is based on distance matrix SVD, because it presents a way how the distance matrix reconstruction should be done, if all pairwise distances are not known. This is typically the case in WSN. It is also relevant to use the shortest Euclidean paths to estimate those missing distances that cannot be directly measured. The shortest paths are usually known as part of the routing information in a multi-hop network. Even though the shortest radio path indicated by the routing information is not always the same as the shortest Euclidean path, because of the variations in the radio environment. The way how the shortest path is used in the presented distance matrix reconstruction makes it feasible, though not always the best estimate for the missing pairwise distances.

MDS will produce the relative coordinates of the sensor nodes as an outcome. In this context the term relative means that the computed coordinates can be rotated, reflected or translated compared to the preferred reference coordinate system. As a consequence, four reference points in 3D or three reference points in 2D are needed in both coordinate systems to be able to make the final transform to the reference coordinates. Error in each pairwise distance estimate in the distance matrix effects the whole localization outcome. The final transform from the localization outcome relative coordinates to the preferred reference coordinates will also affect the error. The analytical presentation of the error behavior and the effects of the error in these operations is a topic for further research.

RSSI-based distance estimation does not require separate hardware for the distance computation. The additional communication load caused by the RSSI-based distance estimation is also smaller what it would be in the case if additional distance estimation hardware or location-aware beacon nodes would be used. The method can be applied with any radio technology, but because the radio channel effects appear differently on different frequencies, the expected accuracy of the RSSI-based distance estimation is also different for different communication technologies.

Publication 2 presents one example about the way how the statistical properties of the measured signal can be utilized to improve the RSSI-based distance estimation. The path loss exponent and the loss constant of the propagation model are fitted by using the received signal strength standard deviation and number of lost packets in each distance as an optimization criteria. Distance estimation results are improved, but the use of this method requires measurements in the network environment in advance, to collect the data for optimization. Moreover, the results are environment specific and in every particular environment the measurements and the respective parameter optimization should be repeated regularly, because the radio environment changes all the time. This makes the method feasible for such environments where planning and measurements can be done in advance before deploying the network. It is less suitable for such cases where the nodes must be rapidly deployed and the network must start to operate immediately.

TASC clustering algorithm presented in Publication 3 has the novelties of truly distributed computation and the ability to cluster the network according to the existing network topology. It is enough for each node to be aware of its two-hop neighborhood, and most of the clustering related computation is also executed there. Selection of clusterheads as well as the number of clusters in the clustering outcome depends on the network topology, and neither of them is defined in advance. The algorithm is suitable for such applications, where the nodes are deployed randomly and the network must automatically initialize itself. It is especially suitable to organize deployments which have a challenging topology, such as the one illustrated in Figure 2. Required node capability to measure distances from its neighbors limits the algorithm use. On the other hand, the fact that the algorithm tolerates remarkable amount of noise in the distance measurements, as indicated in the simulations in Publication 3, suggest that also RSSI-based distance estimation can be applied instead of ultrasonic or other more accurate measurements. Another limitation for the TASC clustering algorithm use is the amount of messaging it requires, which causes certain level

of overhearing. Message scheduling is also required in the communication application to avoid message collisions during the clustering.

Utilization of the node weights, which are computed based on the shortest paths in Publication 3, can be researched further. The fact that they carry information about the network topology suggests that in addition to clustering, there can be also other ways to utilize them in WSN spatial organization and data processing and routing.

Reference broadcast based WSN time synchronization, which uses the recursive clock skew estimation presented in Publication 4, has two main novelties. First, the use of ML-based clock skew estimator (MLE-EIT) suits well for correlated time measurements because it does not produce as big estimation error variance as LSE-methods, which are often used. Second, time-varying transmission delays, which can have a remarkable effect in WSN communication, are taken into account in the computation. Numerically a more stable variation of the estimator is also presented to reduce its numerical sensitivity. In addition to Publication 4 simulations which indicate a better performance of the developed time synchronization method compared to the LSE-RPT and LS linear regression, developed time synchronization is also used in the implementation of device-free localization to the UWASA Node, which is discussed in the context of situational awareness in Publication 9.

6.1.2 Security

Security at all levels of communication, data processing and sharing must be taken into account all the way, from the beginning of the wireless automation system design. In many cases the security functions cannot be implemented to the system afterwards or if they can, the implementation does not allow them to operate in full efficiency. In such a case it might be cheaper and less risky to re-build the whole automation system in the case of security problems instead of trying to implement partial security to the existing one. The lack of proper security has also been one of the main concerns in wireless automation. For this reason it is included in both existing standards, WirelessHART and ISA 100.11a. In the former, the security is forced so that the system cannot be implemented without using also the security functions. In the latter, the use of them is optional.

Security is out of the main focus of this dissertation, but its importance is emphasized in the content, and it is also considered in general level in Publication 5.

6.1.3 Platform Planning and Design

Publication 6 presents a platform-based planning and design process for wireless automation. The process uses a novel combination of product, application and technology platform. It targets to reach as high level of platform genericity as possible to be able to make rapidly different products by utilizing that platform, as illustrated in Figure 18. Actually that principle applies to both application platform and product platform. Once the targeted genericity is reached with the application platform so that it is possible to make rapidly different applications by utilizing it, the platform design must be revised from commercialization and productization point of view. The generic product platform is then reached by making the modifications they may require.

In addition to platform-based planning and design process presented in Publication 6, the expectations and requirements of WSNs were mapped by interviewing industrial and academic experts in the context of GENSEN project (Virrankoski 2012). These interviews were made in year 2011. When comparing the outcomes in that year to the situation on 2018-2019, the following observations can be made. During the time of the interviews there was a common view that the development of wireless automation was in the area of rapid growth of the hype curve, which typically includes also some extra enthusiasm with some unrealistic expectations. It was also expected that the standardization will play one key role in the development of the wireless automation sector. These assumptions have been proceeding in the expected way. WirelessHART and ISA 100.11a standards have remarkably increased industrial investments for wireless automation. The area is also diversifying to other communication technologies, as discussed in subchapter 6.2. As it was assumed in the interviews (Virrankoski 2012), subcontracting based product development and services have created new business, and there are more companies focusing on system integration.

On the other hand, the hybrid solutions of RFID and WSN technologies have not become so common and advanced, what was expected in year 2011, and the development of wireless field buses is still on its way. The convergence and interoperability of different networking technologies and IP-based networking seems to have a bigger role what was expected during the time of the interviews. Many important factors pointed out in the interviews, such as device performance in terms of data sampling; transmission and computation capacity; power consumption and power supply; communication reliability; easiness to configure, assemble and maintain; extendibility and interfacing to the other parts of the automation system, are still important issues to consider in wireless automation design.

Wireless sensor platform the UWASA Node, which is presented in Publication 7, was developed by following the platform approach presented in Publication 6 and the discussed industrial expectations (Virrankoski 2012). The performance of the developed platform was then evaluated trough several applications in several research projects (Publications 8-13). One updated version of the platform and two tailored versions for industrial use were made later on. Targeted higher level of genericity was achieved, but with a cost of increased software complexity. Since there is a wide variety of architectures the platform can support, the software configuration for each of them requires more work than it would require for a couple of pre-specified architectural options. On the other hand, the UWASA Node as presented in Publication 7, is a research platform, where one of the targeted properties is the possibility to make as many different software and hardware configurations as possible by using the same platform. In the versions that are made for industrial use, the platform is typically simplified by keeping the features that are needed and removing the other ones.

6.1.4 Applications

Publication 8 results indicate that WSN is a well suited solution for greenhouse monitoring. Horticulture offers one promising area to develop so-called green technologies to reduce CO2 emissions. Additional CO2 can be used there to increase its content inside the greenhouse. By doing so, a remarkable part of the used CO2 can be tied to the biomass. From the wireless automation point of view, the control of this process requires CO2 sensors, which are feasible to use in wireless sensor nodes. There are some which are currently suitable for that purpose as the one used in Publication 8, but compared to many other types of sensors, their price and power consumption are still relatively high. In addition to monitor the greenhouse, WSN can also be used to control the greenhouse conditions in a distributed manner based on the control spesifications and the measured data. This requires distributed computation in the network and interfacing between the WSN and the actuators that control the greenhouse conditions, such as heating, ventilation, irrigation, CO2 adding and lighting systems.

Indoor situational awareness system presented in Publication 9 was able to produce indoor situation model by fusing the measured data and other sources of information. It was shared in real-time with the friendly forces operating in the building and with the upper echelons of the command chain. It is also beneficial that the presented system sub-entities can operate independently, if some of them are disabled or otherwise not in use. The communication security and its

tolerance to electronic warfare require further development. The COP server should be distributed to several locations to reduce the system vulnerability. More advanced algorithms can also be added to the data processing. Indoor navigation system accuracy can be improved and the matching between digital building interior maps and the measurements of the wearable sensors, including camera, can be included to the indoor navigation. In addition to localizing and tracking persons by using the device-free localization, the location and tracking data can be used to recognize person's activities to extract further information about the situation in the building. In Publication 9, the device-free localization was performed using a WSN that was deployed inside the building. With suitable frequencies the monitoring can be done through the walls, which enables a new type of through-wall sensing. Especially, if the sensor nodes would be mounted to drones that can be sent to surround the building or a certain part of it. Obviously the accuracy of the through-wall sensing will also depend on the thickness and the material of the walls. In addition to the recognition of persons, the radio tomography can also be used to monitor, for example, the way how somebody is breathing. This information can be utilized in advanced data processing to recognize the activities of the monitored persons and, for example, to differentiate between hijackers and hostages in the case of the hostage situation.

DFL can be utilized in many types of applications, since it enables the counting and monitoring of persons in a privacy preserving way by using just radio signal without cameras. These applications can be related e.g. elder people care in assisted living, childcare, people monitoring for elevator control, monitoring of employees presence in risky and restricted access areas in industrial environment etc.

In Publication 10, speaker identification is performed such that acoustic samples are collected by using the wireless sensor nodes. Then the samples are transmitted to PC, where the feature matrix is computed and compared against the feature matrices of the acoustic samples, which are stored in the database. Instead of transmitting the complete measured acoustic samples from the network to the PC, the cepstral coefficients and their first and second order derivatives could be computed in the node. It would remarkably reduce the communication load, because then only feature matrix would be transmitted from the network to the centralized computer. Moreover, it would also improve the system security, because if an unfriendly third party captures the wireless transmission, it would benefit less from the feature matrix compared to the complete acoustic sample.

Data required for the speaker identification can be reduced further based on the fact that human speech exists in a certain frequency band. Only the respective frequencies can be selected from DFT for further processing by using some suitable technique such as Goerzel algorithm. On one hand the amount of data needed for speaker identification can be reduced, but on the other hand the reduction of the data may also weaken the matching accuracy. Figuring out this relationship and finding the best tradeoff would be a topic for further research.

In addition to speaker identification, the technique presented in Publication 10 can be used to identify also other kind of recognizable voices. It could be utilized, for example, in condition monitoring to recognize electric transformer malfunctioning based on the type of its humming or valve malfunctioning based on the voice of its spindle.

The results of Publication 11 indicate that it is possible to estimate the number of persons by applying the BSS to mixed speech signal in the case of low-quality samples, which are collected by using WSN. A remarkable limitation in the presented method is that it requires at least as many separate measurements (as many sensor nodes) as there are speakers. The use of the under constrained case, where there are less measurements than source signals, would be an interesting area for further research. The result of the second experiment in Publication 11 is much weaker than the result of the first one, since the difference of magnitude in the eigenvalues is hardly recognizable there. It is mainly explained by the time synchronization error between the measuring sensor nodes. Once the error increases, it remarkably weakens the result as indicated by the further experiments in Publication 11.

Another field for further research based on Publication 11 results would be the use of BSS to improve the speaker identification result. In Publication 10, the speaker identification was performed by using noisy and low-quality signals as they are. Instead of doing so, the signals could be first separated from the background noise or simultaneous mixed signals by using the BSS and then the speaker identification could be performed by using the separated signals. Presumably this procedure would improve the speaker identification accuracy.

A joint use of frequency converters and WSN, which is presented in Publication 12, can be utilized in many wireless automation applications. Electricity distribution, processes or machines driven by the frequency converters can be controlled by using the WSN measurement data. This enables local and distributed control in the network so that the control algorithm can be executed locally in the sensor nodes and actuators instead of swapping all the data between the centralized control computation and the networked devices.

Ability to collect wirelessly data from the frequency converters provides new technical opportunities especially in such applications, where the frequency converters cannot be cabled, because of the mobility or harsh environment. The data collected from the frequency converters can be used for the system control and condition monitoring. The latter area is in particular interesting, because the analysis of the frequency converter data can provide information about the machine or process what it drives.

Publication 13 presents an energy harvesting solution for the UWASA Node. Developed and implemented solution uses a solar cell, but it is designed so that other power harvesters can be added to the same architecture. By doing so, the power can be extracted simultaneously from different harvesting sources and also from the battery, if needed. If the temporal power consumption is less than the harvested energy at that time moment, the battery is charged. The application indicates that there exist feasible solutions for the energy harvesting. The ability to use simultaneously several energy sources is also a benefit in the developed architecture. Rechargeable battery balances the variations that usually exist in the harvested power. It also helps to avoid the wasting of the harvested energy, because the amount harvested over the immediate consumption can be saved by charging the battery.

If the energy harvesting from the operation environment is preferred as an energy source for the WSN, it must be taken into account in the network and energy harvesting solution planning. The WSN performance requirements and their power need must be balanced with the energy resources which are possible to harvest with the existing or targeted energy harvesting solution in that particular environment. The energy consumption varies remarkably in different applications even if the same sensor platform is used. From the energy consumption point of view, there can be simple measure-and-transmit type of WSN applications on the one end, and applications that require high frequency sampling, intensive computation and high data transmission capacity on the other end.

6.2 Today and in the Future

6.2.1 State of the Art

The results and discussion presented in this dissertation covers a time span for more than ten years. During that time, the WSN technology has been maturing from emerging to partially standardized, as discussed in Chapter 3. The efficiency

of sensor platforms have remarkably increased as discussed in Chapter 4, and many issues related to communication and power supply have been continuously improved. This has created high expectations in the markets and encouraged industry to invest more in wireless automation. Instead of handling the wireless automation as an additional component or extension, it has become a natural part of the automation system options and specifications, and can be shipped and serviced as a part of the system. There are concepts such as IoT, Industrial Internet, Industry 4.0 and system of systems, where WSNs and wireless automation have an important role as well as they have in the whole ongoing digitalization hype. This has also created new type of business, as it was assumed to happen in the expert interviews on 2011 (Virrankoski 2012).

6.2.2 IP-Based Integration

System control and management requires integrated communication between different subsystems from which the system consists of. Since the Internet communication is IP-based and it offers a supreme coverage jointly with cellular networks, the IP-based communication seems to be the technology that integrates the different communication networks also in wireless automation. The local WSN does not necessarily need to be IP-based itself, but it must be interfaced to IP-based networks and support IP-based communication.

This integration enables a wider distribution of system services. For example the data storage, data analytics, user interface and even the computation of control algorithms that utilize the WSN measurement data, can all be services which are offered as a cloud service for the local wireless automation system over the Internet. This requires a careful planning of the communication security and reliability to create fast, robust and secure communication pipes and virtual networks for wireless automation inside the public Internet and cellular networks.

6.2.3 Diversification of WSN Technologies

At the same time when the communication is integrating as IP-based on the higher level, the underlying WSN technology is diversifying. The early view about smart dust is branched into several development paths.

In remote metering and remote monitoring applications, it is enough to measure rarely (like once in an hour) and simply transmit the measured data further without any processing on the node or network level. In these applications the

sensor platform architecture can be very simple, since there is just one or a couple of sensors in the node, and data processing on the node level is not performed. Long-range and low-power communication technologies such as Narrowband IoT (NB-IoT), Sigfox and LoRa are favored. Since the main purpose of the network is to collect data in a wireless manner, a star topology, where each node communicates directly with the gateway and the network nodes do not communicate directly with each other, is preferred. In these applications the power consumption is often very low, which enables several years operation times with a single battery. In some cases it is also easy to harvest the required energy from the environment.

In some wireless automation applications the final target is to fully integrate the WSN properties so that instead of using separate sensor nodes, the sensors, microprocessors and radios are directly integrated into the devices which operate in the system. In these applications the system requirements must be well defined and the applied technology well established so that the possible WSN technology updates do not require major updates to the devices. In addition to short range communication technologies, the integrated solution can enable communication over cellular network. The handy part in these applications is that there is no need for a separate WSN installation, service and integration with the rest of the system. On the other hand, a remarkable part of the redundancy and flexibility, which is usually provided by WSN, cannot be utilized.

There are also such WSN applications, where the network consists of wireless sensor nodes and actuators. They can communicate with each other either directly or over multi-hop path. Wireless communication is interfaced with the other parts of the communication system, and it supports IP-based networking. In these applications each node can be equipped with one or several sensors, and part of the data processing and control algorithms can be executed in the network in a distributed or locally centralized manner. This application type covers also the ones discussed in this dissertation. As discussed in Chapter 4, there are currently several sensor nodes commercially available with most of the software required for their basic use. The computation and communication capacity of the nodes is also increasing. Since the embedded computers are becoming smaller, cheaper and more efficient, the differentiation between them and sensor nodes is blurring. There are embedded computers like the different versions of Rasberry Pi and some others, which can act as sensor platforms and are close to them in terms of size and power consumption, but usually they are more efficient and more expensive.

The application area of “smart dust”, which uses micro- or nanoscale devices with very low energy consumption, also exists. In terms of commercial potential it still belongs more to the category of emerging technologies than commercially available ones, but the technology development in this area is fast. Some other interesting technologies, which may reshape the area of WSNs in the next few years, are the joint use of sensor systems and drones, passive communication technologies, fifth generation cellular mobile communications (5G) and wireless communication between devices using light (Li-Fi).

6.2.4 Role of IEEE 802.15.4 and Open Source Platforms

As discussed in the previous subchapter, WSN technologies are diversifying and there are also other available options than IEEE 802.15.4. There are weaknesses in IEEE 802.15.4 based WSNs, such as limited data transmission capacity, weak penetration through the obstacles once the transmission power is kept on the frame defined for the license-fee ISM band, and problems with the co-existence with WLAN (IEEE 802.11) networks. In practice, the WLAN may easily jam IEEE 802.15.4 communication in 2.4 GHz ISM band, if the network coexistence is not taken into account in the planning of the channel use. Issues related to communication reliability in terms of packet loss rate and time variant transmission delays are also easier to handle in some other frequency bands.

There are also benefits in IEEE 802.15.4, which will keep it among the important communication protocols in wireless automation and IoT in the nearby future. Having a global license-free ISM band of 2400–2483.5 MHz and local license-fee ISM bands of 868.0–868.6 MHz in Europe and 902–928 MHz in North America is a big advantage. So is the fact that IEEE 802.15.4 is supported by two standards for wireless automation; WirelessHART and ISA 100.11a. It also forms the basis of ZigBee, MiWi, 6LoWPAN, Thread and Snap specifications. This makes IEEE 802.15.4 a strong communication standard in the area of low-power wireless short range networks, and it includes also the support for IP-based networking.

There are several sensor platforms supporting IEEE 802.15.4 available by commercial producers. In addition to hardware, they include software for communication and network management, drivers for the sensors etc. Usually there is also product support available by the producer and developer websites supported by the producer and by the user community. This narrows down the role of the open source platforms, but does not dissolve it. The development of node level computation in WSNs for distributed and locally centralized operations and data processing are still on its early levels. Commercial software is

usually closed and connected to certain type of platform, which makes it unsuitable for the research and development work, but open source allows the developer to modify everything and test different things with the platform.

In addition to distributed and locally centralized operations, also many management and control issues related to WSNs are still under intensive research. These include e.g. energy efficiency, time synchronization, handling and compensation of time variant communication delays, improving the communication reliability by applying retransmission requests or estimation of lost data packets, and detecting and correcting outliers from the measured data. Open source platform enables the development and testing of different techniques.

In terms of Publication 6 and Figure 18, the open source platform can be seen as an application platform to enable rapid testing and development of different applications. Once the targeted application is made, the existing commercial platforms suitability for the application commercialization will be evaluated as a part of the productization process. If suitable solution cannot be based on them, new commercial product can be designed by utilizing the open source platform that was used in the application development.

Wireless sensor platform hardware expires technically much faster than software or algorithms behind the software. This must be taken into account in the open source platform design. There must be software compatibility between different versions of hardware so that the software of the older version of platform hardware operates also with newer version of hardware. As a consequence, there should be a certain level of independency between open source platform software and hardware, but some general guidelines about the modular hardware architecture must be kept between the different platform versions, and use them as a guideline also in the software design.

The design of new hardware versions is relatively time consuming and the production of small hardware series is relatively expensive. Thus, it would be beneficial to have compatibility between open source software platform and commercial hardware platform so that the open source WSN developer community would not need to take care of the hardware development. In addition to commercial hardware compatibility, it is also beneficial to have simulators which enable to simulate the open source sensor network as software before actual implementation.

7. CONCLUSIONS

This doctoral dissertation focuses on open source platform development in wireless automation under IEEE 802.15.4 standard. Research method is empirical. First, the state of the art is discussed by giving an overview about WSNs, wireless automation and standardization related to the use of IEEE 802.15.4 based WSNs for wireless automation. Since this work covers a time span for more than ten years, also the development of wireless sensor platforms during that time is included to the introductory part.

In the WSN applications for wireless automation, the network must automatically initialize itself for use, adapt for changes and heal from malfunctioning situations. First part of the results, Publications 1-4, are related to localization, clustering and time synchronization, which all belong to the WSN initialization and control. Security is not in the focus of this dissertation, but it is a necessary part of feasible wireless automation solution and must be taken into account since the early levels of the system design. Because of that, it is discussed in the context of unified privacy preserving model, which is presented in Publication 5.

A platform based approach, which targets to design a generic open source sensor platform, was selected as a design method and presented in Publication 6. The design targets were further focused by interviewing the experts from the academia and industry (Virrankoski 2012). Generic and modular sensor platform, the UWASA Node presented in Publication 7, was developed as an outcome of this process.

The UWASA Node is then used in most of the applications presented in Publications 8-13, though also Sensinode Microseries is used in the greenhouse monitoring presented in Publication 8, and Mica 2 in the BSS presented in Publication 11.

Based on the implementation results, a wireless sensor and actuator network that was based on the UWASA Nodes was a feasible solution for many types of wireless automation applications. It was also possible to interface it with the other parts of the system. Based on these results it can be concluded that the targeted level of genericity was achieved. However, it was also observed that the achieved level of genericity increased the software complexity. Since many different options are possible by using the same platform, each new configuration requires also more work with the software what it would require in the case of a

sensor platform, which would be targeted to a more narrow application sector with more limited number of configuration options.

The development of commercial sensor platforms, which support IEEE 802.15.4 sensor networking, has narrowed down the role of open source sensor platforms, but they are not disappearing. Commercial software is usually closed and connected to certain platform, which makes it unsuitable for research and development work. Even though there exists many commercial WSN solutions and the market expectations in this area are high, there is still a lot of work to do before the visions about IoT are reached, especially in the context of distributed and locally centralized operations in the network. In terms of control engineering, one of the main research issues is to figure out how the well-known control techniques must be applied in wireless automation where WSN is part of the automation system. For example, the distributed and limited WSN computation and communication resources, time-variant communication delays, the existence of missing and misleading data, sensor platform energy supply, and time synchronization will set some new conditions, which must be taken into account in the control design. Open source platforms offer an important tool in this research and development work.

The hardware expires much faster than software and algorithms behind the software. This must be taken into account in the open source wireless sensor platform development. Older software versions must have compatibility with the new hardware so that every hardware update does not automatically require software update. Tools for software based network simulation and configuration can also be used as a part of the open source platform development, but in addition to simulations, also implementations are needed to figure out and test the WSN operation. Open source software compatibility with commercial hardware platforms would be beneficial in research and development.

REFERENCES

Aamodt, K. (2008). Chipcon Products from Texas Instruments. Application Note AN042 (Rev. 1.0).

Acharjya D., Geetha M. (ed.) (2017). Internet of Things: Novel Advances and Envisioned Applications. Studies in Big Data, vol 25. Springer, Cham, ISBN 978-3-319-53470-1.

Achlioptas, D. and McSherry, F. (2001). Fast Computation of Low Rank Matrix Approximations. In the Proceedings of the 33rd Annual ACM Symposium on Theory of Computing, pp. 611–618, 2001.

Ahonen, T. (2008). Greenhouse Monitoring with Wireless Sensor Network. *Master's Thesis, University of Vaasa, Department of Computer Science,* 2008.

Alippi, C. (2014). Intelligence for Embedded Systems – A Methodological Approach. Springer 2014, ISBN 978-3-319-05278-6.

Allan, D. (1987). Time and frequency (time-domain) characterization, estimation, and prediction of precision clocks and oscillators. IEEE Transactions of Ultrasonic Ferroelectric Frequency Control, 1987, 34, (6), pp. 647–654.

Amis, A. D., Prakash, R., Vuong, T. H. P. and Huynh, D. T. (2000). Max-Min D-Cluster Formation in Wireless Ad Hoc Networks. In the Proceedings of IEEE INFOCOM 2000.

Bandyopadhyay, S. and Coyle, E. J. (2003). An Energy Efficient Hierarchical Clustering Algorithm for Wireless Sensor Networks, Proceedingsof IEEE INFOCOM 2003.

Basagni, S. (1999). Distributed Clustering for Ad Hoc Networks. International Symposium of Parallel Architectures, Algorithms and Networks (I-SPAN'99), Fremantle, Australia, June 23-25, 1999.

Bailey, T. and Durrant-Whyte, H. (2006). Simultaneous localization and mapping (SLAM): Part II. IEEE Robot. Autom. Mag., vol. 13, no. 3, pp. 108-117, Sep. 2006.

Beagleboard official website (2018) [Web document]. Available at http://beagleboard.org/

Bertoni, H. L. (1999). Radio Propagation for Modern Wireless Systems. Prentice Hall 1999, ISBN: 0130263737.

Bertsekas, D. (2000). Dynamic Programming and Optimal Control, vol 1. Athena Scientific, 2000.

Bimbot, F., Bonastre, J.-F., Fredouille, C., Gravier, G., Magrin-Chagnolleau, I., Meignier, S., Merlin, T., Ortega-Garcia, J., Petrvovska-Delacretaz, D. and

Reynolds, D. A. (2004). A tutorial on text-independent speaker verification. EURASIP Journal on Applied Signal Processing, vol. 4, pp. 430–451, 2004.

Björkbom, M. (2010). Wireless Control System Simulation and Network Adaptive Control. Doctoral dissertation, Aalto University, School of Science and Technology, ISBN 978-952-60-3461-4.

Bogert, B. P., Healy, M. J. R. and Tukey J. W. (1963). The quefrency analysis of time series for echoes: cepstrum, pseudo-autocovariance, cross-cepstrum and saphe cracking. In the Proceedings of the Symposium on Time Series Analysis, New York, USA, pp. 209–243, 1963.

Chen, J., He, S. and Sun, Y. (2014). Rechargeable Sensor Networks – Technology, Theory and Application. World Scientific Publishing Co., ISBN 978-9814525459.

Cmu cam website (2018) [Web document]. Available at http://cmucam.org

Contiki operating system official website (2018) [Web document]. Available at http://www.contiki-os.org

Cramér, H. (1946). Mathematical Methods of Statistics. Princeton, NJ: Princeton Univ. Press. ISBN 0-691-08004-6.

Cuhac, C. (2011). Camera Integration to Wireless Sensor Node. Master's Thesis, University of Vaasa, Department of Computer Science, Communications and Systems Engineering Group, 2011.

Dargie, W. & Poellabauer, C. (2010). Fundamentals of Wireless Sensor Networks: Theory and Practice. Wiley 2010, ISBN: 978-0-470-97568-8.

Das, M. and Banoj, P. (2011). Prototyping Wireless Sensor Node using FPGA for Mines Safety Application. ACEEE Int. J. on Electrical and Power Engineering, Vol. 02, No. 02, August 2011.

De Loera, J. A., Rambau, J. and Santos, F. (2010). Triangulations, Structures for Algorithms and Applications. Algorithms and Computation in Mathematics. 25. Springer.

Demir, A., Mehrotra, A., Roychowdhury, J. (2000). Phase noise in oscillators: a unifying theory and numerical methods for characterization. IEEE Trans. Circuits Syst. I, Fundam. Theory Appl., 2000, 47, (5), pp. 655–674.

Dissanayake, M. W. M. G., Newman,P., Clark, S., Durrant-Whyte, H. and Csorba, M. (2001). A solution to the simultaneous localization and map building (SLAM) problem. IEEE Trans. Robot. Autom., vol. 17, no. 3, pp. 229-241, Jun. 2001.

Dunkels, A. (2003). Full TCP/IP for 8 Bit Architectures. In the Proceedings of the First ACM/Usenix International Conference on Mobile Systems, Applications and Services (MobiSys), San Francisco, USA.

Dunkels, A. (2004). Contiki – a lightweight and flexible operating system for tiny networked sensors. In the Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks, pp. 455–462.

Dunkels, A., Österlind, F. and He, Z. (2007). An adaptive communication architecture for wireless sensor networks. Proceedings of the Fifth ACM Conference on Networked Embedded Sensor Systems (SenSys), Sydney, AU.

Dunkels, A. (2011). The ContikiMAC Radio Duty Cycling Protocol. Swedish Institute of Computer Science, Technical Report T2011:13, ISSN 1100-3154.

Durrant-Whyte, H. and Bailey, T. (2006). Simultaneous localization and mapping: Part I. IEEE Robot. Autom. Mag., vol. 13, no. 2, pp. 99110, Jun. 2006.

Durrett, R. (2000). Probability: theory and examples (4th ed.). Cambridge University Press. ISBN 0-521-76539-0.

Durvy, M., Abeillé, J., Wetterwald, P., O'Flynn, C., Leverett, B., Gnoske, E., Vidales, M., Mulligan, G., Tsiftes, N., Finne, N. and Dunkels, A. (2008). Making sensor networks IPv6 ready. Proceedings of the Sixth ACM Conference on Networked Embedded Sensor Systems (SenSys) (poster session), Raleigh, NC, US.

Elson, J., Girod, L. and Estrin, D. (2002). Fine-grained network time synchronization using reference broadcasts. ACM SIGOPS Operating Systems Rev., 2002, 36, (SI), pp. 147–163.

Engel, A., Friedmann, A., Koch, M., Rohlfing, J., Siebel, T., Mayer, D. and Koch, A. (2014). Hardware-accelerated wireless sensor network for distributed structural health monitoring. 2nd International Conference on System-Integrated Intelligence: Challenges for Product and Production Engineering; Elsevier ScienceDirect Procedia Technology 15 (2014) 737-746.

Eriksson, L, Elmusrati, M., Pohjola, M. (ed.) (2008). Introduction to wireless automation - Collected papers of the spring 2007 postgraduate seminar, Helsinki University of Technology, Department of Automation and Systems Technology.

Ester, M., Kriegel, H-P., Sander, J. and Xu, X. (1996). A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases With Noise. In the Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining (KDD'96), Portland, Oregon, 1996.

Evans, D. (2011). The Internet of Things – How the Next Evolution of the Internet is Changing Everything. Cisco Internet Business Solution Group (IBSG), April 2011.

FieldComm Group (2018) [Web document]. WirelessHART specification at FieldComm Gorup website. Available at: https://fieldcommgroup.org/

Flammini, A., Ferrari, B., Marioli, D., Sisinni, E. and Taroni, A. (2007). Sensor networks for industrial applications. In the Proceedings of 2nd International Workshop on Advances in Sensors and Interface, 26-27 June 2007, Bari, Italy.

Flammini, A., Ferrari, B., Marioli, D., Sisinni, E. and Taroni, A. (2009). Wired and wireless sensor networks for industrial applications. Microelectronics Journal Volume 40, Issue 9, September 2009, pp. 1322-1336.

Floyd, R. W. (1962). Algorithm 97: Shortest Path. Communications of the ACM. 5 (6): 345.

FreeRTOS official website (2018) [Web document]. Available at https://www.freertos.org

Frotzscher, A., Wetzker, U., Bauer, M., Rentschler, M., Beyer, M., Elspass, S. and Klessig, H. (2014). Requirements and current solutions of wireless communication in industrial automation. In the proceedings of IEEE International Conference on Communications (ICC 2014), W8: Workshop on 5G technologies.

Furui, S. (1981). Cepstral analysis technique for automatic speaker verification. IEEE Transactions on Acoustics, Speech and Signal Processing, vol. ASSP-29, NO. 2, pp. 254–272, April 1981.

Ganeriwal, S., Kumar, R. and Srivastava, M. (2003). Timing-sync protocol for sensor networks. In the Proceedings of the First International Conference on Embedded Networked Sensor Systems, ACM, 2003, pp. 138–149.

Ghiasi, S., Srivastava, A., Yang, X. and Sarrafzadeh, M. (2002). Optimal Energy Aware Clustering in Sensor Networks. Sensors 2002, 2, 258-269.

Goh, K. M., Ong, S. H., Joe, Y. Y., Kusolpalin, P., Moh, W. P. and Ling, K. V. (2012). FPGA based wireless sensor node for distributed process monitoring. In the Proceedings of 7th IEEE Conference on Industrial Electronics and Applications (ICIEA), 18-20 July, 2012, Singapore.

Ham, F. M., Faour, N. A. and Wheeler, J. C. (2000). Infrasound Signal Separation Using Independent Component Analysis, SPIE, Vol. 4055, Orlando, Florida, USA, April 2000.

Han, C-C., Rengaswamy, R. K., Shea, R., Kohler, E. and Srivastava, M. (2005). SOS: A dynamic operating system for sensor networks. *In the proceedings of the 3rd international conference on mobile systems, applications and services MobiSYS'05*.

Hatler, M., Gurganious, D. and Kreegar, J. (2018). 802.15.4 IoT Markets – Zigbee, Thread, 6LoWPAN, Wi-SUN and Others. ON World Research market dynamics report, May 2018.

Henning, M. (2004). A new approach to object-oriented middleware. IEEE Internet Comput., vol. 8, no. 1, pp. 6675, Feb. 2004.

Hill, J. & Culler, D. (2002). Mica: A Wireless Platform for Deeply Embedded Networks. *IEEE Micro 22(6)*, 12-24.

Hill, J., Horton, M., Kling, R. and Krishnamurthy, L. (2004). The Platforms Enabling Wireless Sensor Networks. *Communications of the ACM*, June 2004/Vol. 47, No. 6.

IEEE 802.15.4 documentation (2018) [Web document]. Available at: http://grouper.ieee.org/groups/802/15/pub/TG4Expert.html

Industrial Internet Consortium (2018) [Web document]. Available at: https://www.iiconsortium.org/

ISA100 Wireless Compliance Institute (2018) [Web document]. Available at https://isa100wci.org/

Jakobson, I. (1993). Is Object Technology Software's Industrial Platform? *IEEE Software*, Vol.15, 24-30.

Juan, B. H., Rabiner, L. R. and Wilpon J. G. (1987). On the use of band-pass liftering in speech recognition. IEEE Transactions on Acoustics, Speech and Signal Processing, vol. ASSP-35, no. 7, pp. 947–954, July 1987.

Kanugo, T., Mount, D. M., Netanyahu, N. S., Piatko, C. D., Silverman, R. and Wu, A. Y. (2002). A Local Search Approximation Algorithm for k-Means Clustering. In the Proceedings of the 18th Annual ACM Sympposium on Computational Geometry, 2002, 10-18.

Khaitan, S. K. and McCalley, J. D. (2015). Design Techniques and Applications of Cyberphysical Systems: A Survey. IEEE Systems Journal, Vol. 9. Issue 2, June 2015.

Klein, D., Kamvar, S. D. and Manning, C. D. (2002). From Instance level Constraints to Space-level Constraints: Making the Most of Prior Knowledge in Data Clustering. The Nineteenth International Conference on Machine Learning (ICML-2002), Sydney, Australia, July 8-12, 2002.

Koivo, H. N. and Elmusrati, M. (2010). Systems Engineering in Wireless Communications. Wiley 2010, ISBN 9780470021781.

Korkalainen, M., Tukeva, P., Lindholm, M. and Kaartinen, J. (2012). Hybrid localization system for situation awareness applications. In the Proceedings of 3rdWOWCA, Apr. 2012.

Kumar, C. K. S., Sukumar, R. and Nageswari, M. (2013). FPGA Implementation for Energy Efficiency in Secure Wireless Sensor Node – A Critical Review. International Journal of Emerging Research in Management & Technology, ISSN: 2278-9359.

Lennvall, T., Svensson, S. and Hekland, F. (2008). A Comparison of WirelessHART and ZigBee for Industrial Applications. In the proceedings of 2008 IEEE International Workshop on Factory Communication Systems.

Liao, J., Singh, B. K., Khalid, M. A. S. and Kemal, T. E. (2013). FPGA based wireless sensor node with customizable event-driven architecture. EURASIP Journal on Embedded Systems 2013, 2013:5.

Low, A. (2013). Evolution of Wireless Sensor Networks for Industrial Control. Technology Innovation Management Review, May 2013.

Lymberopoulos, D. & Savvides, A. (2005). XYZ: A Motion-Enabled, Power Aware Sensor Node Platform for Distributed Sensor Network Applications. *In the Proceedings of The Fourth International Symposium on Information Processing*

in Sensor Networks (IPSN'05), Track on Sensor Platform, Tools and Design Methods for Embedded Systems (SPOTS).

Mahmood, A. and Jäntti, R. (2009). Time synchronization accuracy in real-time wireless sensor networks. In the Proceedings of IEEE Ninth Malaysia Int. Conf. on Communications (MICC), 2009.

Maróti, M., Kusy, B., Simon, G. and Lédeczi, A. (2004). The flooding time synchronization protocol. In the Proceedings of the Second International Conference on Embedded Networked Sensor Systems, 2004, ser. SenSys'04, pp. 39–49.

Matlab fminsearch function description (2018) [Web document]. Available at https://www.mathworks.com/help/matlab/ref/fminsearch.html

Matusiak, M., Paanajärvi, J., Appelqvist, P., Elomaa, M., Ylikorpi, M. and Halme, A. (2008). A novel marsupial robot society: Towards long-term autonomy. In the Proceedings of 9th International Symposium DARS, Nov. 2008, pp. 523-532.

McGrath, M. E. (1995). Product Strategy for High-Technology Companies. *New York, Irwin Professional Publishing*.

McQueen, J. B. (1967). Some Methods for Classification and Analysis of Multivariate Observations. In the Proceedings of the Fifth Symposium on Math, Statistics and Probability (pp. 281-297), 1967.

Meyer, M.H. and Lehnerd, A.P. (1997). The Power of Product Platforms: Building Value and Cost Leadership. *New York, NY: Free Press*.

Mills, D.L. (2006). Computer network time synchronization: the network time protocol. CRC Press, 2006.

Moore, G. A., Johanson, P. and Kippola, T. (1998). *The Gorilla Game: an Investor's Guide to Picking Winners in High Technology*. HarperColllins Publishers.

Mica2 datasheet (2003) [Web document]. Available at https://www.eol.ucar.edu/isf/facilities/isa/internal/CrossBow/DataSheets/mica2.pdf

Mica2dot datasheet (2003) [Web document]. Available at https://www.eol.ucar.edu/isf/facilities/isa/internal/CrossBow/DataSheets/mica2dot.pdf

Micaz datasheet (2004) [Web document]. Available at http://www.openautomation.net/uploadsproductos/micaz_datasheet.pdf

Myrsky, M., Maula, A., Saarinen, J. and Kankkunen, I. (2012). Teleoperation tests for large-scale indoor information acquisition. In the Proc. Comput. Intell. Telemat. Control Embedded Syst., vol. 1. 2012, pp. 13-18.

Niculescu, D. and Nath, B. (2004). VOR Base Stations for Indoor 802.11 Positioning. In the Proceeding of IEEE MOBICOM, 2004.

Nixon, M. (2012). A Comparison of WirelessHART and ISA 100.11a. White paper, Emerson Process Management, September 2012.

Nodes; Wikipedia list of wireless sensor nodes (2018) [Web document]. Available at https://en.wikipedia.org/wiki/List_of_wireless_sensor_nodes

Noh, K., Serpedin, E. and Qaraqe, K. (2008). A new approach for time synchronization in wireless sensor networks: pairwise broadcast synchronization. IEEE Transctions of Wireless Communications, 2008, 7, (9), pp. 3318–3322.

Oppenheim A. V. and Schafer R. W. (1968). Homomorphic analysis of speech. IEEE Transactions on Audio and Electroacoustics, vol. 16 no. 2, pp. 221–226, 1968.

Oppenheim A. V. and Schafer R. W. (1989). Discrete-time signal processing. Prentice Hall, Englewood Cliffs, NJ, USA, 1989.

Paavola, M. and Leiviskä, K. (2010). Wireless Sensor Networks in Industrial Automation. Peer-reviewed chapter in Silvestre-Blanes, J. (ed.), Factory Automation, Intech Open, March 1st 2010, ISBN: 978-953-307-024-7, pp. 201-220.

Park, C. and Chou, P.H. (2006). AmbiMax: Autonomous Energy Harvesting Platform for Multi-Supply Wireless Sensor Nodes. In the Proceedings of Third Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks Sep. 25 - 28, 2006, Reston, VA, USA.

Patwari, N. and Agrawal, P. (2008). Effects of correlated shadowing: Connectivity, localization, and RF tomography. In the Proceedings of IPSN 2008, pp. 82-93.

Piedra, A. de la, Braeken, A. and Touhafi, A. (2012). Sensor Systems based on FPGAs and their Applications: a Survey. Sensors 2012, ISSN 1424-8220.

Pister, K. (2001). Smart Dust Project [Web document]. Available at: https://robotics.eecs.berkeley.edu/~pister/SmartDust/

Pottie, G. and Kaiser, W. (2000). Wireless integrated network sensors. Commun. ACM, 2000, 43, (5), pp. 51–58.

Rabiner, L. and Juang B. H. (1993). Fundamentals of speech recognition. New Jersey, Prentice Hall, 1993.

Rao, C. R. (1945). Information and the accuracy attainable in the estimation of statistical parameters. Bulletin of the Calcutta Mathematical Society 37: 81–89.

Rasberry Pi official website (2018) [Web document]. Available at https://www.raspberrypi.org/

Saarinen, J., Paanajïärvi, J. and Forsman, P. (2011). Best-first branch and bound search method for map based localization. In the Proc. IEEE/RJS Int. Conf. Intell. Robot. Syst., Sep. 2011, pp. 59-64.

Sakthidharan G.R. and Punitha A. (2017). Wireless Sensor Network in Automation and Internet of Things. In: Acharjya D., Geetha M. (eds) Internet of Things: Novel Advances and Envisioned Applications. Studies in Big Data, vol 25. Springer, Cham, pp. 173-191, ISBN 978-3-319-53470-1.

Sanchez, R. (1996). Strategic product creation: Managing new interactions of technology, markets and organizations. *European Management Journal*, Vol. 14 pp.121 - 138.

Savvides, A., Park, H., and Srivastava, M. B. (2003). The n-hop Multilateration Primitive for Node Localization Problems. In the Proceedings of IEEE MOBICOM, 443–451, 2003.

Sensinode Microseries datasheet (2007) [Web document]. Available at http://www.ee.oulu.fi/~ikram/Downloads/microseries/sensinode-datasheet-U100R2-20070923.pdf

Sensinode Nanoseries datasheet (2008) [Web document]. Available at http://www.ee.oulu.fi/~ikram/Downloads/nanoseries/

Serpedin, E. and Chaudhari, Q.M. (2009). Synchronization in wireless sensor networks: parameter estimation, performance benchmarks and protocols. Cambridge University Press, 2009.

Shang, Y., Ruml, W., Zhang, Y. and Fromherz, M. (2003). Localization from mere connectivity. In the Proceedings of ACM MobiHoc'03, pp. 201–212.

Shang, Y. and Ruml, W. (2004). Improved MDS-Based Localization. In the Proceedings of IEEE INFOCOM, 2004.

Shen, X., Wang, Z. and Sun, Y. (2004). Wireless sensor networks for industrial applications. In the Proceedings of Fifth World Congress on Intelligent Control and Automation, 15-19 June 2004, Hangzhou, China.

Sohraby, K., Minoli, D., Znati, T. (2007). Wireless sensor networks: technology, protocols, and applications. John Wiley and Sons, ISBN 978-0-471-74300-2.

Srinivasa, S. (2013). Wireless sensor networks open new frontiers for a smarter world. *Texas Instruments white paper*, April 2013.

Stevens, S., Volkman, J. and Newman, E. B. (1937). A scale for the measurement of the psychological magnitude of pitch. Journal of the Acoustical Society of America, vol. 8, pp. 185–190, 1937.

Sundararaman, B., Buy, U., Kshemkalyani, A.D. (2005). Clock synchronization for wireless sensor networks: a survey. Ad Hoc Networks 2005, 3, (3), pp. 281–323.

Sorenson, H. W. (1980). Parameter Estimation: Principles and Problems. New York: Marcel Dekker, 1980.

Timmerman, G. J. and Kamp, P. G. H. (2003). Computerised Environmental Control in Greenhouses. PTC, The Netherlands, pp. 15–124, 2003.

TinyOS official website (2018) [Web document]. Available at https://github.com/tinyos/tinyos-main

Tynan, R., Marsh, D., O'Kane, D, and O'Hare G. M. P. (2005). Intelligent Agents for Wireless Sensor Networks. In the Proceedings of 4th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2005), July 25-29, 2005, Utrecht, Netherlands.

Vacon trademark frequency converters by Danfoss (2018) [Web document]. Available at https://www.danfoss.com/en/search/?query=vacon

Virrankoski, R. & Keskinen, S. (2009). GENSEN: A Novel Combination of Product, Application and Technology Platform Development in the Context of Wireless Automation. *In the Proceedings of 14th International Conference on Productivity & Quality Research (ICPQR 2009)*, October 19-23, Alexandria, Egypt.

Virrankoski, R. (ed.) (2012). Generic Sensor Network Architecture for Wireless Automation (GENSEN). *Proceedings of the University of Vaasa, Reports 174*, Vaasa 2012.

Virrankoski, R. (ed.) (2013). Wireless Sensor Systems in Indoor Situation Modeling II (WISM II). *Proceedings of the University of Vaasa, Reports 188*, Vaasa 2013.

Warshall, S. (1962). A theorem on Boolean matrices. Journal of the ACM. 9 (1): 11–12.

Wilson, J. and Patwari, N. (2010). Radio tomographic imaging with wireless networks. IEEE Transactions of Mobile Computing, vol. 9, no. 5, pp. 621-632, May 2010.

Wu, Y.-C., Chaudhari, Q. and Serpedin, E. (2011). Clock synchronization of wireless sensor networks. IEEE Signal Processing Magazine, 2011, 28, (1), pp. 124–138.

Yigitler, H., Kaltiokallio, O. and Jäntti, R. (2013). A management framework for device-free localization. In the Proceedings of IEEE IJCNN, August 2013.

Younis, O. and Fahmy, S. (2004). Distributed Clustering in Ad-hoc Sensor Networks: A Hybrid, Energy-Efficient Approach. Proceedings of IEEE INFOCOM 2004.

Zaane, O. R. and Lee, C-H. (2002). Clustering Spatial Data in the Presence of Obstacles: a Density-Based Approach. Sixth International Database Engineering and Applications Symposium (IDEAS 2002), Edmonton, Alberta, Canada, July 17-19, 2002.

Xi, J. and Zha, H. (2004). Sensor Positioning in Wireless Ad-hoc Sensor Networks Using Multidimensional Scaling. In the Proceedings of IEEE INFOCOM, 2004.

6LoWPAN wiki (2018) [Web document]. Available at http://6lowpan.tzi.org/.

Distance Matrix Reconstruction from Incomplete Distance Information for Sensor Network Localization

P. Drineas*, A. Javed*, M. Magdon-Ismail*, G. Pandurangan†, R. Virrankoski‡ and A. Savvides‡
*Computer Science Department
Rensselaer Polytechnic Institute, Troy, NY
Email: drinep, javeda, magdon@cs.rpi.edu
†Computer Science Department
Purdue University, West Lafayette, IN
Email: gopal@cs.purdue.edu
‡Electrical Engineering Department
Yale University, New Haven, CT
Email: reino.virrankoski, andreas.savvides@yale.edu

***Abstract*— This paper focuses on the principled study of distance reconstruction for distance-based node localization. We address an important issue in node localization by showing that a highly incomplete set of inter-node distance measurements obtained in ad-hoc node deployments carries sufficient information for the accurate reconstruction of the missing distances, *even in the presence of noise and sensor node failures*. We provide an efficient and provably accurate algorithm for this reconstruction, and we show that the resulting error is bounded, decreasing at a rate that is inversely proportional to $\sqrt{n}$, the square root of the number of nodes in the region of deployment. Although this result is applicable to many localization schemes, in this paper we illustrate its use in conjunction with the popular MultiDimensional Scaling algorithm. Our analysis reveals valuable insights and key factors to consider during the sensor network setup phase, to improve the quality of the position estimates.**

I. Introduction

In the past few years the sensor network community has reached a consensus that knowledge of node locations is unquestionably one of the most desirable attributes of ad-hoc sensor networks. Knowledge of location can support many networking and maintenance services, and more importantly map the sensed data to physical space. Since the manual recording of node positions is a difficult task even for modest sized networks, the community has invested significant effort in creating algorithms that can derive locations based on inter-node measurements.

The simplest and most common embodiment of such algorithms considers the estimation of a coordinate system from a set of pairwise distance measurements among sensor nodes. However, it is well known, that in realistic deployments obstacles and large node separations render the collection of all n^2 distances infeasible. Many of the existing algorithms try to resolve this issue by providing heuristic approximations to the missing distances. The success of such techniques has invariably been measured experimentally. There is an alarming lack of simple algorithms with bounded running time complexity – either centralized or decentralized – that are able to *provably* localize the sensor nodes up to bounded error.

The work in this paper takes a forward step in this direction, by providing a simple and provable algorithm for the accurate reconstruction of the missing pairwise distance measurements. The main contribution of this paper is to show that highly incomplete distance matrices such as the ones obtained in ad-hoc deployments, contain sufficient information to allow the accurate reconstruction of the missing distances, even in the presence of noise. To this end, we describe a provable reconstruction algorithm with bounded error and illustrate its use in conjunction with the popular Multidimensional Scaling (MDS) algorithm [12], [13], [8]. However, we emphasize that this presentation focuses on matrix distance reconstruction. We acknowledge the fact that to obtain more accurate locations an additional iterative refinement phase similar to the ones described in [13] and [14] is necessary. This presentation does not delve into the

details of iterative refinement.

Section III gives an intuitive overview of the main results, followed by a detailed description in Sections IV and V and our evaluation results in Section VII.

II. Related Work

Node localization has been a subject of intense study in the recent literature. The various approaches may be classified based on whether they are assisted or ad-hoc, centralized or distributed, or based on the type of technologies they employ. Some approaches are based on radio received signal strength [15], [13], [10], others employ more accurate distance measurement technologies [11], and others assume a combination of angle and distance measurements [4], [10].

Our work is closely related to studies that use approximations to distance measurements. These include the MDS based approaches described in [12], [13], [8]. Novel distance reconstruction techniques via SemiDefinite Programming formulations (SDP) have been recently proposed in [3], [9], [14]. Our work addresses the same problem. However, to the best of our knowledge, no explicit connection between the accuracy of the reconstruction and the number of sensor nodes in the network has been provided in existing work.

There has been significant recent theoretical work in general *matrix reconstruction* problems, a special case of which is the Euclidean distance matrix reconstruction problem. In particular, Achlioptas and McSherry in [1], [2] proved that given randomly sampled elements of a matrix, it is possible to accurately approximate the spectral characteristics – singular values and singular vectors – of a matrix. Drineas *et. al.* in [5], [6] proved that it is also possible to approximate the spectral characteristics of a matrix by sampling a small constant number of rows and/or columns of a matrix. We refer the reader to the references for further details.

III. Distance Matrix Reconstruction

A. Problem Statement

In a sensor network localization problem, n sensor nodes are placed in the two (or three)[1] dimensional Euclidean space. Every sensor measures its distance (up to noise) to a subset of the other sensors. Given this (incomplete) distance information, the task is to recover the positions of the individual sensor nodes. More formally, let $\mathbf{x}_i \in \mathbb{R}^2$ denote the position of node i, $i \in 1 \dots n$. Let d_{ij} denote the Euclidean distance between nodes i and j for $i, j \in 1 \dots n$, i.e.,

$$d_{ij}^2 = \|\mathbf{x}_i - \mathbf{x}_j\|^2 = \mathbf{x}_i^T \mathbf{x}_i + \mathbf{x}_j^T \mathbf{x}_j - 2\mathbf{x}_i^T \mathbf{x}_j.$$

Let $\mathbf{X}$ denote the $n \times 2$ position matrix whose i^{th} row is $\mathbf{x}_i^T$, and let $\mathbf{D}$ denote the $n \times n$ distance matrix given by $D_{ij} = d_{ij}^2$. We assume that the sensors are distributed on a bounded domain, so $d_{ij} \in [0, d_{max}]$. Estimates $\tilde{d}_{ij}^2 = d_{ij}^2 + \epsilon_{ij}$ are measured for some pairs of nodes, where ϵ_{ij} models the measurement noise. We assume that the noise is zero mean and has bounded variance. However, we do not assume that it is Gaussian. The goal of localization is to recover estimates $\hat{\mathbf{x}}_i \in \mathbb{R}^2$ that are "close", up to rotation/reflection and translation, to the $\mathbf{x}_i$ for all $i \in 1 \dots n$.

Existing algorithms for localization (e.g., the MDS-MAP algorithm of [12], [13]) start by using the incomplete and noisy distance information contained in the $\tilde{d}_{ij}$ to first reconstruct all the distances d_{ij}. The goal of this paper is to give provably accurate algorithms for *reconstructing* the entire distance matrix, given a small number of noisy pairwise distances $\tilde{d}_{ij}^2$. In particular, we obtain estimates $\bar{d}_{ij}^2$ for all $i, j \in 1 \dots n$ for which, modulo our assumptions,

$$\frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} (d_{ij}^2 - \bar{d}_{ij}^2)^2 = O\left(\frac{1}{\sqrt{n}}\right).$$

In words, the squared error per entry drops inversely proportional to the square root of the number of nodes in the sensor network. Thus, we lay a theoretical foundation upon which existing algorithms, such as MDS-MAP, may operate.

Notation. Let $\mathbf{1}_n$ be the n-dimensional vector of ones, and $\mathbf{I}_n$ the $n \times n$ identity matrix. For any matrix $\mathbf{A}$, $\|\mathbf{A}\|_F^2 = \sum_{i,j} A_{ij}^2$ and $\|\mathbf{A}\|_2 = \max_{\|\mathbf{y}\|=1} \|\mathbf{A}\mathbf{y}\|$.

B. MDSLocalize *Using Exact Distances*

To motivate the need for accurate reconstruction of the distance matrix, we can ask whether it is possible to recover the original positions $\mathbf{x}_i$ (up to rotation/reflection and translation), given *all* n^2 pairwise Euclidean distances, without any measurement noise. A SemiDefinite Programing approach used in [3] shows that the answer to this question is affirmative. It has been folklore knowledge that under the same assumptions, MultiDimensional Scaling (MDS) approaches do the same. We summarize the MDS algorithm below, and give a proof of Theorem 1 in the Appendix.

[1] In the interest of space, we only focus on the 2D case. The 3D case is a straight forward extension.

Algorithm MDSLOCALIZE

1) **Centering.** Compute $\boldsymbol{\tau}(\mathbf{D}) = -\frac{1}{2}\mathbf{LDL}$, where $\mathbf{L} = \mathbf{I}_n - (1/n)\mathbf{1}_n\mathbf{1}_n^T$.
2) **SVD.** Compute $\boldsymbol{\tau}_2(\mathbf{D})$, the best rank 2 approximation to $\boldsymbol{\tau}(\mathbf{D})$ using its Singular Value Decomposition, $\boldsymbol{\tau}_2(\mathbf{D}) = \mathbf{U}_2\boldsymbol{\Sigma}_2^2\mathbf{U}_2^T$.
3) **Return** $\tilde{\mathbf{X}} = \mathbf{U}_2\boldsymbol{\Sigma}_2$.

At the second step of MDSLOCALIZE, $\mathbf{U}_2$ is an $n \times 2$ matrix of the top two left singular vectors of $\boldsymbol{\tau}(\mathbf{D})$, and $\boldsymbol{\Sigma}_2$ is a 2×2 diagonal matrix. At the third step, the i^{th} row of $\tilde{\mathbf{X}}$ is the estimate $\tilde{\mathbf{x}}_i^T$.

Theorem 1: MDSLOCALIZE, when applied to the complete, exact distance matrix $\mathbf{D}$ returns estimates of the positions $\tilde{\mathbf{x}}_i$ that are equal (up to rotation/reflection and translation) to the true positions $\mathbf{x}_i$ for all i.

The above theorem immediately suggests an approach when some of the pairwise distances are missing: replace the missing entries by estimates and run MDS on this estimate of $\mathbf{D}$. Indeed, this approach has been suggested and experimentally evaluated in [12], where a missing distance between nodes i and j is *approximated* by its shortest path distance in the sensor network connectivity graph. The hope has always been that if the estimate of $\mathbf{D}$ is accurate enough, then the result of the MDSLOCALIZE procedure will mimic the statement of theorem 1. We will show here that the first step can be accomplished, namely that $\mathbf{D}$ can be reconstructed from partial information with provable accuracy. The analysis of running MDSLOCALIZE on this provably accurate reconstruction will be discussed in upcoming work.

C. Inferring Missing Distances

A crucial question naturally arises. Can one accurately approximate the missing distances, given a small subset of pairwise distances?

Lemma 1: The rank of $\mathbf{D}$ is at most 4.

Proof: Notice that

$$\mathbf{D} = \mathbf{1}_n\mathbf{z}^T + \mathbf{z}\mathbf{1}_n^T - 2\mathbf{X}\mathbf{X}^T, \tag{1}$$

where $\mathbf{z}$ is an $n \times 1$ vector whose i^{th} element is equal to $\|\mathbf{x}_i\|^2 = \mathbf{x}_i^T\mathbf{x}_i$. To conclude, observe that $\mathbf{D}$ is the sum of three matrices of ranks 1, 1, and at most 2. More generally, in d dimensions, the rank of the third matrix is at most d, giving $rank(\mathbf{D}) \leq d + 2$.

$\diamond$

This simple lemma lies at the *heart* of our work. The fact that $\mathbf{D}$ is of rank at most 4 explains, both rigorously and intuitively, the correctness of our algorithm and the quality of our bounds. Intuitively, it states that $\mathbf{D}$ has a lot of structure. Roughly speaking, even though $\mathbf{D}$ has n^2 entries, there exist only 4 linearly independent columns (or rows) in $\mathbf{D}$ or, equivalently, there exist only $8n$ degrees of freedom in $\mathbf{D}$. Thus, a carefully chosen $8n$ entries in $\mathbf{D}$ should suffice to reconstruct $\mathbf{D}$ exactly.

D. Sampling $\mathbf{D}$

As discussed, only $8n$ entries in $\mathbf{D}$ should suffice for reconstruction, and hence localization. As a motivating example, consider an idealized setting, in which we could choose which entries of $\mathbf{D}$ to measure. Suppose we pick 4 linearly independent rows of $\mathbf{D}$, say (without loss of generality) the first 4 rows. This amounts to the unrealistic assumption that we are given all distances from the first 4 sensor nodes to *all* other nodes. Assume also that we are given at least 4 entries from every other row of $\mathbf{D}$, i.e., every sensor is able to compute its distance to at least 4 other sensors (a realistic assumption). The 4 entries in row j ($j > 4$) may be used to determine the linear combination of the first 4 rows that would give the j^{th} row, and hence determine the entire j^{th} row. We know that this process is feasible, since $\mathbf{D}$ has rank at most 4. Thus, the 4 given entries in each row suffice to reconstruct the entire row. Assuming that the measurements are noiseless, the reconstruction of $\mathbf{D}$ is perfect.

The assumption that the first 4 rows are given is clearly out of reach, since this would imply the existence of 4 extremely powerful sensor nodes, which can compute their distance to any other sensor node. In a realistic setting, we do not get to choose the entries of $\mathbf{D}$ that are measured. Instead, we can postulate a reasonable model under which the entries of $\mathbf{D}$ are "sampled", and ask whether these "sampled" entries are sufficient to recover the structure of $\mathbf{D}$, even in the presence of noise. The above discussion highlights two points. (i) $\mathbf{D}$ has a lot of structure, and a carefully chosen small sample of its entries will result in accurate reconstruction. Therefore, (ii) the relevant question is what realistic assumptions on the sampling of $\mathbf{D}$ give accurate reconstruction?

We describe a general, realistic model to answer the above question. Introduce an $n \times n$ sampling matrix $\mathbf{P}$ whose (i, j)-th entry $p_{ij} \in [0, 1]$ denotes the probability that node i successfully measured its exact distance to node j, i.e., d_{ij}^2 is measured with probability p_{ij}, and is unknown with probability $1 - p_{ij}$. The measurements are corrupted, thus we measure $\hat{d}_{ij}^2 = d_{ij}^2 + \epsilon_{ij}$ with

probability p_{ij}. Recall that ϵ_{ij} are independent zero mean, bounded variance random variables.

Our model includes the commonly assumed disk model which sets $p_{ij} \approx 1$ if $d_{ij} \leq R$, and $p_{ij} \approx 0$ otherwise. Here R denotes the sensor radius. Our model implicitly allows for operation in obstructed environments and varying signal propagation models, by allowing more general values for p_{ij}.

E. Assumptions

We need to make some assumptions on the p_{ij} in order to prove that localization is, in principle, feasible. Notice that some assumptions on the p_{ij} are clearly necessary in order to give any provable guarantees for localization. For example, if all but $O(1)$ of the p_{ij} are equal to zero, localization is impossible. We state our assumptions and defer a detailed discussion of their plausibility to Section VI, after the presentation of our reconstruction algorithm.

Assumption 1: All the p_{ij}'s are known.

Even though this assumption sounds quite strong, we will argue that it is essentially implicit in existing literature. More importantly, it is actually feasible to get realistic, accurate estimates of the p_{ij} in practical settings.

Assumption 2: $p_{ij} \geq p_\epsilon > 0$, for all $i, j = 1 \dots n$, for some small positive constant p_ϵ.

In words, we assume that even far away sensors have a very small, non-zero probability of detecting their distance. This assumption might be true as sensor technology improves, or if the sensors are spread over small, bounded regions.

IV. SVD-RECONSTRUCT

We describe the reconstruction algorithm, which we will analyze in Section V. The algorithm is tantalizingly simple, and is motivated by recent important results regarding the reconstruction of low-rank matrices [1], [2].

SVD-RECONSTRUCT takes as input a fraction of the entries of $\mathbf{D}$ that are available, i.e., entries of $\mathbf{D}$ that correspond to pairs of nodes that were able to measure their pairwise distances – recall that $\tilde{D}_{ij} = \tilde{d}_{ij}^2 = d_{ij}^2 + \epsilon_{ij}$ is measured with (known) probability p_{ij}. Thus, the input to SVD-RECONSTRUCT is the matrix $\tilde{\mathbf{D}}$ given by

$$\tilde{D}_{ij} = \begin{cases} d_{ij}^2 + \epsilon_{ij} & \text{with probability } p_{ij}, \\ ? & \text{with probability } 1 - p_{ij}. \end{cases}$$

The ? denotes that the entry is unknown. The first step is to construct a new matrix $\mathbf{S}$ with entries

$$S_{ij} = \begin{cases} \frac{d_{ij}^2 + \epsilon_{ij} + \gamma_{ij}(1-p_{ij})}{p_{ij}} & \text{if } d_{ij} \text{ was detected } (p_{ij}), \\ \gamma_{ij} & \text{otherwise } (1 - p_{ij}). \end{cases}$$

$\mathbf{S}$ is well defined since the p_{ij} are known and non-zero. The γ_{ij} are values representing our "best guess" for the distance between nodes i and j, given that the two nodes were not able to detect their distance. These values naturally model side information that is available in practice. Our algorithm works for any choice for the γ_{ij}, e.g., all γ_{ij} might be set to zero. However, better choices for the γ_{ij} can improve the accuracy of the reconstruction. We will quantify this in equation (4), and in Section VII we will demonstrate the experimental performance of the SVD-RECONSTRUCT algorithm for various choices for the γ_{ij}.

The next step is to construct $\mathbf{S}_4$, the best rank 4 approximation to $\mathbf{S}$ (recall that $\mathbf{D}$ has rank at most 4).

Algorithm SVD-RECONSTRUCT

1) Given $\tilde{\mathbf{D}}$, construct $\mathbf{S}$.
2) Construct $\mathbf{S}_4$, the best rank 4 approximation to $\mathbf{S}$, using the Singular Value Decomposition of $\mathbf{S}$.
3) Run MDSLOCALIZE on $\mathbf{S}_4$ to obtain $\tilde{\mathbf{x}}_i$, $i = 1 \dots n$, which approximate the $\mathbf{x}_i$ up to rotation/reflection and translation.

The entries of $\mathbf{S}$ satisfy two important properties. Their expectation $\mathbf{E}[S_{ij}]$ is equal to d_{ij}^2 for all i and j (recall that the expectation of ϵ_{ij} is equal to zero), and their variance is bounded since the p_{ij} are bounded away from zero; see Section V for details. These two properties will allow us to use the bounds of [1], [2] to prove that $\mathbf{S}_4$, the best rank 4 approximation to $\mathbf{S}$, is "close" to $\mathbf{D}$. More specifically, we shall obtain bounds for $\|\mathbf{D} - \mathbf{S}_4\|_F^2$.

V. ANALYSIS OF SVD-RECONSTRUCT

The main goal of this paper is to lay a formal foundation for localization by giving provably accurate algorithms for reconstructing $\mathbf{D}$ from highly incomplete distance information. We now show that SVD-RECONSTRUCT is one such algorithm. Instrumental to this goal will be the fact that $\mathbf{D}$ has low rank (lemma 1).

The following lemma is crucial to the analysis. Its essential content is that $\mathbf{S}$ is an unbiased estimator for $\mathbf{D}$.

Lemma 2: For all i, j,

$$\mathbf{E}[S_{ij} - D_{ij}] = 0.$$

We give the proof in the Appendix. The lemma holds because of our careful choice of the *scaling factors*

for the entries of $\mathbf{S}$. We now show that $\mathbf{S}_4$ is close to $\mathbf{D}$, which implies that SVD-RECONSTRUCT accurately recovers $\mathbf{D}$.

Lemma 3 (Theorem 1, [2]): Let $\mathbf{S}_4$ be constructed as described in SVD-RECONSTRUCT. Then,

$$\begin{aligned}\|\mathbf{D}-\mathbf{S}_4\|_F &\leq \|(\mathbf{D}-\mathbf{S})_4\|_F \\ &+ 2\sqrt{\|(\mathbf{D}-\mathbf{S})_4\|_F \|\mathbf{D}\|_F}\end{aligned}$$

and also

$$\|\mathbf{D}-\mathbf{S}_4\|_2 \leq 2\|(\mathbf{D}-\mathbf{S})_4\|_2 .$$

The above lemma is essentially Theorem 1 of [2], using the fact that $\|\mathbf{D}-\mathbf{D}_4\|_F = \|\mathbf{D}-\mathbf{D}_4\|_2 = 0$. We now present a bound for $\|(\mathbf{D}-\mathbf{S})_4\|_F$. To prove this bound we first need to bound $\|(\mathbf{D}-\mathbf{S})_4\|_2 = \|\mathbf{D}-\mathbf{S}\|_2$. Towards that end we use Theorem 5 of [2].

Lemma 4 (Theorem 5, [2]): Let σ_S^2 denote an upper bound for the variance of the entries of $\mathbf{S}$, or equivalently, $\mathbf{Var}[S_{ij}] \leq \sigma_S^2$ for all $i,j = 1\ldots n$. Then, with probability at least $1-1/(2n)$, for sufficiently large n,

$$\|\mathbf{D}-\mathbf{S}\|_2 \leq 4\sigma_S\sqrt{2n}, \quad (2)$$

$$\|(\mathbf{D}-\mathbf{S})_4\|_F \leq 12\sigma_S\sqrt{2n}. \quad (3)$$

Combining lemmas 3 and 4 we can easily derive a bound on the quality of $\mathbf{S}_4$ as an approximation to $\mathbf{D}$.

Lemma 5: $\mathbf{S}_4$ is a "good" approximation to $\mathbf{D}$, since with probability at least $1-1/(2n)$,

$$\begin{aligned}\|\mathbf{D}-\mathbf{S}_4\|_F &\leq 12\sigma_S\sqrt{2n} + 8\sqrt{\sigma_S\sqrt{2n}\,\|\mathbf{D}\|_F} \\ \|\mathbf{D}-\mathbf{S}_4\|_2 &\leq 8\sigma_S\sqrt{2n}.\end{aligned}$$

See the Appendix for a proof of the above lemma. We now bound the σ_S term in lemmas 4 and 5. We will use the fact that ϵ_{ij} is zero mean and its variance is bounded by σ_ϵ^2. Indeed (for details see Appendix)

$$\mathbf{Var}[S_{ij}] \leq \frac{2}{p_{ij}}\left((d_{ij}^2-\gamma_{ij})^2+\sigma_\epsilon^2\right).$$

Notice that the quality of the bound improves if γ_{ij} is close to d_{ij}. Overall, using $p_{ij} \geq p_\epsilon$ (Assumption 2),

$$\sigma_S^2 \leq \frac{2}{p_\epsilon}\max_{i,j}\left((d_{ij}^2-\gamma_{ij})^2+\sigma_\epsilon^2\right). \quad (4)$$

The following theorem summarizes our results regarding the accuracy of the reconstruction process, and argues that the average reconstruction error per entry decreases *inversely proportional* to the square root of the number of nodes in the sensor network.

Theorem 2: Let $\mathbf{S}_4$ be constructed as described in the SVD-RECONSTRUCT algorithm. Then, with probability at least $1-1/(2n)$,

$$\|\mathbf{D}-\mathbf{S}_4\|_F \leq 12\sigma_S\sqrt{2n} + 8\sqrt{\sigma_S\sqrt{2n}\,\|\mathbf{D}\|_F},$$

where σ_S^2 is bounded as in equation (4). Let $d_{\max}$ denote the $\max_{i,j} d_{ij}$ over all $i,j \in 1\ldots n$. Since $\|\mathbf{D}\|_F \leq nd_{\max}$, assuming that p_ϵ is any small constant,

$$\|\mathbf{D}-\mathbf{S}_4\|_F^2 \leq O(nd_{\max}^4 + n^{3/2}d_{\max}^3).$$

Thus, the average square error per entry in $\mathbf{S}_4$ is

$$O(d_{\max}^4/n + d_{\max}^3/\sqrt{n}).$$

Assuming that $d_{\max}$ is independent of n, the error decreases inversely proportional to the $\sqrt{n}$.

VI. DISCUSSION

We briefly discuss the impact of the assumptions of Section III-E in light of the SVD-RECONSTRUCT algorithm. Consider Assumption 1. Traditionally [12], MDSLOCALIZE has been run on a reconstructed distance matrix

$$S_{ij} = \begin{cases} d_{ij}^2+\epsilon_{ij} & \text{if } d_{ij} \text{ was detected,} \\ \gamma_{ij} & \text{otherwise,}\end{cases}$$

where γ_{ij} is the shortest path distance between i and j on the sensor network connectivity graph. In the context of constructing $\mathbf{S}$, this corresponds to setting $p_{ij} \approx 1$ if the distance is measured, and $p_{ij} \approx 0$ otherwise. Thus, the traditional setting implicitly assumes that the p_{ij} are known, i.e., p_{ij} is closely approximated by a step function of d_{ij}.

Our setting is more general, since it admits the possibility that the probability for a sensor to detect its distance to another sensor may smoothly decay. In such a situation, one needs to be more careful in selecting S_{ij}. Specifically, the p_{ij} need to be incorporated into S_{ij}. Note that this automatically happens in the traditional setting because of the assumed form for the p_{ij}. The drawback of this more general, and more realistic setting is that one needs to know the p_{ij}. In practice, this is a reasonable requirement, since prior to deploying the sensors, one can gather a great deal of technical information on the sensors. For example, through rigorous repeated experimentation, one can obtain near exact estimates on how a signal transmitted by a sensor degrades as a function of distance. This suffices to derive simple formulas for the probability p_{ij} based on various random models of the background noise. It turns out that such (unbiased, bounded variance) estimates of the p_{ij} suffice. A detailed discussion of relaxing the requirement that the exact p_{ij} are known is deferred to a full version of this paper.

We now turn our attention to Assumption 2, which states that even far away sensors have some arbitrarily

small, though non-zero probability of detecting their distance. As sensor technology improves such an assumption becomes only a mild restriction. In general, p_{ij} is a continuous, non-linear, decreasing function of the distance d_{ij} between the two nodes i and j. Simple models for the detection probabilities can be derived for RF sensors [16], based on the fact that the received power decreases inversely proportional to the square of the distance from the source. Since sensors are deployed in a bounded region, the detection probability among a pair of sensors might become very small, however, it remains bounded away from zero.

One may, however, encounter settings where two sensors have essentially zero probability of detecting their distance. For example, if the sensors are so far apart that the signal to noise ratio is too small, then there is no chance that the sensors will detect their distance. Our results do not strictly apply to this setting in the global sense, however they do apply in the local sense. Specifically, in any "local" region, it is certainly the case that p_ϵ is bounded away from zero. Our results imply that in this local region, which corresponds to a submatrix of the full distance matrix, the distances can be reconstructed accurately. Thus for this particular local region, the positions of the sensors can be recovered in their own local coordinate system. The global localization problem then becomes equivalent to a problem of meshing together several *provably accurate* local "maps" into a single global map, where each local map can be in its own coordinate system.

VII. Evaluation

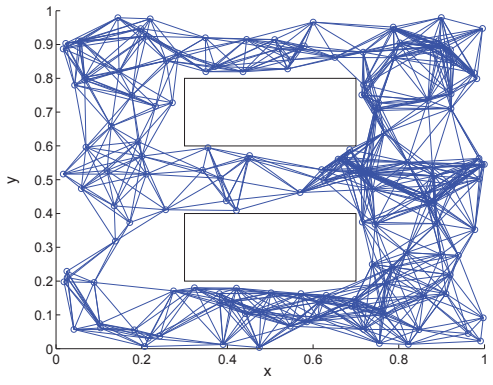


Fig. 1: Example corridor shaped scenario

We evaluate the trends of the reconstruction algorithm on two main types of deployment, uniform and corridor based. We assume that each node detects nodes that are within a radius of $R = 0.165$ with probability one; if two nodes are at distance more than 0.165 the probability that they detect each other is $p_\epsilon = 1/100$. Thus we satisfy Assumption 2, while at the same time the connectivity of the sensor network remains essentially the same.

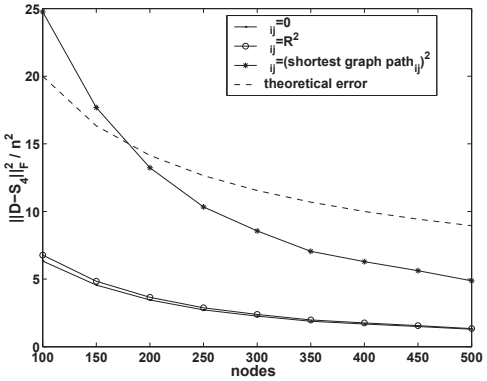


Fig. 2: Uniform Deployment w/o noise

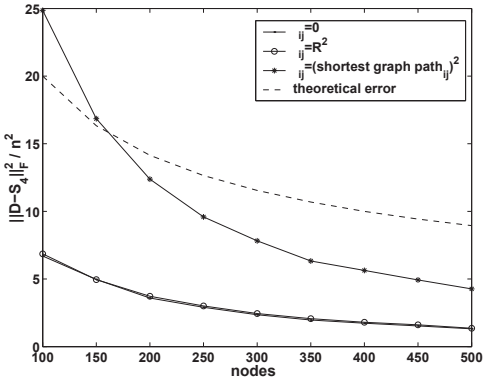


Fig. 3: Uniform Deployment with noise

In the uniform scenarios nodes are randomly scattered in a 1×1 square field following a uniform distribution. In the corridor shaped scenarios, nodes are scattered on a 1×1 square using the same uniform distribution. Corridors are formed by creating two rectangular gaps inside the square field as shown in Fig 1. For each scenario, we evaluate the reconstruction trend for network sizes ranging from 100 to 500 nodes with 10 scenarios for each size. The average connectivity ranges from (roughly) 5 to (roughly) 42. We subsequently plot the average for each size. We evaluate the quality of our reconstruction

for $\gamma_{ij} = 0$, $\gamma_{ij} = R^2$ and $\gamma_{ij} =$ shortest path^2. The reconstruction trends are shown in Figs 2, 3, 4, and 5.

Figures 2 and 4 show the trend when measurements are noise free. Figures 3 and 5 display the same results when distance measurements are corrupted by a noise drawn from a zero mean uniform distribution that is 63% of the actual measurement. Clearly, the plots verify the main result of our work: the accuracy of the localization drops inversely proportional to the square root of the number of nodes in the sensor network. The similarity between the theoretical error bound curve and the curves for the cases $\gamma_{ij} = 0$ and $\gamma_{ij} = R^2$ is indeed striking. As predicted by equation (7), noise does not significantly affect the distance matrix reconstruction error, since the variance of the noise (σ_ϵ^2) is dominated by the first term of equation (7).

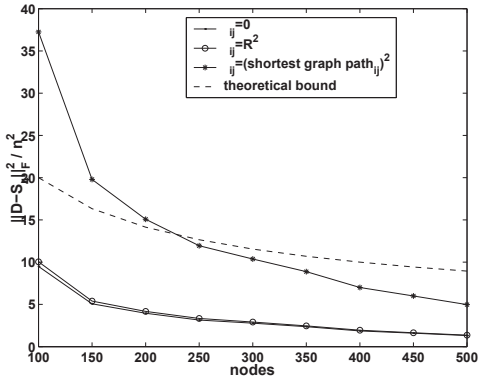


Fig. 4: Corridor Deployment w/o noise

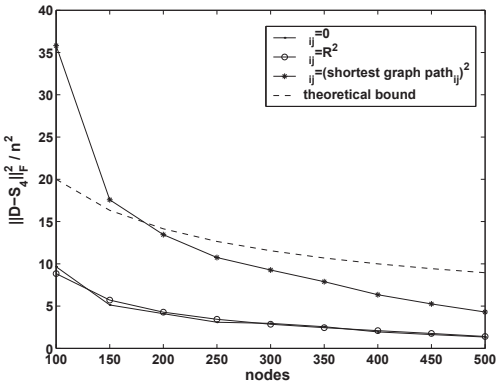


Fig. 5: Corridor Deployment with noise

Finally, we evaluate the SVD-reconstruct algorithm on the following deployment scenario. Consider a situation where two different types of sensors S_1 and S_2 are deployed in adversarial environments, where even though two sensors are within range of each other they might still fail to detect and measure their pairwise distance. Let sensors of type S_1 fail with probability p_1 and sensors of type S_2 fail with probability p_2. These probabilities may be inferred from past deployment experience. We assume that the radius of either type of sensors is R. We scatter sensor nodes of both types uniformly at random over a 1×1 square field.

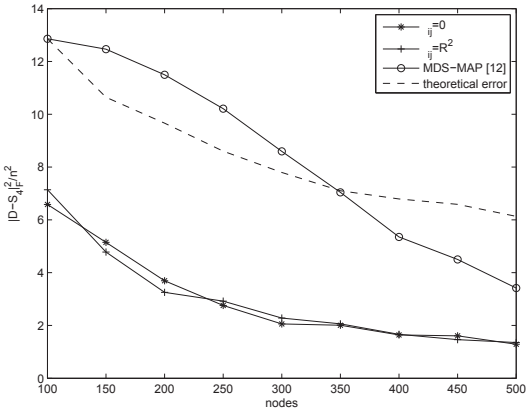


Fig. 6: Comparison with MDS-MAP ($p_1 = \frac{2}{3}, p_2 = \frac{3}{4}, R = 0.1$)

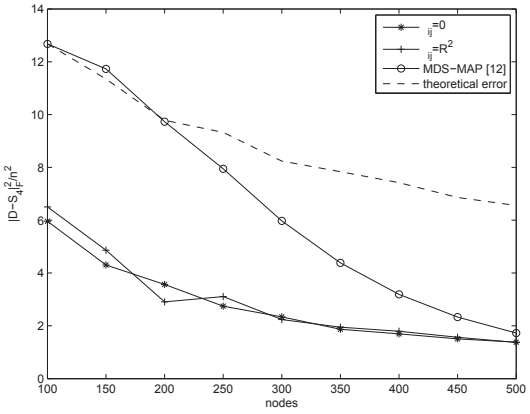


Fig. 7: Comparison with MDS-MAP ($p_1 = \frac{1}{2}, p_2 = \frac{3}{4}, R = 0.1$)

Now consider the 4 possibilities that arise in this setting. If two sensor nodes of type S_1 are within distance

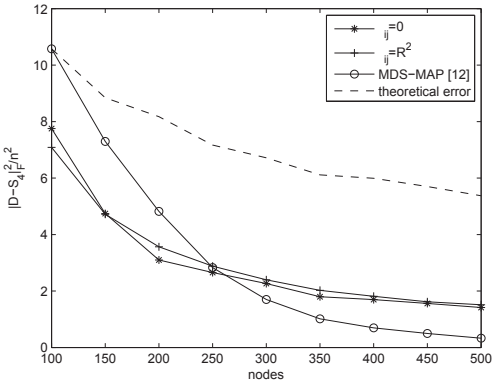


Fig. 8: Comparison with MDS-MAP ($p_1 = \frac{1}{2}, p_2 = \frac{3}{4}, R = 0.165$)

R from each other, they will detect their distance with probability $(1-p_1)^2$; if two sensor nodes of type S_2 are within distance R from each other, they will detect their distance with probability $(1-p_2)^2$; if one sensor node of type S_1 and one sensor node of type S_2 are within distance R from each other, they will detect their distance with probability $(1-p_1)(1-p_2)$; if two sensor nodes of any type are farther than R they will detect their distance with a small, fixed probability $1/100$ (we do not account for individual failure probabilities in this case). Figures 6 and 7 show that for $R = 0.1$ and two different choices for the failure probabilities p_1 and p_2 the SVD-Reconstruct outperforms the MDS-MAP algorithm of [12]. This effect is particularly pronounced in sparse deployments, and is due to the careful rescaling of the known distances by the apriori known failure probabilities. However, as R increases (Figure 8), the comparative advantage of SVD-Reconstruct decreases; in particular, for dense deployments the MDS-MAP algorithm of [12] seems to marginally outperform SVD-Reconstruct.

VIII. Conclusions and Future Work

In this paper we described a first step towards provable algorithms for sensor network localization, by demonstrating that – under some assumptions – reconstruction of Euclidean distance matrices from partial information is, in principle, feasible. Clearly, many important questions remain open. Our current work focuses on three directions. *(i)* We seek to relax the assumptions of Section III-E. *(ii)* We investigate the error bounds of applying the MDSLocalize algorithm on the *reconstructed* distance matrix $\mathbf{S}_4$. *(iii)* We investigate fully distributed, gossip-based protocols for MDSLocalize and SVD-Reconstruct, with provable running time and message size guarantees. *(iii)* We intend to evaluate these algorithms on a real testbed at ENALAB at Yale University.

Acknowledgements: P. Drineas's work is supported by an NSF CAREER award, M. Magdon-Ismail's work is supported by NSF CNS 0323324, and A. Savvides' work is supported by an NSF CAREER award.

References

[1] D. Achlioptas and F. McSherry, *Fast Computation of Low Rank Matrix Approximations*, Proceedings of the 33rd Annual ACM Symposium on Theory of Computing, pp. 611–618, 2001.

[2] D. Achlioptas and F. McSherry, *Fast Computation of Low Rank Matrix Approximations*, submitted.

[3] P. Biswas and Y. Ye, *Semidefinite Programming for Ad-Hoc Wireless Localization*, Proceedings of Information Processing in Sensor Networks, pp. 46 - 53, USA.

[4] K.K. Chintalapoudi, A. Dhariwal, R. Govindan, and G. Sukhatme, *On the Feasibility of Ad-Hoc Localization Systems*, USC Technical Report No. 03-796, 2003.

[5] P. Drineas and R. Kannan, *Pass efficient algorithms for approximating large matrices*, Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 223–232, 2003.

[6] P. Drineas, R. Kannan and M.W. Mahoney, *Fast Monte Carlo Algorithms for Matrices III: Computing a Compressed Approximate Matrix Decomposition*,Yale University, Department of Computer Science, YALEU/DCS/TR-1271, 2004.

[7] D. Goldenberg, A. Krishnamurthy, W. Maness, Y. R. Yang, A. Yound, and A. Savvides, *Network Localization in Partially Localizable Networks*, to appear in Proceedings of IEEE INFOCOM, 2005.

[8] J. Xi, H. Zha, *Sensor Positioning in Wireless Ad-hoc Sensor Networks Using Multidimensional Scaling*, Proceedings of IEEE INFOCOM, 2004.

[9] Tzu-Chen Liang, T. Wang, and Y. Ye, *A Gradient Search Method to Round the Semidefinite Programming Relaxation Solution for Ad-Hoc Wireless Sensor Network Localization*, working paper available at http://www.stanford.edu/ yyye/formal-report5.pdf.

[10] D.Niculescu and B. Nath, *VOR Base Stations for Indoor 802.11 Positioning*, Proceeding of IEEE MOBICOM, 2004.

[11] A. Savvides, H. Park, and M. B. Srivastava, *The n-hop Multilateration Primitive for Node Localization Problems*, Proceedings of IEEE MOBICOM, 443–451, 2003.

[12] Y. Shang, W. Ruml, Y. Zhang, and M. Fromherz, *Localization from mere connectivity*, ACM MobiHoc, pp. 201–212, 2003.

[13] Y. Shang and W. Ruml, *Improved MDS-Based Localization*, Proceedings of IEEE INFOCOM, 2004.

[14] A. M. C. So and Y. Ye, *Theory of Semidefinite Programming for Sensor Network Localization*, working paper available at http://www.stanford.edu/ yyye/local-theory.pdf.

[15] R. Stoleru and J. Stankovic, *Probability Grid: A Location Estimation Scheme for Wireless Sensor Networks*, Proceedings of Sensor and Ad-Hoc Communications and Networks Conference (SECON), 2004.

[16] M. Zuniga and B. Krishnamachari, *Analyzing the Transitional Region in Low Power Wireless Links*, Proceedings of Sensor and Ad-Hoc Communications and Networks Conference (SECON), 2004.

APPENDIX

Proof of Theorem 1: After the first step of the algorithm, $\boldsymbol{\tau}(\mathbf{D})$ is an $n \times n$ matrix whose (i,j)-th entry is equal to the inner product $\left(\mathbf{x}_i - (1/n)\mathbf{1}_n^T\mathbf{X}\right)^T \left(\mathbf{x}_j - (1/n)\mathbf{1}_n^T\mathbf{X}\right)$. In words, the (i,j)-th entry of $\boldsymbol{\tau}(\mathbf{D})$ is equal to the inner product of the coordinate vectors corresponding to the i-th and the j-th sensors, translated in a coordinate system whose origin is the point $(1/n)\sum_{i=1}^n \mathbf{x}_i$. Notice that

$$\mathbf{D} = \mathbf{1}_n\mathbf{z}^T + \mathbf{z}\mathbf{1}_n^T - 2\mathbf{X}\mathbf{X}^T, \tag{5}$$

where $\mathbf{z}$ is an $n \times 1$ vector whose i-th element is equal to $\|\mathbf{x}_i\|^2$. Then,

$$\begin{aligned}\boldsymbol{\tau}(\mathbf{D}) &= -\frac{1}{2}\mathbf{L}(\mathbf{1}_n\mathbf{z}^T + \mathbf{z}\mathbf{1}_n^T - 2\mathbf{X}\mathbf{X}^T)\mathbf{L} = \\ &= \left(\mathbf{X} - (1/n)\mathbf{1}_n\mathbf{1}_n^T\mathbf{X}\right)\left(\mathbf{X} - (1/n)\mathbf{1}_n\mathbf{1}_n^T\mathbf{X}\right)^T\end{aligned}$$

Notice that $\boldsymbol{\tau}(\mathbf{D})$ is a symmetric positive semidefinite matrix of rank at most 2 and its Singular Value Decomposition (computed at the second step of the algorithm) has the same left and right singular vectors. Thus,

$$\tilde{\mathbf{X}} = \mathbf{U}_2\boldsymbol{\Sigma}_\mathbf{2} = \left(\mathbf{X} - (1/n)\mathbf{1}_n\mathbf{1}_n^T\mathbf{X}\right)W, \tag{6}$$

for some 2×2 orthonormal matrix W. Clearly, $(1/n)\mathbf{1}_n^T\mathbf{X} = (1/n)\sum_{i=1}^n \mathbf{x}_i$ is the translation and W is the rotation/reflection. Thus, up to rotation/reflection and translation, we have recovered the original coordinates $\mathbf{X}$.

Proof of Lemma 2:

$$\begin{aligned}\mathbf{E}[S_{ij}] &= \mathbf{Pr}[\epsilon_{ij} = \epsilon] \\ &\cdot \left(\frac{d_{ij}^2 + \epsilon - \gamma_{ij}}{p_{ij}}p_{ij} + (1-p_{ij})\gamma_{ij} | \epsilon_{ij} = \epsilon\right) \\ &= d_{ij}^2 = D_{ij}.\end{aligned}$$

Proof of Lemma 4: The first part of the lemma is an instantiation of Theorem 5 of [2]. For the second part, notice that

$$\begin{aligned}\|(\mathbf{D}-\mathbf{S})_4\|_F^2 &= \sum_{i=1}^4 \sigma_i^2\left((\mathbf{D}-\mathbf{S})_4\right) \\ &\leq 4\sigma_1^2\left((\mathbf{D}-\mathbf{S})_4\right) \\ &= 4\|(\mathbf{D}-\mathbf{S})_4\|_2^2 \\ &= 4\|\mathbf{D}-\mathbf{S}\|_2^2 \\ &\leq 128\sigma_S^2 n,\end{aligned}$$

and the lemma follows by taking square roots of the two sides.

Bounding the variance of the entries of S_{ij} (σ_S^2):

$$\begin{aligned}&\mathbf{Var}[S_{ij}] = \\ =\ &\mathbf{Var}[D_{ij} - S_{ij}] = \mathbf{E}\left[(D_{ij} - S_{ij})^2\right] \\ =\ &\mathbf{Pr}[\epsilon_{ij} = \epsilon]\left(p_{ij}\left(\frac{d_{ij}^2 + \epsilon - \gamma_{ij}(1-p_{ij})}{p_{ij}} - d_{ij}^2\right)^2\right. \\ +\ &\left.(1-p_{ij})\left(\gamma_{ij} - d_{ij}^2\right)^2 | \epsilon_{ij} = \epsilon\right) \\ =\ &\mathbf{Pr}[\epsilon_{ij} = \epsilon]\left(p_{ij}\left(\frac{(d_{ij}^2 - \gamma_{ij})(1-p_{ij})}{p_{ij}} + \frac{\epsilon}{p_{ij}}\right)^2\right. \\ +\ &\left.(1-p_{ij})\left(\gamma_{ij} - d_{ij}^2\right)^2 | \epsilon_{ij} = \epsilon\right) \\ \leq\ &\frac{2(d_{ij}^2 - \gamma_{ij})^2(1-p_{ij})^2}{p_{ij}} + \frac{2\mathbf{E}\left[\epsilon_{ij}^2\right]}{p_{ij}} \\ +\ &(1-p_{ij})\left(\gamma_{ij} - d_{ij}^2\right)^2 \\ =\ &\frac{(d_{ij}^2 - \gamma_{ij})^2(1-p_{ij})(2-p_{ij})}{p_{ij}} + \frac{2\sigma_\epsilon^2}{p_{ij}} \\ \leq\ &\frac{2}{p_{ij}}\left((d_{ij}^2 - \gamma_{ij})^2 + \sigma_\epsilon^2\right).\end{aligned}$$

Notice that the quality of the bound improves if γ_{ij} is close to d_{ij}. Overall,

$$\sigma_S^2 \leq \max_{i,j}\frac{2}{p_{ij}}\left((d_{ij}^2 - \gamma_{ij})^2 + \sigma_\epsilon^2\right). \tag{7}$$

Improving RSSI based distance estimation for 802.15.4 wireless sensor networks

A. Faheem*, R. Virrankoski and M. Elmusrati
Department of Computer Science
University of Vaasa Finland
E-mail: faheem.ahmed@student.uwasa.fi

Introduction

Wireless sensor networks (WSNs) have been receiving a lot of attention recently due to a wide range of potential applications such as environment monitoring, warehouse inventory, object tracking etc. In many cases it is necessary to obtain accurate location information of the nodes. Many techniques based on e.g. Multilateration [1], [2], Multidimensional Scaling [3], [4] have been developed over the years to achieve localization in WSNs. Most of these techniques rely on basic node to node distance, angle or number of hops [5], [6] to achieve full scale localization. This information can be absolute or relative. There are many different parameters that have been used as indicator of distance between nodes e.g. RSSI [6], [7], [8], [9], Time Difference of Arrival (TDoA) [10], Angle of Arrival etc. Normally the nodes used in a wireless sensor network have very little resources. Therefore techniques that utilize small resources without the need for extra hardware, need to be developed. RSSI based localization present one such solution, as the recent advancements in radio hardware make it possible to achieve reliable signal strength indication [11].

In this paper an RSSI based distance estimation technique for 802.15.4 network, based on CC2420 radio core, is discussed. In this approach we use standard deviation (SD) of the RSSI value and the packet loss information as a part of model parameters estimation process. The SD and packet loss limits are optimized along with the curve parameters to achieve minimum distance error. The distance estimator uses these optimized limits as a measure of accuracy of the remote node's estimated distance.

The rest of the paper is organized as follows. In section 2 we describe the experimental setup that has been used. Section 3 gives an overview of the system that we have developed for the optimization of parameters. The results are discussed in section 4. We summarize and conclude the work in section 5.

2. Experimental setup

The NanoRouter N601 from Sensinode was used as gateway node and programmed as FFD for the experiments. Sensinode devices include the radio module RC2301AT from Radiocraft. The module contains CC2431 system on chip (SoC) RF transceiver solution from Texas Instruments (TI). The operation of gateway node is controlled by a pc application using the built-in USB interface (FTDI232B). The NanoSensor N711 was programmed as a RFD. This device has battery pack support for two AA size batteries making it a suitable choice as the RFD. The RFD was displaced by one meter intervals up to 30 meter and RSSI values were collected for 50

measurements at each distance. The experimental data consists of three sets of measurements for outdoor measurements and three sets of measurements for indoor environment.

3. Developed System

Log Distance Path Loss Model is a basic way of estimating path loss as a function of distance between the nodes. The model is normally expressed as following equation.

$$L(dB) = \mathrm{P_o} + 10\ n\ log_{10}\left(\frac{d}{d_o}\right) + X_\sigma$$

Where n is the path loss exponent, d is the distance between transmitter and receiver, X_σ is a Gaussian random variable with standard deviation σ and $\mathrm{P_o}$ is the received power at reference distance d_o.

Chipcon specifies the following formula [12] to compute RSSI.

$$\mathrm{RSSI} = -10\ n\ log_{10}(d) + \mathrm{A}$$

Where n is propagation exponent, d is the distance from the sender and A is the received signal strength at one meter of distance.

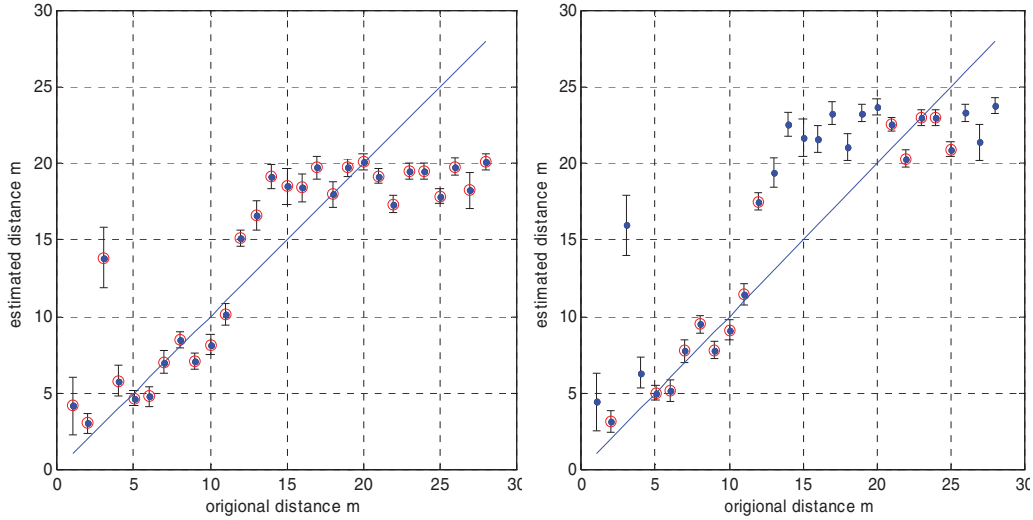
To find the model parameters, the collected RSSI data (1—30 meter, 50 RSSI measurements at each distance) was loaded to an optimizer application developed with MATLAB. The parameter optimizer can be divided into two main parts. The first part is the error calculation function, which takes into account the provided starting point values of the model parameters (SD limit, packet loss limit, path loss exponent and the constant) and calculates the mean square error (MSE) between the original distances and the estimated distances. The second part implements the bounded minimization operation [13] on the error function and tries to minimize the average distance error by optimizing the model parameters. The optimizer returns new set of model parameters which provide the minimum error in the estimated distances.

4. Results and Discussion

We implemented the new optimized model parameters to estimate the distance of a remote node. The blind node collects 50 RSSI measurements from the remote node and tries to map the collected data to a certain distance using the optimized model. Fig. 1 represents the estimated distances between a fixed blind node and a remote node for the range of 1—28 meter.

In Fig. 1(a) it is noticeable that nodes at 14—28 meter are estimated to be at approximate distance of 16—20 meter. The blue line represents the true distances. The points which are far from the line are the false ones. But the blind node cannot distinguish between the false ones and the true ones without knowing the original distances.

In Figure 2(b) we notice that using optimized SD and packet loss limits blind node can effectively eliminate most of the false distance estimates.



(a) (b)

Figure 1: Distance estimates with (a) Curve fitting parameters; (b) Optimized limits and curve parameters.

In figure 2 the percentage distance error with the percentage of nodes that belong to that error margin are displayed along with the percentage of SD and packet loss limits which have been used to achieve these results, are displayed. For example the first four bars in the graph indicate one set of experiment where the error is reduced to 80 percent of the maximum error and 10 % of nodes were excluded by using (optimized) 60 % of maximum SD of collected RSSI and about 18% of maximum packet loss values as filtering parameters.

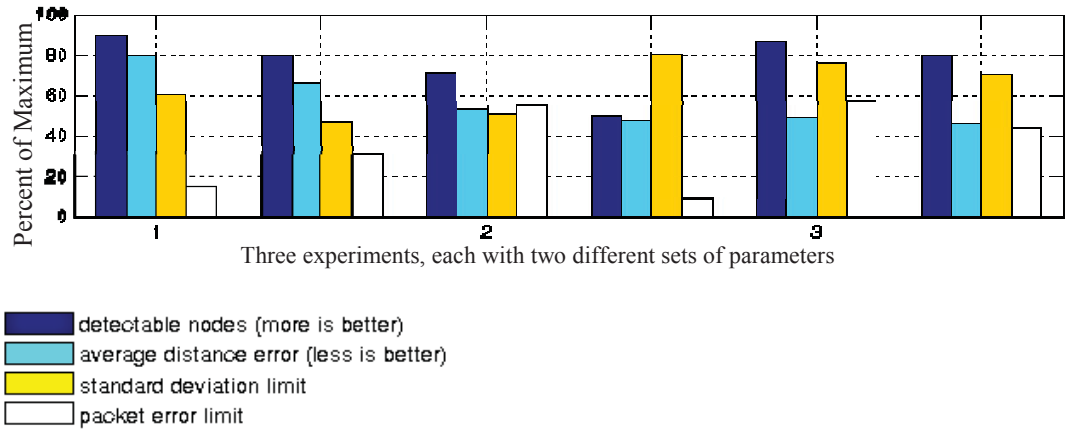


Figure 2: Percentage of average error and corresponding number of nodes using certain percentage of STD and packet loss limits.

5. Conclusion

We proposed a very simple method of eliminating faulty distance estimates by filtering the measured RSSI through optimized SD and packet loss limits. We conducted experiments with CC2431 radio nodes to collect enough RSSI measurements for model parameter estimation. We devised an optimizer to find optimized model parameters and limits. Then we used the experimental data to verify the optimized model and found that our method does help reducing the average distance error by effectively identifying and eliminating those estimates which introduce the most error.

References:

[1] Z. Yang and Y. Liu, "Quality of trilateration: Confidence based iterative localization," In Proc. IEEE ICDCS 2008, Beijing,China, June 17-20, pp.446-453.

[2] F. Thomas and L. Ros, “Revisiting trilateration for robot localization,” *IEEE Trans. Robot.*, vol. 1, no. 1, pp. 93–101, Feb. 2005.

[3] V. Vijayanth , W. Vincent W.S.,"Ordinal MDS-based localisation for wireless sensor networks, International Journal of Sensor Networks, v.1 n.3/4, pp.169-178, January 2006.

[4] P. Drineas, M. Magdon-Ismail, G. Pandurangan, R. Virrankoski, and A. Savvides, “Distance matrix reconstruction from incomplete distance information for sensor network localization,” Yale University Electrical Engineering, Tech. Rep., 2005.

[5] H.Chen, K.Sezaki, P.Deng and H.C.So, "An improved DV-hop localization algorithm for wireless sensor networks," *Proc. IEEE Conference on Industrial Electronics and Applications (ICIEA 2008),* Singapore pp.1557-1561, June 2008,

[6] S. Tian, X. Zhang, P. Liu, P. Sun, and X. Wang, "A RSSI-Based DV-Hop Algorithm for Wireless Sensor Networks," *in Wireless Communications, Networking and Mobile computing*, 2007. WiCom 2007. International Conference, 2007, pp. 2555-2558.

[7] A. Abdalkarim, F. Thorsten and D. Falko, "Adaptive Distance Estimation and Localization in WSN using RSSI Measures,” *In 10th EUROMICRO Conference on Digital System Design - Architectures, Methods and Tools (DSD 2007)*, L¨ubeck, Germany, August 2007. pp 471–478

[8] P. Chuan-Chin and C. Wan-Young, "Mitigation of Multipath Fading Effects to Improve Indoor RSSI Performance," IEEE Sensors Journal, vol. 8, NO. 11, pp. 1884—1886 Nov. 2008.

[9] K. Benkic, M. Malajner, P. Planinsic and Z. Cucej, “Using RSSI value for distance estimation in wireless sensor networks based on ZigBee,” in *Proceedings of 15th International Conference on Systems, Signals and Image Processing - IWSSIP 2008*, pp. 303 – 306, 2008.

[10] S. Schwarzer, M. Vossiek, M. Pichler, and A. Stelzer, "Precise distance measurement with IEEE 802.15.4 (ZigBee) devices," in *Radio and Wireless Symposium, 2008 IEEE*, pp. 779—782, 2008.

[11] K. Srinivasan and P. Levis, “RSSI is under appreciated,” *In Proceedings of the Third Workshop on Embedded Networked Sensors*, 2006.

[12] K. Aamodt, *Chipcon Products from Texas Instruments, Application Note AN042 (Rev. 1.0).*

[13] J. D'Errico, “MATLAB Central,” Sep. 28, 2006. [Online]. Available: http://www.mathworks.com/matlabcentral/fileexchange/8277. [Accessed: Mar. 28, 2010].

TASC: Topology Adaptive Spatial Clustering for Sensor Networks

Reino Virrankoski* and Andreas Savvides
Embedded Networks and Applications Lab (ENALAB)
Yale University, 51 Prospect Street #212, New Haven, CT 06520
Tel. +1-203-432-1275, Fax +1-203-432-0593
reino.virrankoski@hut.fi, andreas.savvides@yale.edu

***Abstract*— The ability to extract topological regularity out of large randomly deployed sensor networks holds the promise to maximally leverage correlation for data aggregation and also to assist with sensor localization and hierarchy creation. This paper focuses on extracting such regular structures from physical topology through the development of a distributed clustering scheme. The Topology Adaptive Spatial Clustering (TASC) algorithm presented here is a distributed algorithm that partitions the network into a set of locally isotropic, non-overlapping clusters without prior knowledge of the number of clusters, cluster size and node coordinates. This is achieved by deriving a set of weights that encode distance measurements, connectivity and density information within the locality of each node. The derived weights form the terrain for holding a coordinated leader election in which each node selects the node closer to the center of mass of its neighborhood to become its leader. The clustering algorithm also employs a dynamic density reachability criterion that groups nodes according to their neighborhood's density properties. Our simulation results show that the proposed algorithm can trace locally isotropic structures in non-isotropic network and cluster the network with respect to local density attributes. We also found out that TASC exhibits consistent behavior in the presence of moderate measurement noise levels.**

I. Introduction

The anticipation of large-scale sensor networks and experience from preliminary deployments has demonstrated the need for meaningful decomposition of large distributed sensor networks into a set of smaller sub-networks. Such decomposition should be conducted in a manner that facilitates sensor node coordination and enhances the feasibility of network management and in-network processing and aggregation of sensor data. In this paper, we explore this issue of network decomposition through the development of a specialized distributed clustering scheme. The scheme we investigate is designed to extract regularity from irregular network topologies by allowing the nodes to organize themselves into groups of locally isotropic (or regular) non-overlapping clusters without requiring the knowledge of node locations. Given the close coupling of sensors to the physical world we advocate that such a classification of sensor nodes according to their spatial attributes would be beneficial from multiple aspects.

Besides the intuitive benefit of improving the ease of network management, the spatial grouping of nodes with respect to regions of close proximity and similar deployment density promotes efficient data aggregation and efficient compression of sensor data. Spatial clustering would also assist transmission power control, since intra-cluster communication requires less transmission power in dense clusters. Moreover, as pointed out by [1] spatial irregularity in sensor sampling can exacerbate the load and cost imbalance between different parts of the network. This is mainly because many of the existing distributed signal processing and compression algorithms assume spatially regular data samples (see Figure 1). This also entails that the spatial grouping of nodes can help reduce the propagation of redundant data inside the network. This argument is further reinforced by the recent results presented in [2].

Despite the fact that clustering has been previously studied both theoretically and in the context of ad-hoc networks [3]–[10], its consideration in the context of sensor networks gives rise to a new problem setup where sensor measurements are used as actual inputs to the problem.

The proposed distributed algorithm does not require node locations but it assumes that nodes are aware of their 2-hop neighborhood. It also assumes that nodes are able to measure distances to their one hop neighbors. We consider both assumptions reasonable. The former is a standard assumption for many neighborhood discovery algorithms whereas the latter is becoming a common

*Visiting Assistant Researcher from the Control Engineering Laboratory, Helsinki University of Technology, Finland.

feature of many sensor network applications, though not all of them. Accurate internode distance measurements in the sensor network domain have been demonstrated using ultrasound in the system described in [13], the MIT Crickets [14] and in the Medusa MK-2 node [15]. In the radio domain, ultra-wide-band ranging systems such as the one offered by Ubisense [16] have already demonstrated accurate distance measurements with small sensor form factors that will be suitable for sensor networks. Moreover camera based schemes such as the one we developed in [24] can accurately measure internode distances without requiring specialized measurement hardware on each node. Finally, we note that the spatial clustering of the network before node localization is actually an advantage for ad-hoc localization. Ad-hoc localization schemes such as [15], [17]–[19] may benefit from the properties of our algorithm to eliminate computation redundancies and geometric error propagation.

The contribution of this paper is the development and characterization of a *Topology Adaptive Spatial Clustering Scheme* (TASC) that operates on combination of node weights and a dynamic density reachability criterion adopted from previous work in the database community [11], [12]. The paper is organized as follows. In the next section we highlight the related work. Section III describes the clustering problem requirements. Section IV provides the details of our weight scheme and density reachability criterion and describes the clustering algorithm. Algorithm evaluation through simulations is presented in Section V. Section VI discusses some additional attributes and Section VII states our conclusions and plans for future work.

II. Related Work

A mathematical framework that has similarities with the network clustering problem is k-means clustering [7]. Some interesting k-means clustering modifications were recently applied to ad hoc clustering in [8], [9], [20]. What makes the setting we investigate different from others is the fact the amount of prior knowledge is smaller than it is in typical k-means applications. Nodes are only able to measure distances to their one hop neighbors, positions are unknown and the network architecture does not offer the centralized knowledge needed for basic k-means algorithm applications.

Basagni in [3] presented a Distributed Clustering Algorithm (DCA) and a Distributed Mobility-Adaptive Clustering (DMAC) algorithm. DCA is suitable for clustering of quasi-static ad hoc networks and DMAC adapts to changes in network topology caused by node mobility. Selection of clusterheads is based on weights. The weights are real numbers that characterize each node feasibility to become a clusterhead. They are based on node connectivity (number of one hop neighbors) or on node mobility such that weights are inversely proportional to node velocity.

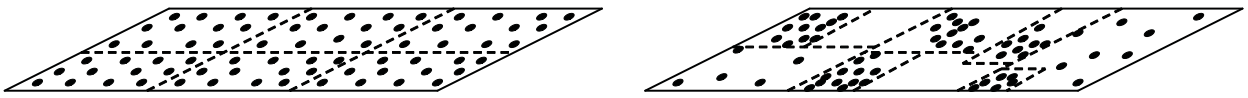
In Max-Min D-Cluster Formation introduced by Amis et al [4] clusterheads are selected such that they form a d-hop dominating set. By definition, if an ad hoc network is modeled as a graph $G = (V, E)$, a set C of vertices is a d-hop dominating set of G if every node in V is at most d ($d > 1$) hops away from a vertex in C. Clusterhead election is based on node id in four logical stages. Since d is an input value to the heuristic, it enables control over the density of clusterheads in the network. Authors also prove that the minimum d-hops dominating set problem is NP-complete.

Chen and Liestman [10] present a zonal algorithm to find weakly connected dominating sets. The algorithm consists of three phases. First, an input graph representing the ad hoc network is partitioned into regions of approximately size x. Then, the distributed algorithm for weakly connected sets is run in each region and finally some additional region border vertexes are added.

An Energy Efficient Hierarchical Clustering Algorithm for Wireless Sensor Networks by Bandyopadhyay and Coyle [5] targets to organize the sensors in clusters such that communication energy consumption is minimized. In single-level clustering, each sensor has same probability p to become a clusterhead. After election each sensor that becomes a clusterhead advertises itself as a clusterhead to all sensors within its radiorange. Advertisement is forwarded to all sensors that are no more than k hops away from the clusterhead. Each sensor joins to the cluster of closest clusterhead. Optimal values of p and k (with respect to communication energy) are computed under the assumption that sensors are distributed as per a homogenous spatial Poisson process.

Younis and Fahmy [6] use hybrid of node residual energy and other parameter, such as node proximity to its neighbors or node connectivity. The clustering goals are network lifetime maximization, scalability and load-balancing. It is assumed that each node has a fixed number of transmission power levels. Transimission power control is further applied to define cluster radius by the transmission power level used for intra-cluster announcements.

Targeting to the clusters that are formed *with respect to existing network topology* is the issue that makes TASC different than any existing sensor network clustering algorithm. If clusterhead selection is based on node id or node connectivity [3], [4] or randomness [5], it does not guarantee that clusterhead location is reasonable in terms of spatial attributes. If one operates with received signal



(a) (b)

Fig. 1. Difference between uniform network clustering and TASC clustering. Density variations do not exist in figure a), because underlying node deployment is uniform. In that case equal cluster size in terms of number of nodes per cluster and in terms of cluster area guarantees balanced clusters. In figure b), non-uniform node deployment is clustered with respect to existing topology such that clustering targets to *minimize density variations in each cluster*. In non-uniform case, one cannot dictate the number of nodes per cluster or cluster area in advance, because both of them depend on the density variations that exist in the network.

strength [6], the correspondence between geographic distances and received signal strength is often weak in real applications.

In the context of clustering less attention is paid for the problem how to cluster network such that clustering improves data compression. We anticipate that the TASC uniform clustering approach will enable *1) different data compression rates in each cluster* and *2) improved overall compression rate in the whole network*, if suitable data compression technique as the one presented in [21] is applied. Furthermore, one is able to achieve savings in data aggregation and communication costs, if as much redundancy as possible can be eliminated in lowest possible hierarchy level in the network and clustering with respect to spatial attributes enables lower transmission power in intra-cluster communication.

III. Clustering Objectives

Our clustering approach is motivated by the requirements of the sensor network domain. More specifically, a clustering algorithm should partition the network so that the nodes inside each cluster have high correlation in sensor measurements and are evenly spaced in order to maximize gains and reduce errors due to ill geometric positioning as in the case of node localization. In non-uniform network, node density variations are *globally big* but there exist subgroups of nodes such that density variations are *locally small*. We assume that *1) each node can measure distances to its one hop neighbors* and *2) each node has knowledge of its 2-hop neighborhood*. We set following main goal to our clustering algorithm:

The main objective of TASC is to cluster non-uniform sensor networks such that relative node density variation in individual clusters is smaller than relative node density variation in the whole network.

The type of non-uniform network clustering that we are targeting is illustrated in Figure 1b and an example of clustering outcome is shown in Figure 2. Density variations are estimated by dividing network area into a set of non-overlapping triangles such that each node locates at least in one triangle vertex like illustrated in Figure 2b. In such triangulation, density variations are indicated by the triangle edge length standard deviation. *Relative density variation*, that is computed by dividing the edge length standard deviation by the average edge length, describes density variations such that the value is independent on actual distances.

Figure 1 shows that in contrast to uniform deployments, in more random deployments one cannot dictate a fixed number of clusters or use a grid construction since that would diminish the exploitation of correlation properties. Instead, TASC requires only the minimum number of nodes in a cluster in order to avoid the creation of single node clusters.

IV. Leader Election and Cluster Formation

A. Algorithm

In algorithm execution, each node considers its 2-hop neighborhood. Other pre-specified parameters are the *required minimum cluster size* and the *density reachability parameter Dr*, that is explained in detail in subsection C. Leader election and cluster formation takes place in five phases:

1) *Each node computes its own weight based on shortest Euclidean paths in its* 2*-hop environment*.
2) *Each node broadcasts its own weight to its* 2*-hop neighborhood, and receives the weights of its* 2*-hop neighbors*.
3) *Each node nominates the node having biggest weight in the density-reahable subset of its* 2*-hop neighbors and broadcasts its nominee to its* 2*-hop neighborhood*.
4) *Each node receives all nominees in its* 2*-hop neighborhood, and elects the closest nominee to its leader*.
5) *Each node that ends up in a cluster where the total number of nodes is smaller than pre-specified minimum cluster size joins to closest cluster, where the number of nodes exceeds the required minimum cluster size*.

To be able to compute its leader, each node must send two messages to its 2-hop neighborhood and receive 2

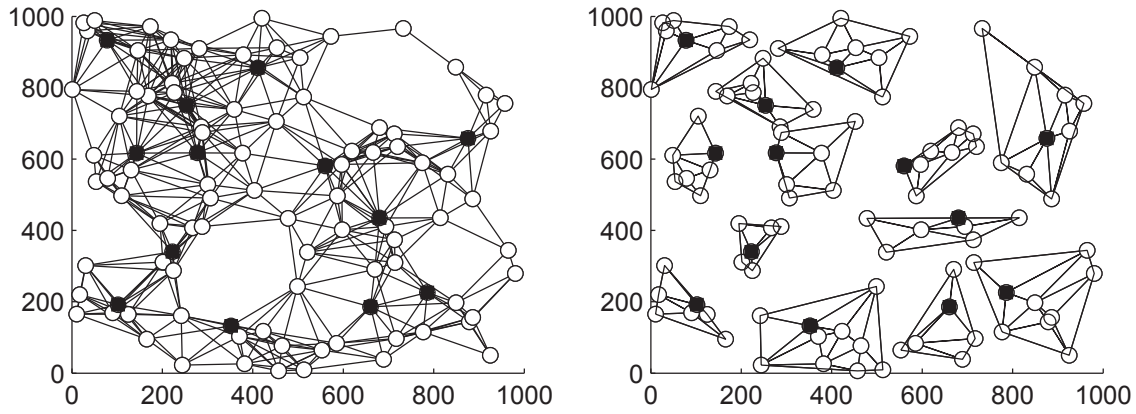


Fig. 2. An example of the clustering outcome when each node considers its 2-hop neighborhood. Figure a) shows node configuration and network connections resulting to clusters shown in figure b). Delaunay triangulation of each cluster is also shown in figure b). Cluster leaders are marked by black square.

messages from each of its two hop neighbors.

B. Weight Computation: Discovering Local Network Structure

The computation of node weights tries to achieve the reverse effect of greedy forwarding in geographic routing [22], [23]. In greedy forwarding, a node found on the path to a packet destination, forwards the packet to its neighboring node with location closest to the location of the destination. Instead of trying to forward traffic to the neighboring node that is closest to the destination, TASC all-pairs-shortest path routing is based on distance measurements to extract information about the network topology. More specifically, node weight is a measurement of two key quantities 1) *the frequency a node is found on the shortest path between pairs of nodes* and 2) *the distance contribution of the edges of that node with respect to the total length of the path*.

Consider the network in Figure 4a. If we define the weights to be the number of times a node is found on the shortest path, then we can compute a weight for each node. Node A for instance can be found on the paths AB, AC, AD and AE hence it will have a weight of 4. Node C is found on eight different paths hence it receives a weight of 8. To construct a proof of this behavior we use the principle of optimality [25]:

If S is the shortest Euclidean path between two nodes, it includes all shortest paths between all pairs of nodes that are located in path S.

Definition 1: Each node in the sensor network gets weight +1 each time the shortest Euclidean path between any pair of nodes in the network crosses that node

Inputs: 2-hop neighborhood, inter-node distance measurements, minimum cluster size, Dr
Output: Leader node

$weight$ = ComputeWeight();
BroadcastToNeighborhood($weight$);
If all $weights$ received:
  Select heaviest density reachable node as $nominee$;
  BroadcastToNeighborhood($nominee$);
EndIf
If all nominations have been received:
  Select the closest $nominee$ as $leader$;
  BroadcastToNeighborhood($leaderID$, $nodeID$);
EndIf
If this node is $leader$:
  Wait until election timeout;
  BroadcastToNeighborhood($clustermembers$, $clustersize$);
EndIf
If cluster size is received:
  If $clustersize <$ minimum cluster size:
    select the closest neighbor for which $clustersize \geq$ minimum cluster size and join its cluster;
  EndIf
  BroadcastToNeighborhood($leaderID$, $clustersize$)
EndIf

Fig. 3. Clustering Algorithm. Each node computes its own weight in its own 2-hop neighborhood, and thus function *ComputeWeight()* is called once in each node. Dr is a parameter for density range computation explained in detail in subsection C. In the pseudocode, $clustersize$ means cluster size in terms of number of nodes.

or ends at it. Paths are assumed undirected in weight computation.

Theorem 1: Let S be the shortest Euclidean path between two nodes, and let $2n+1$ be the total number of nodes in path S. When computing all shortest paths between each pair of nodes in path S and assigning weights to each node in S as presented in Definition 1, the node that is from equal hop distance from both endpoints of path S, gets the biggest weight.

Proof: Observe path S having total number of $2n+1$ nodes, and let c be the node located from equal hop distances from both ends of the path S. Since the total number of nodes in path S is $2n+1$, there are n nodes on both sides of node c. Based on basic routing theory, S includes shortest paths between all pairs of nodes located in S. Thus, the weight of node c is equal to the total number of shortest paths crossing node c and ending at node c:

$$W_c = n \cdot n + 2n = n^2 + 2n \tag{1}$$

Pick then a node g from the path S so that there are $k < n$ nodes from the other side of that node. In that case there are $n + (n-k)$ nodes on the opposite side. The weight of node g is:

$$\begin{aligned} W_g &= k(n + (n-k)) + k + n + (n-k) \\ &= 2nk + 2n - k^2 \end{aligned} \tag{2}$$

When comparing the weights (1) and (2) we get

$$\begin{aligned} W_g < W_c \Leftrightarrow 2nk + 2n - k^2 < n^2 + 2n \Leftrightarrow \\ n^2 - 2nk + k^2 > 0 \Leftrightarrow (n-k)^2 > 0 \end{aligned} \tag{3}$$

That holds always when $0 < k < n$. ■

Corollary 1: If there are 2n nodes in the path S discussed in Theorem 1, two nodes in the middle get both equal biggest weight values.

Proof: The result follows from equations (1)-(3), when total number of $2n$ nodes are used. ■

Theorem 2: When weights in network graph are computed as presented in Definition 1, the node or nodes closest to the network center achieve the biggest weights.

Proof: The proof is a generalization of the discussion presented in equations (1)-(3). Observe M shortest paths that are crossing each other in one node, and mark $N = 2M$. In the symmetric case, the number of nodes in each path is $2n+1$ and all paths are crossing each other in the midmost node c. When all shortest paths between pairs of nodes located in paths M are taken into account in weight computation, the weight of the node c is

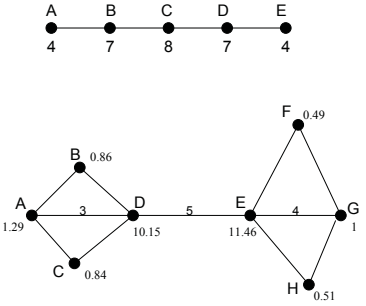


Fig. 4. Weights example.

$$\begin{aligned} W_c = & n \cdot (N-1) \cdot n + n \cdot (N-2) \cdot n + ... \\ & + n \cdot n + N \cdot n = n^2 \textstyle\sum_{i=1}^{N-1} i + Nn \end{aligned} \tag{4}$$

Observe next an asymmetric case, where one of the paths has $n + k + 1$ nodes, where $1 \leq k < n$, when the rest of paths have still $2n+1$ nodes, and paths are crossing each other in node g so that in path M_j there are n nodes on the one side and k nodes on the opposite side of node g. For other paths $M_{i,i\neq j}$, g is still the midmost node. In that case the weight of node g is:

$$\begin{aligned} W_g = & n \cdot (N-2) \cdot n + kn + n \cdot (N-3) \cdot n + kn + ... \\ & + n \cdot n + kn + kn + (N-1)n + k \\ = & n^2 \textstyle\sum_{i=1}^{N-2} i + (N-1)kn + (N-1)n + k \end{aligned} \tag{5}$$

When comparing weights W_c and W_g, we get

$$\begin{aligned} & W_g < W_c \Leftrightarrow \\ & n^2 \sum_{i=1}^{N-2} i + (N-1)kn + (N-1)n + k < n^2 \sum_{i=1}^{N-1} i + Nn \Leftrightarrow \\ & (N-1)kn + k < (N-1)n^2 + n \end{aligned} \tag{6}$$

which is true under assumption $1 \leq k < n$. ■

This is enough to show that in non-uniform deployment, the node that tends to be the midmost related to all shortest communication paths (in terms of hops) gets the biggest weight. The result of Corollary 1 generalizes this result so that if some of the paths have even number of nodes, there can be several nodes with equal biggest weights in the middle.

1) Including Distances in Weight Computation: Although this method of computing weights would decide the central node, it does not give enough information in the cases where the paths are asymmetric such as the paths in the example network shown in Figure

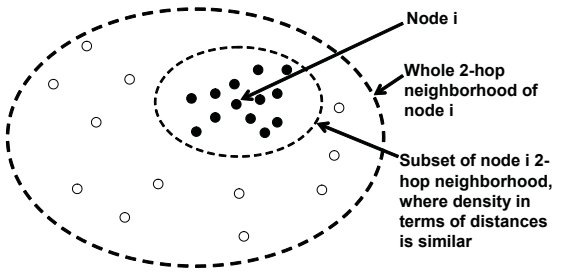


Fig. 5. The effect of density reachability. Node i figures out the subset of its 2-hop neighborhood, where density in terms of distances is similar or higher.

4b. To handle this problem, we augment the weight computation to incorporate distance information. Instead of incrementing the weight by one each time a node is used in a path, we increment the weight as a function of the distance a node contributes to the path. If a node k is found on the path from node i to node j in between nodes a and b, then the weight increment of node k is given by equation (7) where $l_{a,k}$ and $l_{k,b}$ are the lengths of the edges between nodes a and b and node k respectively and $l_{i,j}$ is the length of the whole path from node i to node j.

$$w_{ij} = \frac{l_{a,k} + l_{k,b}}{l_{ij}} \tag{7}$$

An example of this weight computation is shown in Figure 4b.

C. Density Reachability: Grouping Similar Densities

While the information from node weights can be used to identify local centers, we would still like to construct clusters by grouping nodes in regions with similar density attributes. To achieve this goal, in addition to considering weights we need to consider additional means of pulling areas with high node densities towards the center of a cluster. To be able to do so using only distance measurements, each node must find the subgroup of its 2-hop environment, where node density in terms of distances is similar or higher to the node under consideration. The problem is illustrated in Figure 5. Each node seeks around it such subgroup of nodes, where density variations are smaller than density variations in its whole 2-hop environment. A modified version of density reachability, that is traditionally applied in data clustering to cluster spatial data in the presence of obstacles [11], [12], is applied.

The definition of density reachability is based on distance metric called *density range*, that defines the *upper bound of density variations in terms of distances* such that density is considered higher or equal. One can define the *resolution* in which accuracy density reachability differentiates between more and less dense by modifying density range. To make TASC adapt to local density variations, we propose following dynamic density range definition:

Definition 2: The density range r_i of node i with respect to the given density reachability parameter Dr is the smallest disk centered at i that covers $Dr - 1$ other nodes in the vicinity of i.

In the definition, Dr is a constant number of nodes given a priori. When a bigger number is given, density range becomes longer. If density ranges are longer, each node has a bigger upper bound to the distance variations that it accepts to its density reachable set. Thus, when value Dr increases, two changes are happening in the set of density reachable nodes: 1) the set includes more nodes and 2) the set includes bigger density variations. The upper bound is the whole 2-hop neighborhood of node i, and the effect of density reachability diminishes as density range approaches the 2-hop radius. Based on the dynamic definition of the density range, we define a node to be density reachable as follows:

Definition 3: A node j is density reachable from i if there is a path from i to j where the length of every hop l satisfies the constraint: $l \leq r_i$.

Figure 6 shows an example of density reachable set definition. Node i considers its 2-hop neighborhood, and $Dr = 4$. Nodes j, k and the black nodes are density reachable from node i since there exists Euclidean path from node i to these nodes such that the length of each hop in the path is smaller or equal than r_i. Note that for the purposes of our clustering algorithm, density reachability can only expand within the 2-hop neighborhood of each node.

By applying density reachability, each node further limits the number of nodes that it can potentially nominate by considering only density reachable nodes as nomination candidates (see Figure 5). This effect pulls cluster leaders towards most dense groups in the cluster, but nomination among density reachable candidates is still based on weights.

V. Evaluation of Cluster Properties

To characterize the properties of the clustering algorithm, we run a set of simulations on a suite of 100 random scenarios. In each scenario, 100 nodes are deployed on a square deployment field of size 1000 by 1000. The simulation also assumes that the distance measurement range of the node is equal to the communication

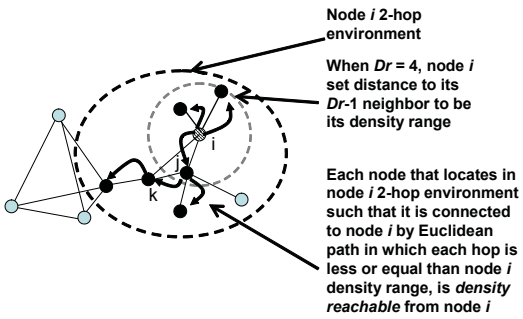


Fig. 6. Example of density reachable nodes selection. Node i select its density reachable nodes in its 2-hop neighborhood when $Dr = 4$. Black nodes indicate the subset of node i 2-hop neighbors, that is density reachable from node i.

range. In practice, we expect that the communication range is greater than the measurement range, so this assumption does not violate the fundamental properties of our clustering algorithm. Each scenario is used five times over different connectivity levels. Each time the connectivity is varied by varying maximum measurement range from 200 to 400 in steps of 50. Respective average node connectivity in each case is 10.31, 15.35, 21.09, 27.32 and 33.80. We note that even though the average connectivity is relatively high, density variations in our simulation scenarios are so high that if maximum communication range (in our simulations equal to maximum measurement range) is less than 200, all nodes are not connected to the network. For most cases, the required minimum number of nodes per cluster is set to 4. To keep communication cost and computational complexity low, each node considers its 2-hop environment. Our simulations are implemented with an in-house version of NeslSim [26], which is implemented in PARSEC. The main role of the NeslSim environment in our work is the enforcing of a distributed implementation of our clustering algorithm. The computation of shortest paths is done using the Floyd-Warshall algorithm running at each node.

1) Cluster Evaluation Metrics: As mentioned in section III, node density variation is given by triangle edge length standard deviation, if cluster area is divided into a set of non-overlapping triangles such that nodes locate in triangle vertices. In addition to node density variation and number of nodes per cluster, we are computing *density per cluster* in terms of nodes/m^2. To be able to compute density per cluster, we must define *cluster area*. We do so using *Delaunay triangulation*.

Delaunay triangulation is a standard triangulation method that we found well suitable for cluster triangulation. By definition, a Delaunay triangulation of a finite set of points in the plane is a triangulation that minimizes the standard deviations of the angles of the triangles, using 60 degrees as the mean. Thus, Delaunay triangulation gives an *optimal planar subdivision in terms of spatial uniformity*. The Delaunay triangulation is related to Voronoi tessellation such that the circle circumscribed about a Delaunay triangle has its center at the vertex of a Voronoi polygon.

We tie the definitions of cluster area, cluster density and node density variation into Delaunay triangulation:

Definition 4: Cluster area is *a sum of cluster Delaunay triangle areas*. The sum of Delaunay triangle areas is equal to the area of the polygon that is defined by outermost Delaunay triangle edges.

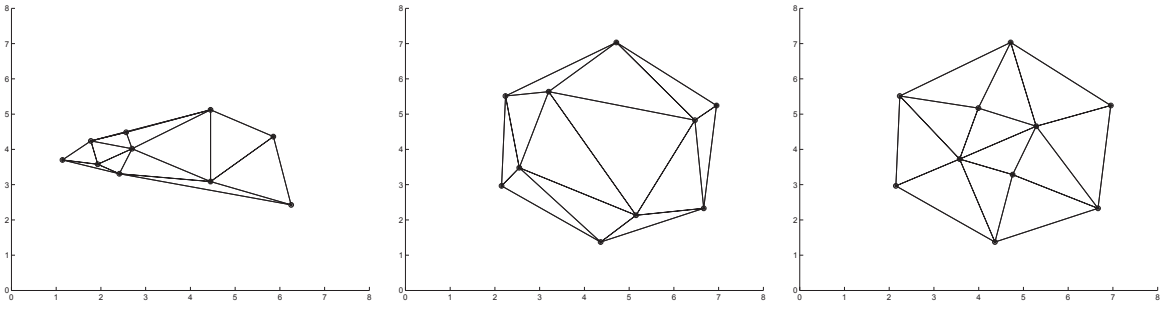
Definition 5: Cluster density (nodes/m^2) is the number of nodes in the cluster divided by cluster area.

Node density variation is characterized by relative standard deviation of Delaunay triangle edges:

Definition 6: Relative node density variation is Delaunay triangle edge length standard deviation in a cluster divided by average Delaunay triangle edge length in that same cluster.

It follows from the definition that a smaller relative node density variation indicates higher degree of uniformity. Cluster shape can be characterized by computing the *distance ratio*, that is minimum distance from polygon center point to node in polygon vertex per maximum distance from polygon center point to node in polygon vertex. Compared to the axial ratio in ellipse fitting, distance ratio gives *worst case* ratio. Three examples of cluster Delaunay triangulation are illustrated in Figure 7.

2) Examining Cluster Uniformity: The first experiment was to evaluate the node density variation in clusters, when distance measurements are assumed noiseless. The density reachability parameter Dr and the required minimum cluster sizes are both set to 4. Figure 8 shows that TASC outcome remains consistent when the network connectivity (the average number of neighbors/node) varies between 10 and 35. This consistency is expected, because connectivity is varied by varying maximum measurement range, but nodes are not moving. The average of Delaunay triangle edges standard deviation per cluster (percentage of the average Delaunay triangle edge length per cluster) computed from 6697 clusters outcome is 0.5211 (52.11%), and the respective standard deviation is 0.123. Comparison between underlying network node density variation and the node density variation in its clusters is illustrated in Figure 9. For each node configuration (Scenario #), the standard deviation of the Delaunay triangle edges of the whole network and the average of that particular network clusters Delaunay



(a) (b) (c)

Fig. 7. Three examples of cluster Delaunay triangulation. Relative standard deviation of Delaunay triangle edge lengths is a) 0.559, b) 0.385 and c) 0.248. Cluster distance ratio is a) 0.462, b) 0.912 and c) 0.912. Since node locations in the convex hull are exactly the same in clusters b) and c), the values of cluster area distance ratio are same in both clusters, but difference in relative triangle edge length standard deviation indicates that nodes are more evenly spaced in cluster c).

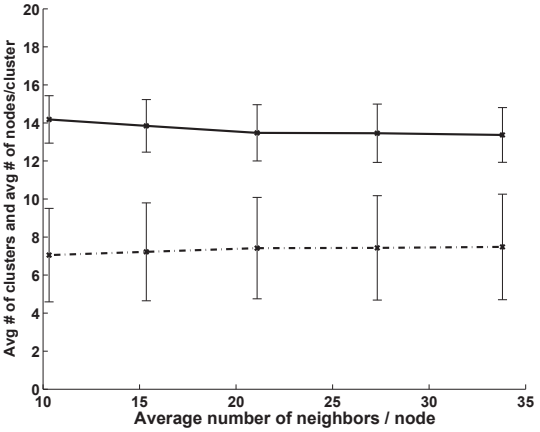


Fig. 8. Average number of clusters (upper solid line) and average number of nodes per cluster (lower dashed line). Standard deviation is shown by errorbars.

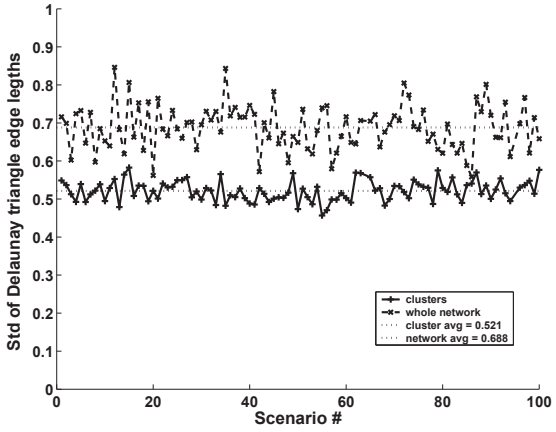


Fig. 9. Comparison between relative node density variation in the network and the average of the relative node density variation in network clusters. Delaunay triangle edge length standard deviation is represented as a percentage of the average edge length per cluster.

triangle edges standard deviation, is shown. The result shows obvious improvement in the degree of uniformity thus verifying that TASC is able to cluster globally non-uniform network into smaller uniform node configurations that exist in the network. Since a non-uniform network includes large density variations and TASC groups nearby nodes together, the cluster size in terms of number of nodes and in terms of cluster area is inversely proportional to cluster density like illustrated in Figure 1b. Our simulation verifies the existence of that trend, and it is shown in Figure 10. The overall average cluster distance ratio computed from 6697 clusters is 0.4966 and respective standard deviation is 0.1619. Those values are enough to show that we are not ending up with flat node chain type of clusters.

3) Cluster Sizes and Density Reachability: Intuitively, one would expect the average cluster size increase with increasing measurement range since the area of the 2-hop neighborhood increases. Instead, the average cluster size remains constant between 7 and 9 nodes in each of the tested cases, as illustrated in Figure 8. This is enforced by the density reachability. For each node i, the subgroup of node i 2-hop neighbors (see Figure 5) depends on node density range. Each node computes its density range based on constant parameter Dr that is given a priori. If the value of Dr is kept constant, changes in maximum measurement range do not change the density reachable subsets (see illustration in Figure 5), because the underlying node configuration remains the same. With no control on the eventual cluster density properties, the cluster sizes increase with measurement range.

As the density range begins to approach the maximum measurement range of the node, the effect of density reachability decays to the point where it cannot differ-

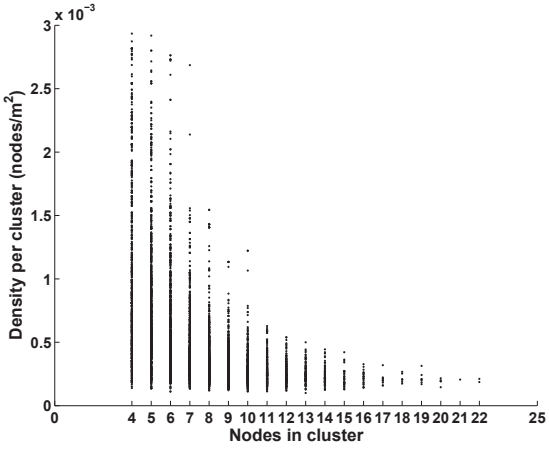


Fig. 10. TASC cluster nodes such that density variations in each cluster are smaller than density variations in the whole network. As a consequence cluster size is smaller in dense areas, but becomes bigger in sprase areas (compare to Figure 1b). Each dot in the figure shows the number of nodes per cluster and respective density per cluster. The overall shape verifies inverse proportionality between area per cluster and density per cluster.

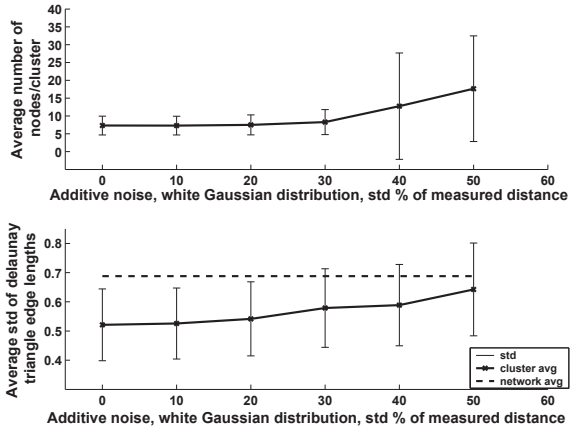


Fig. 11. Measurement noise effect to cluster size and cluster uniformity. Upper figure shows cluster size and cluster size standard deviation, lower figure average Delaunay triangle edge standard deviation (percentage of average Delaunay triangle edge length in each cluster) and average of Delaunay triangle standard deviations over all network scenarios (compare to Figure 9).

entiate among density variations in the vicinity of the node. The size of the hop environment (how many hops) where each node executes the algorithm controls the upper bound of the cluster size. Within the chosen hop environment, that is 2 hops in our simulations, density reachability further limits the cluster size since each node density range is the upper bound of distance variations, that node accepts into its density reachable set. Thus bigger value of Dr increases node density ranges. When the node density range approaches the maximum measurement range, the set of density reachable nodes approaches the entire 2-hop neighborhood of the node. As a consequence, cluster size increases and the resolution in which accuracy TASC cluster the network with respect to local uniformity, becomes weaker.

A. Clustering in the Presence of Measurement Noise

The measurement noise is modeled as additive noise following a white Gaussian distribution that the standard deviation of which is entered as a percentage of the measured distance. The effects of measurement noise on cluster size and cluster uniformity are shown in Figure 11. We are able to obtain consistent cluster sizes with up to such noise level, where additive noise standard deviation is 30% of measured distance. A dramatic change in the cluster size consistency occurs when the noise standard deviation is increased up to 40% of measured distance. Then cluster size variation in terms of number of nodes per cluster becomes huge indicating that algorithm is not able to find density variations with reasonable accuracy. The change is shown by errorbars that indicate cluster size standard deviation in Figure 11. Density variation in clusters stays below whole network average (compare to Figure 9), but Figure 11 shows that the relative density variation in clusters approaches network average such that relative density variation in clusters is not remarkably smaller than it is in the whole network, if noise standard deviation exceeds 30% of measured distance. When the noise standard deviation was increased up to 50% of measured distance, TASC failed to produce separate clusters in 10% of simulated network scenarios.

VI. Discussion

Despite the encouraging results on the behavior of TASC, we acknowledge that there are multiple issues to consider in realistic deployments. First, the parameters of TASC should be adapted to fit the particular application needs. The option of a node running multiple instances of TASC with different parameters is worth of exploring. Second, the timing parameters of the algorithm should be more rigorously defined to comply with an actual deployment. For some systems where incremental deployment makes sense the leader election mechanisms need to be adapted to support the addition and subtraction of nodes from the network. Based on our experience from the simulation behavior and our efforts to build a scalable sensor network testbed, we believe that these changes are possible. In addition to the features described here, weight computation in TASC can reveal important properties of a network topology that should be further investigated. Even though we made the assumption that each

node considers its 2-hop neighborhood, it is possible to generalize the algorithm such that each node considers its n-hop neighborhood for any choice of n. However, the generalization to n-hop environment requires possible changes to density reachability criteria, and if the size of the hop environment increases, also computation and communication costs in each node will increase.

VII. Conclusions and Future Work

Our evaluation has shown that by using the novel combination of weights and density reachability, TASC achieves the desired behavior: *It can decompose large non-uniform networks into smaller locally uniform clusters*. Simulations with noise indicate that TASC tolerates noisy distance measurements up to level, where the standard deviation of Gaussian noise is 30% of measured distance. In addition to the previously mentioned applications, the distribution of weights inside a network can also be used as an indicator for spatial regularity in a specific deployment. One possible research avenue would be to develop and algorithm for making localized decisions on how nodes should reposition themselves to improve sampling uniformity. Another possibility is to repeat the weight-based election process to construct hierarchies. The initial results are encouraging and suggest the more rigorous evaluation of TASC needs in more realistic deployment settings. As part of our future work, we plan to test TASC in the context of our 3-D testbed. The two immediate uses of TASC in our 40-node testbed is to assist with ad-hoc node localization and in radio frequency allocation through the meaningful, spatial decomposition of a dense Zigbee network.

Acknowledgment

This work was partially funded by the National Science Foundation award #0448082 and by scholarships from Nokia Foundation and Emil Aaltonen Foundation.

References

[1] D. Ganesan, S. Ratnasamy, H. Wang and D. Estrin, *Coping with irregular spatio-temporal sampling in sensor networks*, in Proceedings of Second Workshop on Hot Topics in Networks (HotNets-II), Novemeber 2003

[2] S. Pattem, B. Krishnamachari and R. Govindan, *The Impact of Spatial Correlation on Routing with Compression in Wireless Sensor Networks*, Proceedings of the Third International Symposium on Information Processing in Sensor Networks (IPSN'04), April 26 - 27, 2004, Berkeley, California, USA

[3] S. Basagni, *Distributed Clustering for Ad Hoc Networks*, International Symposium of Parallel Architectures, Algorithms and Networks (I-SPAN'99), Fremantle, Australia, June 23-25, 1999.

[4] A. D. Amis, R. Prakash, T. H. P. Vuong, D. T. Huynh, *Max-Min D-Cluster Formation in Wireless Ad Hoc Networks*, Proceedings of IEEE INFOCOM 2000.

[5] S. Bandyopadhyay, E. J. Coyle, *An Energy Efficient Hierarchical Clustering Algorithm for Wireless Sensor Networks*, Proceedings of IEEE INFOCOM 2003.

[6] O. Younis, S. Fahmy, *Distributed Clustering in Ad-hoc Sensor Networks: A Hybrid, Energy-Efficient Approach*, Proceedings of IEEE INFOCOM 2004.

[7] McQueen, J. B., *Some Methods for Classification and Analysis of Multivariate Observations*, Proceedings of the Fifth Symposium on Math, Statistics and Probability (pp. 281-297), 1967.

[8] Kanugo, T., Mount, D. M., Netanyahu, N. S., Piatko, C. D., Silverman, R., Wu, A. Y., A *Local Search Approximation Algorithm for k-Means Clustering*, Proc. of the 18th Annual ACM Symp. on Computational Geometry, 2002, 10-18.

[9] Ghiasi, S., Srivastava, A., Yang, X., Sarrafzadeh, M., *Optimal Energy Aware Clustering in Sensor Networks*, Sensors 2002, 2, 258-269.

[10] Chen, Y. P., Liestman, A. L., *A Zonal Algorithm for Clustering Ad Hoc Networks*, International Journal of Foundations of Computer Science, 14(2):305-322, April 2003.

[11] Ester, M., Kriegel, H-P., Sander, J., Xu, X., *A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases With Noise*, 2nd International Conference on Knowledge Discovery and Data Mining (KDD'96), Portland, Oregon, 1996.

[12] Zaane, O. R., Lee, C-H., *Clustering Spatial Data in the Presence of Obstacles: a Density-Based Approach*, Sixth International Database Engineering and Applications Symposium (IDEAS 2002), Edmonton, Alberta, Canada, July 17-19, 2002.

[13] A. Harter and A. Hopper, *A New Location Technique for the Active Office*, IEEE Personal Communications, vol. 4, No. 5, October 1997, pp.42 - 47.

[14] N. B. Priyantha, A. Chakraborty, H. Balakrishnan, *The Cricket Location-Support System*, Proceedings of 6th ACM Mobicom, Boston, MA, August 2000.

[15] A. Savvides, H. Park and M. B. Srivastava, *The n-hop Multilateration Primitive for Node Localization Problems*, Proceedings of Mobile Networks and Applications, 8, 443-451, 2003

[16] Ubisense website, http://www.ubisense.net

[17] N. B. Priyantha, H. Balakrishnan, E. Demaine, S. Teller, *Anchor-Free Distributed Localization in Sensor Networks*, LCS Tech. Report 892.

[18] Y. Shang, W. Ruml, *Improved MDS-Based Localization*, Proceedings of IEEE INFOCOMM, Hong Kong, March 7-11, 2004

[19] Ji, X., Hongyuan, Z., *Sensor Positioning in Wireless Ad-hoc Sensor Networks Using Multidimensional Scaling*, IEEE Infocom, March 7-11, 2004.

[20] Klein, D., Kamvar, S. D., Manning, C. D., *From Instance-level Constraints to Space-level Constraints: Making the Most of Prior Knowledge in Data Clustering*, The Nineteenth International Conference on Machine Learning (ICML-2002), Sydney, Australia, July 8-12, 2002.

[21] J. Chou, D. Petrovic, K. Ramchandran, *A Distributed and Adaptive Signal Processing Approach to Reduce Energy Consumption in Sensor Networks*, Proceedings of IEEE INFOCOM 2003.

[22] GPSR B. Karp and H.T. Kung, *GPSR: Greedy Perimeter Stateless Routing for Wireless Networks*, Proceedings of Mobicom 2000

[23] LAR Y.B Ko and N. Vaidya, *Location Aided Routing (LAR) in Mobile Networks*, Proceedings of ACM/IEEE MobiCom, pp. 66-75, October 1998

[24] D. Lymberopoulos, A. Barton-Sweeny, A. Savvides, *Sensor Localization and Camera Calibration in Networks of Low-Power Imagers*, Yale ENALAB Technical Report 080501, August 2005

[25] D. Bertsekas, *Dynamic Programming and Optimal Control*, vol 1, Athena Scientific, 2000

[26] NeslSim Website http://www.ee.ucla.edu/ saurabh/NESLsim/

www.ietdl.org

Published in IET Wireless Sensor Systems
Received on 2nd February 2011
Revised on 20th July 2012
doi: 10.1049/iet-wss.2011.0137

ISSN 2043-6386

Recursive clock skew estimation for wireless sensor networks using reference broadcasts

H. Yiğitler[1] A. Mahmood[1] R. Virrankoski[2] R. Jäntti[1]

[1]Department of Communications and Networking, Aalto University, P.O. Box 13000, FI-00076 Aalto, Finland
[2]Department of Computer Science, Communication and Systems Engineering Group, University of Vaasa, P.O. Box 700, FI-65101 Vaasa, Finland
E-mail: huseyin.yigitler@aalto.fi

Abstract: Reference broadcast-based time synchronisation protocols are appreciated by the wireless sensor network community for their low-power demands. The underlying time relation characteristics of the broadcast-based time synchronisation schemes are prone to the effects resulting from the time record correlations. The recursive equivalents of the existing time synchronisation methods have large clock skew estimation error variance since these methods ignore the effect of correlation. In this study, the authors propose a novel recursive clock synchronisation algorithm based on a time relation model that reflects the effect of correlation. The authors utilise the maximum likelihood estimator to reach an asymptotically consistent and efficient clock skew estimator. The authors theoretically evaluate the performance of the developed estimator and compare it with the existing ones. Of the methods studied, the proposed estimator achieved the smallest estimation error variance. Experimental validation suggests an accuracy of less than one tick at synchronisation instants for a 6 h experiment.

1 Introduction

A common notion of time in a distributed system such as a wireless sensor network (WSN) is crucial for applications requiring either synchronous execution or chronologically ordered information. For example, applications involving distributed localisation, tracking, security and structural health monitoring have such requirements. Moreover, some medium access methods such as time division multiple access and energy management methods requiring radio duty cycling can only be used with time synchronised networks.

Each individual sensor in the network has a source of time, a clock, which is essentially a counter driven by a low-cost uncompensated crystal oscillator [1]. The time report of such a clock is different than the actual time because of the 'time offset, clock skew (frequency offset), frequency drift and wideband noise', known as the 'time error model' [2]. Time offset is the difference between the time report and the actual time at a reference instant. The clock skew, frequency drift and wideband noise are mainly due to the non-ideal output of the driving oscillator. Clock skew represents short-term frequency deviations, and frequency drift represents long-term frequency deviations owing to ageing, temperature or supply variations. The wideband noise is analogous to Brownian motion, which is modelled as a Wiener process [3]. This process has statistically independent and bounded variance non-overlapping increments, such that two distinct samples of this process are correlated and the variance of the samples grows unbounded over time. For short-term analysis, the time error is usually modelled without frequency drift [2]. The time error of a clock, ($\mathcal{C}_2$), which is due to time offset and clock skew, is shown in Fig. 1*a*.

The relationship between an unknown clock and a reference clock is also modelled by the time offset, clock skew and wideband noise (the frequency drift is neglected) of each clock, known as the 'time relation model' [4]. The unknown clock can be identified by estimating the time offset and clock skew using the time reports sampled at the same time instants, as depicted in Fig. 1*a*. The compensation for the time offset allows instantaneous synchronisation by masking the effect of other parameters, whereas the synchronous operation is maintained by identifying and correcting the clock skew [5]. Therefore the time synchronisation problem involves determining and compensating for the clock skew after measuring the time difference at a fixed reference instant.

The exchange of time reports over a wireless medium introduces additional uncertainty in the received time information owing to communication delay [4], as illustrated in Fig. 1*b*. The non-deterministic portion of the communication delay is usually considered additive to wideband noise, whereas the deterministic portion contributes to the time offset in the time relation model. In other words, the time relation model has a joint noise process composed of communication delay and joint wideband noise of the clocks. Assuming that the time jitter owing to wireless communication is dominant, the only noise source in the model is communication delay [1]. This assumption can be validated by comparing the second-order statistics of bounded variance processes. In case the

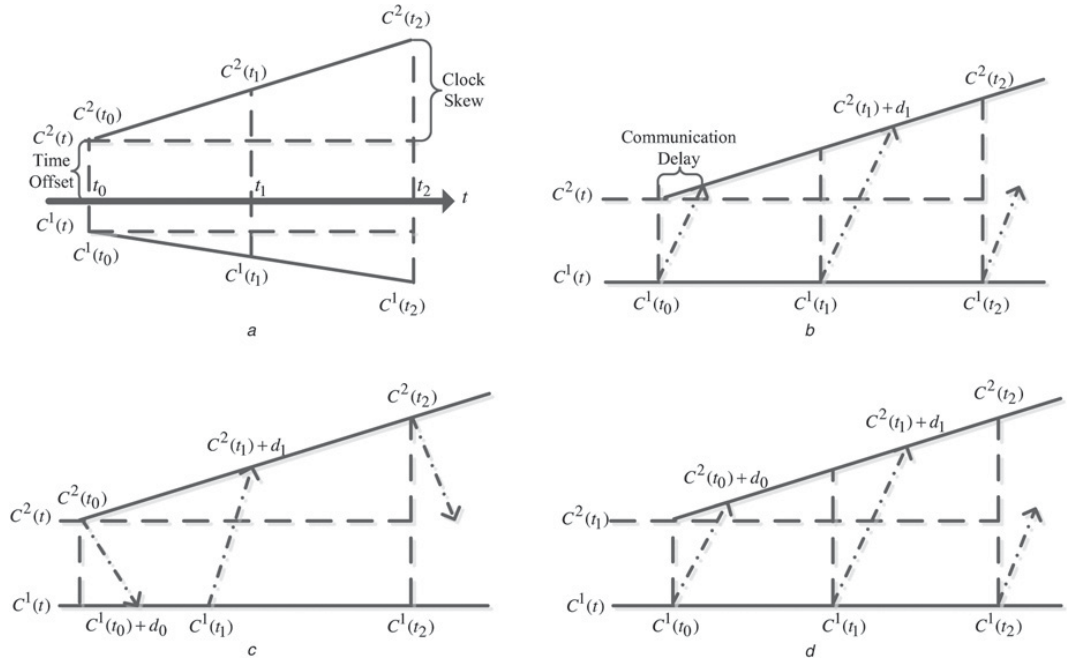


Fig. 1 *Time relation model for different synchronisation scenarios*

a Relation between the time reports of $\mathcal{C}_1$ and $\mathcal{C}_2$
b Effect of wireless communication delay, d, on the relation between the time reports of $\mathcal{C}_1$ and $\mathcal{C}_2$
c Time relation of the handshake-based time synchronisation schemes
d Time relation of the reference broadcast-based time synchronisation schemes

variance of the wideband noise process increases in time, the upper bound of the variance must be much less than the variance of the communication delay. Since the time reports have finite variance disjoint increments, the reference instant of the time reports defines the noise process of the time relation model. Consequently, the communication delay can be assumed to be the dominant noise source if the reference instant can be altered; otherwise, both noise sources must be taken into account.

In general, the network-wide time synchronisation is achieved by a combination of [1, 6]

1. A clock synchronisation algorithm: These algorithms usually estimate the time offset and clock skew based on the local and received time reports [4].
2. A messaging protocol to exchange the time reports among the sensors: These protocols can be classified as 'handshake'-[7–9] and 'broadcast' [5, 10]-based protocols, where, chronologically, the latter preceded the former.

The handshake-based protocols implicitly or explicitly use the time relation model of the network time protocol [7]. A node in the network initiates the handshake by requesting the time report of the reference clock, and then waiting for the response, as depicted in Fig. 1*c*. The unknown clock is identified by using the time increments with respect to the handshake initiation instant. Since the duration of one handshake is small, the communication delay can be assumed to be the dominant noise source. On the other hand, broadcast-based protocols operate based on periodic time report broadcasts of the reference clock, as shown in Fig. 1*d*. The time report broadcasts are referenced to an absolute instant, which may or may not be altered by the synchronisation algorithm. Therefore the time relation model for the broadcast-based messaging schemes in general has two noise sources.

Regardless of the underlying messaging scheme, the time difference between the clocks can be kept within required limits by frequently exchanging the time reports and by compensating for the time offset. The number of transmissions, however, is restrained by the energy constraints of the WSN applications. Pottie and Kaiser [11] demonstrated that the energy required to transmit 1 kb over 100 m is equivalent to the energy required to execute 3 million instructions. This observation has led to the design of broadcast-based protocols along with computationally more complex clock synchronisation algorithms to decrease the number of required transmissions [5, 10, 12, 13]. However, the scarce computational and storage capabilities of sensors limit the practically applicable class of algorithms.

The batch least square estimator (LSE) is usually utilised for clock skew estimation using reference broadcasts [5, 10]. This method requires storing a table of received and local time reports in the memory to fit a line in least-square sense, which demands a significant amount of memory and computational resources. It is well known that this estimator is not efficient for a correlated measurement set [14], that is, the estimation error variance is higher than the lower bound. Yet, for a predetermined table size, the estimation error variance can be kept within limits. On the other hand, the memory and computational requirements are relaxed for the recursive LSE, but it performs poorly when utilised directly for clock skew estimation owing to correlations in the time records. Therefore the effect of time record correlations on the clock skew estimate should be overcome to reach a realisable recursive estimation algorithm. Consequently, the development of such a clock

synchronisation algorithm for a time relation model of reference broadcast schemes motivates this study.

The time relation model for broadcast-based schemes has two noise sources: the joint wideband noise and the communication delay. The effect of wideband noise on the time relation model can only be observed by considering the output of the oscillators. A practical oscillator output has a frequency spectrum spread around localised tones rather than only at discrete frequencies owing to random perturbations in electronic devices. For synchronisation purpose, it is sufficient to model the oscillator output for uncorrelated perturbation sources, which is already shown to exhibit Wiener process behaviour [3]. The oscillator output essentially defines the time error and the time relation models. Consequently, we develop the time relation model based on the oscillator model proposed by Demir *et al.* [3], before considering the effects of wireless communication.

In this paper, we propose a novel recursive clock synchronisation algorithm for reference clock broadcasts. The algorithm is based on a time relation model that captures the characteristics of the oscillator output and communication delay. For this purpose, we first demonstrate the relationship between the oscillator output model in the study by Demir *et al.* [3] and time error model in the study by Allan [2] for the basic time measurement system. Second, we introduce a reference-triggered time relation model, which requires perfectly synchronised recording of the time reports at predetermined intervals of the reference clock. Third, we introduce an external-triggered time relation model for recording the time reports via an external trigger, which is a generalised version of the reference-triggered model. We show that the time relation model of the reference broadcasts, including the wireless communication delay, is equivalent to the external-triggered time relation model. Finally, we overcome the shortcomings of the least squares (LS)-based clock skew estimator by proposing a novel maximum likelihood (ML) based estimator, which is known to be asymptotically consistent and efficient [14]. The ML estimate is used to reach a practically realisable recursive clock skew estimation algorithm based on the external-triggered time relation model.

The rest of the paper is organised as follows. The frequently used symbols throughout the paper are tabulated in Table 1. In Section 2, we provide an overview of the related work. In Section 3, we review the oscillator and time error models and introduce the basic time measurement system. In Section 4, we introduce the time relation models and develop the recursive skew estimator. In Section 5, we compare the performance of the recursive estimator for the externally triggered, incremental time relation model with various skew estimators. In Section 6, the equivalence of external-triggered time relation model and the time relation model for reference clock broadcasts is shown. In Section 7, some practical issues and their solutions are addressed. In Section 8, the developed theory is evaluated experimentally.

2 Related work

In this section, we review the related time synchronisation algorithms. The details of messaging protocols are not within the scope of this paper. The interested reader is referred to works [1, 4, 6] in the reference list and the references therein for a complete overview of the time synchronisation methods for WSN.

The reference broadcast-based time synchronisation protocols for WSN are mostly derived from the studies by Elson *et al.* [5] and Maroti *et al.* [10], which are incorporated with an LS linear regression-based clock synchronisation algorithm. It is experimentally demonstrated that the length of the regression table must be determined by considering the spacing between the entries in the table and the desired level of synchronisation accuracy [13, 15]. In this paper, we show that the entries in the regression table are statistically correlated since the received and local time reports reference an absolute instant. Furthermore, applying the LS linear regression to a set of correlated time values yields a large estimation error variance. Also, we show that the skew estimation error variance is a linear function of the spacing between measurement instants.

A recent study uses two oscillators for time synchronisation [16]: a low-frequency and stable oscillator accompanied by a

Table 1 Major notations

t	Actual time
$\alpha(t)$	phase deviation of the oscillator
ϕ_0	time offset
δ_f	clock skew
ϵ	wideband noise
a	clock skew ratio
$\hat{a}$	clock skew ratio estimate. $\hat{a}(N)$ is the clock skew estimate using N time reports. $\tilde{a}(N)$ denotes the clock skew ratio estimation error at Nth estimate
$\mathcal{C}_i$	the ith clock. $\mathcal{C}_1$ is the reference clock; $\mathcal{C}_2$ is the unknown clock
$\mathcal{O}_i$	the oscillator driving $\mathcal{C}_i$
$\mathcal{Z}_i$	the counter realising $\mathcal{C}_i$
δ_f^i	skew of clock $\mathcal{C}_i$
e_j^i	the time error at jth transition of the oscillator driving $\mathcal{C}_i$
c	the oscillator variance constant. c^1 is the variance constant of $\mathcal{C}_1$, c^2 is the variance constant of $\mathcal{C}_2$
T^i	the period of the oscillator driving $\mathcal{C}_i$. T_0^i denotes the nominal period of the oscillator, $T^i = T_0^i/(1+\delta_{fi})$
m	the value of $\mathcal{Z}_1$ at arbitrary time t. m_j is the value of $\mathcal{Z}_1$ associated with the jth sampling instant. $\hat{m}_j$ is the backward difference of $\mathcal{Z}_1$ values at successive sampling instants, $\hat{m}_j = m_j - m_{j-1}$ and $\hat{m}_1 = m_1$
k	the value of $\mathcal{Z}_2$ at arbitrary time t. k_j is the value of $\mathcal{Z}_2$ associated with the jth sampling instant. $\hat{k}_j$ is backward difference of $\mathcal{Z}_2$ values at successive sampling instants, $\hat{k}_j = k_j - k_{j-1}$ and $\hat{k}_1 = k_1$
C^i	the time report of $\mathcal{C}_i$ at arbitrary time t. C^1 is the time report of the reference clock, $\mathcal{C}_1$; C^2 is the time report of the clock $\mathcal{C}_2$. C_j^i is the time report of $\mathcal{C}_i$ at $t = t_j$. $\hat{C}_j^i$ is the backward difference of the time reports, where $\hat{C}_j^i = C_j^i - C_{j-1}^i$ and $\hat{C}_1^i = C_1^i$

doi: 10.1049/iet-wss.2011.0137

high-frequency oscillator. The low-frequency oscillator generates stable and low-power ticks for a low-granularity clock, while the high-frequency oscillator fills these ticks with high granularity phase information. The achieved synchronisation accuracy is much higher and the power requirement is much lower when compared with the results from the study by Maroti *et al.* [10]. In this paper, we show that this improvement is due to the implementation detail of resetting the high resolution counter for each low-frequency tick, which practically corresponds to the reference-triggered time relation model operating on time report increments.

As stated above, the time error originates from the non-ideal behaviour of the oscillators. This observation has prompted Freris *et al.* [17] to study a stochastic time relation model while including the observed time report correlations in the clock skew model. The proposed model is used for developing a pair-wise synchronisation algorithm for a handshake-based messaging scheme. The proposed model in the study by Freris *et al.* [17] is similar to the model we reach by adapting the oscillator model by Demir *et al.* [3].

The fundamental limits on synchronising network clocks was recently studied by Freris *et al.* [18]. They demonstrated theoretically that the clock skew can be determined arbitrarily well for a handshake-based time report exchange scheme, as earlier demonstrated experimentally in a study by Veitch *et al.* [19]. In this study, we avoid handshake-based time report exchanges owing to energy constraints.

In this study, we assume that stochastic wireless communication delay is an additive white Gaussian noise (AWGN) process. However, it is possible to model this delay component via exponential distribution [20, 21]. The developed recursive algorithm can be adapted for exponential delays by making straightforward modifications. In a study by Sari *et al.* [20], a simultaneous estimation of time offset and skew using the broadcast messages is proposed. However, the developed estimator is a batch-joint ML estimator and Gibbs sampler, and the underlying measurement model does not consider the correlations in the time reports.

3 Basic clock model

3.1 Oscillator model

An oscillator can be modelled as a non-linear autonomous system $\dot{x}(t) = f(x(t))$, which has a non-trivial solution $x_s(t) = x_s(t+T)$ with period T. This non-linear system is perturbed by a small amount to analyse the dynamics. We are interested in perturbations in the form

$$\dot{x}(t) = f(x(t)) + b(t) \tag{1}$$

The solution to this inhomogeneous differential equation is shown by Demir *et al.* [3]

$$x_p(t) = x_s(t + \alpha(t)) \tag{2}$$

where $\alpha(t)$ is defined as 'phase deviation'. If the perturbation is a linear combination of uncorrelated white noise sources, $\alpha(t)$ is a Gaussian process [3], $\alpha(t) \sim \mathcal{N}(\mu, \sigma^2(t))$.

It is possible to show that the mean of the phase deviation is constant, whereas the variance grows linearly in time [3]

$$\sigma^2(t) = ct \tag{3}$$

where c is the empirical oscillator constant.

It has also already been shown that the phase deviation at different time instants is correlated, but jointly Gaussian [3]. The covariance of the phase deviation process is

$$E\{\alpha(t)\alpha(t+\tau)\} = \mu^2 + c\min\{t, t+\tau\} \tag{4}$$

3.2 Time error model

Any oscillator output can be represented by a sinusoidal voltage [2]

$$V(t) = V_0 \sin\left(2\pi f_0\left(t + \phi_0 + \delta_f t + \tfrac{1}{2}Dt^2 + \epsilon(t)\right)\right) \tag{5}$$

where $f_0 = 1/T_0$ is the nominal clock frequency, ϕ_0 is the time offset, δ_f is the clock skew, D is the frequency drift and $\epsilon(t)$ represents random deviations. The 'time error', $\varphi(t) = \phi_0 + \delta_f t + (1/2)Dt^2 + \epsilon(t)$, is composed of short-term (ϕ_0, δ_f, ϵ) and long-term (D) errors. For short-term analysis, it is better to model the time error without frequency drift, $D = 0$ [2].

The phase deviation and the time error can be related to each other by using (5) as a solution for (1). In this case, the solution has a period $T = T_0/(1+\delta_f)$, and $\epsilon(t) = (1+\delta_f)\alpha(t) - \phi_0$ is a Gaussian process since $\alpha(t)$ is Gaussian

$$\epsilon(t) \sim \mathcal{N}((1+\delta_f)\mu - \phi_0, (1+\delta_f)^2\sigma^2(t))$$

Note that the distribution of the time error is characterised by the distribution of $\alpha(t)$, and the oscillator period uniquely defines the clock skew. This observation implies that in case the frequency drift is ignored, the time relation model can be defined in terms of the oscillator model rather than the time error model.

3.3 Time measurement system

A clock is usually composed of a counter driven by a crystal oscillator. The edge-triggered infinite counter increments every low-to-high transition of the oscillator output, that is, the counter increments once in a period. We assume that at $t = 0$, the counter is reset to zero. For this measurement system, the time error at the jth transition is

$$e_j = t - jT \tag{6}$$

where T is the period of the oscillator, and t is the actual time at jth transition. The timing error, e_j, is assumed to have a mean of zero and an increasing variance in accordance with (3)

$$e_j \sim \mathcal{N}(0, cjT)$$

The covariance of the time error between different transitions, $i \neq j$, is

$$E\{e_i e_j\} = cT\min(i, j) \tag{7}$$

Thus, the time error e_j is a Wiener process because of phase deviation.

4 Clock synchronisation

In this section, we introduce the time relation models for clock synchronisation. Based on the time measurement system, we first consider the reference-triggered time relation model and introduce a scalar parameter, 'clock skew ratio', to describe the clock synchronisation problem. We develop the ML estimate of the clock skew ratio for the reference-triggered time relation model. Furthermore, we introduce an external-triggered time relation model, and find the associated clock skew ratio estimate. Finally, we develop the recursive equivalent of the clock skew ratio estimate for the external-triggered time relation model.

4.1 Reference-triggered time relation model

Suppose that we have two independent oscillators, $\mathcal{O}_1$ and $\mathcal{O}_2$, with nominal periods of T_0^1 and T_0^2, respectively. Also, suppose that oscillators $\mathcal{O}_1$ and $\mathcal{O}_2$ are driving counters $\mathcal{Z}_1$ and $\mathcal{Z}_2$, respectively, which are both initialised to zero at $t = 0$. For this measurement model, the reference clock counter, $\mathcal{Z}_1$, is allowed to count exactly m low-to-high transitions of its oscillator output for each time record.

Assuming that at the first sampling instant, the transition counts of $\mathcal{Z}_1$ and $\mathcal{Z}_2$ are m and k, respectively, the relation between the count values is

$$e_m^1 + mT^1 = e_k^2 + kT^2$$

If the aim is to find T^2 with respect to T_0^1, it is possible to define the measurement model as

$$m = \frac{T^2(1+\delta_f^1)}{T_0^1}k + e_{m,k} \tag{8}$$

where the total error, $e_{m,k}$, is

$$e_{m,k} = \frac{1}{T^1}(e_k^2 - e_m^1)$$

Since e^1 and e^2 are independent Gaussian processes, the total error is

$$e_{m,k} \sim \mathcal{N}\left(0, \frac{m(c^1 + c^2)}{T^1}\right) \tag{9}$$

where the variance is defined with respect to the count value of $\mathcal{Z}_1$. The periods of the oscillators can be related to each other with a scalar parameter 'clock skew ratio', a

$$a \triangleq \frac{T^2}{T_0^1}(1+\delta_f^1) \tag{10}$$

Let $\boldsymbol{m}$ be a vector of N reports of $\mathcal{Z}_1$, and let $\boldsymbol{k}$ be the corresponding vector composed of the count values of $\mathcal{Z}_2$. Furthermore, let us assume k_j is recorded at the instant of the jth sample of $\mathcal{Z}_1$. We define the relation between progressive time records in $\boldsymbol{m}$ and $\boldsymbol{k}$ as reference-triggered progressive time (RPT) relation model

$$\begin{aligned} \boldsymbol{m} &= \begin{bmatrix} m, & 2m, & \ldots, & Nm \end{bmatrix}^{\mathrm{tr}} \\ \boldsymbol{k} &= \begin{bmatrix} k_1, & k_2, & \ldots, & k_N \end{bmatrix}^{\mathrm{tr}} \\ \boldsymbol{m} &= a\boldsymbol{k} + \boldsymbol{e} \end{aligned} \tag{11}$$

where $\boldsymbol{e}$ is the time error vector with the components $e_i = e_{im,k_i}$, and the superscript 'tr' represents matrix transpose. The components of the time error vector are jointly Gaussian $\boldsymbol{e} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma})$, and

$$\boldsymbol{\Sigma} = E\{\boldsymbol{e}\boldsymbol{e}^{\mathrm{tr}}\} = \frac{cm}{T^1}\boldsymbol{U}\boldsymbol{U}^{\mathrm{tr}}$$

where $\boldsymbol{U}$ is a lower triangular matrix with all the non-zero entries being unity

$$\boldsymbol{U} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 1 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{bmatrix}$$

Owing to the special nature of Σ, we observe the following properties:

- Determinant of Σ: $|\Sigma| = (cm/T^{-1})$
- Inverse of Σ

$$\boldsymbol{\Sigma}^{-1} = \frac{T^1}{cm}\begin{bmatrix} 2 & -1 & 0 & 0 & \cdots & 0 & 0 \\ -1 & 2 & -1 & 0 & \cdots & 0 & 0 \\ 0 & -1 & 2 & -1 & \cdots & 0 & 0 \\ 0 & 0 & \ddots & \ddots & \ddots & 0 & 0 \\ 0 & 0 & \cdots & -1 & 2 & -1 & 0 \\ 0 & 0 & \cdots & 0 & -1 & 2 & -1 \\ 0 & 0 & \cdots & 0 & 0 & -1 & 1 \end{bmatrix}$$

4.2 Clock skew estimation

An unknown clock can be identified by estimating the clock skew ratio using the RPT defined in (11). The −log likelihood function of a for RPT is

$$L(a) = K + \frac{1}{2}(a\boldsymbol{k} - \boldsymbol{m})^{\mathrm{tr}}\boldsymbol{\Sigma}^{-1}(a\boldsymbol{k} - \boldsymbol{m}) \tag{12}$$

where K is a constant term independent of a. The ML estimate, the optimum a minimising −log likelihood function, is obtained after straightforward manipulations as

$$\hat{a}_{ml}(N) = \frac{m\boldsymbol{\Sigma}_{i=1}^{N-1}(k_{i+1} - k_i)}{k_1^2 + \boldsymbol{\Sigma}_{i=1}^{N-1}(k_{i+1} - k_i)^2} \tag{13}$$

which implies that the ML estimate for the clock skew ratio is a function of the recorded successive time reports of the counters instead of the progressive output of the counters. Therefore the backward difference vectors of the counter

doi: 10.1049/iet-wss.2011.0137

outputs provide notational simplicity

$$\hat{\boldsymbol{k}} = \begin{bmatrix} k_1, & \hat{k}_2, & \ldots, & \hat{k}_N \end{bmatrix}^{\mathrm{tr}} = \boldsymbol{U}^{-1}\boldsymbol{k}$$

$$\hat{\boldsymbol{m}} = m\begin{bmatrix} \mathbf{1}, & \mathbf{1}, & \ldots, & 1 \end{bmatrix}^{\mathrm{tr}} = m\mathbf{1} = \boldsymbol{U}^{-1}\boldsymbol{m}$$

Measurement model (11) is equivalently represented by the reference-triggered incremental time (RIT) relation model.

$$\hat{\boldsymbol{m}} = a\hat{\boldsymbol{k}} + \hat{\boldsymbol{e}} \tag{14}$$

where the noise sequence, $\hat{\boldsymbol{e}}$, is composed of independent and identically distributed random variables

$$\hat{e}_j = (e_j - e_{j-1}) \sim \mathcal{N}\left(0, \frac{cm}{T^1}\right)$$

In other words, the joint density of the measurement noise sequence is $\hat{\boldsymbol{e}} \sim \mathcal{N}(\mathbf{0}, (cm/T^1)\boldsymbol{I}_N)$ where $\boldsymbol{I}_N$ is the $N \times N$ 'identity' matrix. We abbreviate the associated ML estimate (MLE) of the clock skew ratio for RIT as MLE-RIT, which is given by

$$\hat{a}_{ml}(N) = \frac{\hat{\boldsymbol{k}}^{\mathrm{tr}}\hat{\boldsymbol{m}}}{\hat{\boldsymbol{k}}^{\mathrm{tr}}\hat{\boldsymbol{k}}} \tag{15}$$

It should be noted that since the noise vector $\hat{\boldsymbol{e}}$ is composed of independent, identically distributed Gaussian components, the MLE is equivalent to the LSE and minimum variance estimate for a when using the RIT. However, for consistency we refer to this estimate as MLE-RIT.

4.3 External-triggered time relation model

Thus far, we have assumed that every m increment of the reference counter, $\mathcal{Z}_1$, perfectly triggers an instantaneous event for recording the counts of $\mathcal{Z}_2$ to keep the analysis tractable. However, in practice it is more convenient to relax the assumption to a simultaneous recording of two counters, where an external recording trigger is phase-locked to the low-to-high transitions of the oscillator driving $\mathcal{Z}_1$ to prevent additional time jitter on the records. In addition, successive recording events are generated such that the reference counter has incremented at least once, that is, $\hat{m}_j \geq 1$. Therefore the backward difference vector of the $\mathcal{Z}_1$ output records for this measurement system is

$$\hat{\boldsymbol{m}} = \begin{bmatrix} m_1, & \hat{m}_2, & \ldots, & \hat{m}_N \end{bmatrix}^{\mathrm{tr}}$$

For notational simplicity, let us denote the ratio of count increments between successive recording instants with

$$\hat{h}_j \triangleq \frac{\hat{k}_j}{\hat{m}_j}$$

Then, model (14) turns out to be the external-triggered incremental time (EIT) relation model

$$\mathbf{1} = a\hat{\boldsymbol{h}} + \hat{\boldsymbol{e}}_m \tag{16}$$

where $\hat{e}_m$ is also Gaussian, but the components are not identically distributed

$$\hat{e}_m \sim \mathcal{N}\left(\mathbf{0}, \frac{c}{T^1}\mathrm{diag}\left(\frac{\mathbf{1}}{\hat{m}_1}, \frac{\mathbf{1}}{\hat{m}_2}, \ldots, \frac{\mathbf{1}}{\hat{m}_N}\right)\right)$$

The $-\log$ likelihood function of a for EIT in (16) is

$$L(a) = K + \frac{\mathbf{1}}{2}\sum_{j=1}^{N}\frac{\hat{m}_j(1 - a\hat{h}_j)^2}{c/T^1} \tag{17}$$

Then, the ML estimate for the clock skew ratio is the optimum a that minimises (17)

$$\hat{a}(N) = \frac{\sum_{i=1}^{N}\hat{k}_i}{\sum_{i=1}^{N}(\hat{k}_i^2)/(\hat{m}_i)} = \frac{\mathbf{1}^{\mathrm{tr}}\hat{\boldsymbol{k}}}{\hat{\boldsymbol{k}}^{\mathrm{tr}}\hat{\boldsymbol{h}}} \tag{18}$$

We call this estimate the MLE for EIT relation model (MLE-EIT).

4.4 Recursive skew estimator

In order to obtain a recursive estimator, we rewrite (18) as

$$\begin{aligned}\mathbf{1}^{\mathrm{tr}}\hat{\boldsymbol{k}} &= \hat{a}(N)(\hat{\boldsymbol{k}}^{\mathrm{tr}}\hat{\boldsymbol{h}}) \\ &= \hat{a}(N)(\hat{\boldsymbol{k}}^{\mathrm{tr}}\hat{\boldsymbol{h}} + \hat{k}_{N+1}\hat{h}_{N+1} - \hat{k}_{N+1}\hat{h}_{N+1})\end{aligned} \tag{19}$$

The estimate of a for $N + \mathbf{1}$ measurements is

$$\hat{a}(N + \mathbf{1}) = \frac{\mathbf{1}^{\mathrm{tr}}\hat{\boldsymbol{k}} + \hat{k}_{N+1}}{\hat{\boldsymbol{k}}^{\mathrm{tr}}\hat{\boldsymbol{h}} + \hat{k}_{N+1}\hat{h}_{N+1}} \tag{20}$$

Substituting (19) and (20) yields the recursive estimator as

$$\hat{a}(N + \mathbf{1}) = \hat{a}(N) + \frac{\hat{k}_{N+1}}{\hat{\boldsymbol{k}}^{\mathrm{tr}}\hat{\boldsymbol{h}} + \hat{k}_{N+1}\hat{h}_{N+1}}(\mathbf{1} - \hat{h}_{N+1}\hat{a}(N)) \tag{21}$$

5 Comparison of skew estimators

In this section, we compare MLE-EIT with MLE-RIT, a constant estimate for RIT (CE-RIT), and the LSE for RPT (LSE-RPT). It should be noted that the reference-triggered time relation model represents a special case of the external-triggered time relation model. Thus, the clock skew estimates for the reference-triggered model are expected to be better. We are not developing an LSE for an external-triggered progressive time relation model, since it is difficult to analyse and we expect that it will not perform as well as the LSE-RPT.

It is customary to say that one estimator is better than another by comparing estimate-wise and asymptotic values of the first two moments of the estimation error $\tilde{a} = \hat{a} - a$ [14]. The first moment is a measure of the mean deviation from the actual parameter, that is, the 'bias' of the estimate. The variance of the estimation error is a measure of how far the estimate can be from the actual parameter. For unbiased estimators, the ratio of the lower bound to the estimation error variance is defined as the 'efficiency'. If the variance is equal to the lower bound, the estimator is 'efficient'. The asymptotic behaviour of the estimation error variance is a measure of convergence; the estimate must converge to the actual parameter at least in the mean-square sense. If the estimate converges to the actual parameter, the estimator is 'consistent'. Consequently, one estimator compares better with another one if it is unbiased, efficient and consistent whereas the other one is not.

Table 2 summarises the statistical quality measures of these estimators. When considering the practical applicability of the estimators and the underlying time relation models, as well as their characteristics given in Table 2, it is possible to state that MLE-EIT is better than the others.

5.1 Estimation quality analysis for MLE-EIT

The estimation error for MLE-EIT can be obtained by substituting (16) and (18)

$$\tilde{a}(N) = \frac{\hat{\boldsymbol{k}}^{\mathrm{tr}}\hat{\boldsymbol{e}}_m}{\hat{\boldsymbol{k}}^{\mathrm{tr}}\hat{\boldsymbol{h}}} \tag{22}$$

Since $\hat{e}_m$ in (16) is zero mean, MLE-EIT is unbiased. The variance of the estimation error is

$$E\{\tilde{a}^2(N)\} = \frac{\hat{\boldsymbol{k}}^{\mathrm{tr}}E\{\hat{\boldsymbol{e}}_m\hat{\boldsymbol{e}}_m^{\mathrm{tr}}\}\hat{\boldsymbol{k}}}{(\hat{\boldsymbol{k}}^{\mathrm{tr}}\hat{\boldsymbol{h}})^2} = \frac{c}{T^1}\frac{\mathbf{1}}{\sum_{j=1}^{N}(\hat{k}_j^2/\hat{m}_j)} \tag{23}$$

In order to analyse the asymptotic behaviour of the estimation error, it is more convenient to replace $\hat{m}_j$ with the largest possible count increment between the count records, M. In such cases, the estimation error variance always satisfies

$$E\{\tilde{a}^2(N)\} = \frac{c}{T^1}\frac{\mathbf{1}}{\sum_{j=1}^{N}(\hat{k}_j^2/\hat{m}_j)} \le \frac{cM}{T^1}\frac{\mathbf{1}}{\sum_{j=1}^{N}\hat{k}_j^2} \tag{24}$$

The estimation error variance becomes identically zero if $\hat{m}_j = 0$ for all j. This case can only occur if the reference counter does not increment throughout the synchronisation process. In practice, the reference clock is restricted to increment at least once for each count record. If the entries for $\hat{\boldsymbol{k}}$ are all zeros, the estimation error variance is infinite. This condition can only occur in case $\mathcal{Z}_2$ is not incremented between successive records, which makes the clock skew ratio unobservable. If $\sum_{j=1}^{N}\hat{k}_j^2 \ge \mathbf{1}$, it is possible to generalise the upper bound of the estimation error variance

$$E\{\tilde{a}^2(N)\} \le \frac{cM}{T^1}\frac{\mathbf{1}}{\sum_{j=1}^{N}\hat{k}_j^2} \le \frac{cM}{T^1} \tag{25}$$

Since MLE-EIT is unbiased, the lower bound of the estimation error variance is given by the Cramér–Rao inequality, which states that the achievable lower bound is the inverse of 'Fisher information', J [14]. Using the likelihood function in (17), the Fisher information can be calculated as

$$J = -E\left\{\frac{\partial^2(-L(a))}{\partial a^2}\right\} = \left(\frac{c}{T^1}\frac{\mathbf{1}}{\sum_{j=1}^{N}(\hat{k}_j^2/\hat{m}_j)}\right)^{-1} = E^{-1}\{\tilde{a}^2(N)\} \tag{26}$$

Thus, the current estimator is efficient.

If the sum $\sum_{i=1}^{N}\hat{k}_i^2$ increases monotonically, the time record set is sufficiently rich, that is, if $\lim_{N\to\infty}\sum_{i=1}^{N}\hat{k}_i^2 = \infty$, the estimate converges to true skew in mean square. This follows from the fact that the estimation error variance asymptotically approaches zero as the number of time records increases. Consequently, the estimator is consistent [14].

It should be noted that the estimation error variance is linear with M. As the allowed maximum increment of the reference counter increases, the estimation error variance increases. Thus, the estimation error variance can be limited to a suitable value by adjusting the maximum allowable duration between the count records.

5.2 Estimation quality analysis for MLE-RIT

The estimation error for MLE-RIT is

$$\tilde{a}_{ml}(N) = \frac{\hat{\boldsymbol{k}}^{\mathrm{tr}}\hat{\boldsymbol{e}}}{\hat{\boldsymbol{k}}^{\mathrm{tr}}\hat{\boldsymbol{k}}} \tag{27}$$

The mean of the estimation error is zero, whereas the variance

Table 2 Estimation error characteristics of different estimators

Property/estimator	MLE-EIT	MLE-RIT	CE-RIT	LSE-RPT
time relation model	$\mathbf{1} = a\hat{\boldsymbol{h}} + \hat{\boldsymbol{e}}_m$	$\hat{\boldsymbol{m}} = a\hat{\boldsymbol{k}} + \hat{\boldsymbol{e}}$	$\hat{\boldsymbol{m}} = a\hat{\boldsymbol{k}} + \hat{\boldsymbol{e}}$	$\boldsymbol{m} = a\boldsymbol{k} + \boldsymbol{e}$
estimate	$\hat{a}(N) = \frac{\mathbf{1}^{\mathrm{tr}}\hat{\boldsymbol{k}}}{\hat{\boldsymbol{k}}^{\mathrm{tr}}\hat{\boldsymbol{h}}}$	$\hat{a}_{ml}(N) = \frac{\hat{\boldsymbol{k}}^{\mathrm{T}}\hat{\boldsymbol{m}}}{\hat{\boldsymbol{k}}^{\mathrm{T}}\hat{\boldsymbol{k}}}$	$\hat{a}_{ce} = \frac{T_0^1}{T_0^2}$	$\hat{a}_{ls}(N) = \frac{\boldsymbol{k}^{\mathrm{T}}\boldsymbol{m}}{\boldsymbol{k}^{\mathrm{T}}\boldsymbol{k}}$
error	$\tilde{a}(N) = \frac{\hat{\boldsymbol{k}}^{\mathrm{tr}}\hat{e}_m}{\hat{\boldsymbol{k}}^{\mathrm{tr}}\hat{\boldsymbol{h}}}$	$\tilde{a}_{ml} = \frac{\hat{\boldsymbol{k}}^{\mathrm{T}}\hat{\boldsymbol{e}}}{\hat{\boldsymbol{k}}^{\mathrm{T}}\hat{\boldsymbol{k}}}$	$\tilde{a}_{ce} = \frac{T_0^1}{T_0^2} - \frac{\hat{\boldsymbol{k}}^{\mathrm{T}}(\hat{\boldsymbol{m}} - \hat{\boldsymbol{e}})}{\hat{\boldsymbol{k}}^{\mathrm{T}}\hat{\boldsymbol{k}}}$	$\tilde{a}_{ls} = \frac{\boldsymbol{k}^{\mathrm{T}}\boldsymbol{e}}{\boldsymbol{k}^{\mathrm{T}}\boldsymbol{k}}$
bias	0	0	$\frac{T_0^1}{T_0^2} - \frac{\hat{\boldsymbol{k}}^{\mathrm{T}}\hat{\boldsymbol{m}}}{\hat{\boldsymbol{k}}^{\mathrm{T}}\hat{\boldsymbol{k}}}$	0
error variance	$\frac{c}{T^1}\frac{\mathbf{1}}{\sum_{j=1}^{N}(\hat{k}_j^2/\hat{m}_j)}$	$\frac{cm}{T^1}\frac{\mathbf{1}}{\sum_{j=1}^{N}\hat{k}_j^2}$	$\frac{cm}{T^1}\frac{\mathbf{1}}{\sum_{j=1}^{N}\hat{k}_j^2}$	$\frac{\boldsymbol{k}^{\mathrm{T}}\Sigma\boldsymbol{k}}{\boldsymbol{k}^{\mathrm{T}}\boldsymbol{k}}$
unbiased	yes	yes	no	yes
efficient	yes	yes	yes	no
consistent	yes	yes	yes	yes

IET Wirel. Sens. Syst., 2012, Vol. 2, Iss. 4, pp. 338–350
doi: 10.1049/iet-wss.2011.0137

is

$$E\{\tilde{a}_{ml}^2(N)\} = \frac{cm}{T^1}\frac{\mathbf{1}}{\boldsymbol{\Sigma}_{j=1}^{N}\hat{k}_j^2} \tag{28}$$

The estimation error variance decreases as the number of time records increases, since $\lim_{N\to\infty}\boldsymbol{\Sigma}_{i=1}^{N}\hat{k}_i^2 = \infty$. Thus, the estimate converges to the true parameter in mean square, that is, the estimator is consistent. Moreover, it is easy to show that the estimation error variance satisfies Cramér–Rao lower bound with equality, that is, the estimator is efficient.

5.3 Constant estimate for RIT

The constant estimator keeps the estimate at a predetermined constant value regardless of the measurement, $\hat{a}(N) = a^0$. The estimation error for CE-RIT is

$$\tilde{a}_{ce} = a^0 - a = a^0 - \frac{\hat{\boldsymbol{k}}^{\mathrm{tr}}(\hat{\boldsymbol{m}} - \hat{\boldsymbol{e}})}{\hat{\boldsymbol{k}}^{\mathrm{tr}}\hat{\boldsymbol{k}}} \tag{29}$$

The mean of the CE-RIT estimation error is

$$E\{\tilde{a}_{ce}\} = a^0 - \frac{\hat{\boldsymbol{k}}^{\mathrm{tr}}\hat{\boldsymbol{m}}}{\hat{\boldsymbol{k}}^{\mathrm{tr}}\hat{\boldsymbol{k}}} \tag{30}$$

The CE-RIT is asymptotically unbiased if a^0 is selected in accordance with the nominal periods of the oscillators, since, on average, the oscillators have the specified periods.

$$a^0 = \frac{T_0^1}{T_0^2} \tag{31}$$

The estimation error variance is equivalent to the estimation error variance of MLE-RIT, thus CE-RIT is consistent and efficient

$$E\{(\tilde{a}_{ce} - E\{\tilde{a}_{ce}\})^2\} = \frac{cm}{T^1}\frac{\mathbf{1}}{\boldsymbol{\Sigma}_{j=1}^{N}\hat{k}_j^2} \tag{32}$$

The constant estimator can be used if it is possible to reset the counter $\mathcal{Z}_2$ for each m increment of $\mathcal{Z}_1$. If there is an additional time jitter in the records, the bias in the estimates will be large. Consequently, such an estimator can only be used for the systems under nearly ideal conditions. For example, it can be used to synchronise two software clocks in the same microprocessor that are driven by two independent oscillators.

5.4 LSE for RPT

The LSE for RPT in (11) is the optimum a, which minimises $F(a) = (1/2)|a\boldsymbol{k} - \boldsymbol{m}|^2$. Thus, based on the RPT (LSE-RPT), the LSE of the clock skew ratio is

$$\hat{a}_{ls}(N) = \frac{\boldsymbol{k}^{\mathrm{tr}}\boldsymbol{m}}{\boldsymbol{k}^{\mathrm{tr}}\boldsymbol{k}} \tag{33}$$

The estimation error of LSE-RPT is obtained by substituting (11) and (33)

$$\tilde{a}_{ls} = \frac{\boldsymbol{k}^{\mathrm{tr}}\boldsymbol{e}}{\boldsymbol{k}^{\mathrm{tr}}\boldsymbol{k}} \tag{34}$$

The mean of the estimation error is zero, whereas the variance is

$$E\{\tilde{a}_{ls}^2\} = \frac{\boldsymbol{k}^{\mathrm{tr}}\boldsymbol{\Sigma}\boldsymbol{k}}{(\boldsymbol{k}^{\mathrm{tr}}\boldsymbol{k})^2} = \frac{cm}{T^1}\frac{\boldsymbol{\Sigma}_{i=1}^{N}k_i\left(\boldsymbol{\Sigma}_{j=1}^{i}\mathrm{j}k_j + \boldsymbol{\Sigma}_{j=i+1}^{N}\mathrm{i}k_j\right)}{\left(\boldsymbol{\Sigma}_{i=1}k_i^2\right)^2} \tag{35}$$

The estimation error variance converges to zero as the number of time records increases, since the sum in the denominator diverges faster than the one in the nominator. Thus, LSE-RPT is consistent.

Fisher information of LSE-RPT can be found using the likelihood function in (12)

$$\begin{aligned} J &= -E\left\{\frac{\partial^2(-L(a))}{\partial a^2}\right\} = \boldsymbol{k}^{\mathrm{tr}}\boldsymbol{\Sigma}^{-1}\boldsymbol{k} \\ &= \frac{T^1}{cm}\left(k_1^2 + \sum_{j=1}^{N-1}(k_{j+1} - k_j)^2\right) \end{aligned} \tag{36}$$

Then, the Cramér–Rao lower bound is

$$\boldsymbol{k}^{\mathrm{tr}}(\Sigma^{-1}\boldsymbol{k}\boldsymbol{k}^{\mathrm{tr}}\boldsymbol{\Sigma} - \boldsymbol{k}\boldsymbol{k}^{\mathrm{tr}})\boldsymbol{k} \geq 0$$

For non-zero $\boldsymbol{k}$, the equality holds only if $\boldsymbol{k}\boldsymbol{k}^{\mathrm{tr}}$ commutes with Σ, which is only possible for $\boldsymbol{k} = \hat{\boldsymbol{k}}$. Since the last equality contradicts the definition for the vectors, LSE-RPT is not efficient for finite N.

The recursive equivalent of LSE-RPT can be easily derived as

$$\begin{aligned} \hat{a}_{ls}(N+\mathbf{1}) &= \hat{a}_{ls}(N) \\ &\quad + \frac{k_{N+1}}{\boldsymbol{k}^{\mathrm{tr}}\boldsymbol{k} + k_{N+1}^2}(m_{N+1} - \hat{a}_{ls}(N)k_{N+1}) \end{aligned} \tag{37}$$

6 Time synchronisation using reference broadcasts

In the previous analysis, we addressed the time synchronisation problem without time jitter resulting from communication delay. In order to analyse the effects of wireless communication, we consider the basic scenario where the time reports of $\mathcal{C}_1$ are broadcasted every Δ seconds, while the time reports of $\mathcal{C}_2$ are recorded after receiving the broadcasts. We assume that the wireless transmission introduces a deterministic delay, $x(t)$, and an additive white Gaussian jitter, η, on the received time reports.

6.1 Reference broadcast time relation model

The time reports of two independent clocks, $\mathcal{C}_1$ and $\mathcal{C}_2$, at an arbitrary instant, t, are

$$C^1(t) = (\mathbf{1} + \delta_f^1)(t + \alpha^1(t)) + \phi_0^1$$

$$C^2(t) = (\mathbf{1} + \delta_f^2)(t + \alpha^2(t)) + \phi_0^2$$

where ϕ_0^1 and ϕ_0^2 are the time offsets and $\alpha^1(t)$ and $\alpha^2(t)$ are the phase deviations of the clocks $\mathcal{C}_1$ and $\mathcal{C}_2$, respectively. The time reports of these clocks are related to each other via the

actual time t

$$C^1(t) = \frac{\mathbf{1} + \delta_f^1}{\mathbf{1} + \delta_f^2} C^2(t) + (\mathbf{1} + \delta_f^1)\alpha^{1,2}(t) + \phi_0^{1,2} \tag{38}$$

where

$$\alpha^{1,2}(t) = \alpha^1(t) - \alpha^2(t)$$

$$\phi_0^{1,2} = \phi_0^1 - \frac{\mathbf{1} + \delta_f^1}{\mathbf{1} + \delta_f^2}\phi_0^2$$

Note that $\alpha^{1,2}(t) \sim \mathcal{N}(0, (\sigma^1(t))^2 + (\sigma^2(t))^2)$, where $(\sigma^1(t))^2$ and $(\sigma^2(t))^2$ are the variances of phase deviations. If both of the clocks are measuring the time using counters driven by oscillators with nominal periods T_0^1 and T_0^2, the clock relation in (38) can be represented by the following

$$C^1(t) = a\frac{T_0^1}{T_0^2}C^2(t) + \frac{T_0^1}{T^1}\alpha^{1,2}(t) + \phi_{1,2}^0 \tag{39}$$

6.1.1 Progressive time relation model: Let C_j^1 denote the time report of clock $\mathcal{C}_1$ included in the jth broadcast. Suppose that t_j is the actual time when the jth broadcast is received and that the time report of $\mathcal{C}_2$ is recorded as C_j^2. The received reference time is related to the recorded time report of $\mathcal{C}_2$ according to the following equation

$$C_j^1 = a\frac{T_0^1}{T_0^2}C_j^2 + \left(\frac{T_0^1}{T^1}\alpha^{1,2}(t_j) - \eta_j\right) + (\phi_0^{1,2} - x_j) \tag{40}$$

where for the AWGN delay model, $\eta_j \sim \mathcal{N}(\mu_\eta, \sigma_\eta^2)$. Without sacrificing generality we assume that the mean of the stochastic delay is included in the deterministic part, $\mu_\eta = 0$.

The relation between the received reference time and the recorded local time in (38) is compliant with the progressive time model in (8) with the exception of the offset term, $(\phi_0^{1,2} - x_j)$. If we assume that the offset term is compensated by some other means, the concepts derived in Section 4 apply to time synchronisation when using the reference broadcasts. However, we take another approach to derive an incremental time relation model.

6.1.2 Incremental time relation model: The phase deviation has uncorrelated disjoint increments, which implies that $\hat{\alpha}(\tau) = \alpha^{1,2}(t+\tau) - \alpha^{1,2}(t)$ is also uncorrelated with $\alpha^{1,2}(t)$ [3]. Thus, since the communication jitter is AWGN, C_j^i is uncorrelated with $\hat{C}_{j+1}^i$. Then, it is possible to define an incremental time measurement model as

$$\hat{C}_j^1 = a\frac{T_0^1}{T_0^2}\hat{C}_j^2 + \frac{T_0^1}{T^1}\hat{\alpha}_j^{1,2} - \hat{\eta}_j - \hat{x}_j \tag{41}$$

where 'hatted' variables represent backward difference. Furthermore, if we assume that the deterministic communication delay is constant between successive broadcasts, $\hat{x}_j = 0$, the incremental network time measurement model can be simplified as follows

$$1 \simeq a\frac{T_0^1}{T_0^2}\frac{\hat{C}_j^2}{\hat{C}_j^1} + \hat{\beta}_j \tag{42}$$

where $\hat{\beta}_j = (1/\hat{C}_j^1)((T_0^1/T^1)\hat{\alpha}_j^{1,2} - \hat{\eta}_j)$ has a normal distribution.

6.2 Reference broadcast clock skew estimation

6.2.1 Skew estimation for progressive time relation model: The progressive time relation model in (40) can be simplified by assuming that the communication delay is dominant or by ignoring the correlation among the time records

$$\left(\frac{T_0^1}{T^1}\alpha^{1,2}(t_j) - \eta_j\right) \simeq -\eta_j$$

The commonly used time relation model [1, 4] for broadcast-based time synchronisation is equivalent to

$$C_j^1 = a\frac{T_0^1}{T_0^2}C^2(t_j) - \eta_j + \phi_0(t_j) \tag{43}$$

where the total time offset is denoted by $(\phi_0^{1,2} - x_j) = \phi_0(t_j)$.

In case the time offset is compensated by some other means at $t = 0$, (43) is equivalent to (11). Thus, the N sample LSE of the clock skew ratio when using the progressive time relation model is given by LSE-RPT in (33)

$$\hat{a}_{ls}(N) = \frac{T_0^1}{T_0^2}\frac{\mathbf{\Sigma}_{j=1}^N (C_j^2)^2}{\mathbf{\Sigma}_{j=1}^N C_j^1 C_j^2}$$

This estimate is equivalent to the clock skew ratio estimate of LS linear regression with table length N and has large estimation error variance for finite N. This follows from the fact that the variance of the samples of $(T_0^1/T^1)\alpha^{1,2}(t_j)$ monotonically increases with t_j. Thus, we cannot assume that the communication jitter dominates $\alpha^{1,2}(t_j)$ for all t_j.

6.2.2 Recursive skew estimation for incremental time relation model: The network time measurement model in (42) is an incremental time model. In case the reference broadcasts are strictly periodic, and all the broadcasts are received properly, this model is equivalent to the reference-triggered incremental time model. In practice, however, the $\mathcal{C}_2$ record is usually triggered by receiving a broadcast, which is independent of both clocks. The reference broadcasts encounter varying deterministic delays, which changes the spacing between successive receptions. Thus, the external trigger time relation model describes network time synchronisation in a better way when using the reference broadcasts, and the clock skew ratio can be estimated using MLE-EIT in (18) or its recursive equivalent in (21).

The N sample MLE of this model is given by

$$\hat{a}(N) = \frac{T_0^1}{T_0^2}\frac{\mathbf{\Sigma}_{j=1}^N \hat{C}_j^2}{\mathbf{\Sigma}_{j=1}^N \hat{C}_j^2(\hat{C}_j^2/\hat{C}_j^1)} \tag{44}$$

which can be updated recursively by

$$\hat{a}(N+1) = \hat{a}(N) + \frac{T_0^2}{T_0^1}\frac{\hat{C}_{N+1}^2}{\mathbf{\Sigma}_{j=1}^{N+1}\hat{C}_j^2(\hat{C}_j^2/\hat{C}_j^1)}\left(1 - \hat{a}(N)\frac{T_0^1}{T_0^2}\frac{\hat{C}_{N+1}^2}{\hat{C}_{N+1}^1}\right) \tag{45}$$

6.2.3 Synchronised time: For most of the WSN applications, it is crucial to have a network-wide coherent

doi: 10.1049/iet-wss.2011.0137

notion of time. At an arbitrary time instant, $t \geq t_{N-1}$, the time report of $\mathcal{C}_2$ can be translated into reference time by

$$G^2(t) = C^1_{N-1} + \hat{a}(N-\mathbf{1})T^1_0\hat{k}_c \tag{46}$$

where $\hat{k}_c$ is the current reported count of $\mathcal{Z}_2$, initialised at $t = t_{N-1}$.

6.2.4 Time offset estimation: The time offset and the deterministic communication delay are not distinguishable for the time model in (40). However, the total time offset can be estimated using

$$C^1_j - C^1_{j-1} - \hat{a}(j-\mathbf{1})T^1_0\hat{k}_j = \phi^{1,2}_0 + x_j + \eta_j \tag{47}$$

where the phase deviation increment is ignored since we are interested in a narrow time window. If the random communication delay is an independent, identically distributed, zero mean Gaussian sequence, the offset estimate, $o = \phi^{1,2}_0 + x_j$, is

$$\hat{o}(N) = \frac{1}{N-1}\sum_{i=2}^{N}(C^1_j - C^1_{j-1} - \hat{a}(j-\mathbf{1})T^1_0\hat{k}_j) \tag{48}$$

It should be noted that two measurements (broadcasts) are enough to estimate the total clock offset provided that the clock skew ratio estimate has already converged.

7 Practical issues

7.1 Numerical sensitivity

The clock skew ratio is expected to be very close to T^1_0/T^2_0 since the drift is usually a slowly varying process. Hence, for a finite word length processor, the numerical error sensitivity of the estimator in (21) is high. We use (21) to develop a numerically more stable estimator

$$\hat{b}(N+\mathbf{1}) = \hat{b}(N) + \frac{\hat{k}_{N+1}}{\hat{\boldsymbol{k}}^{\mathrm{T}}\hat{\boldsymbol{h}} + \hat{k}_{N+1}\hat{h}_{N+1}}(r_{N+1} - \hat{h}_{N+1}\hat{b}(N)) \tag{49}$$

where

$$\hat{b}(N) = \frac{\mathbf{1}}{T^0_2}(\hat{a}(N) - \mathbf{1}) \tag{50}$$

$$r_N = \frac{\mathbf{1}}{T^0_2}(\mathbf{1} - \hat{h}_N) \tag{51}$$

The estimate for $\hat{b}(N)$ is less sensitive to numerical errors, since T^2_0 is usually within the range of microseconds making $\hat{b}(N)$ around a million times greater than $\hat{a}(N)$.

7.2 Time data consistency

The deterministic communication delay is time varying owing to sources indicated by Maroti *et al.* [10]. The local time of the receiver node may jump ahead of the received reference time more than the model can tolerate, which results in an incorrect update of the clock skew ratio estimate. Thus, the received time data needs to be checked for consistency.

We define an empirical parameter to check the consistency of the received reference time

$$\left|\hat{C}^1_j - a\frac{T^1_0}{T^2_0}\hat{C}^2_j\right| \leq K_{\max} \tag{52}$$

If this inequality fails, the received time data is discarded. Since our development does not assume periodic reception of the reference broadcasts, the skew ratio is updated whenever a valid reference broadcast is received. $K_{\max}$ depends on the beacon period and platform dependent parameters, such as the indication type for the received data; at which layer of the communication stack the timestamp is appended to the transmitted data. Thus, it should be adjusted according to the application accuracy requirements and communication software implementation.

8 Experimental evaluation

8.1 Test platform

The developed theory is verified using a wireless sensor platform. The platform is built on CC2431 by Texas Instruments and driven by a 32 MHz crystal oscillator. It is programmed to run FreeRTOS real-time kernel [22] and NanoStack communication stack [23]. Local clock is implemented as a 16-bit counter, which is configured to increment with a nominal period of 4 μs. The communication stack is modified to timestamp incoming and outgoing beacon frames as described by Maroti *et al.* [10]. The details of the measurement setup are described in a study by Mahmood and Jäntti [13].

A measurement scenario with two nodes, Node A and Node B, is set up based on the described platform. Node A sends a synchronisation beacon with the current local time (counter value) every $\Delta = 1$ s. Node B compensates for the time offset of its clock by using the reference time transmitted by Node A for the first five beacons by calculating the mean difference. Later, the clocks of the nodes are left undisciplined, that is, the counters of Node A and Node B remain free running.

8.2 Experimental results

In this section, the time is represented in terms of counter ticks (1 tick = 4 μs for our platform) in order to make the results independent of the resolution of the counters. We evaluate the performance of a skew estimation algorithm with the difference between the received reference time record of node A and the skew-compensated time record of node B, which is referred by 'Time Error' in this section. Hence, all the results are in terms of variation of time error with broadcast index, j.

$$\text{Time error} = C^1_j - \hat{a}C^2_j$$

where $\hat{a}$ is the clock skew estimate of any skew estimation algorithm associated with the jth broadcast. In our experiment, the time error without skew compensation, $\hat{a} = 1$, grows by 2.4 μs/s [13].

Fig. 2 shows the variation of time error with the clock skew ratio estimated with different algorithms. The skew estimates for the recursive MLE-EIT in (21), the recursive LSE-RPT in (37), and the LS linear regression with a table length of 8 are used to correct the local time for each

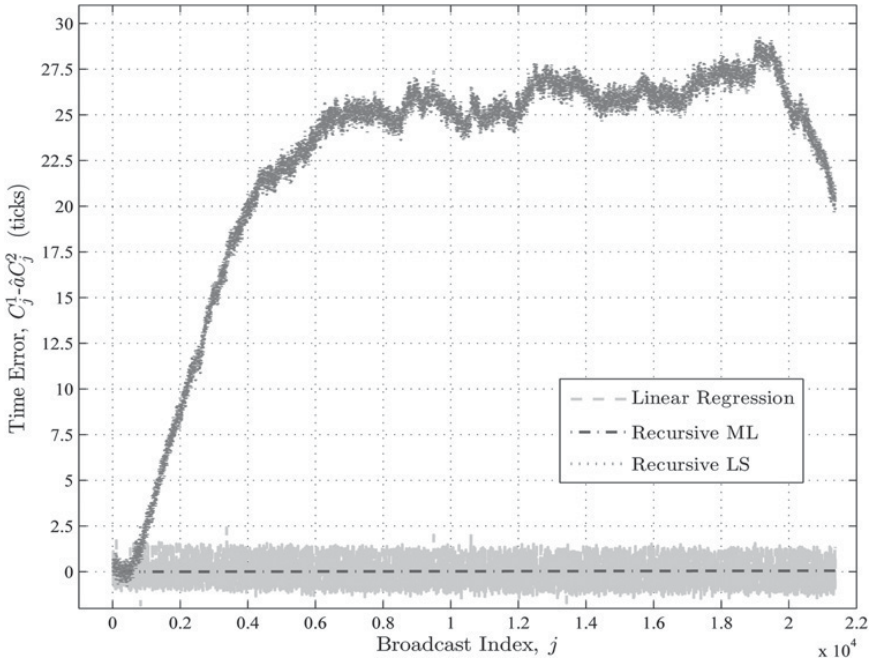


Fig. 2 *Comparison of the time error variation with different skew estimation algorithms*

broadcast. The LS linear regression method is implemented in the same way as Maroti *et al.* [10] did in their study. The results clearly show the superior performance of the developed algorithm. Considering the fact that the regression table length is low and the estimation update period is the same as the broadcast period, the lower performance of the linear regression estimate is mainly due to a high response time and estimation error variance. The lower performance of the recursive LSE compared with the developed estimator reveals a need to reformulate the synchronisation problem.

Fig. 3 shows the time error variation with different $K_{\max}$ values in (52). Note that $K_{\max}$ is also expressed in ticks for this experimental setup. The clocks are synchronised by compensating for the skew using the recursive MLE-EIT. The more relaxed the constraint when increasing the $K_{\max}$, higher the time error. Thus, a careful adjustment of $K_{\max}$ is an important aspect of synchronisation accuracy.

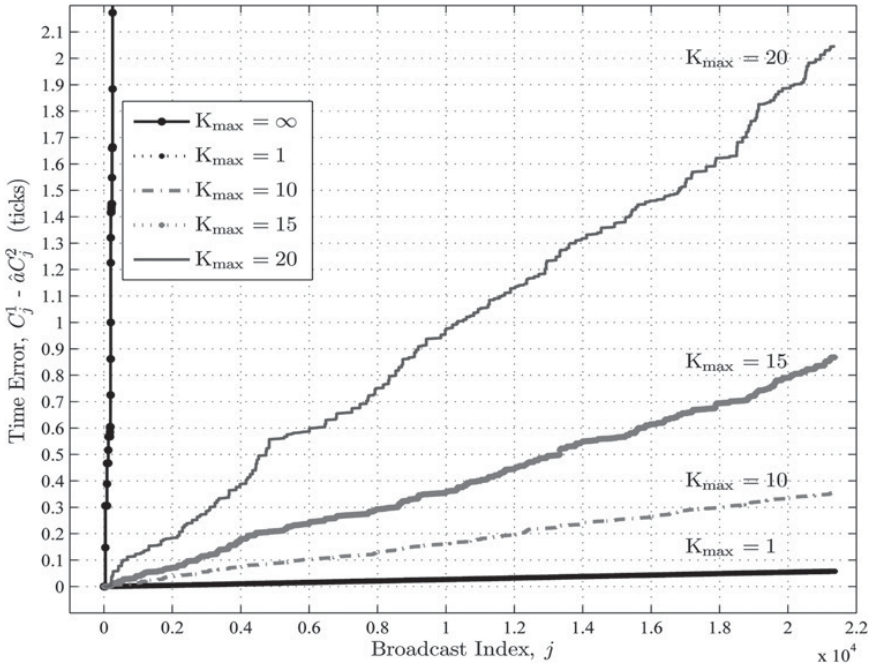


Fig. 3 *Time error variation for the skew estimate of recursive MLE-EIT with K_{max} in (52)*

IET Wirel. Sens. Syst., 2012, Vol. 2, Iss. 4, pp. 338–350
doi: 10.1049/iet-wss.2011.0137

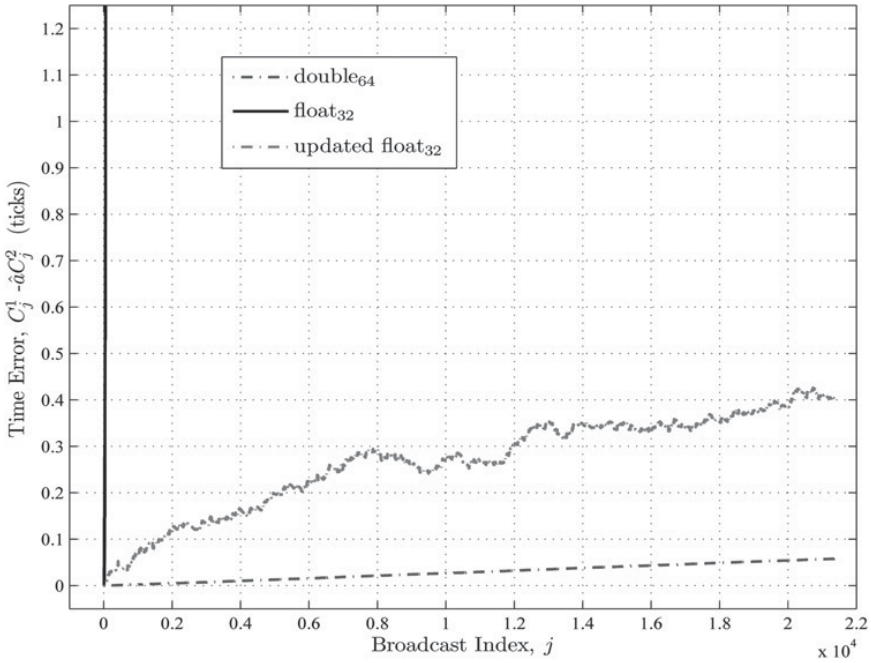


Fig. 4 *Time error variation for the skew estimate of recursive MLE-EIT with different numerical resolutions ($K_{max} = 10$ ticks)*

The numerical sensitivity of the recursive MLE-EIT is demonstrated in Fig. 4. The recursive MLE-EIT is implemented for 64-bit double precision and 32-bit single precision. The numerically stable equivalent estimator presented in (49) is implemented only for 32-bit single precision. The time error drastically increases as the implementation precision is changed from double precision to single precision owing to numerical sensitivity. However, the numerically stable equivalent has a time error close to the error for double precision implementation. Thus, the numerically stable estimator is more suitable for low-precision, embedded microprocessor implementations.

The proposed algorithm has a superior performance, while the requirements are in the order of regression method with the table length 1. However, the developed estimator is sensitive to time data inconsistency and numerical precision, for which the regression-based method is stable. The immunity of the LS regression-based method increases

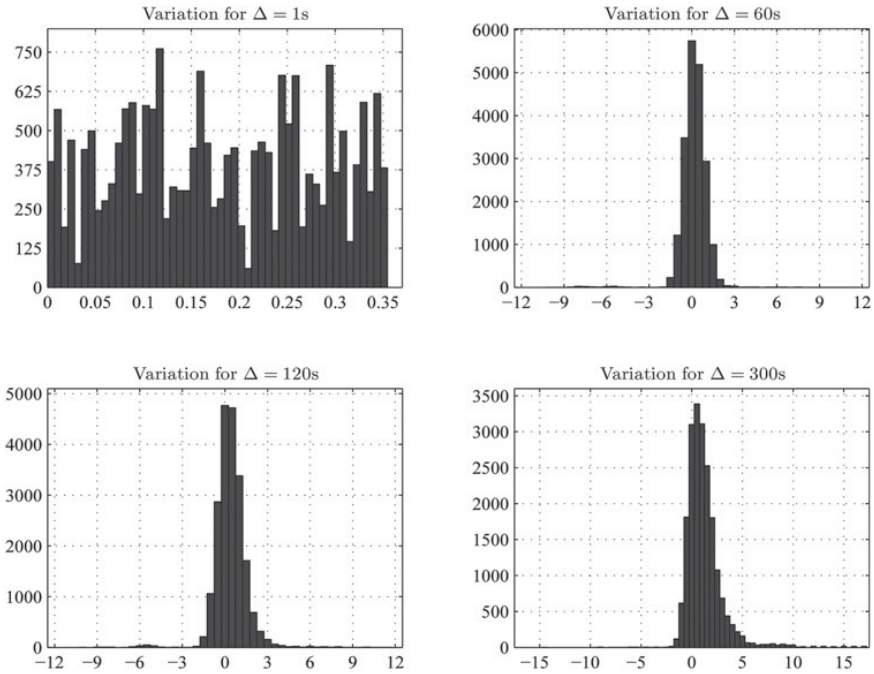


Fig. 5 *Time error distribution between successive reference broadcasts for different synchronisation periods ($K_{max} = 10$ ticks)*

Table 3 Statistical parameters of the results shown in Fig. 5

Δ	μ	σ	$\Pr\{\lvert m_i - \hat{a}k_i\rvert > 3\sigma\}$
60	0.176	0.961	1.12%
120	0.408	1.100	1.708%
300	1.640	3.954	1.960%

as its table length increases, at the cost of greater memory and computational requirements, a larger estimation error variance and a longer response time. The proposed numerically stable, equivalent recursive estimator relaxes the numerical sensitivity as shown in Fig. 4.

The time error distribution between successive beacons for different synchronisation periods is shown in Fig. 5. For this experiment, the time error is calculated at every 1 s for the skew estimate of the latest broadcast. The time error of the time record i after the jth broadcast is given by the following equation

$$\text{Time error}(j+i) = C^1_{j+i} - \hat{a}(j)C^2_{j+i}$$

where the time record index i is initialised to 0 after receiving a new broadcast, and $\hat{a}(j)$ is the skew estimate associated with the jth broadcast.

The quantitative statistical parameters of the time error distributions in Fig. 5 are given in Table 3, where Δ is the period in seconds, μ is the mean, σ is the standard deviation and the last column is the probability of having a time error larger than 3σ. These quantitative values can be used to determine the synchronisation period for the required level of accuracy.

9 Conclusions

In this paper, we studied a novel recursive clock skew estimation algorithm for reference broadcast-based time synchronisation; to the best of the authors' knowledge, this is the first study to make such an attempt. The realistic oscillator output model implies that the progressive time records of the clocks are correlated, which must be reflected by the time relation model. We demonstrated that if the time report correlation is not taken into account, the recursive clock skew estimate will have a large estimation error variance, resulting in poor synchronisation accuracy. We proposed two time relation models to investigate and mitigate the effects of time record correlation on the clock skew estimate. The time relation models are used for developing the recursive clock skew estimator. The numerical sensitivity and time data consistency issues of broadcast-based time synchronisation are addressed for the developed algorithm. We also demonstrated that the proposed solutions for these practical issues can be used when implementing the recursive clock skew estimator.

Broadcast-based synchronisation protocols such as reference-broadcast synchronisation (RBS) and flooding time synchronisation protocol (FTSP) can utilise the developed recursive skew estimator without needing to update the underlying messaging scheme. Such a modification would provide higher synchronisation accuracy, relaxed memory and computational requirements, and a better response time. Furthermore, the energy spent by the synchronisation algorithm can be substantially decreased by utilising two oscillators. A unified approach to synchronising clocks with two oscillators needs to be expanded upon in a separate study.

10 References

1 Serpedin, E., Chaudhari, Q.M.: 'Synchronization in wireless sensor networks: parameter estimation, performance benchmarks and protocols' (Cambridge University Press, 2009)
2 Allan, D.: 'Time and frequency (time-domain) characterization, estimation, and prediction of precision clocks and oscillators', *IEEE Trans. Ultrason. Ferroelectr. Freq. Control*, 1987, **34**, (6), pp. 647–654
3 Demir, A., Mehrotra, A., Roychowdhury, J.: 'Phase noise in oscillators: a unifying theory and numerical methods for characterization', *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, 2000, **47**, (5), pp. 655–674
4 Wu, Y.-C., Chaudhari, Q., Serpedin, E.: 'Clock synchronization of wireless sensor networks', *IEEE Signal Process. Mag.*, 2011, **28**, (1), pp. 124–138
5 Elson, J., Girod, L., Estrin, D.: 'Fine-grained network time synchronization using reference broadcasts', *ACM SIGOPS Oper. Syst. Rev.*, 2002, **36**, (SI), pp. 147–163
6 Sundararaman, B., Buy, U., Kshemkalyani, A.D.: 'Clock synchronization for wireless sensor networks: a survey', *Ad Hoc Netw.*, 2005, **3**, (3), pp. 281–323
7 Mills, D.L.: 'Computer network time synchronization: the network time protocol' (CRC Press, 2006)
8 Ganeriwal, S., Kumar, R., Srivastava, M.: 'Timing-sync protocol for sensor networks'. Proc. First Int. Conf. on Embedded Networked Sensor Systems. ACM, 2003, pp. 138–149
9 Noh, K., Chaudhari, Q., Serpedin, E., Suter, B.: 'Novel clock phase offset and skew estimation using two-way timing message exchanges for wireless sensor networks', *IEEE Trans. Commun.*, 2007, **55**, (4), pp. 766–777
10 Maróti, M., Kusy, B., Simon, G., Lédeczi, A.: 'The flooding time synchronization protocol'. Proc. Second Int. Conf. on Embedded Networked Sensor Systems, 2004, ser. SenSys'04, pp. 39–49
11 Pottie, G., Kaiser, W.: 'Wireless integrated network sensors', *Commun. ACM*, 2000, **43**, (5), pp. 51–58
12 Noh, K., Serpedin, E., Qaraqe, K.: 'A new approach for time synchronization in wireless sensor networks: pairwise broadcast synchronization', *IEEE Trans. Wirel. Commun.*, 2008, **7**, (9), pp. 3318–3322
13 Mahmood, A., Jantti, R.: 'Time synchronization accuracy in real-time wireless sensor networks'. IEEE Ninth Malaysia Int. Conf. on Communications (MICC), 2009, 15–17 December, pp. 652–657
14 Sorenson, H.W.: 'Parameter estimation: principles and problems' (M. Dekker, 1980)
15 Aoun, M., Schoofs, A., van der Stok, P.: 'Efficient time synchronization for wireless sensor networks in an industrial setting'. Proc. Sixth ACM Conf. on Embedded Network Sensor Systems, 2008, pp. 419–420
16 Schmid, T., Dutta, P., Srivastava, M.B.: 'High-resolution, low-power time synchronization an oxymoron no more'. Ninth ACM/IEEE Int. Conf. on Information Processing in Sensor Networks, 2010
17 Freris, N., Borkar, V., Kumar, P.: 'A model-based approach to clock synchronization'. IEEE Proc. 48th IEEE Conf. on Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conf. CDC/CCC 2009, 2009, pp. 5744–5749
18 Freris, N., Graham, S., Kumar, P.: 'Fundamental limits on synchronizing clocks over networks', *IEEE Trans. Autom. Control*, 2011, **56**, (6), pp. 1352–1364
19 Veitch, D., Babu, S., Pàsztor, A.: 'Robust synchronization of software clocks across the internet'. Proc. Fourth ACM SIGCOMM Conf. on Internet Measurement, 2004, pp. 219–232
20 Sari, I., Serpedin, E., Noh, K., Chaudhari, Q., Suter, B.: 'On the joint synchronization of clock offset and skew in RBS-protocol', *IEEE Trans. Commun.*, 2008, **56**, (5), pp. 700–703
21 Chaudhari, Q., Serpedin, E., Qaraqe, K.: 'Some improved and generalized estimation schemes for clock synchronization of listening nodes in wireless sensor networks', *IEEE Trans. Commun.*, 2010, **58**, (1), pp. 63–67
22 FreeRTOS, http://www.freertos.org/, FreeRTOS
23 Nanostack, http://www.sensinode.com, Sensinode

doi: 10.1049/iet-wss.2011.0137

Unified Information Privacy Preserving Model

Bahaa Eltahawy and Reino Virrankoski
University of Vaasa, Department of Computer Science, Communications and Systems Engineering Group
P.O.Box 700, FI-65101, Vaasa, Finland
{bahaa.eltahawy, reino.virrankoski}@uva.fi

***Abstract* – Incomplete security solutions are causing a remarkable shortage that prevents the industrial Internet and different kinds of data service applications to become more common. Privacy is a vital asset to individuals; with it only they can possess control over their native information and data. The current dependence on the Internet, data networks, and cellular communication created the need for a robust security and privacy standardization, that can keep systems safe, well-secured and adopt the users right to the privacy. In this paper, we disassemble the problem to its main components by providing the different parameters required for privacy preservation. We also point out the main shortages in the current deployments, and provide clarification for the different terms that can precisely describe the privacy issues. As a main outcome of this work, we introduce a unified model for information privacy preservation. Additionally we provide a set of technical recommendations regarding the way how the model should be implemented.**

Keywords—privacy; safety; security; industrial internet; digitalization; communication parties; conflict of interests and costs

I. INTRODUCTION

The rapid development of computation and communication systems has opened completely new opportunities for industry, commerce and private customers. Many services can be offered to users over different network devices such as table computers or smartphones, and the combined use of local area networks and the Internet has enabled the concept of industrial Internet. Advanced communication plays an important role also in many other technology concepts, such as Smart Grid in electricity distribution and production. Once the communication systems have become more diverse and complicated, the amount and types of security risks have also increased.

Safety, security and privacy measures are a need to protect users and their assets. Telecommunications and data technologies can play a double role here, as they can provide a means to protect users, or in contrast cause harm by revealing users' information, either unintentionally or maliciously, also with or without users' awareness and permission.

The current architecture of Internet and internetworks allows data to traverse through many networks in the way to its destination. These networks can be different in terms of trustworthiness and acceptable security measures. The existing architecture provides many benefits in connectivity and reachability, but causes also many threats in data protection and data control. As a consequence, advanced measures for privacy are needed to protect end users' data. Unfortunately, the new privacy scope faces many conflicts and un-clarities, due to regulations, as well as technical limitations. That is because, different parties participating in the communication process have different interests, demands, procedures and standards. If the end-users are private customers, they are typically the least involved in the system development, and they are mostly enforced to follow certain policies with almost no options available. There is also a risk that improvements in privacy may increase system complexity and deployment costs.

In this paper, we disassemble the problem to its main components by providing the different parameters required for privacy preservation. We also point out the main shortages in the current deployments, and provide clarification for the different terms that can precisely describe the privacy issues. As a main outcome of this work, we introduce a unified model for information privacy preservation. Additionally we provide a set of technical recommendations regarding the way how the model should be implemented.

The rest of this paper is organized as follows. Safety is discussed in Section II and the different areas of security in Section III. Section IV presents the security architecture, and privacy is discussed in Section V. We present the new unified privacy preserving model in Section VI and make final conclusions in Section VII.

II. SAFETY

Safety [1] is the practice to keep systems safe, which means freedom of harm, specifically the unintentional one. Systems can be vulnerable due to many reasons. The unintentionality plays a role with incidents such as fire accidents, loss of vital assets and documents, outage incidents, and others. Safety is the practice to take the right actions to avoid such situations.

Safety as itself can be divided into physical and data safety. Physical safety concerns assets in their physical form, for example machinery, buildings, files, and data storages. Safety considerations should be taken in consideration in the early levels of the to the system design before the implementation. The purpose of system usage and the right actions and plans must be followed to keep these systems in their original form. For example, cooling systems and fire suppression must be installed prior servers installation, access points must be locked with the right enclosures, etc. Data safety concerns those actions that should be taken to keep users' data safe, which means that the data must remain maintainable, achievable, available, and intact. Data safety protects against data loss and

data leakage. It can be achieved by implementing Data Loss Prevention (DLP) technologies, that include operations like control over data, encryption, supervision, and filtering [2].

To this extent we can summarize safety as the core of the protection process, and it is a matter of the design phase. Proper design and implementation can bring the system to the desired safety level. However, as a part of the solutions for safety, security and privacy must also be considered.

III. SECURITY

Security is the freedom from danger or threat [3]. This definition is a broad one, and to be specific, security depends solely on the system upon description. Security is divided into several categories, including personal, operations, communications, network, and information security [4]. Here we concern information security by the meaning of security. According to Straub et al., "Information security protects the availability, integrity, confidentiality, and authenticity of information and underpins such societal goods as privacy, the protection of digital identity, and the protection of intellectual property. This is performed by deploying a dynamic system of measures taken to protect data, information, and systems from unauthorized use or a disruption due to a human agency or a natural threat" [5]. The classical security model (CIA model) comprised of confidentiality, integrity, and availability measures [6]. The later Parkarian Hexad model included utility, possession, and authentication [7]. In the current applications and services, also non-repudiation is a vital parameter to consider due to e-commerce and e-banking operations [8].

A. Utility

Utility refers to the usefulness, and worthiness of the exchanged data [9]. Utility depends on the deployed applications and the data formats applications utilize. For this reason, utility is not a concern for operators, because it is an abstract concept and a property of applications. Utility can be seen as a lack of standardization between peers. It can affect the bandwidth and costs if not well addressed.

B. Availability

Availability is the readiness and reliability of resources to be accessible upon request [10]. Different resources have to be available and accessible to establish a session, e.g. IDs, addresses, locations, databases, privileges, etc. Availability plays the comprehensive role to the security model since other operations are meaningless if resources are not available. As a part of the availability, system must be able to handle incidents of unexpected failures, by considering link redundancy and aggregation, and redundant standby infrastructure solutions. The second main issue is the intentional attacks targeting resources. Those can be repulsed by isolating the sources of the attacks, and by hiding communication by means of anonymity.

C. Integrity

Integrity concerns the delivery of data in its original form. Information integrity refers to freedom, trustworthiness and dependability of information [11]. In other words, it is the consistency and the assurance of data against any sort of modification or alternation [12]. In integrity protection, the applied mechanisms mainly depend on the authentication procedures, varying from simple checksums to sophisticated cryptographic algorithms as hashing methods. For robust security, integrity is imperative to both data and control frames. Separation between authentication and integrity mechanisms would significantly enhance the security practices. Another practice is the separation between domains, i.e. network access, network security, user domain, and application security [12].

D. Possession

Data possession or control protection concerns protecting and controlling data in the communication devices [13]. Regardless of the communication session, devices typically store valuable data. Once a device is lost, significant amount of user's data can be compromised. Possession shall be completely independent of the network and its security standards. Consequently, passwords, tokens, biometric IDs or other methods are mandatory to provide strict access, as well as data encryption, and automatic data erase for predefined cases. Another solution is to store devices' data off premises; i.e. use services as cloud storage rather than storing data locally.

E. Confidentiality

Confidentiality directly concerns the content and the access to user's data. It is related to users' privacy, which will be discussed later. According to IEEE standards, confidentiality is the property of information that is not made available or disclosed to unauthorized individuals, entities, or processes [14]. Confidentiality is protected by means of cryptography, and it is mostly deployed in the access layer since the core network is assumed to be trustworthy [15]. The main difficulty the confidentiality mechanisms face is the issue of compatibility with the older systems. This prevents hybrid networks and devices from taking advantage of the advanced security mechanisms, thus leaving a hole for attackers to exploit system weaknesses.

F. Authentication

Authentication, authorization, and accounting (AAA) concepts provide means to identify users and to approve their permissible activities. Authentication validates the user's identity, authorization examines the user's privileges, services and resource permissions, and accounting keeps tracking of the user's activity for further considerations and security countermeasures [10] [16]. The authentication is performed along with the integrity and the ciphering procedures, because integrity provides a means to ensure the consistency of the authentication procedure. The integrity must be protected, while ciphering protects the exchange of the authentication data [17]. Unfortunately, most authentication mechanisms exchange unprotected messages first during the authentication and key management (AKM) phase. These messages can include usernames, security association parameters and other parameters, which can create a gate for intruders to access the system. Fortunately, public key infrastructure (PKI) applications can overcome these situations, but they still suffer from the lack of sufficient standardization.

G. Non-repudiation

Non-repudiation is a crucial concept for businesses and legalization. According to ISO standard, repudiation is the denial by one of the entities involved in a communication of having participated in all or part of the communication [18]. Non-repudiation is required to prevent an entity from denying a communication activity, to prevent abandoning responsibility and accountability of own actions by means of verification. This can be performed by digital signatures with PKI systems.

H. Existing Shortages

Still, the main shortages in the current communication systems are the lack of standardizations of the pre-defined parameters, compatibility with weaker legacy systems, AKM exchanges of vital data in clear, and the fact that the security structure is often independent of end-users and their demands.

IV. Secrutiy Domains

For implementation, the 3GPP has divided the security architecture into security domains [19]. We have adopted this architecture, and also introduced two extra domains, *device domain security* and *data domain secu*rity. This architecture can work as a security basis over all IP-based networks.

1) *Network Access Security:* Features between the user's device and the first node to the network. This is a relay to handle security functions, e.g. authentication.

2) *Network domain security:* Features between the different network nodes, e.g. tunneling, and secure routing protocols by means of cryptography.

3) *User domain security:* A means to validate the user to the device upon usage.

4) ***Device Domain Security****:* This extends the User domain security to include facilities for remote access, tracking and locating, erasing, accounts and permissions, functionality control, and all other policies concernig devices themselves.

5) *Application domain security:* Secure data exchange between applications.

6) ***Data Domain Security****:* This is the set of policies that ensure data protection within the device, e.g. using AntiVirus and AntiMalware, patching and updates.

7) *Visibility and configurability:* Information about the applied security level, encrypiton, network capabilities and services. It allows the user to participate to the security process so that he can choose his parameters, and accept or reject a service or a certain access.

V. Privacy

It is well-known that even if the highest safety and security standards are applied, some information can still be leaked to outsiders. For example time, location, session activity, and many indicators to the communicating parties as well. This can enable the outsiders to extract valuable information using a method known as connect the dots. In a worse scenario, interception can be done internally, i.e. by a trusted service. This later exact case urges the need to answer the following questions: Who is allowed to access an entity's data? And under which acceptable circumstances? This goes beyond the safety and security concepts discussed so far. To answer these questions, we go to the more generalized frame, privacy.

A. Privacy, Definitions and Theories

Research interests related to privacy have been growing rapidly since the beginning of computerization. Main reasons behind this development are the new demands and concerns in privacy caused by the digitalization, and the moral issues regarding technology and sciences. Privacy protects users' private data from being disclosed, or connected to draw a figure about their activities and their personalities [20]. Still, this is not absolute. There are some lines where privacy needs to be revealed under some criteria and by the right entities.

Privacy can be explained according to two theories: privacy control theory, and restricted access theory [21]. The privacy control theory proposes that privacy can be preserved if a person has control over his information and the way how it is spread. On the contrary, the restricted access theory relies on that the privacy can be preserved by restricting what others can access, based on secrecy, anonymity and solitude. Both theories were contradicted by the privacy control/restricted access theory [22]. It stated that controlling information in the cyberspace is unfeasible. However, according to privacy control / restricted access theory, it is a must that the right entity at the right time can access the information. This combines the advantages of both previous theories by stating that an entity can control information and restrict others from accessing it, while it is still accessible by the right entities whenever needed under the right conditions. Moreover, such a concept of privacy-policies was introduced, where they are flexible to be set according to the situation. The above mentioned theories summarize the privacy definition as: privacy is a right for individuals, as they hold the right to control their own information and the right to restrict others from accessing it, as long as no harm can be caused to others with this information.

B. Elements of Privacy

To preserve privacy, five elements need to be protected: data and traffic, identities, locations and mobility, time, and existence [23].

Traffic and exchanged data between entities should be protected against others. This can be achieved by using the cryptographic functions of the security procedures. Identity is how a person defines oneself to the world, describing his individuality, sort, and relation among others. An identity is used to relate a user to his own activities, interests and privileges. Therefore, users' identities must be protected. This can be practically unfeasible since identities are used for session establishment. However, the use of temporal independent identities and a level of randomization can be a feasible solution for that. This generally is the concept of anonymity.

Related to privacy, location and mobility is a crucial concern. Many systems, services, and applications keep tracking records about users, to be used to provide services on demand. However, there is no enough transparency about this process and data usage. Location records can be used to draw a general picture about a person's behavior, and disclosure of such information is a serious threat. Location Based Services (LBS) and other applications can use this data maliciously if they are not well trusted. Location privacy can be achieved by implementing blurring and obfuscation techniques, or by applying K anonymity servers and mix zones. These techniques will partly hide the exact locations of the users.

Time can be used with the other elements to precisely identify users' activities. Consequently, time privacy is required to protect against disclosure of activity times. Times of transactions can be hidden by randomly sending junk data at random instances to cause a sort of illusion about the exact times of events, though it will increase the amount of traffic.

The last factor to consider is the existence privacy, which protects entities by hiding them from surveillance. This task is not easy due to the nature of communication and its limitations. A strict control of node visibility and the use of pseudonyms can still provide an adequate level of existence privacy.

C. Privacy Relations

Privacy as a set of relations comprises anonymity, unlinkability, unobservability and undetectability [24]. These relations maintain privacy by hiding users' identities, as well as any indication, relation or connection about their activities, to protect them from information leakage.

Briefly, pseudonymity is the use of traceable anonymous identities rather than the real ones. Anonymity is the ability to combine an anonymous identity with a recognizable one in such a way that the identity cannot be identified from a set of identities. Anonymity can be achieved by the continuous use of pseudonyms or by providing less information than the amount what is needed for identification. Unlinkability is the inability to detect a link or relation between two identities or between two activities. Undetectability is the inability to detect the existence of an identity or its participation to certain activity. Finally, unobservability is the inability to observe a user and its activities. This rather implies undetectability and anonymity. Anonymity itself comprises sender and receiver anonymity to protect the communicating peers, as well as relationship anonymity to rather hide any information about the communication and the involved active users, i.e. unlinkability. These elements and their relations can be written as follows:

Unobservability → *Undetectability*
Unobservability → *Anonymity* → *Pseudonymity*
Sender/Receiver Anonymity → *Relationship Anonymity.*
Sender/Receiver Unobservability → *Relationship Anonymity.*

It is clear that the maintaining of unobservability is the sufficient condition to preserve privacy. The drawback is that it requires systems with high complexity. As a consequence, it is difficult to achieve unobservability in practice. A better easy to implement mechanism is to provide anonymity, in addition to spread dummy meaningless traffic to enable undetectability. Still, deploying these elements collectively is more conceptual than applicable, and in any way absolute privacy can be achieved in reality only up to certain levels [25]. If privacy levels span on a scale from 1 to 6 as presented in Table 1, then level 5 indicating beyond suspicion would be the best achievable deployment in reality. In contrast, exposed or provably exposed are the worst cases, which are common in web and in some of the VoIP applications.

Table 1. Privacy levels span on a scale from 1 to 6 [25].

6	5	4	3	2	1
Absolute Privacy	Beyond Suspicion	Probable Innocence	Possible Innocence	Exposed	Provably Exposed

D. Parties, Rights and Responsibilities, Conflict and Costs

Within a communication session, four parties with different rights and responsibilities are included [26]. The first and second parties are individuals or entities establishing the communication session. They are users who use the provided services and typically they do not hold control on any of the communication factors. The third party is the one managing the communication environment. This party can be the operator network, a monitoring organization, or the governmental authorities and policy officials. The fourth party consists of all other entities, which do not participate in the communication session and do not get any information about it by default. This can include the public, or in a worst case a malicious attacker.

Parties have different rights and responsibilities, which is the main reason behind the conflict of interests. The first party has the right to privacy, to hold control over their own information, and to restrict others access. This party has to take into account the spreading of their own information, and the responsibility associated with it. The first party also holds the right to acquire the level of privacy that suits for them, according to the benefits they gain or the threats they might face [27]. However, as a part of the society, the first party holds the responsibility not to cause harm with the rights they hold, because the privacy can be broken by any suspicious harm or danger. The second party is a replicate of the first one, with the same rights as they are sharing a communication relation. By default, the first party trusts the second party, which gives the second party responsibility against spreading information.

The third party (the one managing the communication environment) has an important role in privacy since it has access to the resources of the communication facility. Third party has the responsibility to protect the communication between communicating parties, also to protect other parties' data, including the stored personal information. Similarly, third party holds a right to protect the society from any sort of danger that other parties might cause. The fourth party on contrast to the previous ones has the least responsibilities and rights, as they have the right to access the authorized data only.

Conflicts do not arise only because of the differences between these four parties associated with the communication session. There are also other conflicts regarding the meaning of

privacy itself, including [28]:

1) Cultural conflicts: The culture and its understanding and acceptance for privacy values, for individuals and society.

2) Organizational conflicts: Different places have different perspectives regarding privacy.

3) *Individual conflicts:* No common view about privacy.

4) *Structural conflicts:* The structure of privacy systems might conflict with safety, security and other modules.

5) Communication conflicts: Tracking, mobility and other services will face difficulties as well.

6) Price conflicts: Increased costs due to complexity.

7) *Efficiency and quality conflicts:* Data collection and extraction for evaluation processes conflicts with privacy.

8) Operational conflicts: Limitation of shared resources and information affect operations and increase difficulties.

9) *Standardization conflicts:* Different standards exist, but they do not currently cover the area of privacy that well. New standard development is a time consuming and difficult task.

10) *Expansion conflicts:* The standards should be forced across all networks, which might affect expansion plans.

In addition to these conflicts, it is necessary to consider crime fighting and privacy conflicts. Privacy, as all rights, can also be misused. That is because of the protection it provides against information collection, which in turn can encourage illegal actions and behavior. This later increases the criminal activities and certainly causes harm to the society [21]. It is a must while drawing the general lines for privacy to consider such cases, to allow lawful interception under the legal form.

VI. Unified Privacy Preserving Model

Based on the presented discussion and analysis, we ended up formulating the Unified Privacy Preserving Model. The semantic architecture of the model is shown in Figure 1. In the proposed model, we could combine the essential components that are required to preserve privacy in the communication systems. The given model comprises both the theoretical and the practical perspectives. The theoretical perspective defines the main elements that should be considered to preserve privacy, and the practical perspective provides recommendations on how to implement these elements.

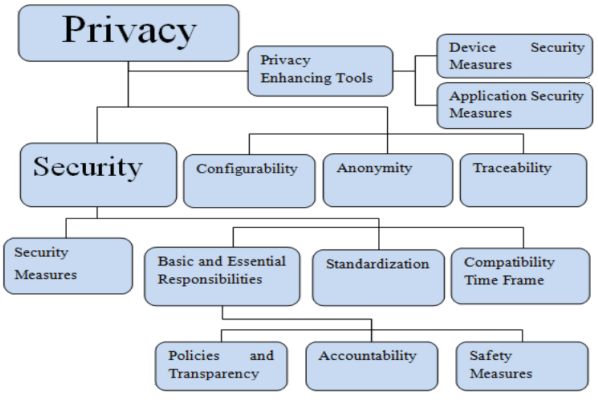


Figure 1. The proposed privacy preserving model.

A. Privacy from the Theoretical Perspective

The main elements that should be considered to preserve privacy are the following ones:

1) The transparency of the policies: Clear policies with well-defined procedures, functions, and actions are required. Policies have to clarify the disclosure situations, purposes and responsibilities upon disclosure. Also, the changes of the applied policies should be acknowledged and agreed upon applicaiton.

2) *Accountability:* Responsibility for own actions and behavior.

3) *Safety measures*: Operators' responsibility to maintain safety, which includes physical and data safety practices.

4) *Standardization:* Security measures and standards should be agreed. Applied mechanisms cannot be left as an implementation option for operators.

5) *Compatibility:* Upgrade time frame for legacy systems, applications, and protocols should be clearly declared.

6) *Security measures:* Considerations for actions targeting users, systems and data, also applying the main security measures discussed previously, will ensure secure communication. Authorities and governments on the other hand also hold a responsibility against the harm that might be caused to users or by users, thus to take the legal considerations.

7) *Configurability:* Ability to hold control over own data and security functions should be revised. Flexibility to configure and control the different privacy parameters according to users' preferences should be granted.

8) Anonymity: To protect users and to hide their activities.

9) *Traceability:* Unlike anonymity, linkage is allowed but only controlled by authorities under the legal form. This specefically restricts privacy so that society can provide its protection.

10) *Application Security measures:* Restricting applications from gathering, storing, or exchanging data about users without declaration and usage transparency. Also, anonymity should be applied upon collecting data.

11) *Device Security:* Mechanisms to restrict access to devices and stored data. Also rules to remotely control devices under the legal form.

B. Privacy Practical Perspective Recommendations

From the practical perspective, we provide here a set of general recommendations to consider in the implementation of the hypothetical elements:

1) Upgrading the non-standard algorithms and functions to the latest stable standard.

2) Time frame to upgrade legacy components. Also hybrid network compatibility capabilities should be revised.

3) Access network and AKM mechanisms are the weakest ones of the security phase, thus they need further consideration.

4) Digital Signatures are required to provide data origin authentication and non-repudiation guarantees.

5) PKI implementation to protect confidentialtity, and to provide lawful interception only for legal authorites.

6) Multi-homing solution is an acceptable to provide for anonymity and communication reliability.

7) IPv6 upgrade is required since it affords the means for multi-homing and also security services, that do not exist in IPv4.

8) Host Identity Protocol (HIP) implementation to provide anonymity, multi-homing, and also interception for legal authoroties.

9) IPSec,TLS/SSL, and/or other security protocols should be implemented to provide means for tunneling and application security.

10) Other security protocols should be considered, as Secure Real-Time Transport Protocol (SRTP), SRTP Control Protocol (SRTCP), Secure Distributed Anonymous Routing Protocol (SDAR), and Onion Routing mechnisms.

11) Application Security level is required to protect users from attacks targeting data within devices.

VII. CONCLUSTION

In this paper, we gave a discussion about the issue of privacy, its related measures, and the difficulties that are faced when trying to achieve privacy in the current communication systems. Based on the discussion we provide a general privacy preserving model and a set of technical recommendations for its implementation. These results can be utilized when thinking the privacy solutions for the rapidly growing number of applications for industrial internet and private customers. "It is a necessity to protect individuals and societal right to privacy, yet a complete preserving system does not and cannot exist, because privacy is never absolute".

REFERENCES

[1] "safety." Dictionary.com Unabridged. Random House, Inc. 1 Oct. 2015. <Dictionary.com http://dictionary.reference.com/browse/safety>.

[2] Wang Xingkui; Peng Xinguang, "Research on data leak protection technology based on TPM," in *Mechatronic Sciences, Electric Engineering and Computer (MEC), Proceedings 2013 International Conference on* , vol., no., pp.2354-2358, 20-22 Dec. 2013

[3] security. Oxford Dictionaries. Oxford University Press, n.d. Web. 4 July 2013. < http://www.oxforddictionaries.com/definition/english/security>.

[4] Whitman, Michael E. & Herbert J. Mattord (2010). *Principles of information security*. Boston: Cengage Learning.

[5] Straub, Detmar W., Seymour E. Goodman & Richard Baskerville (2008). Information security: policy, processes, and practices. New York: M.E. Sharpe.

[6] Saltzer, Jerome H. & Michael D. Schroeder (1975). The protection of information in computer systems. Proceedings of the IEEE, 1975, 63.9: 1278-1308.

[7] Parker, Donn B. (1998). Fighting computer crime: A new framework for protecting information. New York: John Wiley & Sons, Inc.

[8] ISO (1989). ISO 7498-2, Information processing systems-Open Systems Interconnection – Basic Reference Model – Part 2: Security Architecture. ISO. Geneva, Switzerland.

[9] Andress, Jason (2011). The basics of information security: understanding the fundamentals of InfoSec in theory and practice. Massachusetts: Elsevier.

[10] Stamp, Mark (2006). Information Security Principles And Practice. Hoboken, New Jersey: John Wiley & Sons.

[11] Geisler, Eliezer, Paul Prabhaker & Madhavan Nayar (2003). Information integrity: an emerging field and the state of knowledge. In: Management of Engineering and Technology, 2003. PICMET'03. Technology Management for Reshaping the World. Portland International Conference on. IEEE, 2003. 217-221.

[12] Lei, Wu & Song Xiao Ting (2009). Information integrity and its protection in networks. In: 2009 5th Asia-Pacific Conference on Environmental Electromagnetics. 2009. 238- 241.

[13] Ateniese, Giuseppe, Roberto Di Pietro, Luigi V. Mancini & Gene Tsudik (2008). Scalable and efficient provable data possession. In: Proceedings of the 4th international conference on Security and privacy in communication networks. ACM, 2008.

[14] IEEE (1998). IEEE standards for local and metropolitan area networks: standard for interoperable LAN/MAN security (SILS) specification; IEEE standard 802.10. IEEE Standard Press.

[15] Horn, Günther, Klaus Muellerand & Bart Vinck (1999). Towards a UMTS security architecture. ITG FACHBERICHT ,1999: 495-500.

[16] Convery, Sean (2007). Network Authentication, Authorization, and Accounting: Part One. The Internet Protocol Journal [online] 10:1 [cited 2 Oct. 2013] Available from Internet: <URL: http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_10-1/101_aaa-part1.html>.

[17] Tanenbaum, Andrew S. & Maarten Van Steen (2007). Distributed systems principles and paradigms. Ed 2. New Jersey: Prentice Hall.

[18] ISO (2009). ISO/IEC 13888-1: Information Technology Security Techniques-Non repudiation-Part 1: General. Available from World Wide Web: <URL: https://www.iso.org/obp/ui/#iso:std:iso-iec:13888:-1:ed-3:v1:en>.

[19] 3GPP. Security Architecture (3GPP TS version 33.102 Release 4). Available from World Wide Web: <URL: http://www.etsi.org/deliver/etsi_ts/133100_133199/133102/04.01.00_60/ts_133102v040100p.pdf>.

[20] Horniak, Virginia (2004). Privacy Of Communication-Ethics And Technology. Master Thesis, Mälardalen University, 2004. Available from World Wide Web: <URL: http://www.idt.mdh.se/utbildning/exjobb/files/TR0390.pdf>.

[21] Spinello, Richard (2006). Cyberethics: Morality and law in cyberspace. Massachusetts: Jones & Bartlett Learning.

[22] Moor, James H. (1997). Towards a Theory of Privacy I1', in the Information Age. Computers and Society, 1997, 27.3: 27-32.

[23] Candolin, Catharina (2005). Securing military decision making in a network-centric environment. Available on World Wide Web: <URL: http://lib.tkk.fi/Diss/2005/isbn9512279819/isbn9512279819.pdf>.

[24] Pfitzmann, Andreas & Marit Hansen (2005). Anonymity, unlinkability, unobservability, pseudonymity, and identity management-a consolidated proposal for terminology. Available from World Wide Web: <URL: http://dud.inf.tu-dresden.de/literatur/Anon_Terminology_v0.28.pdf>.

[25] Chao, Gao (2009). Study on Privacy Protection and Anonymous Communication in Peerto-Peer Networks. In: Multimedia Information Networking and Security, 2009. MINES'09. International Conference on. IEEE, 2009. 522-525.

[26] Mason, Richard O. (2000). A tapestry of privacy, A meta-discussion.

[27] Graham, Ian. "Putting Privacy in Context--An overview of the Concept of Privacy and of Current Technologies." Retrieved June 13 (1999): 2013.

[28] Noam, Eli M. (1995). Privacy in Telecommunications: Markets, Rights, and Regulations. Part I. New Telecom Quarterly, 3.2: 52-59.

GENSEN: A Novel Combination of Product, Application and Technology Platform Development in the Context of Wireless Automation

Reino Virrankoski
University of Vaasa, Department of Computer Science,
Telecommunication Engineering Group
P.O. Box 700, FI-65101 Vaasa, Finland
Tel. +358-6-324-8694, Fax. +358-6-324-8467
reino.virrankoski@uwasa.fi

Simo Keskinen
University of Vaasa, Department of Production,
P.O. Box 700, FI-65101 Vaasa, Finland
Tel. +358-6-324-8482, Fax. +358-6-324-8467
simo.keskinen@uwasa.fi

Keywords: multi use platform, wireless automation, wireless sensor networks

Abstract

During the last decade, several wireless sensor platforms capable to use commercial sensors have been developed for wireless monitoring and automation. However, the field of wireless automation applications is still immature, and there are no complete generic solutions widely available. Many applied wireless solutions have been developed just for cable replacement in simple star topology setups. They measure and transmit data without any processing or any system adaptivity. These kind of “dummy” wireless sensor networks usually run into troubles with limited transmission capacity and limited energy resources if they are applied to wireless automation, where the application might also pose such requirements for the sensor networking platform that time-consuming application-specific tailoring and configurations would be needed. Many of the developed network architectures have been small with just a couple of nodes manually placed in a single-hop star topology. They are unfeasible for big automation systems, because they neither scale up with the number of nodes nor adapt to the changes in the network topology. Compatibility with the rest of the automation system is also forgotten such that many of the wireless automation application prototypes have became stand alone systems instead of integrating to the rest of the automation system. Usually the most expensive and time-consuming phase in the product development is the process to develop a platform up to such a level that allows a fast production of different applications on top of the platform. In this paper we describe a product-making process in the context of research and development project Generic Sensor Systems for Wireless Automation (GENSEN). In this project a combination of product, application and technology platform development is applied to create an application platform that enables a fast production of different kinds of wireless automation applications. Our solution will fill the existing gaps between the current wireless sensor networking platforms and the various needs of different applications of wireless automation. The developed platform enables automatic network configuration and data processing in the network. It supports complex, multi-hop network structures and scales up to hundreds of sensor nodes.

Introduction

Wireless sensor systems enabled by wireless communication provide new opportunities for industrial automation. Since wireless sensor systems are remarkably cheaper than the cabled ones, a bigger number of measurement points can be applied. This allows us to collect more precise and more redundant data from the system, which enables advanced control system development. Moreover, wireless connection provides us access to such places that are difficult or impossible to cable, such as rotating machine parts, several measurement points in one motor, high-voltage locations in an electric power transmission system and locations in the middle of greenhouse flora.

There are several challenges to be solved to make the wireless sensor systems fill the automation requirements. The data transmission rate must be high enough to fill the real time operation requirement. The energy supply to the wireless sensor nodes must be organized in such a way that it does not require continuous human intervention. Since the data transmitted over the wireless channel is typically incomplete and can also include misleading information, automatic fault diagnostics must be applied to be able to eliminate measurement outliers and to complete missing information. In time-critical applications one must also be able to handle the time-varying transmission delays (Eriksson 2008).

If we have such an application that includes tens or even hundreds of measurement points, like a sensor node deployment in a modern greenhouse (Ahonen 2008), automatic network initialization is required. Automatic self-healing from the possible network malfunctioning situations is also needed to make the system reliable. Tens or hundreds of measurement points can produce a huge amount of data. Since energy consumption and data transmission rate are both remarkable bottlenecks in the practical feasibility of sensor networks, and since each sensor node is equipped with microprocessor and some memory, some data processing should be performed locally. Instead of using the network just to collect the data, one should compress the data in the network such that only useful part of the information will be transmitted all the way from the local area network to the centralized network control. It is also possible to perform event-based networking such that the nodes are in the low-power operating mode most of the time and switch on their radios just if they have something important to transmit. In addition to single sensor data compression, one must also perform data fusion in such systems, which use information measured by several different types of sensors. This kind of data fusion is typically performed either in a centralized knowledge after collecting all the data from the network or in a locally centralized manner in network cluster heads if a hierarchical network architecture is applied.

Typically a wireless sensor network, which operates under IEEE 802.15.4 communication protocol, forms just a part of the communication architecture applied in automation system. Some other parts can be based on IEEE 802.11 wireless communication, or they can be cabled. Thus, one design challenge is to make the wireless sensor network compatible with the rest of the system. It is, for example, known that IEEE 802.11-based WLAN network can be an efficient jammer to IEEE 802.15.4-based network. Thus, in the research point of view, we must pay attention to interference minimization and radio resource management. Even without an existence of other wireless communication network, these issues become important in harsh environments such as electric grid substations or spaces that include a lot of metal.

There are a huge amount of industrial sensors as well as several sensor networking platforms consisting of wireless sensor nodes (sometimes called sensor motes) commercially available (Crossbow 2009, Polastre 2005, Lymberopoulos 2005, Imote2 2009, Sensinode 2009). However,

one is still missing such a generic wireless sensor network concept, which enables a rapid production of different kinds of automation applications. Quite often the applications need to be built from the scratch and tailoring the nodes, protocols and other software takes a lot of time and effort. We are targeting to fill this lack in GENSEN-project, and produce a generic sensor network concept for wireless automation. We will apply such sensor platforms produced by Telos (Polastre 2005) and Sensinode (Sensinode 2009), which support Contiki operating system (Contiki 2009). We will also consider Sensinode Microseries and Nanoseries platforms up to such versions of Nanoseries, which still offer an open source code. To guarantee the compatibility with the other parts of the automation system, our solution supports IP-based networking with IPv4 and IPv6. This is accomplished by using a 6LoWPAN compliant protocol stack (either NanoStack or Contiki µIPv6) (6LoWPAN 2009). In some applications we may also develop our own hardware which is fully compatible with the pre-mentioned ones. A sensor development is out of the focus of this project because we are applying commercially available sensors.

The core part of the technology platform development is the creation of novel algorithms for networking and data processing and their implementation to the protocol stack. Related to the product platform, special attention will be paid to system reliability and overall system usability including user interfaces. Also product covering to create different system parts an appearance suitable for marketing will be considered. Application platform is developed through five different applications: trolley crane automation, wind turbine generator monitoring and control, cattlehouse automation, greenhouse automation and distributed energy production system monitoring and control. In the case of each application, customer needs, application specific technical requirements, application environment and system integration to other subsystems will be evaluated by using various technical and customer satisfaction criteria. Quality Function Deployment (QFD) can be used as a one tool in this process.

After going over the basics of the platform development and related work, we describe the way how product development is done by using a combination of product, technology and application platforms in the context of GENSEN-project. The project is funded by Finnish Agency for Technology and Innovation (Tekes) and by the participating companies. Participating research units come from the University of Vaasa, Helsinki University of Technology and Seinäjoki University of Applied Sciences. Finally we give a discussion about the benefits achieved through combined platform development process and point some directions to future work.

Related Work

Platforms

Typically automation system consists of operatively independent units called modules. Hardware, software and mechanical architecture must all be taken into account in the modular structure. Also production structures are based on modular (or platform) architecture in which the modules can also be called platforms. In modular production architecture one can freely combine different types of platforms. This property can be exploited in several production sectors where the physical compatibility is the only requirement.

The situation is much more challenging in intelligent embedded systems, such as wireless sensor networks. The associated challenges can also be seen as an opportunity, if one can systematically foresee the current and close future requirements and direct his product development accordingly. Several requirements must be filled to make the modules compatible

with certain generation products and to enable transmission from current to next generation. These requirements can be fulfilled by applying platform structuring and platform-based planning.

Previously a term platform was used in the context of technology to call physical construction elements such as bridges, skeletons, metal plates etc. On 1990s the term became common also in electronics. At the beginning it was used to characterize software architectures and later also other application areas of electronics. Software architecture platform is illustrated in Figure 1.

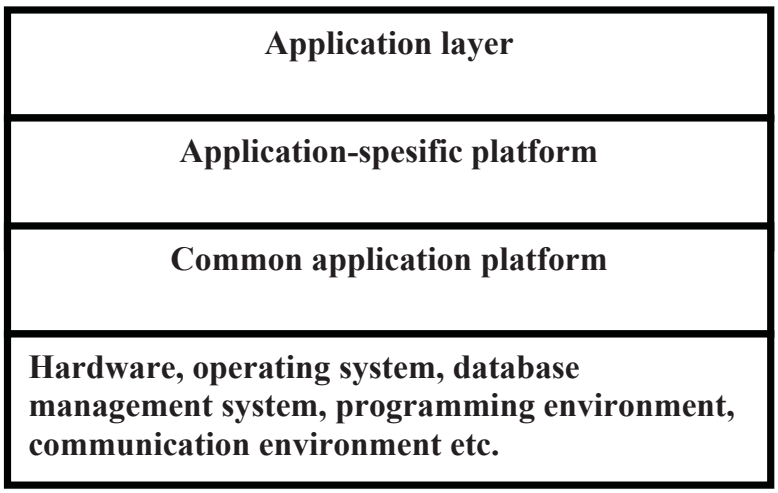


Figure 1: Software architecture platform (Jakobson 1993)

There exists several platform definitions and characterizations (Maier & Lehnerd 1997, McGarth 1995, Sanchez & Machoney 1996). A product platform can be defined to be the common part in certain product group, family or generation. Different kinds of modifications as well as practices to improve process and information management can be built on top of the product platform. It can also be seen as a state up to which the process is already brought to enable the production of customer-specific applications as fast and efficiently as possible, like it is illustrated in Figure 2. Developing a product up to platform level is more expensive and time consuming than customization. Once the product platform exists, customization processes can be done fast, efficiently and parallel for different customers.

In company and company networks point of view platform concept has a rich applicability. Some of the main advantages are as follows:

- Platform concept enables product management and provides tools for product lifetime support.
- By applying the platform concept, one can take advantage of both mass production and customization benefits. In other words the platform concept enables mass customization.
- In addition to products, the platform concept can also be applied to processes, services and to the whole business concepts.

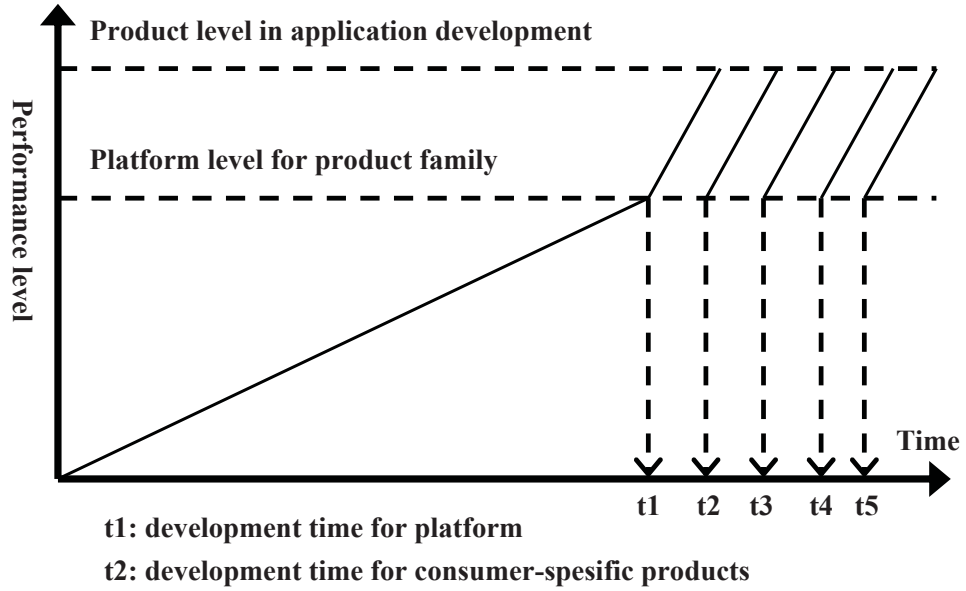


Figure 2: Product platform and customization processes (Saaranen 1998)

It the context of wireless sensor networks, Hill et al. (Hill 2004) presented a classification to four different platform classes. Even though the hierarchical classification they presented provides one way to divide heterogeneous sensor network to different classes, it does not meet the requirements of wireless automation. Less attention is paid to the data processing in sensor network during data aggregation and too much is counted for TinyOS operating system (TinyOS 2009). TinyOS does not offer such a generic level that it would enable fast and easy modifications depending on the particular application needs. It is also suffering about weak documentation and uncoordinated application development. Moreover, it does not offer compatibility with other embedded systems typically used as a part of the automation system. The role of IP-based networking is also underestimated. IP based networking with IPv4 or IPv6 is playing an important role when wireless and cabled parts of the automation system are integrated together. In wireless sensor and actuator networks this support can be offered, for example, by 6LoWPAN compliant protocol stack (6 LoWPAN 2009).

Wireless Automation Standards

Recently, the Hart Communication Foundation (HCF) released the WirelessHART (WirelessHART 2009) standard for wireless automation and some field devices and sensor equipment supporting the specification are already available on the market. The core protocol of WirelessHART is the TSMP (time-synchronized mesh protocol) [Reference], which defines the communication slots in time and frequency for the nodes in the network. The main features of WirelessHART include: time-synchronized communication, self-organization and self-healing, frequency hopping (and channel blacklisting), security (encryption, authentication, integrity), and full mesh networking (path redundancy). WirelessHART is the first standard on wireless communication in industrial automation. Another standard, the ISA100.11a is also coming up during the year 2009 (Isa 2009). It has similar features as WirelessHART, but it is targeted to systems with faster cycle times. The WirelessHART standard currently specifies a minimum communication cycle time of 1 s, and the equipment on the market can support 4 s

communication rate. It is obvious that the standard as such is not useful in all applications, because of the long duty cycles. Typically in the control systems we consider the duty cycles can be up to 100 times faster. Nevertheless, TSMP has several appealing properties and techniques that would be useful if implemented as light-weight versions on the wireless sensor networking platforms mentioned above.

Combined Platform Development in GENSEN

In GENSEN project we combine the requirements set by three different platform development points of view. It is important to notice that these platform requirements effect each other and the best generic solution can be achieved by evaluating the platforms against each other since the early stages of the development process.

Applications

In GENSEN we build five different types of applications: wireless part of the trolley crane control system, monitoring and control of wind turbine generators, monitoring and control of distributed energy production system, greenhouse automation and cattlehouse automation. The application development will be done with the companies who participate to the GENSEN project and operate in the pre-mentioned areas. It is obvious that the system requirements of each of these applications differ from each other, but the common part of the requirements can be found by comparing them to each other. By doing so we can find the level up to which the common technology platform can be developed. This level can be increased by following the maximal requirements, but it must be done in such a way that it does not cause useless additional costs or useless burden in the applications with lower requirements. The definitions of the application requirements offer a lots of detailed information for the application platform development, such as network performance requirements in terms of lifetime, reliability and data throughput; sensor node size, cost and performance requirements; the requirements of the physical environment; the requirements of the compatibility with the rest of the automation system etc.

Application Platform

We are developing such application platform, which allows a fast production of applications as illustrated in Figure 2. In practice it means that the main objective is to develop a generic wireless sensor network architecture, which scales up for different numbers of sensor nodes, enables advanced networking and data processing and is easy to equip with different types of existing commercial sensors depending on the particular application needs. Advanced networking properties such as multi-hop support, automatic network configuration, re-configuration and advanced data processing are available in the application platform such that they do not need to be tailored separately for different applications. The purpose of this architecture is to enable flexible configuration of sensor networks to be applied in various applications with dramatically less tailoring compared to the state of the art solutions. The core of the concept is a flexible protocol stack developed especially for the needs of time-critical applications, such as wireless automation. The design of the generic wireless sensor network architecture requires research and development in the following main tracks:

1) Technology platform development
    - Integration and definition of generic, compatible hardware components (nodes, sensors, radios)
    - Software and driver libraries for the hardware (including protocol stacks)

- Configuration tool to integrate the application design

2) Protocol development
   - A flexible communication and networking protocol stack for the generic sensor network platform
   - Implementing the useful and energy-efficient parts of the time-synchronized mesh protocol used for example in WirelessHART
   - Better support for asynchronous sampling, data compression, data fusion and control algorithms as required in the technical requirements of the application platforms

3) System validation and testing
   - Development of five applications
   - Interaction between application platform and technology platform development

4) Evaluation of the commercialization capabilities
   - Marketing opportunities of the developed technology platform and application platforms
   - Technical requirements set by the commercialization process

The important requirements we set for the applied sensor platforms are open software and IP-based networking with IPv4 and IPv6 protocol support. We apply Sensinode Microseries and such versions of Sensinode Nanoseries which still provide open software (Sensinode 2009). In addition to further development of Sensinode Nanoseries we will also consider such platforms, which support Contiki operating system (Contiki 2009). Contiki is an open source, highly portable, multi-tasking operating system for memory-efficient networked embedded systems and wireless sensor networks A typical Contiki configuration can have 2 kilobytes of RAM and 40 kilobytes of ROM memory. Contiki provides IP communication, both for IPv4 and IPv6 protocols. Contiki is developed by a group of developers from industry and academia lead by Adam Dunkels from the Swedish Institute of Computer Science (SICS). The Contiki team consists of sixteen developers from SICS, SAP AG, Cisco, Atmel, NewAE and TU Münich. Currently Telos is offering support for Contiki operating system, and Sensinode is going to announce its own Contiki support during the year 2009.

The work which will be done in the context of sensor platforms consists of the software implementation of the developed algorithms and the software and hardware integration. We will use commercially available sensors in our application platform. The types of the applied sensors depend on the particular application needs. We may also develop our own hardware solutions compatible with the pre-mentioned sensor platforms to make them more useful in variety of applications. The main reasons for the need of adding certain hardware components is that many of the sensor network hardware platforms have only single microcontroller that handles both the sampling, data processing and communication protocol related tasks. The system is energy efficient, but the architecture imposes constraints on the utilized applications and protocols. For instance, the node cannot sample data and transmit at the same time. More flexible platform could be achieved by using two microcontrollers where one is dedicated to sampling and data processing and the other is dedicated to running the communication protocols. In addition, the solution based on two microcontrollers could be made very flexible in terms or utilized radio system. An API could be defined for the application to access the communication module. Same application system (sensors and processing) could be utilized

with various radios depending on the communication needs (required radio range, level of interference etc.). We simplify the classification presented by Hill et al. (Hill 2004) to respond better to the needs of wireless automation by proposing a sensor network architecture consisting of two kinds of nodes: 1) energy efficient small capacity nodes that could be utilized as simple sensors or relays, and 2) high capacity flexible nodes that could serve as soft sensors, cluster heads or gateways.

To manage the configurations of the sensor network hardware, software, protocols and algorithms in a particular application, we will develop a configuration tool, which will contain all the sensor drivers, communication protocols and application layer algorithms as well as configuration options of the nodes in libraries. With the tool, the appropriate hardware/software configuration is defined easily with the specifications of the communication duty cycles and mechanisms. The configuration tool will be integrated to the PiccSIM simulator (Kohtamäki 2009), which is a complete co-simulation and co-design environment for wireless networked control systems and wireless automation.

We are targeting to offer an application platform for such automation systems that may include tens or hundreds of measurement points. The flexibility the wireless communication provides must be preserved such that one can easily add, remove or re-locate sensor nodes in the network based on demand on the fly. Thus, after the initial deployment the network must automatically configure itself. The network configuration includes routing, localization, time-synchronization and the assigning of frequencies and network addresses. Clustering can also be included to the configuring operations, if hierarchical network architecture is applied (Virrankoski 2005). Once the initial configuration is completed, the network must maintain that information and also adapt to the possible changes in the network. During the network operation, some nodes may start to malfunction, some may run out of power, some may be removed and some new nodes may be added. There may also exist network level malfunctioning. Hence, network monitoring and diagnostics, together with self-healing mechanisms are important to ensure reliable operation and to identify the sources of malfunctioning. The network solution we develop applies IEEE 802.15.4 communication protocol. The system must support the co-existence of IEEE 802.15.4 and IEEE 802.11 because of the widely spread wireless local area networks, which apply the latter protocol.

The beacon mode of the IEEE 802.15.4 MAC supports network synchronization and slotted CSMA with guaranteed time slots which enable some devices to do contention free transmissions. In real-time applications, synchronized collision free transmissions are needed to minimize transmission delays and jitter. However, the design objective of the beacon mode has been power saving by introducing inactive periods for sleeping. The beacon mode of IEEE 802.15.4 has been found to be very complex and have low throughput, see e.g. (Ko 2006). Hence, we believe that there is a need for MAC protocol that is designed especially for real-time applications.

The TSMP protocol has been designed with the above restrictions in mind. The key properties of the protocol are: 1) time synchronized communication, 2) frequency hopping, 3) automatic node joining and network formation, 4) fully redundant mesh routing, 5) secure message transfer, hence fulfilling all the requirements posed above. The downsides of the protocol are the need for the coordinator node that continuously assigns the time and frequency slots and takes care of node joining etc. Besides the duty cycles achieved, for example, with WirelessHART are too slow for our usage. We will adjust the protocols applied in our application platform such that it better suits the applications we consider and such that the achievable duty cycle will be enhanced. We will define and propose a “TSMP-Lite Protocol” and implement it to the

application platform and the configuration tool with simultaneous support for 6LoWPAN (IP-based networking in WSNs).

Applied network topologies can include sub-networks consisting of very simple, small-size, short range, low-cost and non-standard sensor nodes. These kinds of sub-networks are connected to the rest of the system via bridges with the interface including the standard platforms.

So far many wireless automation applications are still focusing on data collection. In these cases the whole wireless communication acts as a cable replacement; all data is collected to the centralized sink, control decisions are made there and then control commands are sent back to the actuators. There are many reasons why this is not an efficient way to operate. First, sensor nodes have scarce resources that should not be wasted to the continuous transmission of such a data, which is not necessarily relevant. Second, the useless redundancy of the data is already a burden in many systems. Even though the sink has remarkably higher computation and memory resources compared to any of the nodes in the network, useless amounts of redundancy can still cause longer delays and a remarkable waste of resources in the system. Third, bigger amounts of data will increase the risk of packet losses and other data transmission errors in the network. These errors will then increase the network latency and they may even lead to serious network malfunctioning.

We apply spatial correlations and blind compression methods to eliminate the useless redundancy from the data as near the measurement points as possible. We will pay attention to both node level and cluster level data compression. If there is more than one level of hierarchies in the network, the hierarchical structure can also be exploited to create different abstraction levels of the data.

We will also test position assisted data compression: By using the location information, every physical position or area generates only one data flow independent on the actual number of sensor nodes in the area. Once this method is applied, the network can be addressed by positions instead of individual node addressing.

One way to reduce the amount of transmitted data is adaptive and asynchronous sampling. Instead of sampling in a constant frequency, each node can adapt its own sampling rate based on the measured data. In the condition monitoring, for example, there is no need for high sample rate or data transmission, if everything works as expected. If the measured data starts to indicate exceptional behavior, the node should automatically report about the change and also increase its sample rate to be able to figure out more precisely what is actually happening. The amount of node and network resources sacrificed to trace down the malfunctioning must depend on the seriousness of the observed phenomenon.

Usually more than one type of sensor data is required in advanced control systems. The sensor nodes we are applying enable simultaneous use of several types of sensors. Thus, we need data fusion algorithms to combine the multi-sensor data to be able to perform the correct control operations. Data fusion can be done in a partially distributed (or locally centralized) manner in the network cluster heads, or it can be done in a fully centralized manner in the network sink.

We consider data fusion methods typically applied in control engineering, such as Kalman Filter, particle filters, fuzzy computation and Hidden Markov Models. If these methods are applied in a centralized manner, one challenge is the real-time requirement. The data fusion computation should not cause so long delays to the network operation, that it would seriously violate the system real-time requirement. In locally centralized computation one challenge is to find such

light-weight implementation of pre-mentioned methods that it can be executed with a limited memory and computation resources.

Product Platform

Usually the slowest and most expensive part in a product development is the period during which the application platform will be developed up to such a level, which allows a fast production of different types of applications. Once the platform is completed, the application development is much faster and cheaper. Developed application platform can be used with a wide spectrum of automation and monitoring applications such that in each particular case only minor modifications are needed to produce the particular application.

Final step is the development of product platform. Developed application platform itself as well as each of the five validation applications we produce during the project can be commercialized. Some key requirements of the product platform, such as market opportunities, costs, system usability, standardization and possibilities to offer technical support for the application must be taken into account in the application platform development. On the other hand, mandatory requirements rising from the technology platform as well as practical issues observed during the application platform development will guide the development of the product platform.

Conclusions and Future Work

In this paper we made a survey to the field of wireless automation and presented a way to combine technology platform, application platform and product platform development such that it can be done in parallel. Moreover, the platform development processes interact with each other in beneficial way as illustrated in Figure 3.

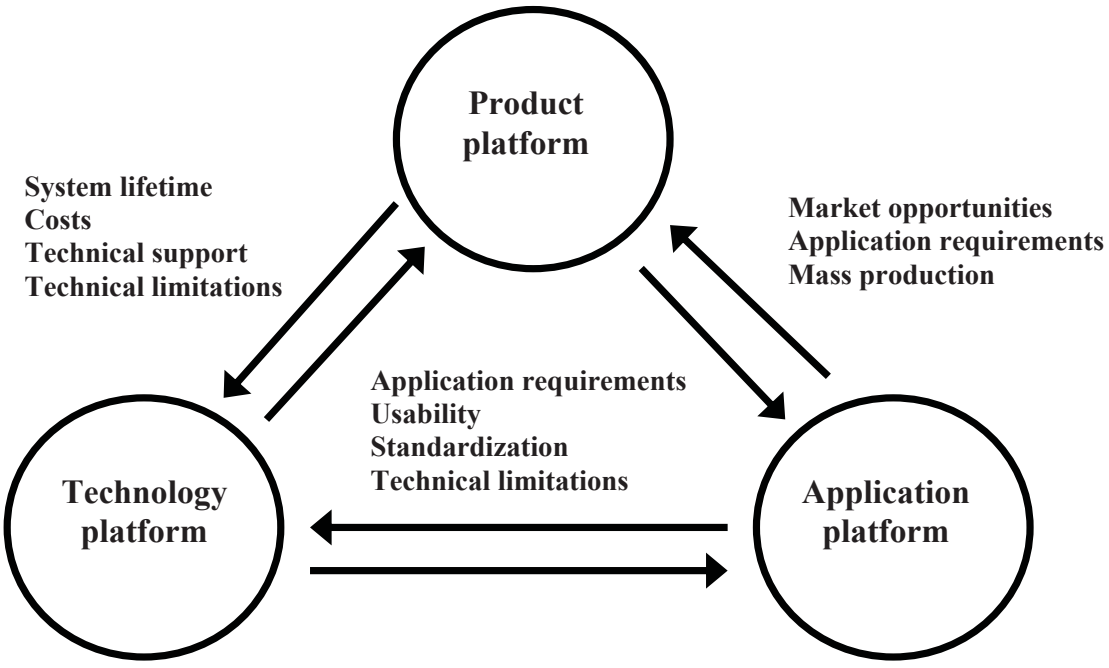


Figure 3: Interaction between platform development processes

As a part of the GENSEN-project on 2009-2010, we will evaluate the validity of the presented combined platform development. We will also clarify the interactions between technology platform, application platform and product platform development. Some marketing research in the discussed application field will be performed to be able to specify the steps to the developed application platform commercialization.

References

Ahonen, T., Virrankoski, R. and Elmusrati, M. (2008), Greenhouse Monitoring with Wireless Sensor Network, in the proceedings of 2008 IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications, October 12-15, 2008, Beijing, China.

Contiki website (2009), http://www.sics.se/contiki/

Crossbow website (2009), http://www.xbow.com

Eriksson, L, Elmusrati, M., Pohjola, M. (ed.) (2008), Introduction to wireless automation - Collected papers of the spring 2007 postgraduate seminar, Helsinki University of Technology, Department of Automation and Systems Technology.

Hill, J., Horton, M., Kling, R. and Krishnamurthy, L. (2004), The Platforms Enabling Wireless Sensor Networks, Communications of the ACM, June 2004/Vol. 47, No. 6.

Imote2 datasheet (2009),
http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/Imote2_Datasheet.pdf (2009)

Isa (2009) website, http://www.isa.org

Jakobson, I. (1993), Is Object Technology Software's Industrial Platform?, IEEE Software, Vol. 15, 24-30.

Ko, L., Liu, Y. and Fang, H. (2006), "Design and Implementation of IEEE 802.15.4 Beacon-enabled Network Devices," In Proc. IEEE PERCOMW'06.

Kohtamäki, T., Pohjola, M., Brand, J. and Eriksson, L. M. (2009), PiccSIM Toolchain – Design, Simulation and Automatic Implementation of Wireless Networked Control Systems. IEEE International Conference on Networking, Sensing and Control, Okayama City, Japan, March 26-29, 2009.

Lymberopoulos, D. and Savvides, A. (2005), XYZ: A Motion-Enabled, Power Aware Sensor Node Platform for Distributed Sensor Network Applications, to appear in the proceedings of Information Processing in Sensor Networks (IPSN), SPOTS track, April 2005

McGrath, M. E. (1995), Product Strategy for High-Technology Companies, New York, Irwin Professional Publishing.

Meyer, M.H. and Lehnerd, A.P. (1997) The Power of Product Platforms: Building Value and Cost Leadership, New York, NY: Free Press.

Polastre, J., Szewczyk, R. and Culler, D. (2005), Telos: Enabling Ultra-Low Power Wireless Research, The Fourth International Conference on Information Processing in Sensor Networks: Special track on Platform Tools and Design Methods for Network Embedded Sensors IPSN 2005: 364-369, April 25-27, 2005

Saaranen, J. and Keskinen, S. (1998), Product and Process Platform – a New Concept for Management, 7th International Conference on Productivity & Quality Research (ICPQR'98), Miami, USA.

Sanchez, R. (1996), "Strategic product creation: Managing new interactions of technology, markets and organizations", European Management Journal, Vol. 14 pp.121 - 138.

Sensinode website (2009), http://www.sensinode.com

TinyOS website (2009), http://www.tinyos.net (2009)

Virrankoski, R., Savvides, A. (2005), TASC: Topology Adaptive Spatial Clustering for Sensor Networks, Yale University, ENALAB, in the proceedings of the 2nd IEEE International Conference on Mobile Ad Hoc and Sensor Systems (MASS'05), November 7-10, 2005, Washington DC, USA.

WirelessHART; HART Communication Foundation website (2009), http://www.hartcomm.org

6LoWPAN website (2009), http://www.6lowpan.org

Biographical Notes

Mr. **Reino Virrankoski** (Lecturer of Telecommunications, M.Sc.) received his M.Sc. in Mathematics from the University of Helsinki on 2000 and he is a Ph.D.-student in Control Engineering at Helsinki University of Technology. On 2000-2007 he was working as a Graduate Researcher at Helsinki University of Technology. During that time he also worked as a Visiting Assistant Researcher at Yale University on 2004-2005. Currently he holds a faculty position as a Lecturer of Telecommunications at the University of Vaasa. Mr. Virrankoski's main research interests are communication and control in telecommunication systems and in wireless sensor networks, wireless automation, localization and controlled mobility and wireless networks in defense and security.

Mr. **Simo Keskinen** (Senior Researcher, Licentiate of Economics) is currently working as Senior Researcher in the Department of Production at University of Vaasa. His research focuses on development of infrastructure networks, technology management, production and product development. He has been working in industrial R&D tasks in Salora Oy (later a part of Nokia) and in teaching tasks in the polytechnics of Oulu and Vaasa.

Stackable Wireless Sensor and Actuator Network Platform for Wireless Automation: The UWASA Node

Huseyin Yigitler, Reino Virrankoski, and Mohammed Salem Elmusrati
University of Vaasa, Department of Computer Science,
Communications and Systems Engineering Group
P.O. Box 700, FI-65101 Vaasa, Finland
{husali, reino.virrankoski, mohammed.elmusrati}@uwasa.fi

***Abstract*—Diversity of wireless automation application demands causes a tendency to develop different platforms for different applications, which yields long and uneven development periods even for similar applications. Consequently, there is an emerging need for generic architecture for wireless automation both in software and hardware level. In this work, we are introducing a modular and stackable wireless sensor platform, the UWASA Node. The UWASA Node is designed to provide reasonable level of processing power and memory for moderate level algorithms, and necessary means for controlling and monitoring power consumption. It is designed to support applications that require very low-power single processor wireless sensor, up to applications that have processing demands which can be met by multiple processors stacked one top of another.**

***Index Terms*—Wireless Sensor Networks, Wireless Automation, Wireless Sensor Network Platform**

I. Introduction

Wireless automation can be described as performing automation tasks over the environment, which is covered by sensors that exchange information over wireless links, possibly forming multi-link paths. These kinds of applications, traditionally, operate on wired networks due to stringent time and reliability constraints imposed by the control, data fusion, monitoring etc. requirements. Recent advances in wireless communication protocols, miniaturization in micro-electronic circuits and decentralized algorithms have enabled option for true replacement of wired networks with wireless networks at the cost of solving reliability, delay, limited resource, and security problems associated with the wireless communication [1].

Wireless Sensor and Actuator Networks (WSANs) provide means to intelligently acquire and change the state of the physical world. These networks are composed of low cost, low power, short distance and multifunctional sensor nodes [2]. Employment of WSANs for automation provides cost effective replacements of cables, easier and faster deployment compared to wired systems, while providing reconfiguration and expansion abilities, as well as providing extension to applications which are impossible for cabled systems [3]. The cost effective replacement allows application of large number of measurement points, allowing development of self-organized, inherent colloborative processing, self-healing systems.

In general, wireless automation applications are divided into two categories, real-time and event driven. Although this distinction allows classifying some requirements of each application category, there are some properties that should be fulfilled in common. For example, applications falling in either category put stringent constraints on the timing (or latency) and reliability of the network [4].

Utilization of WSANs for wireless automation applications is limited due to problems associated with either wireless communication, or resource limited nature of the network nodes. The WSAN nodes are usually designed as low-power wireless communication enabled devices that can operate for very long time span with a limited energy battery. This requirement put constraints on the node hardware, composed of a low process power, low memory processor and a short communication range, low bandwidth wireless transceiver. However, the wireless automation applications usually require time synchronized operation, which demands higher process power to meet measurement update rate of the application. Some automation algorithms require buffering and processing of chronological data chunk, which requires higher memory. On the other hand, wireless communication causes reliability, un-deterministic delays and security problems, which are parts of the communication protocol development.

The diversity of wireless automation application demands causes a tendency to develop different platforms for different applications, which yields long and uneven development periods [5]. This is true for hardware, software, and algorithm development stages. However, the development period can substantially be reduced by considering carefully engineered solutions for each level of development.

Although, in general, hardware development is necessarily

This work is supported in part by the Finnish Funding Agency for Technology and Innovation (TEKES), and is developed as a part of the GENSEN project.

application dependent, requiring re-design for different applications, it is possible to define common requirements for broad class of wireless automation applications. A wireless sensor platform that is intended for wireless automation should have enough resources for efficient data fusion, compression, aggregation, and to run control algorithms, while providing low-power operation options. It is also possible to develop a node hardware that provides basic wireless communication, moderate level of process power and memory, power management options, and extendable peripheral options. In this case, the application dependent hardware development would be left as relatively simple task of attaching simple slave boards. Such hardware architecture would be used as a backbone for most automation applications, while providing fast and easy way of adopting itself for different application requirements.

In this work, we are introducing a modular and stackable wireless sensor platform, the UWASA Node, developed within the context of GENSEN project, [5], which aims to be a generic hardware platform up to a level that it allows fast adaptation for development of different wireless automation applications. The proposed hardware architecture is designed to adopt itself for different levels of application hardware demands by providing means of stacking relatively simple slave boards. Although the idea is simple, providing modular structure causes some problems that should be solved in hardware level.

The remaining part of this presentation contains two sections. In section II, necessary details of hardware model and basic considerations are given. The section III is reserved for detailed explanations of hardware modules.

II. The UWASA Node Hardware Model

There is an emerging need for generic sensor architecture for wireless automation, since the existing solutions fail to fulfill the requirements of broad class of applications, causing slow and uneven application development periods. Within the context of GENSEN project, the UWASA Node, a Modular and Stackable WSN node is designed, which has hardware architecture shown in Figure 1.

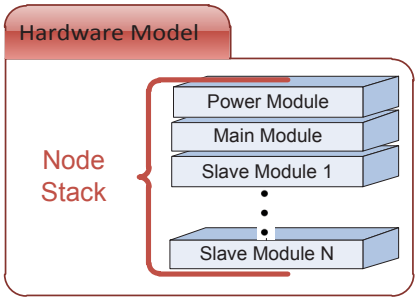


Figure 1: Hardware Model of the UWASA Nodes

As shown in Figure 2, an intended application must have at least Power Module and Main Module. In addition to those, a node may contain some Slave Modules to adopt itself to application needs, or to act as a backbone router or translation gateway in the network. Regardless of types and number of slave modules, all modules are interconnected with circuits provided by Hardware Stack Connector.

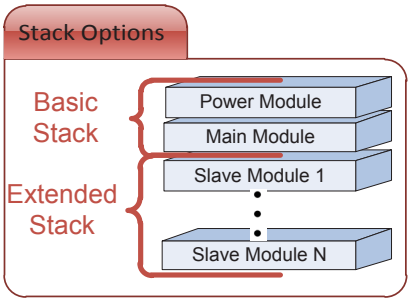


Figure 2: Stack options of the UWASA Nodes

A. Main Design Issues

The UWASA Node is designed to support applications that require very low-power single processor wireless sensor, up to applications that have processing demands that can be met by multiple processors stacked one top of another. Such a broad application support is achieved by intrinsic support for multiple processing units in single hardware stack, while providing a feature to turn off the power of a built-in processor. Although this provides necessary means for extendable and generic solution, it arises some problems that need to be addressed.

The first problem is related to power supervisory need inside the node hardware. In order to manage the power consumption of the node, there is a need to develop hardware components that allow power state management. This need is solved by the control interfaces provided by the Power Module, and a unified software architecture that allows power state control of each processor of main module and slave modules.

The second problem is related to time management inside the node stack. For a single processor hardware module, there is a single time measurement mechanism, which allows network time synchronization through relatively simple algorithms. The UWASA Node architecture provides means to manage in-node timing either by main controller or by RF controller, or both. All processors in the node stack are obliged to be connected to *heart-beat* signal provided by either RF controller or main controller, which allows proper time stamping and periodic operations.

The third problem is complicated software demand of the node, because multiple different processors require different software to be developed. This problem is solved by middleware like software, which provides hardware abstraction up to a level that most of the application software development is over the provided APIs.

The fourth problem is related to application software development. The hardware should support fast application software development by providing necessary support for debug interfaces. Within the context of hardware development a *Development Board* has also been designed. The development board contains sufficient number of peripheral interfaces, and debug emulators for the Main Module

processors.

B. Supply Options and Power Management Interfaces

The nodes can be deployed in environments that can provide multiple power sources. The UWASA Node supports two independent supplies, one of them being a rechargeable battery; the other one being any source meeting the requirements (preferably an infinite energy source). The UWASA Node is equipped with dynamic power path management hardware, which efficiently utilizes two supplies. If the infinite energy source can supply more power than the instantaneous demand of the node, the battery starts to be recharged.

The power module is equipped with 10 independent regulators. Two of these regulators have fixed voltage outputs to supply the main module. Other eight regulators have adjustable output voltage levels, and they can be disabled by logic controls. This feature allows proper power control of the slave modules.

The power module contains battery fuel gauge, which can be accessed by the main controller. This feature allows battery energy monitoring, and generation of accurate low power alerts. Moreover, this gauge can be used for quantification of power demand of application tasks, which is crucial for determination of energy demand of a specific application.

C. Time Synchronization and Management

For proper data acquisition and algorithm operation, node has to guarantee synchronized operation between its processing units. Moreover, the whole node should support network level time synchronized operation. The in-node time management is performed by a heart-beat signal provided by the Main Module. This signal is used for time stamping of each data chunk that will be exchanged inside the node. Being simple, yet, this architecture provides glueless time management. The network level time synchronization is handled by the Main Module. The output of the network level time synchronization algorithm is used for compensation of skew of the time measurement sub-systems inside the node, and also it is used for updating the heart-beat signal source period.

III. Hardware Design Details

A. Main Module

The Main Module is the master module, and contains two processing units to allow transparent wireless communication while providing necessary means for stackable hardware architecture. The basic hardware blocks of the Main Module are shown in Figure 3.

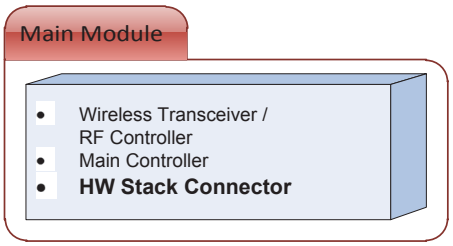


Figure 3: The basic hardware blocks of the Main Module

The Main Module is responsible for wireless communication, managing in-node data exchange, performing data processing and decision making, while supervising power interfaces and managing in-node power mode. The main components of the main module are shown in Figure 4.

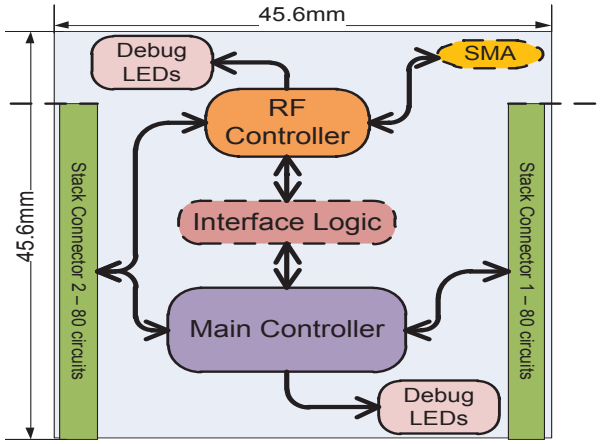


Figure 4: The components of the Main Module

The main features of the main module can be summarized as follows.

1. RF Controller and Main Controller are both programmable microcontrollers; RF Controller has 8051 based, 8-bit processor that runs at either 16 MHz or 32 MHz, whereas Main Controller has ARM7TDMI-S 32-bit processor that can run at up to 72 MHz.
2. RF Controller has integrated IEEE802.15.4 MAC, integrated positioning engine; whereas Main Controller has integrated Ethernet MAC, and USB2.0 Full Speed device, as well as many standard serial interfaces.
3. RF Controller can switch off the power of Main Controller, when excessive process power is not needed; whereas, the Main Controller can control enable states and voltage level of power sources through the interfaces provided by the Power Module.
4. Individual peripherals of Main Controller and RF front end of RF controller can be disabled.
5. Operating frequencies of both RF Controller and Main Controller can be adjusted.
6. Time synchronization is solved by the means of heart-beat signaling provided by either RF Controller or Main Controller. Network level time synchronization is ensured by the RF Controller.

B. Power Module

Power Module is responsible for providing power related interfaces to all node modules. The basic hardware blocks of the Power Module are shown in Figure 5.

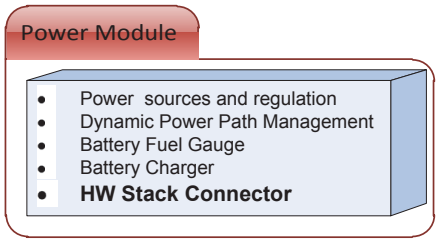


Figure 5: The basic hardware blocks of the Power Module

This module contains power source path management, battery monitor, battery charger, voltage regulators, and power control logic block. Those components are shown in Figure 6.

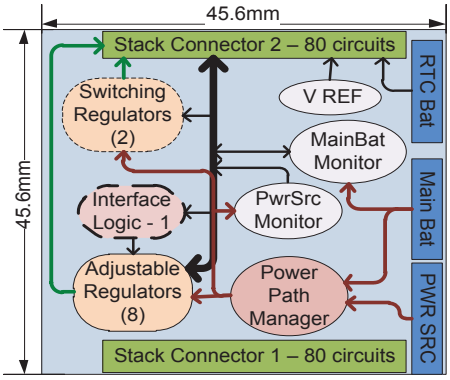


Figure 6: The components of the Power Module

The main features of the Power module can be summarized as follows,

1. The node supports LiIon – LiPo battery connection. The power module is designed to operate within the safety limits of LiIon – LiPo batteries.
2. The power module contains integrated LiIon – LiPo battery charger, which allows on-line charging the battery when the other source is connected to a supply.
3. Two independent supply options are provided through Power Module; the whole node supports dynamic power path management when the node is supplied via battery and other source (e.g. solar panel, vibration based power generator, USB connection, etc.).
4. Instantaneous (power drain)/charge status of the battery can be monitored.
5. The voltage level on the main power line of the node can be monitored.
6. 10 independent power regulators are provided on the Power Module. 8 of them with adjustable output voltage levels; where 4 of them are switching regulators, and other 4 being linear regulators. Those regulators should be used to supply slave modules, to allow main controller to turn on/off whole slave module hardware, to minimize quiescent currents

C. Slave Modules

Three types of Slave Modules are defined depending on the purpose,

1. *Active Slave Module:* This type of Slave Module has its own processing unit. Consequently, data processing – driver related issues are handled on the slave module itself.
2. *Passive Slave Module:* This type of Slave Module does not contain its own processing unit, but has driver related hardware of Sensor/Actuator, which is connected to one of the interface of the Main Module on the Hardware Stack Connector.
3. *Extension Slave Module:* This type of Slave Module is an extension to Main Module. For example, SRAM connected to EMIF interface of the Main Module is this type of Slave Module.

The Slave Modules are intended to be the hardware implementations of the applications. The basic hardware blocks of three types of slave modules are shown in Figure 7.

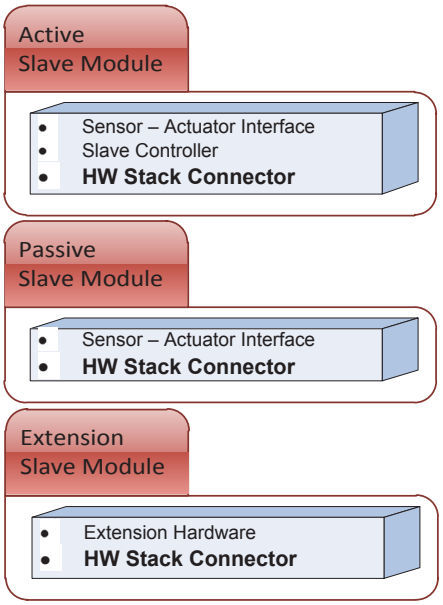


Figure 7: The basic hardware blocks of the Slave Modules

D. Development Board

In order to simplify development stage, and provide hardware-less development option, a Development Board is designed. This board aims to provide essential development interfaces for both RF Controller and Main Controller. The content of this board is shown in Figure 8.

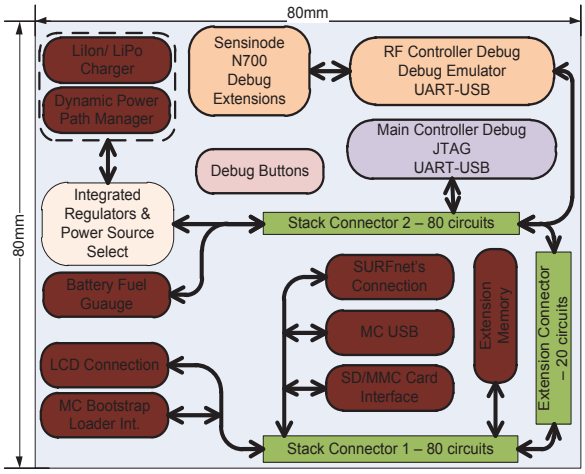


Figure 8: The functional blocks of the Development Board

REFERENCES

[1] V.C Gungor and G.P. Hancke, "Industrial wireless sensor networks: Challenges, design principles, and technical approaches," *Industrial Electronics, IEEE Transactions on*, vol. 56, no. 10, pp. 4258--4265, 2009.

[2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393-422, 2002.

[3] R. D. Poor, "Reliable wireless networks for industrial applications," in *RF & Wireless Technologies*.: Newnes, 2008, pp. 423-434.

[4] Jean-Philippe Vasseur and Adam Dunkels, *Interconnecting Smart Objects with IP: The Next Internet*.: Morgan Kaufmann, 2010.

[5] Reino Virrankoski and S Keskinen, "GENSEN: A Novel Combination of Product, Application and Technology Platform Development in the Context of Wireless Automation," in *Proceedings of the 14th International Conference on Productivity & Quality Research (ICPQR 2009)*, Alexandria, Egypt, 2009.

Greenhouse Monitoring with Wireless Sensor Network

Teemu Ahonen, Reino Virrankoski and Mohammed Elmusrati
University of Vaasa
Department of Computer Science
Telecommunication Engineering Group
P.O. Box 700, FI-65101, Vaasa, Finland
Tel. +358-6-324 8111
Fax. +358-6-324 8677
{teemu.ahonen, reino.virrankoski, mohammed.elmusrati}@uwasa.fi

Abstract — **In modern greenhouses, several measurement points are required to trace down the local climate parameters in different parts of the big greenhouse to make the greenhouse automation system work properly. Cabling would make the measurement system expensive and vulnerable. Moreover, the cabled measurement points are difficult to relocate once they are installed. Thus, a Wireless Sensor Network (WSN) consisting of small-size wireless sensor nodes equipped with radio and one or several sensors, is an attractive and cost-efficient option to build the required measurement system.**

In this work, we developed a wireless sensor node for greenhouse monitoring by integrating a sensor platform provided by Sensinode Ltd. [1] with three commercial sensors capable to measure four climate variables. The feasibility of the developed node was tested by deploying a simple sensor network into Martens Greenhouse Research Foundation's greenhouse in Närpiö town in Western Finland. During a one day experiment, we collected data to evaluate the network reliability and its ability to detect the microclimate layers, which typically exist in the greenhouse between lower and upper flora. We were also able to show that the network can detect the local differences in the greenhouse climate caused by various disturbances, such as direct sunshine near the greenhouse walls. This article is our first step in the area of greenhouse monitoring and control, and it is all about the developed sensor network feasibility and reliability. Data analysis, control solutions and more complex network setups will be left to be the main directions of our future work.

I. INTRODUCTION

The most important factors for the quality and productivity of plant growth are temperature, humidity, light and the level of the carbon dioxide. Continuous monitoring of these environmental variables gives information to the grower to better understand, how each factor affects growth and how to manage maximal crop productiveness [2]. The optimal greenhouse climate adjustment can enable us to improve productivity and to achieve remarkable energy savings - especially during the winter in northern countries [3].

In the past generation greenhouses it was enough to have one cabled measurement point in the middle to provide the information to the greenhouse automation system. The system itself was usually simple without opportunities to control locally heating, lights, ventilation or some other activity, which was affecting the greenhouse interior climate. This all has changed in the modern greenhouses. The typical size of the greenhouse itself is much bigger what it was before, and the greenhouse facilities provide several options to make local adjustments to the lights, ventilation, heating and other greenhouse support systems. However, more measurement data is also needed to make this kind of automation system work properly. Increased number of measurement points should not dramatically increase the automation system cost. It should also be possible to easily change the location of the measurement points according to the particular needs, which depend on the specific plant, on the possible changes in the external weather or greenhouse structure and on the plant placement in the greenhouse.

Wireless sensor network (WSN) can form a useful part of the automation system architecture in modern greenhouses. Wireless communication can be used to collect the measurements and to communicate between the centralized control and the actuators located to the different parts of the greenhouse. In advanced WSN solutions, some parts of the control system itself can also be implemented in a distributed manner to the network such that local control loops can be formed. Compared to the cabled systems, the installation of WSN is fast, cheap and easy. Moreover, it is easy to relocate the measurement points when needed by just moving sensor nodes from one location to another within a communication range of the coordinator device. If the greenhouse flora is high and dense, the small and light weight nodes can even be hanged up to the plants' branches.

WSN maintenance is also relatively cheap and easy. The only additional costs occur when the sensor nodes run out of batteries and the batteries need to be charged or replaced, but the lifespan of the battery can be several years if an efficient power saving algorithm is applied.

In this work we took the very first steps towards the wireless greenhouse automation system by building a wireless measuring system for that purpose and by testing its feasibility and reliability with a simple experimental setup. We integrated three commercial sensors to Sensinode's sensor platform [1]. By using these sensors, we are able to measure four parameters, which are crucial in greenhouse climate adjustment: temperature, relative humidity, light irradiance and air carbon dioxide content. The platform uses 6LoWPAN protocol, which allows us to send compressed IPv6 packets over IEEE 802.15.4 networks.

II. RELATED WORK

The Rinnovando group [4] is doing research work in a tomato greenhouse in the South of Italy. They are using Sensicast devices for the air temperature, relativity humidity and soil temperature measurements with wireless sensor network. They have also developed a Web-based plant monitoring application. Greenhouse grower can read the measurements over the Internet, and an alarm will be sent to his mobile phone by SMS or GPRS if some measurement variable changes rapidly. The Rinnovando group has a test bed in 20 x 50 meters tomato greenhouse. In their test bed, six nodes are deployed into two rows 12.5 m apart from each other. One mesh node works as a repeater and improves the throughput of the communication. Bridge node gathers data from other sensor nodes, which transmit the measurements of temperature and relative humidity in one minute intervals [4].

Liu et al. [5] have developed and tested a WSN prototype for environmental monitoring inside the greenhouse. They are using a star topology network of Crossbow MICAz motes. The motes measure temperature, humidity and soil moisture, and send their measurements to the sink node in five minutes intervals. Sink node is a combination of MICAz mote and MIB510 board with data terminal. The terminal with ARM processor module shows the latest measurements in LCD-screen inside the greenhouse and delivers the data to the main PC by using GSM module. The central PC located further apart from the network takes care of data logging and processing. Mote programming and data receiving is possible through the RS-232 serial interface provided by MIB510 board. The Received Signal Strength Indicator (RSSI) values over the distance between nodes with different antenna heights and polarization angles were compared to each other. Based on the results it was possible to conclude that the longest communication range was achieved when nodes had same orientation and maximal antenna height. The temperature difference in experimental measurement between two nodes, where one node was placed in the center of the greenhouse and another near the greenhouse wall, indicates the existence of the microclimate layers [5].

III. PLANT DEVELOPMENTAL FACTORS

The productivity of the greenhouse depends on many different factors. Many research projects are focusing to these factors and their interdependencies. Grower can set the reference values to certain environmental variables, and then the greenhouse automation system targets to keep the variables in these values. The optimal levels of water and fertilizer can also be defined [5].

Carbon dioxide (CO_2) is a natural gas, which is dangerous for humans in high concentrations, but a lifeline for trees and plants. The air consists of nitrogen, oxygen and carbon dioxide. In the photosynthesis process, the plants convert CO_2, water and light into glucose and oxygen according to

$$6H_2O + 6CO_2 + light \quad C_6H_{12}O_6 + 6O_2 .$$

Thus, CO_2 is an important greenhouse climate variable, which enhances the growth of the plants. Sunshine and lights increase the amount of carbon dioxide. During the summer, the greenhouse gets the CO_2 it needs from the natural air, when ventilation and roof windows are open [3]. In northern countries this opportunity does not exist during the winter. Grower can use pure extra CO_2, or he can produce more carbon dioxide by CO_2 burner. Some greenhouse heating systems re-circulate their CO_2 emissions into the greenhouse making double advantages for the producer [2]. The use of external CO_2 offers also a way to tie the carbon dioxide collected from some industrial process to the biomass grown in the greenhouse instead of emitting it to the atmosphere.

The optimal greenhouse air temperature depends on the intended level of the photosynthetic activity. Each plant species has its own optimal values of air temperature and active radiation of light, which enable the maximum photosynthetic activity (see Figure 1). Soil temperature plays also an important role. Conduction heat transfers directly to the soil structure and through convection between the plant roots and water flow around them.

A main concern in humidity and temperature control is to provide the best conductivity to active movement of water and nutrients through the plant. Humidity control is also an important tool to prevent the spread of plant diseases in greenhouses. Normally, the range of healthy relative humidity for the plants is from 50% to 70%. High air moisture reduces the required plant watering frequency. The greenhouse automation uses the watering and misting system, if the air moisture decreases under the targeted level [6].

Temperature and humidity are closely linked together in a greenhouse. Cold air has a lower moisture-holding capacity than warmer air, and therefore the decrease of the relative humidity is a sign of increased air temperature [3]. Transpiration rate tells how many grams plant's leaf surface called stomata releases water vapour per minute.

The greenhouse protects the plants from the extreme weather conditions. However, if the period of daylight prevents the photosynthetic activity, the plants do not grow. Horticultural lighting allows the grower to extend the growing season. It enables a year-round producing of plants or makes it possible for the grower to start sowing in early spring and continue season till the first frost. Plants need about 10-12 hours light to improve growth. When the plants

are producing flowers or fruits the supplemental need of light per day increases up to 16 hours. Figure 1 shows the photosynthetic activity in different wavelengths of light radiation [2].

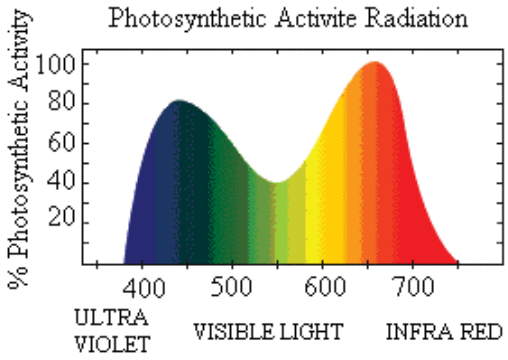


Fig. 1. Photosynthetic activity in different wavelengths of light radiation. Figure from [2].

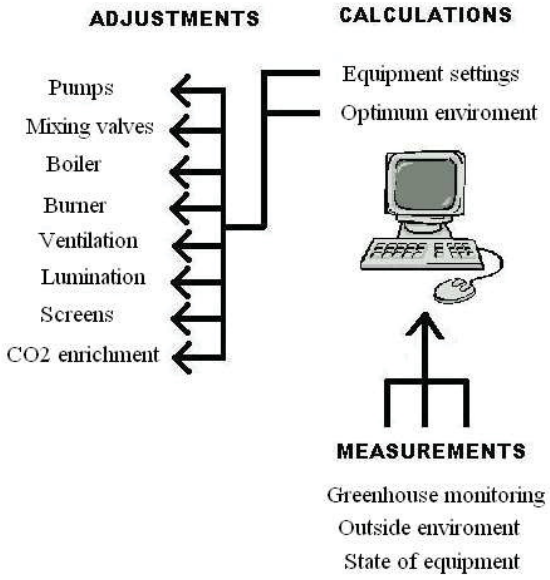


Fig. 2. Tasks in greenhouse environmental control. Figure from [2].

IV. GREENHOUSE CONTROL

Greenhouse monitoring and control can be divided into three main tasks: Measuring, calculating and adjusting [2]. These three tasks have their own functionalities as presented in Figure 2.

The measured values of the greenhouse climate variables are first converted from analog to digital and then transmitted to the computer. Because of the high moisture in the greenhouse, the computer is normally located outside. Signal provided by the sensors is normally weak. Without signal amplifier cabled sensor units cannot transmit the data correctly. Wireless sensor network does not have such problems. Measured data can be sent directly to the gateway node which is plugged in to the computer (see Figure 5), or it can be transmitted in a multi-hop manner via router nodes, if the distance between the measuring nodes and the computer exceeds the length of a single radio link. Besides data collecting and control calculation, the computer also presents the climate variable values and statistics on the screen for the user. The computer runs the greenhouse climate control algorithm, and the new values for the control signals are computed typically in every 15-60 seconds.

Control output signals from the computer have low voltage (24 volts). Each output is connected to electronic relay, which switches the equipment under its control on or off through the second relay, which gives to the device the input voltage it needs. The control system is illustrated in Figure 2. Computer computes the intermediate time from the output signal and then determines how long each relay is turned on [2].

A modern greenhouse can consist of several parts which contain their own local climate variable settings. As a consequence, several measurement points are also needed.

V. EXPERIMENTAL SETUP IN A GREENHOUSE

A. The Greenhouse Environment

We made our experiments in Martens Greenhouse Research Center's greenhouse in the Närpiö town in Western Finland [7]. The size of the greenhouse was 18 x 80 meters and in its traditional control system it has only one cabled measurement unit in the middle.

Greenhouse's moist climate and dense flora are similar to the surroundings of a jungle. This kind of environment is challenging both for the sensor node electronics and for the short-range IEEE 802.15.4 wireless network, which communication range is much longer in open areas. Therefore, we limited the distances between communicating nodes to 15 meters in our deployment.

B. Sensor Nodes

The wireless sensor node we used was Sensinode's Micro.2420 U100 (see Figure 3) [6]. It operated as a basic measuring node with a CC2420 802.15.4 RF-transceiver and a MSP430 Microcontroller. The gateway node was a combination of U100 node and USB serial adapter board (Micro.USB U600) [1]. Sensors were soldered to a board equipped with the needed components (resistors, capacitors and operation amplifier). Then the sensor board (see Figure 4 on the left) was plugged in to the U100 node through its I/O pins. The node and two 1,5V AA-batteries acting as a power source were sheltered by a plastic box (80*55*33mm) to prevent them from the humidity. Sensor board was placed on the top of the box and sensitive electrical components were protected from the moisture by a plastic coating spray.

Finally, the whole board was enveloped by ESD plastic sachet leaving only the heads of the sensors outside.

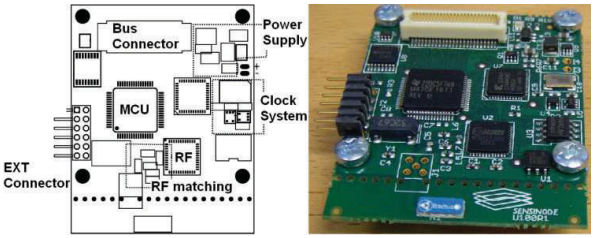


Fig. 3. Sensinode's Micro.2420 U100. The node is equipped with ZigBee radio, but it operates under 6LoWPAN protocol.

Sensinode's devices are based on 6LoWPAN protocol, which enables transmission of compressed Internet Protocol version 6 (IPv6) packets over IEEE 802.15.4 networks [8]. Sensinode's Nanostack protocol provides the use of 6LoWPAN and a standard Socket API for accessing the network. It works in 2,4GHz ISM band and offers 250 kbps data rate [9].

C. Sensors

Fast response time, low power consumption and tolerance against moisture climate made SHT75 relative humidity and temperature sensor [10] a perfect solution for the greenhouse environment. Temperature accuracy of the sensor is ±0.3 °C and the accuracy of the relative humidity under ±2 %. Communication between SHT75 sensor and node is similar to IIC interface developed by Philips. Data and clock line are the same in both cases, but SHT75 has only one pull-up resistor between data and power supply line.

Luminosity was measured by TAOS TSL262R [11], which converts light intensity to voltage. Unstable output signal is handled by low-pass filter to get correct luminosity values.

We mounted irradiance, temperature and humidity sensors into four nodes, but Carbon dioxide sensor was tricky because it sets special requirements for the input voltage and the response time. Figaro's TGS4161 [12] carbon dioxide sensor (see Figure 4 on the right) was the alternative, which was the most compatible with low voltage sensor node. CO_2 measuring takes longer time than other measurements and CO_2 sensor voltage supply must be within ±0.1V from the 5 Volts. The carbon dioxide value can be read from the output voltage. Operation amplifier raises the voltage level of otherwise weak signal from the sensor. We end up to left the TGS4161 to be implemented in its own node which can also act as a router node in a multi hop network, which will be part of the future work.

D. Node Deployment and Network Architecture

We applied a simple star topology, where four nodes with temperature, luminosity and humidity sensors measured climate variables and communicated directly with the gateway node. The gateway node acted as a coordinator and received the measured data from the sensor nodes. It was located in the greenhouse entrance hall because the humidity there was 20-30% lower than inside the greenhouse. A laptop computer was connected to the gateway node by USB-cable.

Martens greenhouse was divided into vertical blocks and the nodes monitored one block at a time. Figure 5 illustrates how the sensor nodes were deployed to the greenhouse block. The idea of the vertical deployment was to get a better understanding of the microclimate layers which typically exist in the greenhouse, and to figure out what kind of differences occur in the climate between lower and upper flora.

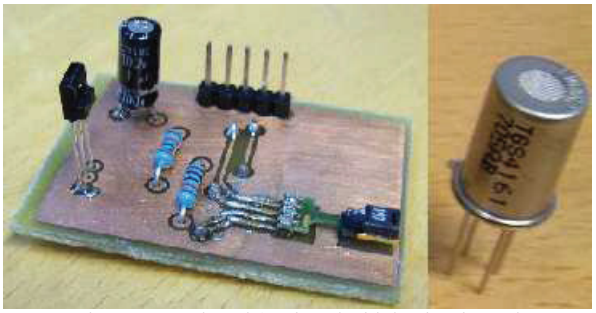


Fig. 4. Sensor board (equipped with luminosity and temperature/humidity sensors) and carbon dioxide sensor.

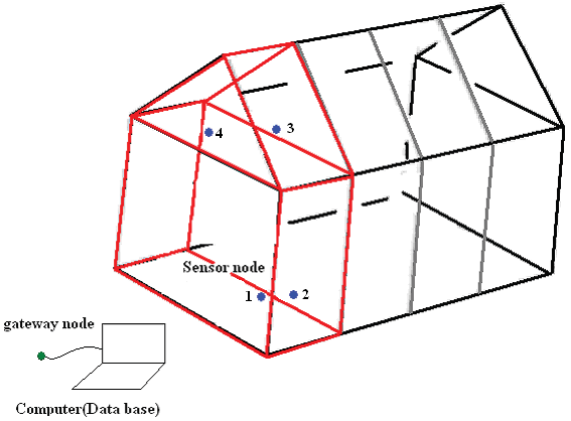


Fig. 5. Experimental setup in Martens greenhouse.

Node 1 (see Figure 5) was placed 490 cm away from the glazed side wall of the greenhouse. It was hanged in 120 cm height and the distance to the edge of dense tomato foliage was 410 cm. Node 2 had 180 cm distance to the side wall and it was placed at the height of 176 cm. That location was a shadowy spot where the nearest lamp was broken. The length between the first plants and the device was 174 cm.

Node 3 measured the crown layer in 310 cm height just above the Node 1. Node 4 was in the middle of the greenhouse block 930 cm away from the side wall and hanged at a height of 295 cm. The distance from the node to the edge of the foliage was 135 cm. Node 2 and Node 3 are shown in Figure 6.

Periodical sleep and wake modes were applied. In its turn, each node woke up and turned on its radio for 15 seconds, and went then back to sleep and turned off its radio for 255 seconds (4 min 15 s). At a time, only one of the four nodes equipped with temperature, luminosity and humidity sensors was reading data from the sensors and waiting data request from the coordinator. The coordinator took care of data requesting, and the other nodes were only able to answer to the request. Thus, the coordinator acted as a master device, which polled data from the sensor nodes in certain time periods. Collisions between other node transmissions were easily avoided in this way.

Fig. 6. The node 2 and node 3 (inside red squares) in a greenhouse test setup.

VI. RESULTS

In our experimental setup, four nodes were deployed to one greenhouse block to gather information about the differences in climate variables between lower and upper flora. Each node red temperature, humidity and irradiance values once in four minute intervals over three hours. During the experiment, the coordinator sent 200 data requests, and each sensor node responded 50 times. Ten packets with readings were either lost or received incorrectly. That indicated 5% data loss rate in terms of packets. The maximal communication range, 15 meters was figured out in individual test where the distance between the coordinator and the sensor node inside the greenhouse dense flora was increased until the connection was lost. We also observed that the reliable range in terms of tolerable packet loss was approximately 10 meters. Compared to our previous experiment in an open parking lot, the reliable communication range fell to one third in the greenhouse's dense flora.

A fickle weather on the measurement day affected the results. The sun was shining for a half an hour in the beginning of the test and later on during shorter periods of the day. The greenhouse environmental control system, Priva[13], adjusted the ventilation, heating and misting according to new samples once in 15 minutes. Priva's measurement box located in the middle of the greenhouse, and the block where our sensor nodes were deployed, was in the greenhouse's south end.

The temperature values measured by four wireless sensor nodes are shown in Figure 7. Local temperature values were strongly influenced by the sunshine at the beginning of the experiment period. Node 1 was far away from the greenhouse ceiling and from both greenhouse walls. Thus, the temperature stayed stable in its area for most of the time. The 15 minutes sampling period in the greenhouse control system explains, why temperature raised over 30 Celsius in some spots before the control system opened the roof windows. Later on, a partly cloudy weather balanced the results between the nodes for the rest of the experiment.

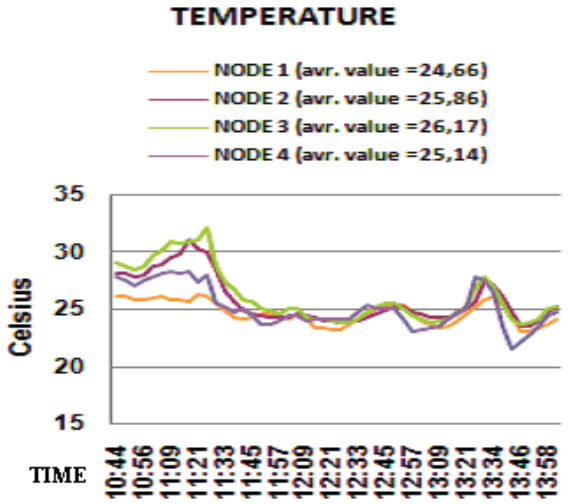


Fig. 7. Temperature measurent results.

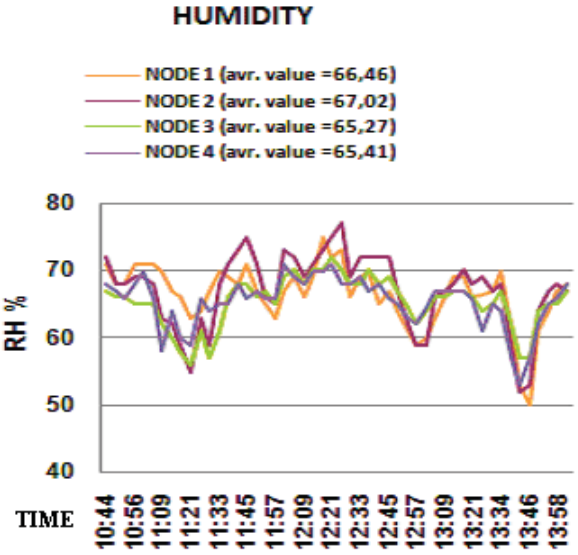


Fig. 8. Relative humidity measurement results.

Node 3 and Node 4 were both placed on the crown layer of the tomato growth. A slant ceiling of the greenhouse made the distance between Node 4 and the ceiling three meters longer than distance between Node 3 and the ceiling. Therefore, the measurements provided by node 4, indicated one degree lower temperature. Node 2 was located near the side wall of which the sun was heating raising the temperature measured by the node.

The relation between the humidity and temperature was explained in Chapter III. The measurements collected by the nodes verified the fact that the lowering of the relative humidity increases the air temperature and vice versa. Figure 8 shows the changes in relative humidity between four nodes. Comparison between temperature and humidity values (Figures 7 and 8) shows how variables are linked together. For example, two distinct drops in humidity are clearly to be seen in the Figure 8. Temperature values increased at the same time when moisture dropped, as shown in Figure 7. Relative humidity did not differ much between the nodes. Node 1 and Node 2 were placed on a shadowy spot, and they measured a little bit higher moisture than nodes on the upper layer.

VII. CONCLUSIONS AND FUTURE WORK

In this work, we integrated three commercial sensors with Sensinode's sensor platform to measure four environmental key variables in greenhouse control. The system feasibility was verified in a simple star topology setup in a tomato greenhouse. We achieved up to 10 meter communication range with tolerable 5% packet loss. Because of the high humidity and dense tomato growth, the reliable communication range was reduced to one third of the respective communication range in open space. The measurements also indicated that the system is able to detect the local differences in the greenhouse environment, such as different climate layers which exist from greenhouse bottom to the top.

High moisture forced to consider the possible damages and to protect sensitive boards carefully. When running the experiments, another board damaging factor was noticed. The pollen from the tomato flowers colored one of the black plastic boxes yellow. Small particles of the pollen could also block the measuring component of the sensors, affecting the measuring results.

Applied 15 seconds wake periods between 4 min 15 s sleep periods fulfilled the requirements of the energy-efficient wireless sensor network architecture. Each sensor node was receiving and sending packets in its own turn according to the polling of the coordinator node. The sleep time of the node was 93.75%, which could be increased over 97.50% by shortening the operation time from 15s to five seconds.

Sensors were turned on all the time. Both, SHT75 humidity/temperature sensor and TSL262R light irradiance sensor are suitable for the low power nodes. Especially, the SHT75 with low current sleep mode and accurate sensors is well suitable for wireless sensor nodes powered by batteries. Irradiance sensor does not have the sleep mode at all, and to save energy it have to be turned off most of the time.

In the nearby future, we will develop a multi-hop network to cover the entire greenhouse. We will also attach probes to the nodes so that the wireless nodes can be used to measure soil moisture and possibly other parameters from the flower pots, but still be flexibly moved with the pots or from one pot to another. We are also considering the option to implement the CO2 sensor to the network by connecting it to the plug-in router node.

In addition to networking in data collecting purposes, we will develop the control part and close the wireless control loop. The control commands will be counted in a centralized or locally centralized manner, and then transmitted wirelessly to the actuators located to the different parts of the greenhouse. Required local control implementations suggest us to use DSP-units with some of the wireless sensor nodes.

REFERENCES

[1] Sensinode (2007). OEM Product catalog. [Online]. Available: http://www.sensinode.com/pdfs/sensinode-catalog-20071101.pdf.

[2] G. J. Timmerman and P. G. H Kamp, "Computerised Environmental Control in Greenhouses," PTC, The Netherlands, Page(s): 15–124, 2003.

[3] Greenhouse guide. (Referred 20.04.2008). [Online]. Available: http://www.littlegreenhouse.com/guide.shtml.

[4] M. Mancuso and F. Bustaffa, "A Wireless Sensors Network for Monitoring Environmental Variables in a Tomato Greenhouse," presented at 6th IEEE International Workshop on Factory Communication Systems in Torino, Italy, June 28-30, 2006.

[5] H. Liu, Z. Meng and S. Cui, **"** A Wireless Sensor Network Prototype for Environmental Monitoring in Greenhouses," presented at Wireless Communications, Networking and Mobile Computing 2007 (WiCom 2007), International Conference on 21-25 Sept. 2007 Page(s): 2344 – 2347.

[6] M. Åberg Secher, "Kasvihuone," Otava, Helsinki, Finland, Page(s): 25–80, 1998.

[7] Martens Greenhouse Research Center Web-page, http://www.martens.fi (referred 17.5.2008).

[8] G. Montenegro and N. Kushalnagar, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks," Internet-Draft, IETF, Septemper 2007.[Online]. Available: http://www.ietf.org.

[9] Sensinode (2007). NanoStack manual v1.0.1. [Online]. Available: www.sensinode.com.

[10] Sensirion (2007). SHT1x / SHT7x Humidity & Temperature Sensor v.3.0.1.[Online].Available:http://www.sensirion.com/en/pdf/product_information /Data_Sheet_humidity_sensor_SHT1x_SHT7x_E.pdf.

[11] Texas Advanced Optoelectronic Solutions Inc. (2003). TSL260R, TSL261R, TSL262R Light to voltage optical sensors. [Online]. Available:http://www.roborugby.org/docs/Taos-TSL260R.pdf.

[12] Figaro engineering inc. (2003). TGS 4161 - for the detection of Carbon Dioxide.[Online].Available:http://www.figarosensor.com/products/4161pdf.

[13] PRIVA - Greenhouse Environmental Control Systems. [Online]. Available: http://www.priva.ca/.

IEEE Access
practical innovations : open solutions

Received September 9, 2013, accepted October 8, 2013, date of publication October 25, 2013, date of current version November 1, 2013.

Digital Object Identifier 10.1109/ACCESS.2013.2287302

Localization Services for Online Common Operational Picture and Situation Awareness

MIKAEL BJÖRKBOM[1], JUSSI TIMONEN[3], HÜSEYIN YIĞITLER[1], OSSI KALTIOKALLIO[1], JOSÉ M. VALLET[1] GARCÍA, MATTHIEU MYRSKY[1], JARI SAARINEN[1], MARKO KORKALAINEN[4], CANER ÇUHAC[2], RIKU JÄNTTI[1], REINO VIRRANKOSKI[2], JOUKO VANKKA[3], AND HEIKKI N. KOIVO[1]

[1]School of Electrical Engineering, Aalto University, Aalto FI-00076, Finland
[2]Department of Computer Science, University of Vaasa, Vaasa 65200, Finland
[3]Department of Military Technology, National Defence University, Helsinki 00870, Finland
[4]VTT Technical Research Centre of Finland, Espoo 02044, Finland

Corresponding author: M. Björkbom (mikael.bjorkbom@aalto.fi)

ABSTRACT Many operations, be they military, police, rescue, or other field operations, require localization services and online situation awareness to make them effective. Questions such as how many people are inside a building and their locations are essential. In this paper, an online localization and situation awareness system is presented, called Mobile Urban Situation Awareness System (MUSAS), for gathering and maintaining localization information, to form a common operational picture. The MUSAS provides multiple localization services, as well as visualization of other sensor data, in a common frame of reference. The information and common operational picture of the system is conveyed to all parties involved in the operation, the field team, and people in the command post. In this paper, a general system architecture for enabling localization based situation awareness is designed and the MUSAS system solution is presented. The developed subsystem components and forming of the common operational picture are summarized, and the future potential of the system for various scenarios is discussed. In the demonstration, the MUSAS is deployed to an unknown building, in an ad hoc fashion, to provide situation awareness in an urban indoor military operation.

INDEX TERMS Localization, mapping, networks, situation awareness.

I. INTRODUCTION

Urban situation awareness and especially localization information is important in many applications. Operations, such as search-and-rescue, military operations, urban combat, hostage situations, emergency situations, indoor fire, or earthquake damaged buildings, rely on localization information, as the map of the environment and location of targets in a possible unknown area is needed. Combining information from several subsystems is a key aspect in these perilous applications. Knowing where things are and combining several sources of information, enables context aware data gathering, analysis and decisions, and aid in situation awareness.

In this paper, a novel solution is presented, called Mobile Urban Situation Awareness System (MUSAS), which is an integrated system that provides localization services of several types to enable situation awareness with focus on an urban environment. The target use of the proposed MUSAS is an operation in an urban environment where locations of own field team members, persons and objects are of key importance. The operation environment is typically partly unknown, which require mapping and localization of objects.

A general use case scenario for the MUSAS is an operation in an urban environment as shown in Fig. 1. A field team performs some task based on the instructions from the mission leader and upper echelon. A common operational picture (COP) [1] of the situation is formed, by the COP server and the MUSAS operator managing the system, using data from several subsystems deployed in the field. The COP is relayed to all the parties involved: the field team, mission leader, and upper echelon, to assist them in performing their tasks. Field team members have a hand-held device for interfacing with the COP. The COP contains information of locations of objects and targets of the task, typically humans, overlaid on a map of the environment, to assist in situation awareness.

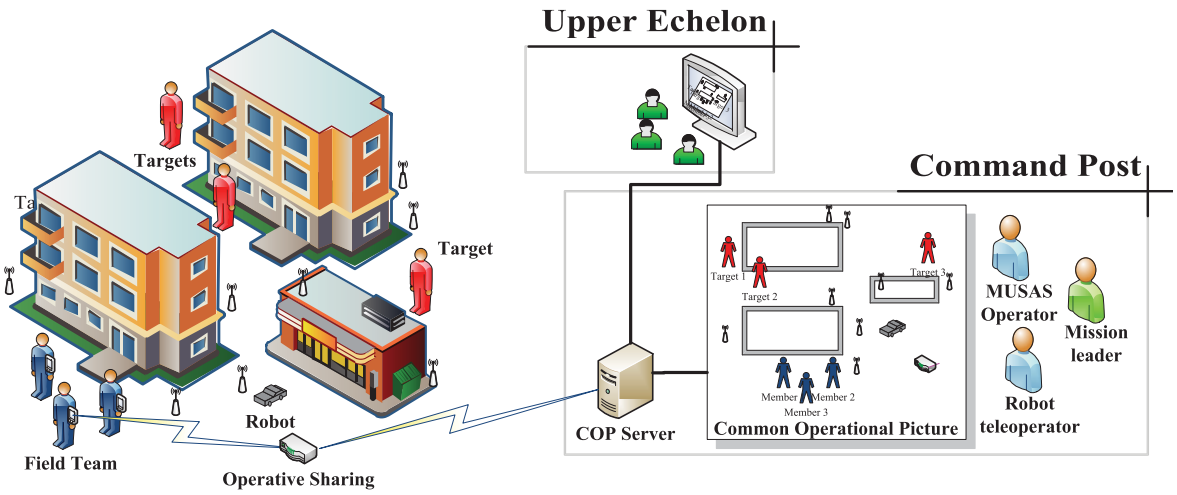


FIGURE 1. General use case example for the MUSAS and entities involved.

A. OBJECTIVES AND CONTRIBUTIONS

A key contribution of the MUSAS is providing a system for online localization based situation awareness using multiple localization and mapping methods. Compared to other similar systems, the MUSAS does not assume or rely on anything of the target environment. The MUSAS builds up its own infrastructure using Wireless Sensor Network (WSN) and Wireless Local Area Network (WLAN) technologies. It maps the unknown area and updates the knowledge as entities are localized. Location information of moving targets is tracked and updated to the COP model and all users. The system can operate both outdoors and indoors and has through wall observation capabilities.

The contributions of this work include describing the general design of an online system for producing and integrating information for a common operational picture, based on mapping an unknown environment and appending several localization information sources. A survey of existing solutions and relevant technologies are done. The subsystems are presented, including their technical details and relevant literature. An implementation is presented and the experiences from a test demonstration are discussed. Other issues related to situation awareness, such as data associating and clustering, object recognition and feature extraction, target identification and tracking, and prediction, are not considered.

In this section, the objectives of the MUSAS, related situation awareness solutions, and the contributions of this paper are described. In Section II, a general system description of a localization based situation awareness system is done, and feasible localization solutions suitable for the use case scenario are identified. The proposed MUSAS architecture and an overview of the implementation are presented in the following sections. In Section III, the robot system is described, including mapping an unknown area using simultaneous localization and mapping (SLAM) by the robot. In Section IV, the localization subsystems are described in more detail with the information they produce for making the COP. Wireless sensor node localization is treated in Section IV-A. Object localization has also been implemented, both for cooperating objects or persons, in Section IV-B, and for non-cooperating persons. For the non-cooperating case, radio tomography can be used, as presented in Section IV-C. In Section V the experiences from a test deployment and the use of the MUSAS in urban combat situations is described. Finally a short conclusion is given with some notes on the use of such systems in other scenarios. A technical report of the system with more detailed information on the implementation can be found in [2].

B. COMMON OPERATIONAL PICTURE

According to [3], situation awareness consists of several levels. The first level is perception or sensing. In the second level, comprehension is built from the observed data, as meaning is assigned to each piece of information and the relations between the components are inferred. In the third level, the situation implications are projected or predicted into the future. In this work only the first two levels are considered, where data is gathered by several entities and fused to some comprehensible picture of the situation. The task of the user is then to decide actions or predict the future based on the produced situation picture.

A common operational picture displays all gathered and combined data from several sources in a single presentation to the user [4]. The information is merged into a common frame of reference and visualized on a screen from where it is easy to comprehend the current situation. The main task of COP systems is thus to bring together data from different subsystems and present that into an overview for enabling situation awareness of a variety of users and different teams [1].

The early studies of COPs were carried out in the 1980's [4]. A major milestone is the development of a large group display to enable situation awareness in military command posts [3]. COPs have been successfully utilized in situations such as large scale natural disasters [5] and terrorist acts, where COPs have had a large impact on reducing human casualties.

A COP is often associated with geographical data, for instance in a combination with a Geographic Information System (GIS), as typical applications are tied to a possible large geographical area. Available maps, blue prints and floor plans can serve as a backdrop to pin the location based information to real-world coordinates and tie them to the environment.

C. SITUATION AWARENESS

Most of the situation awareness literature concerns military cases. The Joint Vision document from 2001 [6], highlights the importance of information superiority throughout the battlefield. Situation awareness of individual soldiers is an important issue, and different armies around the world are developing their future soldier concepts. The target is to create a soldier, who is not only a warrior, but also an active information creator and consumer. The report [7], summarizes some different programs. For example, the Future Soldier program is an international endeavor, led by the USA, to create a soldier of the 2030 [8]. An example of a networked system of systems is the Future Combat Systems (FCS) which links 18 different systems into an operating entity [9].

The Common Operating Picture Software/Systems (COPSS) for emergency management is presented in [10]. This system supports a four dimensional COP and focuses on Shared Situation Awareness (SSA) and supports multiple information sources. Research on a Small Unit Operations Situations Awareness System (SUO SASS) is presented in [11], which has similar aspects as the MUSAS, in terms of ad hoc networks and location services focusing on soldiers. The use of commercial-off-the-shelf (COTS) products in tactical environments is studied in [12]. Especially, an implementation in Android environment, similar to the MUSAS, is studied in [13].

D. SENSOR NETWORKS FOR EMERGENCY SITUATIONS

There are numerous wireless sensor network solutions envisioned for disaster and emergency situations where an infrastructure for data exchange is not readily available. In such scenarios, a WSN can be deployed in ad hoc fashion and provide the means for information exchange and other sensing purposes.

In disaster scenarios, scalable and heterogeneous network solutions for situation management are required. DistressNet [14] provides such a solution and it offers: ad hoc wireless architectures for communication, data exchange to improve situation awareness, and collaborative acoustic sensing for human detection. The system has also multiple solutions for localizing the nodes with the purpose of topology-aware routing and congestion control.

The VigilNet [15] system targets military surveillance, exploiting sensor networks to track targets in areas of interest. The authors consider the setup and operation requirements of the network. In addition, the importance of node localization is considered and Global Positioning System (GPS) is used to fulfill the task. VigilNet targets long term operation and thus energy constraints have an essential role in the system design. On the contrary, the system presented in this paper is deployed for short time intervals and therefore, energy consumption of the nodes does not have to be considered in the system design.

Diamond and Ceruti [16] discuss a military COP model and system architecture for modern warfare. The use of commercial and COTS wireless devices, the diverse sensing possibilities of the devices, and data fusion of different information are seen as effective ways to improve situational awareness for military purposes. Such augments in situational awareness enable new combat paradigms for modern warfare. In contrast to the hypothetical investigation of [16], an actual implementation is presented in this paper.

E. MAPPING AND SEARCH-AND-RESCUE ROBOTICS

Reconnaissance and mapping of an unfamiliar area using a mobile robot, discussed in more detail in Section III, is indispensable, if it is unsafe for humans to enter. Mapping is needed to be able to navigate, operate, and localize the sensed information. The mapping of damaged buildings in an earthquake situation using both ground and aerial robots is presented by [17], where the mapping results of several robots are combined to produce a three dimensional map. Similar robots could be integrated in the MUSAS, with the addition of other subsystems, delivering various other information sources, such as localization of people and objects.

An EC project, Building Presence through Localization in Hybrid Telematic Systems (Pelote) [18], [19], studied the control of a human-robot team in a fire fighting scenario. The proposed solution consisted of a fire-fighter localization system [20], teleoperated robots [21] and an information fusion scheme to synthesize a common model from acquired data. One of the key contributions of the project was that it was experimentally shown that position information is critical in maintaining common situation awareness among the distributed team.

Similar to the MUSAS, Pelote emphasizes the importance of location based information. However, the MUSAS differs from Pelote in that it does not assume a priori information about the target environment. Furthermore, the MUSAS is built upon wireless sensor networks, which extend the range of applicable use case scenarios and enable new positioning possibilities, such as non-cooperative device free localization (DFL).

II. SYSTEM ARCHITECTURE

The target of the MUSAS is to provide a common operational picture for the command post, to the field team and share it

with the upper echelon. This is accomplished by combining information from several different subsystems into a single view. In this section, the general system design and components of the implemented MUSAS is presented. This section is concluded by a description on how the COP information is distributed and presented to the user to aid in situation awareness.

A. GENERAL SYSTEM OVERVIEW

A common operational picture is the visual representation of the up-to-date state of the operation. In this case, focusing on localization information of each entity. The COP includes, but is not limited to, the positions of targets and field team members, and the status of individual assets with respect to a common frame of reference, i.e. a map of the area.

To achieve a COP, information from several online localization systems, backdrop information, such as geographical data and operative information, must be integrated and distributed to all users, as summarized in Fig. 2. The COP server forms the COP model based on inputs provided by all subsystems, and it shares the resultant model with the upper echelon and with the field team using the operative sharing subsystem. The backdrop information subsystem provides basic information related to the operation and the environment. Based on the localization systems, online situation and localization information are formed, and updated the COP model to the current state of the situation. The operative sharing subsystem allows transferring and displaying the generated COP model to the field team, and conveying status updates from the field team to the COP server. Similarly, the upper echelon subsystem provides means for conveying the COP model to the command post, and delivers executive commands to the COP server.

B. INFORMATION SHARING AND INTEGRATION

The COP model must support integration of data gathered from multiple sources. In the MUSAS, various types of information are provided by different subsystems such as mapping information from the robot, and position based content from the team member and target localization subsystems as shown in Fig. 2. Transferring information from

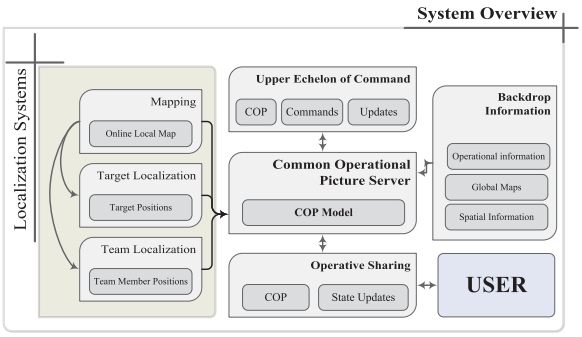


FIGURE 2. General localization based common operational system overview.

the individual subsystems to the COP server and sharing the up-to-date COP model with the upper echelon and users requires a sophisticated networking paradigm. The networking demands can be conveniently fulfilled by abstracting the network away and utilizing a distributed object system architecture. This solution abstracts the underlying technologies to independent functional entities and the integration of the subsystems is done by using a common data sharing framework.

Interactions among distributed object systems is generally enabled by utilizing object-oriented middleware, such as Common Object Request Broker Architecture (CORBA), Remote Method Invocation (RMI) [22] and Internet Communication Engine (ICE) [23]. Middleware, such as CORBA and ICE simplify the development of a distributed system. In addition, they allow independent development efforts of the subsystems, as they support multitudes of operating systems and programming languages.

Considering the diverse requirements of the MUSAS subsystems and the time constraints of data integration and sharing, ICE emerges as the best alternative. This particular middleware architecture is augmented by several services, including a publisher-subscriber topic based event distribution system called IceStorm. Using the IceStorm service, information exchange among the subsystems can be implemented as asynchronous event invocations in topic subscribers. The COP server and subsystems are thus interfaced by abstracted topics defined in and managed by IceStorm.

A fundamental need for a system supporting spatial situation awareness is a subsystem for binding the information from various sources to real world locations. Part of the integration process is associating and combining the position information from the individual localization subsystems to geographic information. Geographic layers, such as maps and blueprints, provide a global coordinate system for the various subsystems. Thus, the location information of the subsystems is inserted into the COP model and delivered to the users in conjunction with the geographic information.

Geographic Information System is one of the well-studied comprehensive solutions, which offers a closed infrastructure and a variety of functions for this purpose. GIS provides means to present the information in layers to aid visual cognition. Further, GIS offers a framework for integrating positioning information generated by the other localization subsystems. By using this framework, the COP server is able to increase the abstraction level of individual objects. The information of individual subsystems is not anymore an object with x- and y-coordinates that are bound to its local coordinate system. Rather, it has a location based on real world coordinates and a certain type, symbol and additional information provided by the COP model.

C. SYSTEM TECHNOLOGIES AND OPERATION

The selected technologies for each subsystem in the MUSAS are depicted in Fig. 3, with brief motivation of the selections in this subsection. Further details are given in Section IV.

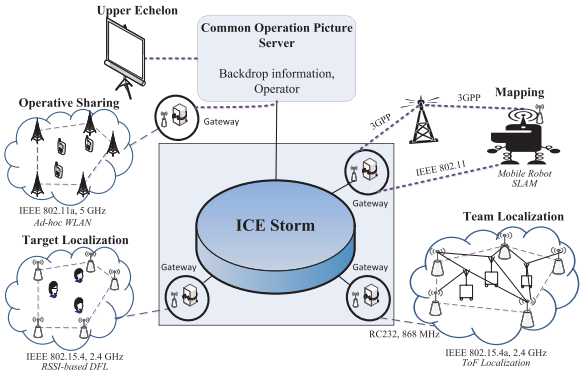


FIGURE 3. The MUSAS system implementation and the utilized technologies.

A common operational picture requires an accurate and up-to-date map of the operating environment, with a common notion of reference and direction. Since, in most of the considered scenarios, this knowledge is not available a priori, a mobile robot, which can generate the map while localizing itself is the most suitable solution among the alternatives, as demonstrated in Pelote [19]. Thus, in the MUSAS, a mobile robot, which is capable of simultaneous localization and mapping, is utilized to generate the map of the environment.

Wireless Sensor Networks can be successfully used to measure spatially distributed data, as a large number of nodes can be distributed in the area of interest. Therefore, an ad hoc WSN is a suitable solution employed in the MUSAS, where relying on existing infrastructure is not possible, due to several reasons, such as damaged and potentially unreliable existing systems.

The target and team localization subsystems aim at estimating the location of assets in the monitored environment. Despite the fact that both systems can be implemented based on visual or radar sensors, the limitations imposed by the cluttered environment and the cost, leverage radio based localization systems. Therefore, the proposed system is built on top of low-cost wireless networks.

An IEEE 802.15.4 network is employed for localizing non-cooperative targets using radio tomographic imaging [24], [25]. The IEEE 802.15.4 nodes are localized using the mobile robot to enable ad hoc deployment. The robot is connected by a versatile multi-radio gateway to support remote operation. For localizing team members, wearable sensors based on IEEE 802.15.4a, time of flight, and inertial sensors are used. To share the COP information to the users, an ad hoc IEEE 802.11a network is used. Gateways for each network are connected together with a wired local area Ethernet network and the ICEStorm publish-subscribe service is used to pass information to the COP server and the other subsystems.

The proposed solution is composed of different wireless communication technologies, some of which may operate on the same frequency band. Therefore, to not interfere with one another, the medium access of these technologies must either be synchronized, or they must be operated in non-overlapping frequencies. In the proposed system, the latter is mostly employed. Subsystem with overlapping frequencies, communicates in turns.

Many of the localization subsystems utilize location information from the other subsystems as depicted in Fig. 4. As an example, radio tomographic imaging requires that the location of the nodes are known. However, in most of the considered use-case scenarios, the node locations are not known a priori. One solution to this problem is to use the robot as a mobile beacon to locate other nodes of the network as described in Section IV-A. Another solution is to equip the robot with a node deployment system and distribute the nodes in desired positions as the robot explores the environment. It is to be noted that these two solutions are not complementary and can be used side by side. In the MUSAS, both options are utilized.

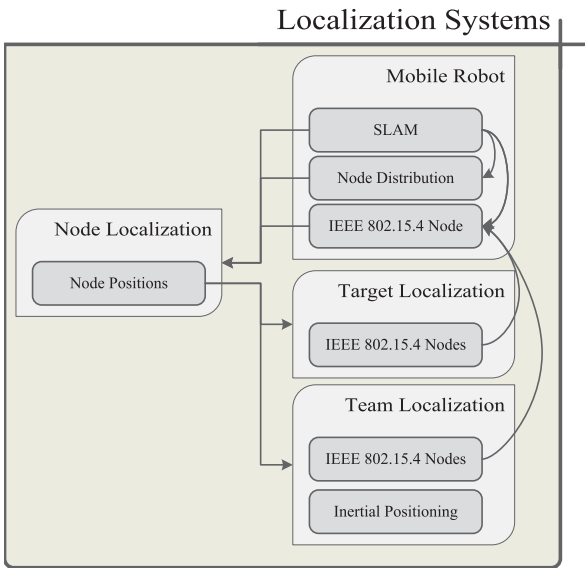


FIGURE 4. Localization systems information flow.

D. COMMON OPERATIONAL PICTURE SERVER

The main task of the COP server is to produce the COP model, which includes all information that is significant for supporting situation awareness. The COP server encapsulates multiple functions, such as hosting relevant backdrop information, geographical information system, as well as publishing the formed COP. These entities are presented in Fig. 5, which is a detail view of the COP server block in Fig. 2. The COP server hosts also multiple services needed by the system, such as information sharing and operative sharing services. A command and control application is running as a front end application for the COP server, which provides a user interface for the command post operator.

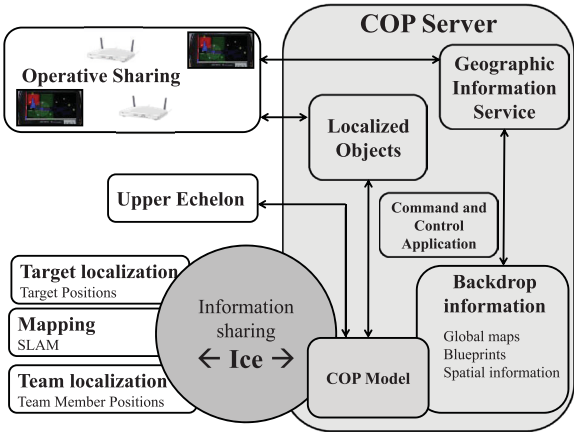


FIGURE 5. COP server framework.

E. PRESENTATION

The COP is presented to the mission leader and upper echelon on a large group display, whereas the field team members are shown a scaled down version in a hand-held device. In either case, a user can zoom in and inspect detailed information associated with a region or object of interest.

The COP presented to the mission leader and the MUSAS operator is shown in Fig. 6. In the depicted scenario, the robot is heading forward in a corridor of an unknown building, simultaneously updating a SLAM generated map. The MUSAS operator identifies the blueprint of the environment and marks it appropriately. The color of the rooms can be changed according to the situation. Additionally, rooms, objects and events can be marked with appropriate NATO APP6B symbols and other polygon shapes, all referencing to local coordinates or real world coordinates (MGRS, WGS84). It is also possible to display the map partially transparent on top of a satellite map, to match it with the surroundings. This mode reveals shapes of the terrain and different targets such as monuments hidden in a forest, improving the situational perception.

The mobile application for the field team members, shown in Fig. 7, is created on an Android platform. Android is chosen because it allows easy deployment on new devices using the same operating system, and makes it possible to use a wide range of COTS products. The application is designed to be as simple as possible for a field team member to perceive the current operational picture. The hand-held application contains only a selected set of features which are presented in Table 1. Common use cases are moving the map, zooming the map, and adding a new object. Every feature is available by using only one hand, including opening the carrying pouch, where it is attached on the torso of the field team member.

F. OPERATIVE SHARING

Sharing the COP information to hand-helds of the field team members is accomplished by using a mobile IEEE 802.11a (WLAN) based ad hoc capable, battery powered, access point network. The network, depicted in Fig. 8(a) and Fig. 3, enables flexible deployment and independence from external infrastructure. No special configuration is needed for the network and it acts as a normal WLAN network for the hand-held devices. The access point, pictured in Fig. 8(b), can automatically connect and join to the existing network access points in the field. When deploying the system, it can be placed anywhere, because it is battery driven. Furthermore,

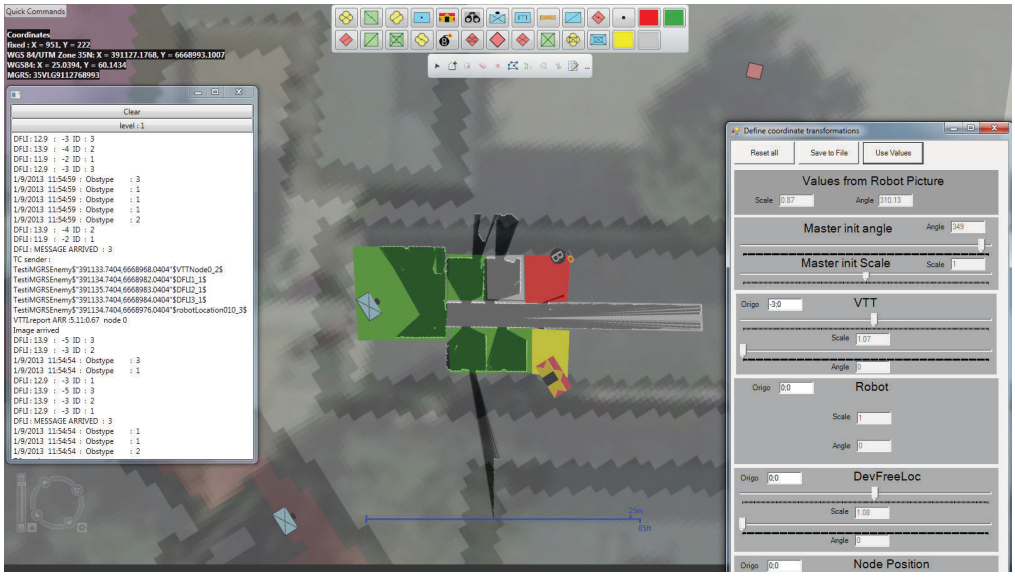


FIGURE 6. Command and control server application user interface.

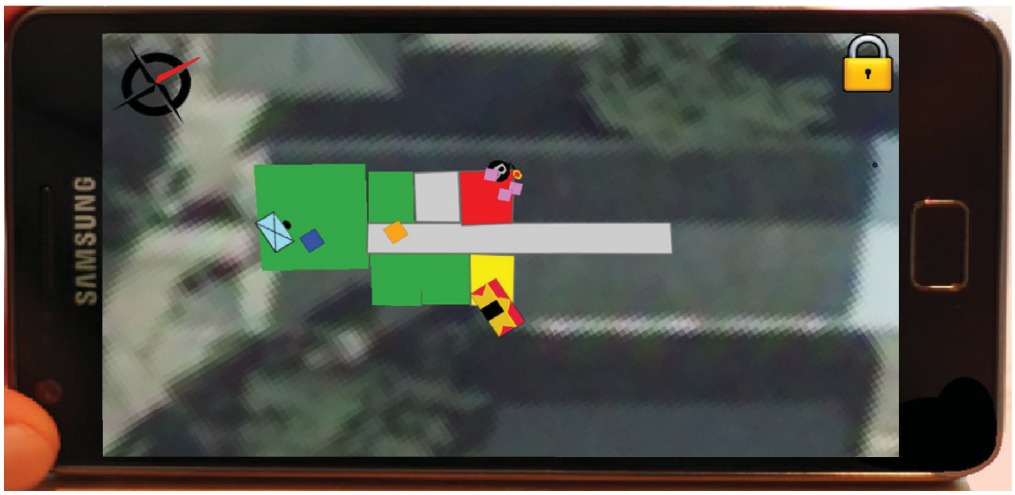


FIGURE 7. Hand-held device graphical user interface for assisting in situation awareness of field team members.

to expand the coverage, existing access points can be moved or new access points can be added.

III. ONLINE INFORMATION ACQUISITION USING A MOBILE ROBOT

To operate in an unknown environment, reconnaissance to collect data and map the area is necessary. The map information, discovered objects, and other information are localized to the local map coordinates and further to global coordinates. For mapping and reconnaissance purposes, the MUSAS uses a mobile robot with SLAM capabilities. In this section, a short description of the mobile robot system is presented. The system components required for control, and how the location information provided by the robot is used in the system, is described.

A. OVERVIEW

A mobile robot features many benefits in the use cases of the MUSAS. Most importantly, it can be deployed to gather information about an unknown situation without risking human lives and the robot is in a central role in creating a common frame of reference for the system.

The remote-controlled robot, shown in Fig. 9, is used as an exploring scout. The robot builds a metric map of the environment while localizing itself against the map. The robot is a tracked platform, weighting approximately 100kg and carries along 100Ah of energy as well as sensors and sufficient computation power. Further details about the robotic system can be found in [26]. In the MUSAS, a laser range finder and dead reckoning for creating the map are used. A camera with a pan-tilt-unit is provided for the teleoperator. In addition, the robot is equipped with a communication subsystem, which enables communication with the robot practically in all environments, without the need for an existing infrastructure.

To build up the localization and sensing infrastructure, treated in Section IV, the teleoperator can deploy wireless sensors into strategic places in the environment, using a wireless sensor node distribution subsystem integrated to the robot. The node deployment is controlled over ICEStorm. Whenever a node is deployed, the information, including the known location of the deployed node, is published to ICEStorm with a timestamp. Further, the robot communicates with the rest of the wireless network, and localizes nodes with unknown positions, deployed by other means, as explained in Section IV-A.

B. COMMUNICATION AND CONTROL

The robot is controlled by teleoperating from the command post. The laser range finder data, the image from the camera, the calculated position and the constructed map of the area is sent to the teleoperation station display shown in Fig. 10. The calculated position of the robot and the constructed map is distributed from the teleoperation station to the COP server by using ICE, as shown in Fig. 3.

As a communication link between the robot and the teleoperation station, two multi-interface routers are used. The routers are especially designed for critical applications where broadband and reliable connectivity and largest possible coverage is needed. They have multiple different kinds of radio terminals, such as 3G HSPA, CDMA450/2000, WiMAX, Wi-Fi, LTE, Flash-OFDM, TETRA (Trans-European Trunked Radio, a radio specifically designed for use by government agencies and emergency services) or satellite, which can be used depending on the situation. The router monitors continuously all installed Wide Area Network (WAN) radios and switches to another radio if one fails or the quality of service is below a user specified threshold. In addition, the routers support virtual private networking, which enables secure and seamless connection, independent of the used radio technology.

TABLE 1. Hand-held device functions.

Feature	Options
Map	Satellite or Raster
User views	Targets, team members, events, rooms (with colors), and compass
Marking objects	Event, Hostile, Unknown, Neutral
Gestures	Pinch - Zoom map Pinch + Rotate - Rotate map Short press - North up Long press - Select satellite or raster map
Compass	Displays North

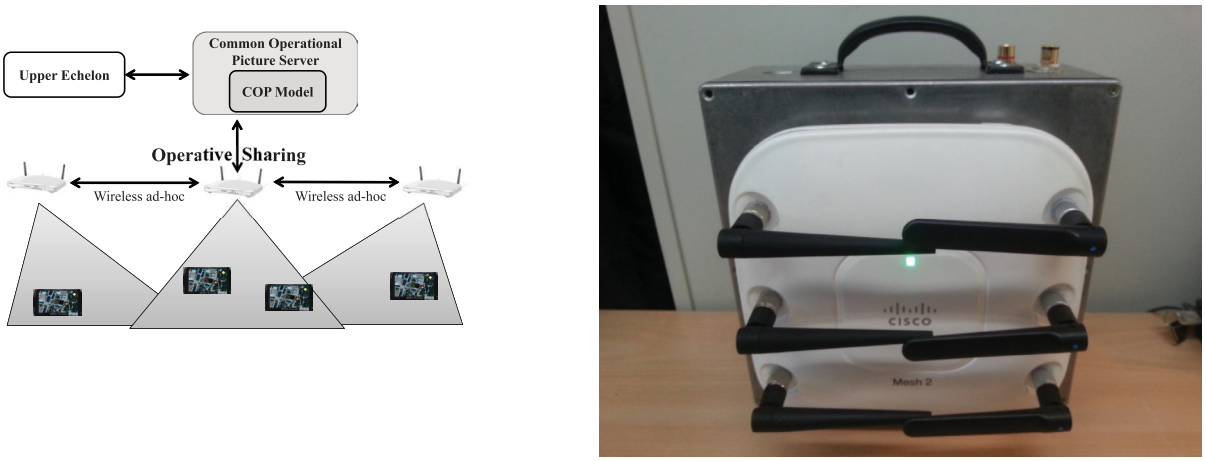


FIGURE 8. Operative Sharing (a) connecting the COP server and the field team member hand-helds using an ad hoc WLAN. (b) WLAN access point with batteries.

As a communication architecture, GIMnet [27], [28], which is a service-based communication middleware for distributed robotic applications, is used. From an application point-of-view, GIMnet provides a virtual private network where all participating nodes may communicate point-to-point using simple name designators for addressing. Using the multi-interface routers and the communication architecture, the system provides the possibility to seamlessly control the robot from virtually any remote location. The setup is mostly the same as in [29].

C. SIMULTANEOUS MAPPING AND TRACKING

Simultaneous localization and mapping, is a well-studied field, and there are several approaches for solving it [30]–[32]. Here, the requirements are to map an arbitrary environment in real time, without changing the frame-of-reference during mapping. Because of these requirements, the problem is approached using a grid-based mapping and tracking (or Maximum Likelihood SLAM) method. The approach incrementally builds an occupancy grid through two steps: 1) Tracking, which maximizes the observation likelihood given the map, and 2) mapping, which fuses the observation with the map into the pose provided by the tracking step. This approach does not employ a loop-closing mechanism, and therefore is referred to as mapping and tracking, in order to distinguish it from a full SLAM solution.

FIGURE 9. The mobile robot unit used for exploring, mapping and node localization in the MUSAS.

The mapping step is a trivial occupancy update step using known pose and laser scanner data with a line model [33]. The tracking step uses a globally optimal search algorithm introduced in [34] for finding the best pose in the map. The search algorithm branches the pose space, with an objective to minimize the point distance to occupied map cells. The solution is bound by using an efficient approximation of the upper and lower bounds of the objective. The algorithm has

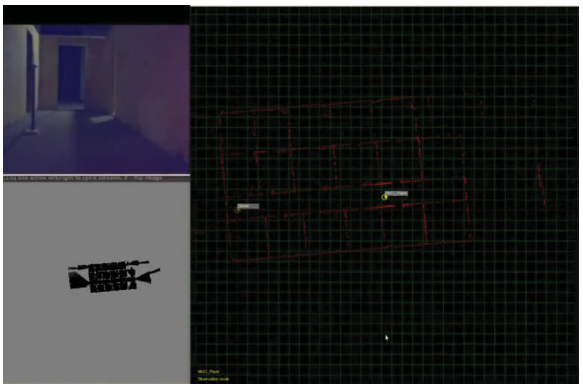


FIGURE 10. Teleoperation view for mobile robot.

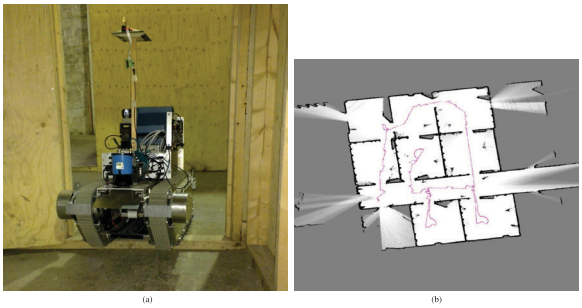


FIGURE 11. (a) The robot in the test environment. (b) An example map from the test scenario.

been shown to provide robust, sub-resolution pose estimates even with very large search spaces [34] and being able to map accurately even in the presence of large loops [29]. In this use-case, the map is incrementally built, and thus the search space is relatively small. The robot mapping and tracking inside the target area is shown in Fig. 11(a).

Fig. 11(b) provides an example map from the test scenario. The map is built in real-time by the robot and shows an exploration through eight rooms. The map is published to the other subsystems using ICEStorm as an image every 10 seconds. The map is then used in the command post and overlaid with the a priori map and global geographical information in the COP server. The map is also provided to the robot operator in order to help in keeping spatially oriented while driving the robot, as shown in Fig. 10. The pose of the robot is published to ICEStorm continuously, for the other subsystems, specifically the robot operator and the node localization system.

IV. SYSTEMS FOR LOCALIZATION

Localization of wireless nodes in Wireless Sensor Networks has been researched extensively, because in spatially distributed systems, sensor data is only meaningful if the location of its origin is known. In the MUSAS, not only node locations are needed, but also locations of field team members, targets and other objects and events, as well as their position in relation to a map. The following subsections briefly present the localization subsystems of the developed MUSAS and explain their technical details and how they produce the required localization information. The interactions of the localization subsystems are described in Section II-C.

A. NODE CALIBRATION AND LOCALIZATION

Due to the ad hoc nature of emergency and rescue situations, the localization systems used in the MUSAS cannot depend on pre-installed infrastructure in the target site. Thus the WSNs used have to be deployed ad hoc. In the most general case, nodes will be placed in random or unknown positions. Once the network has been deployed, the task is then to estimate the position of the nodes, such that the information measured through their sensors can be associated to the known locations.

There are many existing localization methods for WSN [35]. In this work, a maximum likelihood (ML) algorithm based on radial received signal strength (RSS)-distance models is used. Using RSS as a primary source of information for localization has advantages and drawbacks. First, the circuitry to measure RSS is low-cost and most of the radio chips on the market provide an RSS indicator. On the other hand, RSS can be significantly affected by obstacles, and as a consequence, localization using RSS is known to be considerably inaccurate in cluttered environments. However, this sensitivity can be exploited to detect and track objects or persons by monitoring changes in the RSS as is done in Section IV-C. Thus, the same source of information can be used to both locate nodes and track people.

In contrast to RSS-distance model based methods, time based methods using radio signals, such as ultra wide band radios, are less sensitive to the presence of obstacles and gives more accurate position estimates [36]. However, they require expensive circuitry to measure time. Additionally, ranging using time based methods requires dedicated time slots, which can be a limiting factor for tracking [37].

In order to effectively localize the nodes deployed in unknown positions, the MUSAS uses the robot as a mobile beacon. While the robot is exploring the environment, it is communicating with the nodes of the WSN. The robot position is known at all times, and therefore every RSS measurement can be associated to a unique beacon position. Each of the measurements can then be thought of as coming from a fixed beacon placed at the position of the robot at the measurement instant [38]. The advantage of a moving beacon with respect to a limited number of fixed beacons, is that the amount of measurements can be much larger and richer, which allow the localization algorithms to produce more accurate position estimates.

The performance of the localization algorithm depends strongly on the ability of the model to make good predictions of the RSS. In cluttered environments, the RSS can vary significantly, and thus the RSS is modeled as a random

variable. Perhaps the most used RSS-distance model is the log-normal model, which describes the RSS as a normally distributed variable with a mean, decaying proportionally to the logarithm of the distance and with a variance characterizing the variability of the observed RSS [39]. The decaying factor and the standard deviation are well known to depend strongly on the particular environment, and need thus to be estimated. However, the local inhomogeneity of the environment and the hardware differences among the nodes influence significantly the model parameters, which in turn have a strong negative effect on the localization accuracy [40]. Thus, instead of using one model for all the nodes, each node has its own model whose parameters are tuned specifically for that node and the environment.

Because the MUSAS is designed for ad hoc situations, it is not possible to assume the availability of models calibrated a priori or to calibrate the models before the operation. Therefore, algorithms that calibrate the model simultaneously as the node locations are being estimated are needed.

The problem of simultaneous node localization and model parameter estimation can be posed using ML or least-squares (LS) principles, leading in general to a nonlinear optimization problem. The problem can then be solved using any standard nonlinear optimization techniques, such as grid based or Newton-Raphson based. When using the log-normal model, the dependency on the model parameters is linear. Recognizing that the ultimate goal is position estimation, the model parameters can be seen as nuisance parameters, which can be eliminated using the principle of separable least squares [41]. Thus, the search space is reduced to the coordinates of the nodes.

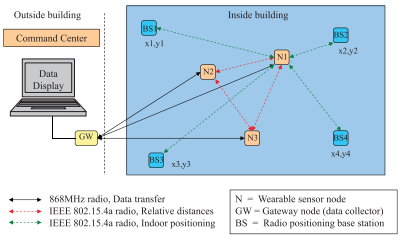
Another conceptually simple approach for simultaneous localization and model calibration is a recursion consisting of 2 steps: starting from an initial guess on the model parameters, first estimate the positions of the nodes. Then, using the estimated positions, re-estimate the model parameters, and start the cycle again. This idea has been proposed in [42] using fixed beacons. In [38] the same principle is exploited using a robot as a mobile beacon to locate the nodes of a WSN in three different environments. With the system used in the MUSAS, a mode localization accuracy of 47 cm was achieved in a large uncluttered space and approximately 1 meter accuracy in a semi-open lobby and a typical office environment [38].

B. TEAM LOCALIZATION SYSTEM

During operation, it is beneficial to know where own team members are located at any given time. This information can be used in operative planning and execution to increase effectiveness and direct the operation where necessary. For the MUSAS, a team localization system exploiting wearable sensors is developed to produce location information of own team members. In addition, the developed system also provides information about the physical state of the person who is wearing the sensor.

Localization of people has been studied extensively, and various different technologies have been proposed [43]–[47]. Commercially ready solutions for outdoor localization already exist such as GPS and GLONASS. On the contrary, indoor localization is more challenging since line-of-sight to GPS satellites is not available and readily available solutions fulfilling the MUSAS requirements do not exist. In most use-case scenarios of the MUSAS, the team operates both indoors and outdoors. Therefore, the proposed system is designed to have a set of complementary positioning technologies that enable localization in versatile urban environments.

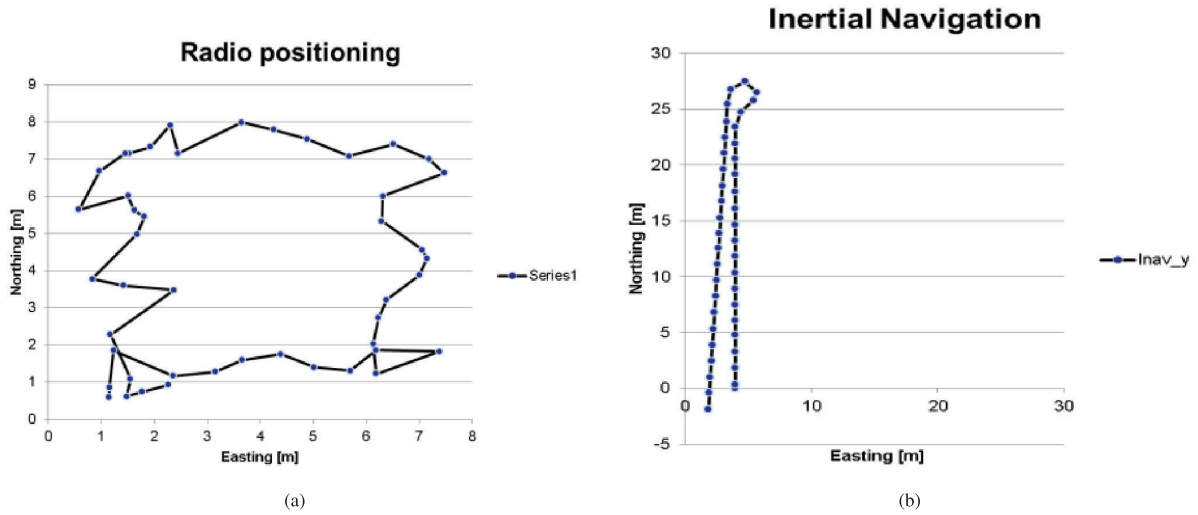
The developed system is based on wearable wireless sensor nodes, which are installed on the clothing and equipment of the team members. Outdoors, the location estimates are provided by GPS. Indoors, the localization is carried out by exploiting either inertial navigation, radio based solutions or both simultaneously. Physical condition monitoring is



(a)

(b)

FIGURE 12. **(a) The architecture of the team localization system. (b) Wearable sensor node installation on a soldier.**



(a) (b)

FIGURE 13. **(a) Radio and (b) Inertial navigation in deployment environment.**

implemented by an inertial based activity recognition algorithm that is able to classify some common activities during operation such as: walking, standing, ascending or descending stairs. The algorithm provides the general intensity level of the current activity.

The team localization system, shown in Fig. 12(a), uses inertial navigation and radio based ranging for localization in indoor environments. Ranging is optional and utilized only if radio positioning base stations are deployed in the environment. Each wearable sensor node has an embedded microcontroller based computing unit for running the localization algorithms, radios for data transmission and ranging, and an IMU (Inertial Measurement Unit) with a 3D gyroscope, magnetometer and acceleration sensors for inertial navigation. The wearable sensors are installed on the back of the person as shown in Fig. 12(b), the antennas and IMU on the shoulders, and the acceleration sensors are placed on the right and left boots. Nanotron 2.4 GHz, IEEE 802.15.4a short range radios are used for radio based ranging and relative distance measurement between team members. Wireless communication with the MUSAS is performed using the RC232, 868 MHz RC1180HP long range radios. The wearable sensors are described in more detail in [48].

Inertial navigation of the system is based on estimating the step length using acceleration data gathered from the boots. This information is combined with heading information provided by the gyroscope and magnetometer. Radio based localization relies on time-of-flight (TOF) based distance measurements to fixed base stations, with known locations.

Both localization methods have been implemented separately in the proposed system. The accuracy of the radio-based localization system depends on the used positioning algorithm and the operating environment. Highest accuracy is achieved in unobstructed environments and in line-of-sight conditions. The accuracy decreases in cluttered environments where multipath propagation is common. Inertial navigation is bound to drift during operation and needs regular position and heading corrections. Radio based positioning does not drift, and in future developments, the inertial navigation drift will be compensated by data fusion algorithms taking benefit of GPS or radio based positioning estimates, when available.

Fig. 13 shows some test results gathered during the deployment. Using radio positioning, a test walking trip is done near the walls inside a room approximately $90m^2$. The radio positioning base stations are installed at the corners of the room. The radio based system is capable of localizing a person with an accuracy of 2 m. In the inertial navigation test, a back and forth route was done in a corridor. In the activity recognition test, a stair case was walked, first downstairs and returning back to the start position, as indicated in Fig. 14. The activity recognition algorithm classifies different types of activities. The current type of activity is indicated in color in the end user interface.

C. DEVICE FREE LOCALIZATION

The MUSAS requires localizing targets in the operation area, rendering a need for utilizing a non-cooperative positioning technology that can operate in various ambient conditions. Device-free localization (DFL) is an emerging technology based on RSS measurements of a dense wireless network. DFL fulfills the target localization requirements of the MUSAS, since it is independent of ambient conditions such as lighting, temperature, humidity, etc., it can operate in

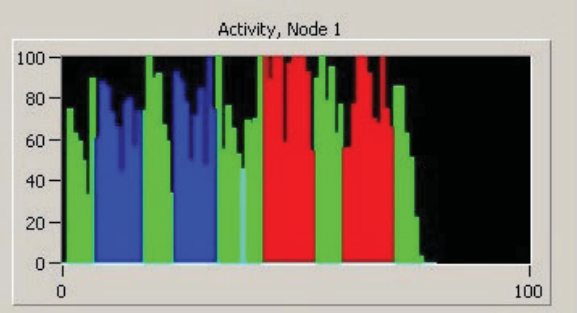


FIGURE 14. Activity recognition test results (green=level walk, blue=descending the stairs, red=ascending the stairs).

obstructed environments, and it can be used in through-wall scenarios. Most notably, this technology does not require that the targets to be localized carry any device.

DFL is based on the fact that wireless communication is affected by people [49], [50], which can be observed in RSS measurements of low-cost wireless devices [51]. Generally, a change in RSS is observed when the link line of two communicating nodes is blocked. Further, the presence of a person causes correlated changes in nearby links, enabling a collaborative localization effort. Since the radio is used for extracting localization information, these systems are referred to as radio frequency (RF) sensor networks [52].

One approach to RSS-based DFL is to estimate the changes in the propagation field of the monitored area, and then form an image of this field, a process referred to as radio tomographic imaging (RTI) [24], [25]. The formed image can then be used to infer the locations of people within the deployed wireless network as shown in Fig. 15(a). Use cases of the MUSAS, set strict demands on the used wireless sensor network and the RSS-based DFL system operation. In the following, these demands are addressed and the applied solutions introduced.

A network monitoring and management framework is essential to manage a WSN as argued by Tolle et. al. [53]. In addition, numerous works have shown that the communication conditions vary significantly over time [54], making network management mandatory to ensure functionality in the long-run. Network management serves two purposes in RSS-based DFL: first, the network can be configured easily, reducing the deployment time; second, it offers the possibility to adapt to changing communication conditions, for instance, the network can change the frequency channel of operation if needed. For these reasons, a network monitoring and management framework is designed and utilized for the purpose of the MUSAS [55].

Similarly, as in the case of the node localization system, the locations of the sensors and RSS-based DFL could be calculated simultaneously as proposed in [56]. However, the MUSAS, take advantage of the robot and the proposed solutions in Section IV-A for obtaining the node locations, and then performs DFL using the known positions of the nodes.

Most RSS-based DFL algorithms require that the RSS statistics are known when the link line is not obstructed by a person. In the current case, there is no possibility for empty-area calibration, thus the system must learn the RSS statistics while running and adapt to the changing environment. Several possibilities exist: first, methods that do not require calibration could be applied [57]; second, online algorithms capable of learning the RSS-statistics when the link is not affected by a person could be used [58], [59]; or third, methods for online calibration could be applied [60], [61]. The methods proposed in [61] are used in the MUSAS.

In an urban environment, it is not always possible to deploy sensors inside the same space where the targets are located. Therefore, through-wall localization capability is desired, which is enabled by the RF-based approach. Previous DFL attempts in through-wall scenarios have used variance-based RTI (VRTI) [57], [62]. However, VRTI is not able to localize a stationary target, since it is based on a windowed variance of the RSS. Kernel distance-based RTI (KRTI) has been demonstrated to localize both stationary and moving targets, even through walls of a building [63]. In the MUSAS, the algorithms presented in [64] are exploited, where a multi-scale spatial model and a novel measurement model are utilized. The results demonstrate high accuracy localization (0.3 m) in a through-wall environment as shown in Fig. 15(b).

It is often required to localize and track multiple targets. In [65]–[67], particle filters are used to track multiple targets simultaneously. However, these works assume that the number of targets is known a priori and that the target trajectories do not intersect. These systems struggle also in estimating the locations in real-time, because of the complexity of particle filters. The above drawbacks are addressed in [68], in which machine vision algorithms are adapted for the purpose of imaging-based DFL, and exploited in the MUSAS. The algorithms are able to estimate the number of people correctly 97% of the time. Furthermore, experiments demonstrate that the system is capable of tracking up to four targets with intersecting trajectories with an average error of 0.55 m or lower in a cluttered indoor office.

V. TEST DEPLOYMENT IN AN URBAN HOSTAGE SITUATION

The implemented MUSAS system was tested, demonstrated and evaluated in an urban military training facility at Santahamina, Finland, in November 2012 as described in this section. The experiment was conducted in a testing yard, consisting of a plywood maze for training troops in urban area warfare. A platoon of soldiers, specialized in urban area warfare, served as a field team and as hostile forces, targets. The evaluation case was a hostage situation, where hostile forces and hostages resided in an unknown indoor environment.

In this event, the system formed a COP using a mobile robot, the device-free localization system, and the wearable sensor nodes. The network was built and localized automat-

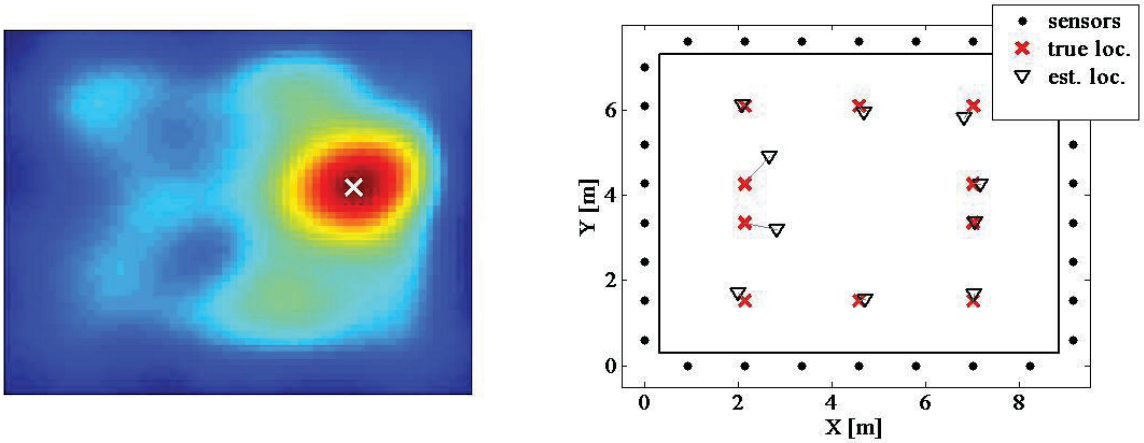


FIGURE 15. **(a) The estimated RF propagation field image. The estimated distribution coincides with the true location of the person (white cross). (b) The position estimates obtained with RSS-based DFL in a through-wall scenario.**

ically as the troops advanced inside the building. During the test a WLAN infrastructure network of approximately 300 × 600 meters was achieved, including the interior of the building, by using only four access points. The soldiers were able to carry the hand-held devices and expand the WLAN network when needed. The robot used two different 3G connections for the remote operator, to ensure connection during the operation. 20 IEEE 802.15.4 sensor nodes where used for DFL. The robot deployed 5 nodes inside the building during the demonstration. Three team localization beacon nodes were used to localize the field team.

The indoor environment map was built online as the robot mapped the building. The MUSAS produced real-time results and delivered information to the field team, including the map, the locations of individual solders and other localized objects and relevant information. In the COP model, rooms were colored red if hostile elements were in a room. After the space was cleared out of danger, the color was changed to green.

Attaching the mobile devices to the soldier's equipment and using it during action were evaluated. Two options were studied: attachment to the left hand (for a right-handed user) and to the upper left torso, using a specific pouch. The first impression was that the hand attachment was better, but the torso attachment proved more reliable. The device is vulnerable when used in the hand, consuming more of the user's attention and also possibly preventing other activities during battle. The torso attachment is slightly more difficult to reach, but, on the other hand, the device is well protected and unobtrusive. After some training, the soldiers got used to carrying and using the device attached to the torso. Later, the mobile device will probably be developed to fit to this attachment more effectively. The soldiers also used glows, specially designed for tactical use with touch screen capability.

During the tests it was recognized that it is inconvenient for the soldier to operate the hand-held device displaying the COP when in action. For this reason, the device was only used for supporting situation awareness, not for active use, such as marking discovered objects to the COP model. During the tests, a short movie was shot [69], which explains the operational concept of the MUSAS. The users gave good feedback about the usability of the mobile devices and also on the speed of the system. In further test the system can be used to evaluate the use of a common operational picture for situation awareness in critical tasks and operations.

VI. CONCLUSIONS

The presented framework provides a novel and scalable solution for creating, hosting and delivering a common operational picture in a multisensory environment focused on localization and position based information in an urban environment. The proposed system is demonstrated by the implemented MUSAS and tested in a realistic urban environment in a military hostage situation.

Compared to other similar systems, the MUSAS focuses on multiple localization services and localized information presentation. The system can be deployed in search-and-rescue and earthquake disaster situations to map the environment and localize people. It has also applications in police hostage situation, indoor fire-fighting scenarios and military operations.

The next step in research is to use a distributed server architecture [70] and distributed computation, to increase modularity and robustness of the overall system. The MUSAS has the architectural solutions which enable distribution of vital services throughout the network and subsystems. Future plans for development include also the implementation of a 3D environment model for localization, as well as improved views for the Android devices.

REFERENCES

[1] M. D. McNeese, M. S. Pfaff, E. S. Connors, J. F. Obieta, I. S. Terrell, and M. A. Friedenberg, "Multiple vantage points of the common operational picture: Supporting international teamwork," in *Proc. 50th Annu. Meeting Human Factors Ergonom. Soc.*, 2006, pp. 467–471.

[2] R. Virrankoski, "Wireless sensor systems in indoor situation modeling II (WISM II)," Dept. Comput. Sci., Univ. Vaasa, Vaasa, Finland, Tech. Rep. 188, 2013.

[3] M. R. Endsley, "Toward a theory of situation awareness in dynamic systems," *J. Human Factors Ergonom. Soc.*, vol. 37, no. 1, pp. 32–64, 1995.

[4] R. S. Hager, "Current and future efforts to vary the level of detail for the common operational picture," Naval Postgraduate School, Monterey, CA, USA, 1997.

[5] A. Deschamps, D. Greenlee, T. J. Pultz, and R. Saper, "Geospatial data integration for applications in flood prediction and management in the Red River Basin," in *Proc. Geosci. Remote Sens. Symp.*, vol. 6. Jan. 2002, pp. 3338–3340.

[6] U. G. P. Office, *Joint Vision 2020*. Washington, DC, USA: Government Printing Office, 2001.

[7] (2012). *Programmes at a Glance* [Online]. Available: http://www.soldiermod.com/volume-10/pdfs/articles/programmes-overview-may-2013.pdf

[8] A. Taylor, *Future Soldier 2030 Initiative*. New York, NY, USA: US Army RDECOMM, 2009.

[9] R. Dietterle, "The future combat systems (FCS) overview," in *Proc. Military Commun. Conf.*, vol. 5. Oct. 2005, pp. 17–20.

[10] R. Balfour, "Next generation emergency management common operating picture software/systems (COPSS)," in *Proc. LISAT*, May 2012, pp. 1–4.

[11] L. J. Williams, "Small unit operations situation awareness system (SUO SAS): An overview," in *Proc. Military Commun. Conf.*, vol. 1. Oct. 2003, pp. 13–16.

[12] V. Kaul, C. Makaya, S. Das, D. Shur, and S. Samtani, "On the adaptation of commercial smartphones to tactical environments," in *Proc. Military Commun. Conf.*, Nov. 2011, pp. 7–10.

[13] N. Suri, L. Pochet, J. Sterling, R. Kohler, E. Casini, J. Kovach, *et al.*, "Middleware, and applications for portable cellular devices in tactical edge networks," in *Proc. Military Commun. Conf.*, Nov. 2011, pp. 7–10.

[14] S. M. George, W. Z. C. H. Myounggyu Won, Y. O. L. A. Pazarloglou, R. Stoleru, and P. Barooah "Distressnet: A wireless ad hoc and sensor network architecture for situation management in disaster response," *IEEE Commun. Mag.*, vol. 48, no. 3, pp. 128–136, Mar. 2010.

[15] T. He, S. Krishnamurthy, L. Luo, T. Yan, L. Gu, R. Stoleru, *et al.*, "Vigilnet: An integrated sensor network system for energy-efficient surveillance," *ACM Trans. Sensor Netw.*, vol. 2, pp. 1–38, Feb. 2006.

[16] S. M. Diamond and M. G. Ceruti, "Application of wireless sensor network to military information integration," in *Proc. 5th IEEE Int. Conf. Ind. Informat.*, vol. 1. Jun. 2007, pp. 317–322.

[17] N. Michael, S. Shen, K. Mohta, Y. Mulgaonkar, V. Kumar, K. Nagatani, *et al.*, "Collaborative mapping of an earthquake-damaged building via ground and aerial robots," *J. Field Robot.*, vol. 29, no. 5, pp. 832–841, Sep./Oct. 2012.

[18] F. Driewer, H. Baier, K. Schilling, J. Pavlicek, L. Preucil, N. Ruangpayoongsak, *et al.*, "Hybrid telematic teams for search and rescue operations," in *Proc. IEEE Int. Workshop Safety, Security, Rescue Robot.*, May 2004, pp. 2–4.

[19] M. Kulich, J. Kout, L. Preucil, R. Mazl, J. Chudoba, J. Saarinen, *et al.*, "PeLoTe—A heterogeneous telematic system for cooperative search and rescue missions," in *Proc. IEEE/RSJ IROS*, Sep. 2004, pp. 1–8.

[20] J. Saarinen, S. Heikkila, M. Elomaa, J. Suomela, and A. Halme, "Rescue personnel localization system," in *Proc. IEEE Int. Workshop Safety, Security Rescue Robot.*, Jun. 2005, pp. 218–223.

[21] N. Ruangpayoongsak, H. Roth, and J. Chudoba, "Mobile robots for search and rescue," in *Proc. IEEE Int. Workshop Safety, Security Rescue Robot.*, Jan. 2005, pp. 212–217.

[22] J. Lee, "Enabling network management using Java technologies," *IEEE Commun. Mag.*, vol. 38, no. 1, pp. 116–123, Jan. 2000.

[23] M. Henning, "A new approach to object-oriented middleware," *IEEE Internet Comput.*, vol. 8, no. 1, pp. 66–75, Feb. 2004.

[24] N. Patwari and P. Agrawal, "Effects of correlated shadowing: Connectivity, localization, and RF tomography," in *Proc. Int. Conf. IPSN*, 2008, pp. 82–93.

[25] J. Wilson and N. Patwari, "Radio tomographic imaging with wireless networks," *IEEE Trans. Mobile Comput.*, vol. 9, no. 5, pp. 621–632, May 2010.

[26] M. Matusiak, J. Paanajärvi, P. Appelqvist, M. Elomaa, M. Ylikorpi, and A. Halme, "A novel marsupial robot society: Towards long-term autonomy," in *Proc. 9th Int. Symp. DARS*, Nov. 2008, pp. 523–532.

[27] J. Saarinen, A. Maula, R. Nissinen, H. Kukkonen, J. Suomela, and A. Halme, "GIMnet—Infrastructure for distributed control of generic intelligent machines," *Provider (Server)*, vol. 586, no. 29, pp. 525–530, 2007.

[28] A. Maula, M. Myrsky, and J. Saarinen, "GIMnet 2.0-enhanced communication framework for distributed control of generic intelligent machines," in *Proc. 1st IFAC Conf. Embedded Syst., Comput. Intell. Telemat. Control*, 2012, pp. 62–67.

[29] M. Myrsky, A. Maula, J. Saarinen, and I. Kankkunen, "Teleoperation tests for large-scale indoor information acquisition," in *Proc. Comput. Intell. Telemat. Control Embedded Syst.*, vol. 1. 2012, pp. 13–18.

[30] M. W. M. G. Dissanayake, P. Newman, S. Clark, H. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localization and map building (SLAM) problem," *IEEE Trans. Robot. Autom.*, vol. 17, no. 3, pp. 229–241, Jun. 2001.

[31] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: Part I," *IEEE Robot. Autom. Mag.*, vol. 13, no. 2, pp. 99–110, Jun. 2006.

[32] T. Bailey and H. Durrant-Whyte, "Simultaneous localization and mapping (SLAM): Part II," *IEEE Robot. Autom. Mag.*, vol. 13, no. 3, pp. 108–117, Sep. 2006.

[33] H. P. Moravec, "Sensor fusion in certainty grids for mobile robots," *AI Mag.*, vol. 9, no. 2, pp. 61–74, 1988.

[34] J. Saarinen, J. Paanajärvi, and P. Forsman, "Best-first branch and bound search method for map based localization," in *Proc. IEEE/RJS Int. Conf. Intell. Robot. Syst.*, Sep. 2011, pp. 59–64.

[35] F. Seco, A. R. Jimenez, C. Prieto, J. Roa, and K. Koutsou, "A survey of mathematical methods for indoor localization," in *Proc. IEEE Int. Symp. Intell. Signal Process. WISP*, Aug. 2009, pp. 9–14.

[36] N. Patwari, A. O. Hero, M. Perkins, N. S. Correal, and R. J. O'Dea, "Relative location estimation in wireless sensor networks," *IEEE Trans. Signal Process.*, vol. 51, no. 8, pp. 2137–2148, Aug. 2003.

[37] G. E. Garcia, L. Muppirisetty, and H. Wymeersch, "On the trade-off between accuracy and delay in cooperative UWB navigation," in *Proc. IEEE WCNC*, Apr. 2013, pp. 1603–1608.

[38] J. Vallet, O. Kaltiokallio, M. Myrsky, J. Saarinen, and M. Bocca, "Simultaneous RSS-based localization and model calibration in wireless networks with a mobile robot," *Proc. Comput. Sci.*, vol. 10, pp. 1106–1113, Aug. 2012.

[39] S. Seidel and T. Rappaport, "914 MHz path loss prediction models for indoor wireless communications in multifloored buildings," *IEEE Trans. Antennas Propag.*, vol. 40, no. 2, pp. 207–217, Feb. 1992.

[40] J. Vallet, O. Kaltiokallio, J. Saarinen, M. Myrsky, and M. Bocca, "On the sensitivity of RSS based localization using the log-normal model: An empirical study," in *Proc. 10th WPNC*, 2013, pp. 1–6.

[41] F. Gustafsson and F. Gunnarsson, "Localization in sensor networks based on log range observations," in *Proc. 10th Int. Conf. Inf. Fusion*, Jul. 2007, pp. 1–8.

[42] R. Zemek, D. Anzai, S. Hara, K. Yanagihara, and K.-I. Kitayama, "RSSI-based localization without a prior knowledge of channel model parameters," *Int. J. Wireless Inf. Netw.*, vol. 15, nos. 3–4, pp. 128–136, 2008.

[43] A. Amanatiadis, D. Chrysostomou, D. Koulouriotis, and A. Gasteratos, "A fuzzy multi-sensor architecture for indoor navigation," in *Proc. IEEE Int. Workshop IST*, Jul. 2010, pp. 452–457.

[44] H. Muller, C. Randell, and A. Moss, "A 10 mW wearable positioning system," in *Proc. 10th IEEE ISWC*, Oct. 2010, pp. 47–50.

[45] S. Holm, "Hybrid ultrasound–RFID indoor positioning: Combining the best of both worlds," in *Proc. IEEE Int. Conf. RFID*, Apr. 2009, pp. 155–162.

[46] R. Tenmoku, M. Kanbara, and N. Yokoya, "A wearable augmented reality system for navigation using positioning infrastructures and a pedometer," in *Proc. 2nd IEEE/ACM ISMAR*, Oct. 2003, pp. 344–345.

[47] S. Lee, B. Kim, H. Kim, R. Ha, and H. Cha, "Inertial sensor-based indoor pedestrian localization with minimum 802.15.4a configuration," *IEEE Trans. Ind. Informat.*, vol. 7, no. 3, pp. 455–466, Aug. 2011.

[48] M. Korkalainen, P. Tukeva, M. Lindholm, and J. Kaartinen, "Hybrid localization system for situation awareness applications," in *Proc. 3rd WOWCA*, Apr. 2012.
[49] H. Hashemi, "The indoor radio propagation channel," *Proc. IEEE*, vol. 81, no. 7, pp. 943–968, Jul. 1993.
[50] H. Hashemi, "A study of temporal and spatial variations of the indoor radio propagation channel," in *Proc. 5th IEEE Int. Symp. Pers., Indoor Mobile Radio Commun., Wireless Netw., Catch. Mobile Future*, vol. 1. Sep. 1994, pp. 127–134.
[51] K. Woyach, D. Puccinelli, and M. Haenggi, "Sensorless sensing in wireless networks: Implementation and measurements," in *Proc. 4th Int. Symp. Model. Optim. Mobile, Ad Hoc Wireless Netw.*, 2006, pp. 1–8.
[52] N. Patwari and J. Wilson, "RF sensor networks for device-free localization: Measurements, models, and algorithms," *Proc. IEEE*, vol. 98, no. 11, pp. 1961–1973, Nov. 2010.
[53] G. Tolle and D. Culler, "Design of an application-cooperative management system for wireless sensor networks," in *Proc. 2nd Eur. Workshop Wireless Sensor Netw.*, 2005, pp. 121–132.
[54] A. Cerpa, J. Wong, L. Kuang, M. Potkonjak, and D. Estrin, "Statistical model of lossy links in wireless sensor networks," in *Proc. 4th Int. Symp. Inf. Process. Sensor Netw.*, 2005, pp. 81–88.
[55] H. Yigitler, O. Kaltiokallio, and R. Jäntti, "A management framework for device-free localization," in *Proc. IEEE IJCNN*, Aug. 2013.
[56] X. Chen, A. Edelstein, Y. Li, M. Coates, M. Rabbat, and A. Men, "Sequential Monte Carlo for simultaneous passive device-free tracking and sensor localization using received signal strength measurements," in *Proc. 10th Int. Conf. IPSN*, 2011, pp. 342–353.
[57] J. Wilson and N. Patwari, "See-through walls: Motion tracking using variance-based radio tomography networks," *IEEE Trans. Mobile Comput.*, vol. 10, no. 5, pp. 612–621, May 2011.
[58] Y. Zheng and A. Men, "Through-wall tracking with radio tomography networks using foreground detection," in *Proc. IEEE WCNC*, Apr. 2012, pp. 3278–3283.
[59] A. Edelstein and M. Rabbat, "Background subtraction for online calibration of baseline RSS in RF sensing networks," *IEEE Trans. Mobile Comput.*, vol. 99, p. 1, 2012, doi: http://doi.ieeecomputersociety.org/10.1109/TMC.2012.206
[60] O. Kaltiokallio, M. Bocca, and N. Patwari, "Follow @grandma: Long-term device-free localization for residential monitoring," in *Proc. IEEE 37th Conf. LCN Workshops*, Oct. 2012, pp. 991–998.
[61] M. Bocca, O. Kaltiokallio, and N. Patwari, "Radio tomographic imaging for ambient assisted living," in *Evaluating AAL Systems Through Competitive Benchmarking* (Communications in Computer and Information Science), vol. 362, S. Chessa and S. Knauth, Eds. Berlin Heidelberg, Germany: Springer-Verlag, 2013, pp. 108–130.
[62] Y. Zhao and N. Patwari, "Noise reduction for variance-based device-free localization and tracking," in *Proc. 8th Annu. IEEE Commun. Soc. Conf. SECON*, Jun. 2011, pp. 179–187.
[63] Y. Zhao, N. Patwari, J. M. Phillips, and S. Venkatasubramanian, "Radio tomographic imaging and tracking of stationary and moving people via kernel distance," in *Proc. 12th Int. Conf. Inf. Process. Sensor Netw.*, 2013, pp. 229–240.
[64] O. Kaltiokallio, M. Bocca, and N. Patwari. (2013, Feb.). *A Multi-Scale Spatial Model for RSS-based Device-Free Localization* [Online]. Available: http://arxiv.org/abs/1302.5914
[65] J. Wilson and N. Patwari, "A fade-level skew-laplace signal strength model for device-free localization with wireless networks," *IEEE Trans. Mobile Comput.*, vol. 11, no. 6, pp. 947–958, Jun. 2012.
[66] F. Thouin, S. Nannuru, and M. Coates, "Multi-target tracking for measurement models with additive contributions," in *Proc. 14th Int. Conf. Inf. Fusion*, 2011, pp. 1–8.
[67] S. Nannuru, Y. Li, M. Coates, and B. Yang, "Multi-target device-free tracking using radio frequency tomography," in *Proc. 7th Int. Conf. ISSNIP*, 2011, pp. 508–513.
[68] M. Bocca, O. Kaltiokallio, N. Patwari, and S. Venkatasubramanian, "Multiple target tracking with RF sensor networks," *IEEE Trans. Mobile Comput.*, vol. PP, no. 99, p. 1, 2013, doi: 10.1109/TMC.2013.92.
[69] (2013, Apr.). *Finnish Combat Camera Team* [Online]. Available: http://www.youtube.com/watch?v=9v1fFIHRWGE
[70] J. Timonen, "Distributed information system for tactical network," in *Proc. 3rd Workshop Wireless Commun. Appl.*, Vaasa, Finland, Apr. 2012.

MIKAEL BJÖRKBOM received the M.Sc. degree with distinction in control engineering from the Helsinki University of Technology, Finland, in 2006, and the Doctoral degree from Aalto University, Finland, in 2010. He continues as a postdoctoral researcher and the research coordinator for the Wireless Sensor Systems Group, Aalto University, which consists of about 15 researchers. His main research interests include wireless process control, adaptive control, and simulation of wireless control systems.

JUSSI TIMONEN is a Ph.D. Student and Researcher with the Finnish National Defence University, Department of Military Technology. His main research areas are information fusion and visualization of common operational picture. The main fields of research are urban area warfare and cyber warfare.

HÜSEYIN YIĞITLER received the B.Sc. and M.Sc. degrees in electrical engineering from Middle East Technical University, Ankara, Turkey, in 2004 and 2006, respectively. He has been pursuing the Ph.D. degree with the Department of Communications and Networking, Aalto University School of Electrical Engineering, since 2011. His current research interests include real-world wireless sensor network development and deployments, wireless automation, network management, and indoor localization technologies.

OSSI KALTIOKALLIO received the B.Sc. and M.Sc. degrees in electrical engineering from Aalto University, School of Electrical Engineering, Helsinki, Finland, in 2011. He is currently pursuing the Ph.D. degree with the Department of Communications and Networking, Aalto University School of Electrical Engineering. He is a member of the Wireless Sensor Systems Group, Aalto University. His current research interests include indoor propagation channel, real-world wireless sensor network deployments, signal processing, and indoor localization and tracking technologies.

JOSÉ M. VALLET GARCÍA received the M.Sc. degree with distinction in automation and electronics from the Carlos III University of Madrid, Spain, in 2002. He is a Researcher and Ph.D. candidate at Aalto University. His main research areas are home automation, fuel cell automation, and localization systems in WSN.

MATTHIEU MYRSKY received the M.Sc. degree in automation technology from Aalto University in 2010. He continued as a Ph.D. student with the Finnish Centre of Excellence, Generic Intelligent Machines Research Group, Department of Automation and Systems Technology, Aalto University. His main research interests include machine abstraction, multi-machine systems, path planning, and system integration.

JARI SAARINEN received the M.Sc. and Ph.D. degrees in automation technology from the Helsinki University of Technology in 2002 and 2009, respectively. He has acted as a Senior Researcher with the Center of Excellence in Generic Intelligent Machines, Aalto University, Finland, and as a Researcher with the Centre for Applied Autonomous Sensor Systems, Örebro University, Sweden. His research interests include long-term autonomy, 3-D perception, localization, and mapping.

MARKO KORKALAINEN received the M.Sc. degree in electrical engineering from the University of Oulu in 2003. Since 2004, he has been with Industrial M2M Systems, VTT, as a Research Scientist. His research interests include wireless sensor network technologies and embedded sensor systems.

CANER ÇUHAC was born in Tekirdağ, Turkey, in 1986. He received the B.E. degree in electronics and communication engineering from Yildiz Technical University, Istanbul, Turkey, in 2008, and the M.Sc. degree in telecommunication engineering from the University of Vaasa, Finland, in 2011.

He joined the Department of Computer Science, University of Vaasa, in 2010, as a Research Assistant, and he became a Researcher in 2011. His current research interests include wireless embedded systems, image processing, and automation.

RIKU JÄNTTI (M'02–SM'07) is an Associate Professor (tenured) of communications engineering and the Head of the Department of Communications and Networking, Aalto University School of Electrical Engineering, Finland. He received the M.Sc. degree (with distinction) in electrical engineering and the D.Sc. degree (with distinction) in automation and systems technology from the Helsinki University of Technology (TKK) in 1997 and 2001, respectively. Prior to joining Aalto (formerly known as TKK) in 2006, he was a Professor Pro Tem at the Department of Computer Science, University of Vaasa. He is an Associate Editor of the IEEE Transactions on Vehicular Technology. His research interests include radio resource control, spectrum management, and performance optimization of wireless communication systems.

REINO VIRRANKOSKI is a Senior Researcher with the University of Vaasa, Finland. After graduating from the University of Helsinki in 2000, he was a Researcher with the Helsinki University of Technology (now Aalto University) from 2000 to 2007, as a Visiting Assistant Researcher with Yale University from 2004 to 2005 as a Lecturer of telecommunications with the University of Vaasa from 2007 to 2010, and has been a Senior Researcher with the University of Vaasa since 2010. His research interests cover communication and control in wireless networks, wireless automation, and wireless networks in defense and security.

JOUKO VANKKA received the M.S. and Ph.D. degrees in electrical engineering from the Helsinki University of Technology in 1991 and 2000, respectively, and the bachelor's degree in social sciences from Helsinki University in 1994. He was a Researcher at the Helsinki University of Technology from 1994 to 2005. Since 2005, he has been with the Finnish Defence Forces. He has been a Professor of military technology with National Defence University since 2012.

His current research interests include cyber warfare, satellite communications, and situation awareness. He is the author or co-author 90 technical papers in the area of software radio and communication systems. He is the author of *Direct Digital Synthesizers: Theory, Design and Applications* (Kluwer Academic Publishers, 2001), *Digital Synthesizers and Transmitters for Software Radio* (Springer-Verlag New York, 2005), *Tactical Networks* (Finnish, 2009), and *Cyber Warfare* (Editor, 2013).

HEIKKI N. KOIVO (S'67–M'71–SM'86) received the B.S.E.E. degree from Purdue University, West Lafayette, IN, USA, and the M.S. degree in electrical engineering and the Ph.D. degree in control sciences from the University of Minnesota, Minneapolis. He is an Emeritus Professor at Aalto University, formerly Helsinki University of Technology (HUT), Espoo, Finland. Before joining HUT in 1995, he served in various professorial positions at the University of Toronto, Toronto, ON, Canada, and Tampere University of Technology, Tampere, Finland. His research interests include study of complex systems, adaptive and learning control, mechatronics, microsystems, and cyber-physical systems. He has authored or co-authored over 400 scientific publications. He has been the principal investigator in more than 100 research projects. He is a member of the Editorial Boards of the *Journal of Intelligent and Fuzzy Systems* and the *Journal of Intelligent Automation and Soft Computing*.

He was an Associate Editor of the IEEE Transactions on Robotics and Automation and a member of the Administrative Council of the IEEE Robotics and Automation Society. He is a fellow of the Finnish Academy of Technology.

Text and Language Independent Speaker Identification By Using Short-Time Low Quality Signals

Maurizio Bocca*, Reino Virrankoski**, Heikki Koivo*

* Control Engineering Group
Faculty of Electronics, Communications and Automation
Helsinki University of Technology (TKK)
P.O.Box 5500, FI-02015 TKK, Finland
Tel. +358-9-451-5215, Fax +358-9-451-5208
{maurizio.bocca, heikki.koivo}@tkk.fi
** Telecommunication Engineering Group
Department of Computer Science
University of Vaasa
P.O.Box 700, FI-65101 Vaasa, Finland
Tel. +358-6-324-8694, Fax. +358-6-324-8467
reino.virrankoski@uwasa.fi

Abstract—*Several speaker identification applications that exploit voice signals recorded by using wireless networks of small, low-power acoustic sensors are becoming feasible. However, the acoustic signals provided by these devices have typically lower signal-to-noise ratio compared to wired microphone systems. In this paper, we present a text and language independent speaker identification algorithm based on a cepstral speech parameterization method. We analyze the robustness of the algorithm when the quality of the recorded voice signals is decreased. We also investigate how the number of cepstral coefficients considered in the extracted feature vector, and the resolution of the Discrete Fourier Transform affect the algorithm performance. To make the application as close to real-time as possible, we propose a light-weight classification technique based on a simple –yet effective– similarity measure.*

1. INTRODUCTION

It is nowadays possible to supply several personal items such as mobile phones, laptops, magnetic keys, electronic wallets, or guns, with voice sensing capability by using miniaturized acoustic sensors. By exploiting the uniqueness of the human voice, the access to such personal items can be limited only to their owners. Furthermore, in high-security applications, speaker identification can be part of the person biometric detection.

If we target to create a model of the ongoing situation inside an unknown building, a wireless network of nodes equipped with acoustic sensors can provide useful information, e.g. in military, police, and rescue operations. The acoustic signals provided by the network can be exploited in many ways, including speaker identification. The voice signals collected by small and unnoticeable acoustic sensors can be matched against already existing databases to detect the presence of those potentially dangerous individuals who have already been classified by the authorities. The speaker identification algorithm must also be able to point out if the person whose voice has just been recorded is not already present in the database. This would allow the authorities expanding the number of records included in their database for possible future critical situations.

The above mentioned indoor situation modeling system must be rapidly deployable to an unknown building interior, and must also operate in real-time. This forces us to minimize the delays caused by communications and computation. To fulfill these strict real-time requirements, we ignore methods that are computation intensive or require a priori information about the features of the environment. Instead, we propose a light-weight algorithm based on Mel-Frequency Cepstral Coefficients (MFCCs).

However, in wireless sensor networks (WSN), the applicable sampling frequency as well as the length of the sampling period is strictly limited by the scarce resources, in terms of computational power and memory size, respectively, of the sensor nodes. In this case, a speaker identification algorithm has to operate with noisy and short-time signals. Therefore, an important question concerns the minimum requirements for the quality of the recorded signals to perform the speaker identification task with a significant accuracy.

In this paper, we present a computationally light-weight speaker identification algorithm. Next, we analyze how its performance is affected by the quality of the recorded voice

signals, in terms of applied sampling frequency and length of the sampling period. The algorithm is based on a frequency-plane analysis by using MFCCs. We also investigate how the number of considered mel-cepstral coefficients, and the number of bins used in the Discrete Fourier Transform (DFT) affect the algorithm performance. The number of available MFCCs is upper limited by the number of bins used in the DFT. Finally, we introduce a light-weight –yet effective– threshold-based method to determine if the voice under investigations does not refer to a person present in the database. We study how the applied value of the threshold affects the overall algorithm performance.

The paper is organized as follows. In the next section, we discuss the related work. Section 3 describes the proposed speaker identification algorithm, while simulation setup and results are presented in section 4. Conclusions and directions for future work are given in section 5.

2. Related Work

Different types of features, such as fingerprints, face traits, iris, and voice, have been used in biometric identification systems. Speaker identification algorithms are composed of two parts: the first extracts one or more feature vectors from the voice signal, while the second computes some similarity measure between the feature vector extracted from the signal under investigations and the ones stored in the database. The decision about the identification is based on the computed similarity [1] [2] [3].

An optimal characterizing feature must have maximal inter-speaker (signals of different individuals) and minimal intra-speaker (signals of the same person) variation. Also, it must be robust against voice disguise and mimicry, and against distortion and noise in the signal. The variability of the channel and of the environment is one of the most important factors affecting the accuracy of speaker identification algorithms. Several techniques, such as feature warping [4] and feature mapping [5], have been proposed to contrast it.

MFCCs have been extensively used in speech recognition, speaker identification and music-related applications. Seddik et al. [6] fed a neural network classifier with the MFCCs extracted from the speaker phonemes. A method to reduce the training time of the neural network is presented in [7]. In [8], MFCCs are used for the identification of singers: the singing introduces much larger variability compared to the normal speech, and it also includes much higher frequency components. MFCCs are also used by Eronen and Klapuri [9] in a musical instrument recognition application. In [10], Eronen analyzes and compares the effectiveness of several types of features to recognize different musical instruments. The best results are obtained with two sets of MFCCs.

Gaussian mixtures models (GMMs) [11] have been the state-of-the-art text independent speaker identification algorithm for many years. Support Vector Machines (SVMs) have also been used in speaker identification applications [12].

We introduce a light-weight speaker identification algorithm and evaluate how the quality of the recorded signals affects its accuracy. The feature vector characterizing the speaker is composed by the MFCCs and by their first and second order temporal derivatives. We analyze the effect of the number of considered cepstral coefficients and of the resolution of the DFT. Our results define the minimum requirements for the wireless acoustic sensors to collect voice signals that enable a successful identification.

3. Cepstral Parameterization Process

The applied speaker parameterization method is based on cepstral analysis as described in [1] [3]. In (7), we propose a light-weight method to separate the MFCCs vectors related to speech portions of the signal from the ones corresponding to silence or background noise.

A speech signal of N samples is first collected to vector $\boldsymbol{x} = [x(1),..., x(N)]$. The high frequencies of the spectrum, which are reduced by the human speech production process, are enhanced by applying a filter to each element $x(i)$ of $\boldsymbol{x}$:

$$x_p(i) = x(i) - \alpha x(i-1), \quad i = 2,\ldots,N. \qquad (1)$$

The enhanced speech signal vector is called $\boldsymbol{x_p}$. The pre-defined parameter α usually belongs to range [0.95, 0.98] [3]. The signal is then windowed with a Hamming window of $L_w = t_w f_s$ points, where t_w is the time length of the window (30 msec), and f_s is the sampling frequency of the signal. The shift between two consecutive windows is set to 2/3 of the window length.

The DFT is applied to each window of the signal. The results are then collected to matrix **T**. Each column of **T** contains N_{bins} elements, where N_{bins} is the number of bins applied in the DFT. Since this transform provides a symmetric spectrum, only the first half of each column of **T** is preserved. We get a matrix **F**, which contains only the first $N_{bins}/2$ rows of **T**.

The power spectrum, which represents the portion of the signal power included within given frequency bins, is computed by squaring the norm of each element in **F**:

$$\mathbf{P_w} = \left[\left| \mathrm{F}(i,j) \right|^2 \right] \quad i = 1,\ldots,\frac{N_{bins}}{2} \quad j = 1,\ldots N_w. \qquad (2)$$

The frequencies located in the range of human speech are further on enhanced by multiplying the power spectrum matrix $\mathbf{P_w}$ by a filterbank matrix $\mathbf{B_f}$. Thus, we get a smoothened power spectrum matrix $\mathbf{P_s} = \mathbf{P_w}\,\mathbf{B_f}$.

$\mathbf{B_f}$ represents a filterbank of triangular filters whose central frequencies are located at regular intervals in the so-called mel-scale. The conversion from the mel-scale to the normal frequency one is done according to [13]:

$$f_{Hz} = 700\left(10^{\frac{Fmelscale}{2595}} - 1\right). \tag{3}$$

The mel-scale filterbank reduces the random variation in the high-frequency region of the spectrum by progressively increasing the bandwidth of the mel-filters.

After having transformed $\mathbf{P_s}$ into decibels ($\mathbf{P_{db}}$), the MFCCs are computed by applying the Discrete Cosine Transform (DCT) to each column vector in $\mathbf{P_{db}}$. The main advantage of this transform is that it converts statistically dependent spectral coefficients into statistically independent cepstral coefficients [14] [15] [16]. The elements of the mel-cepstral matrix $\mathbf{C_p}$ are calculated as:

$$\mathbf{C_p}(k,l) = a(k)\sum_{i=1}^{\frac{N_{bins}}{2}} \mathrm{P_{db}}(i,l)\cos\left(\frac{\pi(2i-1)(k-1)}{N_{bins}}\right) \tag{4}$$

where $1 \le k \le N_{cep}$, $1 \le l \le N_w$, and

$$a(k) = \begin{cases} \sqrt{\dfrac{N_{bins}}{2}} & ,k = 1 \\ \sqrt{\dfrac{4}{N_{bins}}} & ,2 \le k \le N_{cep} \le \dfrac{N_{bins}}{2} \end{cases} \tag{5}$$

In (4), N_{cep} is the number of cepstral coefficients that are considered. The number of elements of each column of $\mathbf{P_{db}}$, $N_{bins}/2$, represents the upper limit for the number of available MFCCs ($N_{cep} \le N_{bins}/2$).

The first cepstral coefficient of each window is ignored since it represents only the overall average energy contained in the spectrum. The rest of the coefficients are centered by subtracting the mean of the respective mel-cepstral vector. We get the centered mel-cepstral matrix $\mathbf{C}$. The lowest and highest order coefficients are de-emphasized by multiplying each column of $\mathbf{C}$ by a smoothening vector $\mathbf{M}$. By doing so, we get a smoothened mel-cepstral matrix $\mathbf{C_s}$. The elements of $\mathbf{M}$ are computed according to:

$$M(i) = 1 + \frac{N_{cep} - 1}{2}\sin\left(\frac{\pi i}{N_{cep} - 1}\right), \tag{6}$$

where $i = 1, \ldots, N_{cep} - 1$ [17].

We compute a normalized average vector of $\mathbf{C_s}$, such that each value $C_N(i)$ in the vector $\mathbf{C_N} = [C_N(1) \ \ldots \ C_N(N_w)]$ is the mean of the respective column in $\mathbf{C_s}$, normalized to range [0,1]. We are able to separate the windowed mel-cepstral vectors related to speech portions of the signal in $\mathbf{C_s}$ from the ones corresponding to silence or background noise by using the overall mean of $\mathbf{C_N}$ as a criterion. Thus, the matrix $\mathbf{C_{sp}}$, containing only the useful mel-cepstral vectors, is:

$$\mathbf{C_{sp}} = \left[C_s(j) \,|\, C_N(j) \ge \mu(\mathbf{C_N})\right] \quad j = 1, \ldots, N_w, \tag{7}$$

where j denotes the jth mel-cepstral vector of matrix $\mathbf{C_s}$ and $\mu(\mathbf{C_N})$ is the overall average of $\mathbf{C_N}$.

The final mel-cepstral coefficients $\mathbf{C_{cep}}$ are computed by taking the row-wise average of $\mathbf{C_{sp}}$:

$$\mathbf{C_{cep}} = \begin{bmatrix} \mu\{C_{sp}(1,1), \ \ldots \ C_{sp}(1,n)\} \\ \vdots \\ \mu\{C_{sp}(N_{cep}-1,1), \ \ldots \ C_{sp}(N_{cep}-1,n)\} \end{bmatrix}, \tag{8}$$

where n (with $n \le N_w$) is the number of mel-cepstral vectors selected from $\mathbf{C_s}$ into $\mathbf{C_{sp}}$ according to (7).

The information carried by $\mathbf{C_{cep}}$ is extended to capture the dynamics of the speech by including the temporal first and second order derivatives of the smoothened mel-cepstral matrix $\mathbf{C_s}$. The elements included in the first order temporal derivative matrix $\mathbf{\Delta C_s}$ are computed as:

$$\Delta C_s(i,j) = \frac{\sum_{k=-\Theta}^{\Theta} k C_s(i, j+k)}{\sum_{k=-\Theta}^{\Theta} k^2}, \tag{9}$$

where $1 + \Theta \le j + k \le N_w - \Theta$ and $1 \le i \le N_{cep} - 1$. As in (9), the second order temporal derivative $\mathbf{\Delta\Delta C_s}$ is obtained by computing the first order temporal derivative of $\mathbf{\Delta C_s}$ [3] [18]. $\mathbf{\Delta C_{cep}}$ and $\mathbf{\Delta\Delta C_{cep}}$ are computed from the matrices $\mathbf{\Delta C_s}$ and $\mathbf{\Delta\Delta C_s}$ by following the same procedure as in (7)-(8).

Finally, the MFCCs and their first- and second order temporal derivatives are collected into the feature vector $\mathbf{F_s}$:

$$\mathbf{F}_s = \left[\mathbf{C_{cep}}^T \quad \mathbf{\Delta C_{cep}}^T \quad \mathbf{\Delta\Delta C_{cep}}^T\right]. \tag{10}$$

$\mathbf{F_s}$ has $3 \cdot (Ncep - 1)$ elements and characterizes the speaker.

4. SIMULATIONS AND RESULTS

4.1. Setup

The simulations are performed in Matlab. Our self-collected database includes 15 languages and 60 individuals (45 men, 15 women), for a total of 190 signals, with length varying between 8 and 10 seconds. Each signal was recorded with a commercially available wired microphone (Labtec desk mic 534). To guarantee the text and language independency of

the algorithm, each person was recorded for a minimum of two times while speaking freely, and possibly using different languages. The signals were recorded in different indoor environments (e.g. office rooms, corridors, halls): this fact introduces variability in the recorded level of background noise and in the space reverberation, conditions known as channel variability.

The whole database was divided into two parts: the first (15 languages, 45 individuals, 36 men, 9 women, 140 samples) was used to study the accuracy of the algorithm in assigning the correct identity to the sample under investigations. The second part of the database (10 languages, 15 individuals, 10 men, 5 women, 50 samples) was exploited to analyze the performance of the algorithm in determining if the signal under investigations did not refer to a person included in the database. In simulations, each signal was matched against all the other signals of the database. Given the presence of 2-4 signals related to the same person, we were able to estimate the accuracy of the algorithm.

As similarity measure between the extracted feature vectors (10) of the voice signals, we chose the Euclidean distance. In our simulations, this similarity measure differentiated the feature vectors better than others, such as the Manhattan and Chebyshev distance, or the Pearson correlation coefficient.

4.2. The Effect of N_{bins} and N_{cep}

In the first group of simulations, realized with the first part of our database, we set the length of the sampling period to 8 seconds and the sampling frequency to 8 kHz: these values represent the best available quality of the recorded voice signals. Next, we varied the number of bins (N_{bins}) used in the DFT from 128 to 2048, and the number of cepstral coefficients (N_{cep}) considered in the computation from 10 to 1024 (with $N_{cep} \leq N_{bins}/2$). The 78% maximum accuracy in the identification was reached with N_{bins} = 512 and N_{cep} = 100.

We observed that the accuracy of the algorithm is marginally affected by the value of N_{bins}, while N_{cep} plays a big role. As shown in Figure 1, for any value of N_{bins}, the best accuracy is obtained when N_{cep} = 100. The performance rapidly decreases when N_{cep} is further on reduced. In fact, the lower order MFCCs are heavily affected by the random spectral variations and slowly varying additive noise distortion. On the contrary, when N_{cep} is increased, the performance of the algorithm first slightly decreases, and then levels off. This is explained by the fact that the higher order MFCCs carry less informative content than the lower order ones, and they tend to overlearn the spectral features of the voice signal.

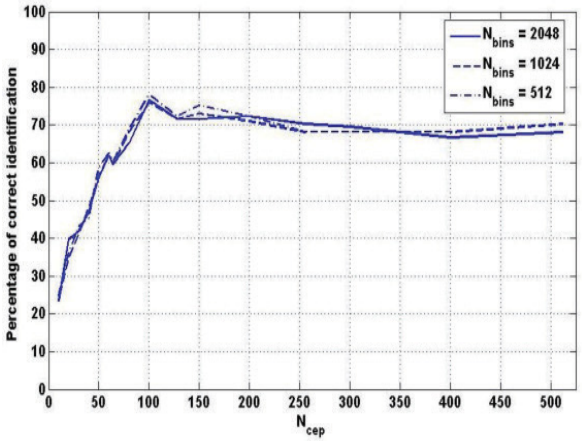


Figure 1 – The effect of N_{cep} on the algorithm accuracy (L = 8 seconds, f_s = 8 kHz).

4.3. The Effect of L and f_s

In the second group of simulations, we kept constant N_{bins} = 512 and N_{cep} = 100 (best configuration), and we varied the length of the sampling period (L) from 8 to 2 seconds, and the sampling frequency (f_s) from 8 kHz to 200 Hz. In this way we were able to test the robustness of the algorithm with short-time low quality signals, such as the ones typically recorded by wireless sensor nodes.

Figure 2 shows the results of this second set of simulations. The accuracy of the identification weakens linearly when f_s is reduced from 8 kHz to 2 kHz (for L = 8 seconds and f_s = 2 kHz, we still get 62.5%). When f_s is further on reduced, the accuracy of the identification rapidly collapses.

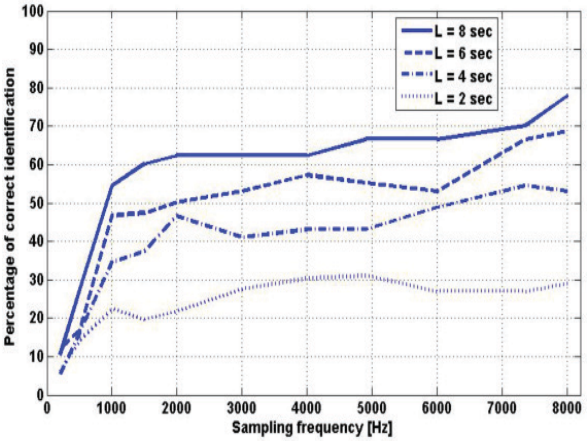


Figure 2 – The effect of f_s on the algorithm accuracy (N_{bins} = 512, N_{cep} = 100).

Finally, the algorithm performance weakens linearly when L is shortened from 8 to 2 seconds. With L = 6 seconds and f_s = 8 kHz, the accuracy is still 70%. The combined effect of the two parameters, f_s and L, is shown in Figure 3.

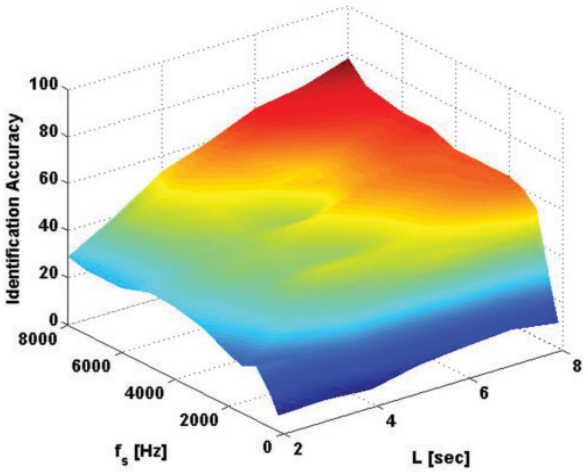


Figure 3 – The combined effect of f_s and L on the algorithm accuracy (N_{bins} = 512, N_{cep} = 100).

4.4. The Detection of Signals Related to Individuals not Included in the Database

We used the second part of the database to evaluate the capability of our speaker identification algorithm to detect those voice signals related to individuals not yet included in the database.

The light-weight method we propose is based on a threshold value (T_{hr}), calculated from the mean (μ_{cor}) and the standard deviation (σ_{cor}) of the distances of the correct identifications registered in the first two sets of simulations, adjusted with a pre-defined parameter (m):

$$T_{hr} = \mu_{cor} + m \cdot \sigma_{cor} \tag{11}$$

If the minimum distance found between the feature vector extracted from the signal under investigations and the ones extracted from the signals included in the first group of the database (known identities) is larger than the threshold, then the voice signal is classified as referring to a new person not yet included in the database.

In the end, we evaluated the accuracy of the algorithm both in correctly classifying the signals related to individuals already included in the database (P_{DB}), and in detecting those signals corresponding to individuals not yet included in the database (P_{NotDB}). The results are shown in Figure 4.

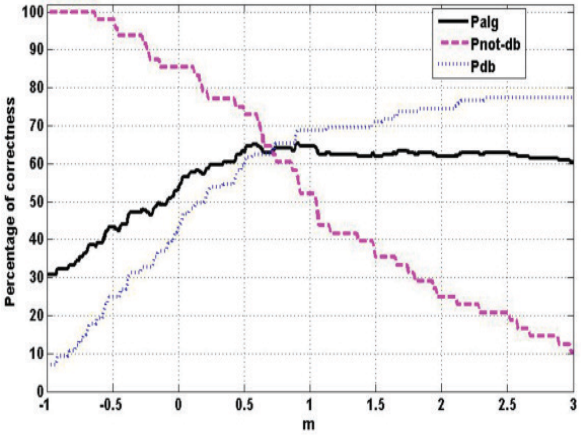


Figure 4 – The effect of m on the algorithm accuracy

The parameter m defines the value of the threshold. When T_{hr} is considerably smaller than μ_{cor} (negative values of m), the algorithm misclassifies as related to individuals not yet included in the database most of the voice signals (high P_{NotDB}, low P_{DB}). On the contrary, when T_{hr} is considerably larger than μ_{cor} (positive values of m), the algorithm is not able to recognize those signals corresponding to individuals not yet included in the database (low P_{NotDB}, high P_{DB}). The maximum overall accuracy (P_{ALG} = [65%,70%]) is reached when m ranges in the interval [0.5,1].

5. CONCLUSIONS AND FUTURE WORK

The proposed speaker identification algorithm is based on speech parameterization by using cepstral analysis. In the feature vector extraction process, we introduced in (7) a light-weight method to separate the portions of the signal related to speech from the ones corresponding to silence or background noise.

The algorithm was first tested to evaluate its accuracy in correctly classifying the voice signals included in a database of known identities. We discovered that with signals having a maximum length of 8 seconds and sampling frequency of 8 kHz, the best accuracy (78%) is obtained with N_{bins} = 512 and N_{cep} = 100. The use of more MFCCs in the computation rather weakens than improves the accuracy. This result does not improve consistently when the applied resolution of the DFT is increased.

With N_{bins} = 512 and N_{cep} = 100 (optimal configuration), the accuracy of the identification stays above 60% with signals 8 seconds long and with sampling frequency ranging from 1.5 to 8 kHz. With f_s between 7 and 8 kHz, the accuracy is in the range between 70 and 80%.

Then, we introduced in (11) a light-weight threshold-based method to determine if the voice under investigations does

not refer to any person present in the database. We analyzed how the applied value of the threshold affects the overall algorithm accuracy, which remains in the range between 65 and 70%.

In future work, we will study how the algorithm accuracy can be improved by modifying the feature vector extraction process. In case of mixed signals (two or more individuals talking simultaneously), we will first separate the different components (individuals) by using a Blind Signal Separation technique based on Independent Component Analysis. Then, we will process the separated signals with the identification algorithm-

Finally, we will collect voice signals with wireless acoustic sensors, both in the single-speaker and multi-speaker case, and we will again evaluate the accuracy of our algorithm.

6. REFERENCES

[1] S. Furui, "Cepstral analysis technique for automatic speaker verification," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. ASSP-29, NO. 2, pp. 254–272, April 1981.

[2] S. Furui, "Comparison of speaker recognition methods using statistical features and dynamic features," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. ASSP-29, NO. 3, pp. 343–350, June 1981.

[3] F. Bimbot, J-F. Bonastre, C. Fredouille, G. Gravier, I. Magrin-Chagnolleau, S. Meignier, T. Merlin, J. Ortega-Garcia, D. Petrvovska-Delacretaz, and D. A. Reynolds, "A tutorial on text-independent speaker verification," *EURASIP Journal on Applied Signal Processing*, vol. 4, pp. 430–451, 2004.

[4] J. Pelecanos, and S. Sridharan, "Feature Warping for Robust Speaker Verification", *in ODYSSEY-2001*, Crete, Greece, pp. 213-218, June 18-22, 2001.

[5] D. Reynolds, "Channel Robust Speaker Verification via Feature Mapping", *in Proc. ICASSP 2003*, Hong-Kong, pp. II-53-56, April 6-10, 2003.

[6] H. Seddik, A. Rahmouni, and M. Sayadi, "Text independent speaker recognition using the mel frequency cepstral coefficients and a neural network classifier," *in Proc. of ISCCSP 2004*, pp. 631–634, 2004.

[7] L. Rudasi and S. A. Zahorian, "Text independent talker identification using neural networks," *in Proc. of ICASSP*, vol. 1, pp. 389–392, 1991.

[8] A. Mesaros and J. Astola, "The mel-frequency cepstral coefficients in the context of singer identification," *in Proc. of ISMIR 2005*, London, UK, September 11-15, 2005.

[9] A. Eronen and A. Klapuri, "Musical instrument recognition using cepstral coefficients and temporal features," *in Proc. ICASSP 2000, Istanbul*, June 5-9, 2000.

[10] A. Eronen, "Comparison of features for musical instrument recognition," *in Proc. of WASPAA'01*, New Platz, NY, USA, pp. 19–22, October 21-24, 2001.

[11] D. Reynolds, T. Quatieri, R. Dunn, "Speaker Verification Using Adapted Gaussian Mixture Models", *Digital Signal Processing*, vol. 10, no. 1-3, 2000.

[12] V. Wan, and S. Renals, "Speaker Verification Using Sequence Discriminant Support Vector Machines", *in IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 2, March 2005.

[13] S. Stevens, J. Volkman, and E. B. Newman, "A scale for the measurement of the psychological magnitude of pitch," *Journal of the Acoustical Society of America*, vol. 8, pp. 185–190, 1937.

[14] B. P. Bogert, M. J. R. Healy, and J. W. Tukey, "The quefrency analysis of time series for echoes: cepstrum, pseudo-autocovariance, cross-cepstrum and saphe cracking," *in the Proceedings of the Symposium on Time Series Analysis, New York, USA*, pp. 209–243, 1963.

[15] A. V. Oppenheim and R. W. Schafer, "Homomorphic analysis of speech," *IEEE Transactions on Audio and Electroacoustics*, vol. 16 no. 2, pp. 221–226, 1968.

[16] A. V. Oppenheim and R. W. Schafer, *Discrete-time signal processing*, Prentice Hall, Englewood Cliffs, NJ, USA, 1989.

[17] B. H. Juan, L. R. Rabiner, and J. G. Wilpon, "On the use of band-pass liftering in speech recognition," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. ASSP-35, no. 7, pp. 947–954, July 1987.

[18] L. Rabiner and B. H. Juang, *Fundamentals of speech recognition*, New Jersey, Prentice Hall, 1993.

Estimating the Number of Persons in an Unknown Indoor Environment by Applying Wireless Acoustic Sensors and Blind Signal Separation

Maurizio Bocca*, Cristian Galperti**, Reino Virrankoski* and Heikki N. Koivo*
*Control Engineering Laboratory
Helsinki University of Technology (TKK)
{maurizio.bocca, reino.virrankoski, heikki.koivo}@tkk.fi
**Electronics and Information Department
Politecnico di Milano
galperti@elet.polimi.it

Abstract— **Figuring out the number of persons and their locations in an unknown indoor environment is one of the first tasks that must be done in military, rescue or intelligence operations. If there is no functioning security system in the building, one must use a monitoring system, which is rapidly deployable but simultaneously as invisible as possible. Wireless sensor network is an attractive candidate to fill such requirements. Many sensor types are suitable for indoor situation modeling, but there are also several technical challenges rising from the limited energy and communication resources of the sensor nodes. This paper focuses on estimating the number of persons in an unknown indoor environment by using low-power acoustic sensors. The number of persons is estimated by applying blind signal separation to the acoustic signal collected by the sensor nodes.**

I. Introduction

During the latest years, rising attention has been paid to such systems that are capable to monitor situation at indoors in military, rescue, intelligence or police operations [1]. Compared to the battlefield, building interior is more complicated because many monitoring systems typically used in the battlefield environment are not feasible indoors. Furthermore, instead of looking to the grouping of forces or other macro-scale actions happening outdoors, one must figure out the locations and the actions of individual persons. That sets high accuracy requirements to the indoor situation modeling.

Figuring out what is happening in the building is not an issue, if the building is equipped with a solid security system. However, we can not assume that such a system always exists. Even in the case if the building has a security system, it can be disabled by enemy forces in the military situation, or by fire or by earthquake in the catastrophe situation. Thus, a monitoring system, which consists of such components that are rapidly deployable and attract as little attention as possible, is needed. Such requirements make wireless sensor networks an attractive choice. However, problems rising from the distributed nature of wireless sensor networks and problems rising from the limited energy resources of the sensor nodes must be solved to make them feasible for indoor situation modeling.

In this paper we present an application that estimates a number of people by using the acoustic signal collected by the wireless sensor nodes equipped with a microphone. Collected signal is a mixture of several voices talking simultaneously. A whitening phase of the Independent Component Analysis is applied to figure out the number of voices in the mixed signal. In addition to estimating the number of speakers, the result can be further applied to reconstruct the independent components, which are the voices of the individual persons, from the mixed signal.

On the hardware side, the application requires a sample rate that is high enough to produce acoustic samples with feasible quality. Accurate time synchronization between the nodes is also required to be able to correctly fuse together the data transmitted by different nodes before applying the whitening.

The paper is organized as follows. In the next section we highlight the related work. Section III describes the way how the number of source signals is computed. Section IV provides the details of the system applied in the experiments. Proposed computation method is evaluated through simulations and experiments in Section V. Section VI states our conclusions and plans for future work.

II. Related Work

A. Independent Component Analysis

Independent Component Analysis (ICA) targets to find a linear representation of the nongaussian data so that the components are statistically independent. Assume that we have n sources transmitting simultaneously and h observing units ($n \leq h$). Each observation we get is a mixture of the transmitted signals and it can be written as

$$x_i = a_{11}s_1 + a_{12}s_2 + \ldots + a_{1n}s_n, \quad (1)$$

where x_i is the observation of the i:th observing unit and the source signals are noted by s_i. With all observations, we get an equation group

$$x = As, \quad (2)$$

where x is the vector of observations, A is the mixing matrix and s is the vector of transmitted signals. If we have a set of observations over the discrete time $t = 1, \ldots, k$, equation (2) can have the same form, but so that x and s are $h \times k$ matrices that include all observations.

The statistical model (2) is called Independent Component Analysis or ICA-model. The independent components s_i are latent variables that cannot be directly observed. After estimating the mixing matrix A, the independent components can simply be reconstructed by solving s from (2):

$$s = A^{-1}x. \quad (3)$$

The starting point of the ICA is the assumption that the source signals $s_i(t)$ and $s_j(t)$ are statistically independent at each time moment t. However, it does not matter if the signals have overlapping spectra. It is also assumed that A is a square matrix, but that assumption is made for simplicity and it can sometimes be relaxed [2], [3].

In the Blind Signal Separation (BSS) scenario, we want to find out the original signal components but we can only observe the mixtures and we have a very little, if any, prior knowledge about the number of source signals and about the structure of the mixing matrix. ICA has become one of the most popular methods to solve the BSS problem.

There exists a wide range of applications of ICA, such as biosignal processing [4], [5], finding hidden factors from financial data [6], [7], reducing the noise in the natural images [8] and signal processing in telecommunications [9]. The area of telecommunications is related to our presentation, but instead of cellular network, we are operating in the context of more resource-limited wireless sensor network.

B. Sensor Network Applications

In the context of wireless sensor networks, acoustic sensing is used in a wide range of applications from monitoring and detection to localization. In many applications, the duty of the sensor nodes is just to detect the existence of the acoustic pulse without analyzing the details of the detected signal. More advanced acoustic sensing is applied in the environmental monitoring and in the context recognition.

One of the research projects in the Center for Embedded Networked Sensing at UCLA [10] is the developing an ability to recognize and localize individual species from their vocalization. The design, analysis and testing of acoustic arrays for localizing bird vocalizations of different species is discussed in the recently reported results [11]. Kwan et al. has applied Hidden Markov Models (HMM) and Gaussian Mixture Models (GMM) to the acoustic signal to monitor and classify birds near the airport and in some critical locations in the airspace to reduce the risk of birdstrikes to the airplanes [12]. In [13], a hybrid sensor network for Cane-Toad monitoring is presented. Cane-Toads are detected based on their vocalization.

In their recent article, Eronen et al. [14] investigate the feasibility of an audio-based context recognition system. They propose a system framework, in which 27 different types of urban environments are defined based on their acoustic properties. Then the defined environments are further classified to six different higher level classes. The algorithm is able to detect the surrounding environment and the shift from one type of environment to another. Environment detection is made by using the sensed acoustic signals and HMMs, which are trained by using the acoustic data collected from each type of environment a priori. ICA is also applied to extract a set of statistically independent vectors from the training data. The authors argue that such an algorithm will enable portable devices to adapt automatically to the changing surroundings.

The way how our work differs from the previous ones reported in the context of sensor networks is that instead of detecting only the existence of the acoustic signal or figuring out only the higher level features of a mixed signal, we are tracing down the individual sources, i.e. the independent components, of the observed mixed signal. This presentation discusses just about estimating the number of persons by using the mixed acoustic signal, but the reconstruction of the individual voices

will be the final target and the natural continuation of this work.

III. Computing the Number of Source Signals

The proceeding of standard ICA can be divided to the preprocessing phase and the separation phase. During the preprocessing, the measured data is first centered by subtracting its mean from it. Then a linear transformation is applied to the observation matrix X so that we obtain a new observation matrix $\tilde{X}$, which is white. In other words, the components of $\tilde{X}$ are uncorrelated and their variances are equal to unity. Measurement vectors $\tilde{x}_i$ can be whitened by applying a transformation

$$v_i = V\tilde{x}_i. \tag{4}$$

In (4), the matrix V is called the whitening matrix. One of the most straightforward ways to compute the whitening matrix is by applying the Principal Component Analysis. Once we know C_x, the covariance matrix of the measured data, the whitening matrix is given by

$$V = \Lambda^{-1/2}E^T, \tag{5}$$

where V is k x h matrix and Λ is a diagonal matrix having the eigenvalues of the covariance matrix C_x in its diagonal in a decreasing order such that $\Lambda = diag[\lambda_{1_{C_x}} \dots \lambda_{k_{C_x}}]$. Associated principal eigenvectors c_{p_i} are collected to matrix $E = [c_{p_1} \dots c_{p_k}]$.

Assume that we have h measured mixed signals and each of them has k measurements. Thus, we get a measurement matrix

$$X(k) = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1k} \\ \vdots & \vdots & \ddots & \vdots \\ x_{h1} & x_{h2} & \dots & x_{hk} \end{bmatrix}. \tag{6}$$

By subtracting the mean of each measurement vector from it, we get a set of measurement vectors $\tilde{x}_i(k) = (x_i(k) - \overline{x}_i(k))$ and a respective measurement matrix $\tilde{X}(k) = [\tilde{x}_1(k) \dots \tilde{x}_h(k)]^T$. By definition, the covariance matrix of $\tilde{X}(k)$ is

$$\begin{aligned} Cov\{\tilde{X}(k)\} &= E\{\tilde{X}(k)\tilde{X}(k)^T\} = \\ C_x = [c_{ij}] &= \left[E\{(\tilde{x}_i - \overline{\tilde{x}}_i)(\tilde{x}_j - \overline{\tilde{x}}_j)^T\}\right]_{ij}, \end{aligned} \tag{7}$$

where $i, j = 1, \dots, h$. Since the mean of each observation vector $\tilde{x}_i$ is set to zero, the values in (7) become

$$\begin{aligned} E\{(\tilde{x}_i - \overline{\tilde{x}}_i)(\tilde{x}_j - \overline{\tilde{x}}_j)\} &= E\{\tilde{x}_i\tilde{x}_j\} \\ = \textstyle\sum_{n=1}^{k} \frac{1}{k}\tilde{x}_{in}\tilde{x}_{jn} &= \tfrac{1}{k}(\tilde{x}_{i1}\tilde{x}_{j1} + \dots + \tilde{x}_{ik}\tilde{x}_{jk}). \end{aligned} \tag{8}$$

Thus, the covariance matrix (7) of $\tilde{X}(k)$ becomes

$$\begin{aligned} C_x &= \begin{bmatrix} E\{\tilde{x}_1(k)\tilde{x}_1(k)^T\} & \dots & E\{\tilde{x}_1(k)\tilde{x}_h(k)^T\} \\ \vdots & \ddots & \vdots \\ E\{\tilde{x}_h(k)\tilde{x}_1(k)^T\} & \dots & E\{\tilde{x}_h(k)\tilde{x}_h(k)^T\} \end{bmatrix} \\ &= \frac{1}{k}\begin{bmatrix} \tilde{x}_{11}^2 + \dots + \tilde{x}_{1k}^2 & \dots & \tilde{x}_{11}\tilde{x}_{h1} + \dots + \tilde{x}_{1k}\tilde{x}_{hk} \\ \vdots & \ddots & \vdots \\ \tilde{x}_{h1}\tilde{x}_{11} + \dots + \tilde{x}_{hk}\tilde{x}_{1k} & \dots & \tilde{x}_{h1}^2 + \dots + \tilde{x}_{hk}^2 \end{bmatrix} \\ &= \frac{1}{k}\tilde{X}(k)\tilde{X}(k)^T. \end{aligned} \tag{9}$$

If the number of source signals is q and $q \leq h$, the q largest eigenvalues of the covariance matrix (9) are some combination of the source signal powers added to the noise power, and the remaining $h - q$ eigenvalues correspond to only noise. If the signal-to-noise ratio is large enough, there is a remarkable difference of magnitude between the eigenvalues which are representing the source signals and the eigenvalues which are representing only noise [15]. Thus, in the whitening step (5) one can estimate the number of source signals q based on the difference of magnitude in the eigenvalues. This information can be used to compress the data such that we only choose the q largest eigenvalues to be used in (5) and set the remaining $h - q$ eigenvalues, that are representing only noise, to zero. In the case of mixed speech signals, the estimated value of q gives us the estimate of the number of persons in the monitored space.

IV. Applied System Architecture

A. Hardware

The experiments are performed by using Mica2 motes equipped with a Crossbow MTS300CA sensor boards [17]. The mote with the sensor board is illustrated in the Figure 1. MTS300CA includes a low power microphone NS LMC567 [18] that can be used for general acoustic recording.

B. Operation

The TinyOS components for Mica2 can normally sample at a frequency up to 200 Hz. By introducing the HighFrequencySampling component [19], the sampling frequency can reach the 6.67 kHz, after turning off the wireless radio of the Mica2 while sampling. Each node starts the acoustic sampling in response to an external message broadcasted by the base station. The message specifies the number k and the interval m

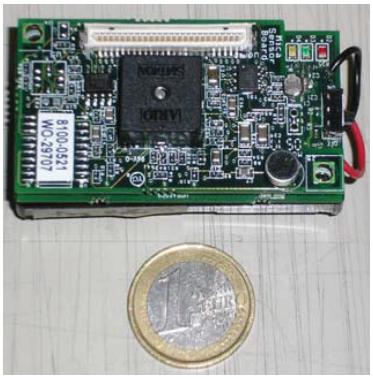


Fig. 1. Mica2 with MTS300CA Sensor Board

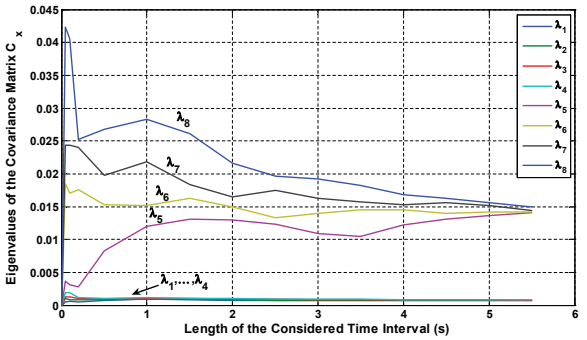


Fig. 2. The behavior of the covariance matrix eigenvalues when there were four source signals and eight observed mixed signals. Sampling frequency was kept on 8 kHz and the total length of the sampling period is 5.5 seconds.

(in microseconds) of the samples. By doing this we target to achieve the best possible accuracy in time synchronization between the nodes. During the sampling, the samples are logged over the EEPROM memory of the Mica2. After the sampling phase, the data is collected wirelessly such that the base station starts downloading data from the addressed node by transmitting another message. To avoid the lost of packages during the downloading of the data, we enabled the acknowledgement (ACK) control system.

V. Simulations and Experiments

A. Computing the Number of Source Signals

In the first experiment, we used four recordings of simultaneously played different acoustic sources recorded at a frequency of 8 kHz with 8 bits per sample. There were two speech signals, a rock-band playing music and a police car siren. Then we created four additional acoustic signals by mixing the previous ones and noise and ensured the statistical independence by computing that the measurement matrix had a full rank $rank(X) = 8$. Next we processed the measurement data and computed the covariance matrix C_x according to (6)-(9). The eigenvalues of C_x became

$$\lambda = \begin{bmatrix} 0.01640 \\ 0.01435 \\ 0.01411 \\ 0.01370 \\ 0.00080 \\ 0.00079 \\ 0.00074 \\ 0.00072 \end{bmatrix}. \quad (10)$$

In (10) we can clearly see the expected difference of magnitude between fourth and fifth eigenvalue indicating that there exits four different acoustic source signals in the observed mixtures. Figures 2 and 3 show the behavior of the covanriance matrix eigenvalues during the sampling period having a sampling frequency of 8 kHz and a total length of 5.5 seconds. Figure 3 indicates that when such a relatively high sampling frequency is applied, the difference of magnitude in the eigenvalues shows up during the first second of the sampling.

In our second experiment, we were using three Mica2 motes sampling in 3 kHz frequency and two human voices talking simultaneously. The mixed signals were recorded and buffered by the motes and after sampling transmitted wirelessly to the base station. The resulting covariance matrix eigenvalues were

$$\lambda = \begin{bmatrix} 0.04426 \\ 0.02415 \\ 0.00952 \end{bmatrix}. \quad (11)$$

The magnitude of difference between the second (0.02415) and third (0.00952) eigenvalue is still detectable even though it is not so big what it was in the first experiment. That result was expectable, since in the second experiment only 3 kHz sampling rate and MTS300CA Sensor Boards were used compared to the 8 kHz sample rate and good quality microphones used in the first experiment. However, the result shows that it is possible to achieve a signal quality that is good enough to figure out the number of source signals and to continue with the signal separation by using resource-constrained wireless sensor nodes.

B. The Effect of Time Synchronization

We investigated also the effect of the time synchronization accuracy by using the three signals recorded in the second experiment. The time synchronization

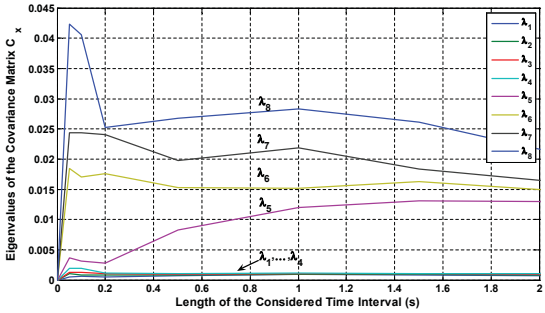


Fig. 3. The behavior of the covariance matrix eigenvalues during the first two seconds of the sampling period. The setup is the same than the one illustrated in Figure 2.

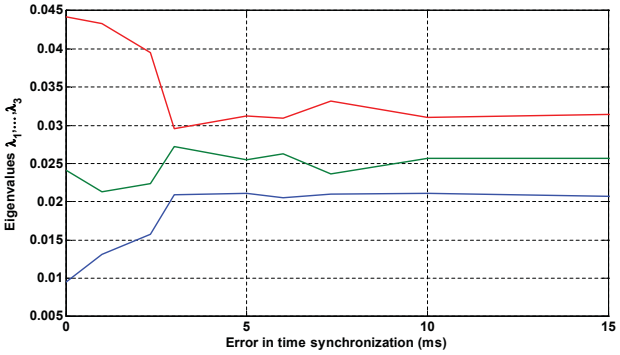


Fig. 4. Time synchronization error effect to the covariance matrix eigenvalues. Three Mica 2 motes are applied to collect the acoustic data by using 3 kHz sampling frequency. Maximum time synchronization error between the sampling motes is increased from 0 to 15 milliseconds.

error between the signals was increased from zero to 15 milliseconds. The behavior of the covariance matrix eigenvalues is illustrated in Figure 4. There is no detectable difference of magnitude in the covariance matrix eigenvalues after the time synchronization error increases to three milliseconds.

C. Periodic Noise Caused by the EEPROM Writing

By analyzing the recordings we were able to detect the presence of a periodic noise component illustrated in Figure 5. We identified that the component is caused by writing the already collected samples to the EEPROM memory of the Mica2. Since the writing to EEPROM happens periodically, it temporarily requires more power resources and that periodic writing causes the periodic noise component to the microphone.

We also strongly believe that the periodic noise component illustrated in the upper part of Figure 7 at page 5 on [13] is caused by the same reason since the

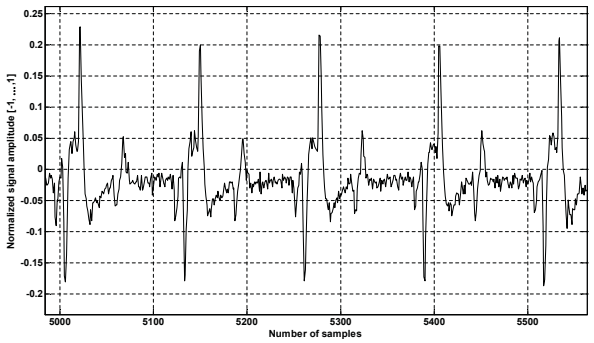


Fig. 5. A periodic noise component caused by the EEPROM writing.

authors have been using the same hardware and the noise component looks the same than the one illustrated in Figure 5.

VI. Conclusions and Future Work

In this paper, we have demonstrated the feasibility of ICA to the processing of acoustic signals in the context of wireless sensor networks. One possible application to estimate the number of persons by using the whitening phase of ICA is demonstrated by using simulations and a simple experimental setup that allows us to listen over a short period of time, transmit the acoustic signals to the base station and then perform the offline analysis.

We found out that one can extract useful results by using a 3 kHz sampling frequency that can be used with the wireless sensor nodes. However, once the sampling frequency is lowered from the values typically used with the wired acoustic systems, the system becomes more dependent about the accurate time synchronization. Based on our experiments, the results collected by using 3 kHz sample rate become useless if the error in time synchronization exceeds three milliseconds.

We are working on adding the Timing-sync Protocol for Sensor Networks (TPSN) [20] to our application. By applying the TPSN, the nodes first synchronize their clocks to the one of the base station. Once the synchronization phase is completed, the nodes start sampling simultaneously and use their clocks to time stamp the samples.

To be able to listen over longer periods of time, we suggest such a clustered network architecture, where the nodes in different clusters alternate periodically between three modes: 1) sampling, 2) transmitting the samples from the node buffer to the base station and 3) repeating the time synchronization. That sort of setup allows us to keep on listening as long as there are sufficient amount of energy resources left in the sensor nodes. Developing

and demonstrating such a system will be a continuation of this work. Moreover, since the resources of Mica2 are pretty limited, using a more resource-rich node such as Telos [16] would enable more robust performance.

In the algorithmic side there are two interesting paths for the further research. First one is the reconstruction of the individual source signals by using ICA. Right now we are just presenting a way to estimate the number of sources, but reconstructing the individual source signals would enable us to match the speech samples against the samples stored in the police and intelligence databases. By doing so, one can define if some person whos speech sample is already stored in the database is present among the speaking persons. The second area for the further research is the underconstrained case. The assumption that we must have at least as many observed mixed signals as we have sources is one of the basic assumptions in the signal separation performed by using ICA. However, the methods to separate the signals in the underconstrained case, i.e. in such a case in which there are more source signals than observed mixed signals, are under active research. Finding such a method that handles the underconstrained case and that is feasible to the sensor networks would considerably relax the overall system requirements.

References

[1] *Statement by Dr. Tony Tether*, Defence Advanced Research Projects Agency, Submitted to the Subcommittee On Emerging Threats And Capabilities, Committee on Armed Services, United States Senate, March 9, 2005

[2] A. Hyvarinen and E. Oja, *Independent Component Analysis: Algorithms and Applications*, Neural Networks, 13(4-5):411-430, 2000

[3] A. Hyvarinen, J. Karhunen and E. Oja, *Independent Component Analysis*, Jhon Wiley & Sons, New York, 2001

[4] R. Vigario, *Extraction of Ocular Artifacts from EEG Using Independent Component Analysis*, Electroenceph. Clin. Neurophysiol., 103(3):395-404

[5] R. Vigario, V. Jousmaki, M. Hamalainen, R. Hari and E. Oja, *Independent Component Analysis for Identification of Artifacts in Magnetoencephalographic Recordings*, Advances in Neural Information Processing Systems, vol. 10, pp. 229-235, MIT Press, 1998

[6] A. D. Back and A. S. Weigend, *A First Application of Independent Component Analysis to Extracting Structure from Stock Returns*, Int. J. On Neural Systems, 8(4):473-478, 1997

[7] K. Kiviluoto and E. Oja, *Independent Component Analysis for Parallel Financial Time Series*, In the Proceedings of International Conference on Neural Information Processing (ICONIP98), vol. 2, pp. 895-898, Tokyo, Japan, 1998

[8] A. Hyvarinen, *Sprase Code Shrinkage: Denosing of Nongaussian Data by Maximum Likelihood Estimation for Independent Component Analysis*, Neural Computation, 11(7):1739-1768, 1999

[9] T. Ristaniemi and J. Joutsensalo, *On the Performance of Blind Source Separation in CDMA Downlink*, In the Proceedings of International Workshop on Independent Component Analysis and Blind Signal Separation (ICA99), pp. 437-441, Aussosis, France, 1999

[10] CENS Website: *http://research.cens.ucla.edu/*

[11] C-E. Chen, H. M. Ali and H. Wang *Design and Testing of Robust Acoustic Arrays for Localization and Enhancement of Several Bird Sources*, In the Proceedings of Information Processing in Sensor Networks (IPSN06), April 19-21, Nashville, Tennessee, USA, 2006

[12] C. Kwan, K. Ho, G. Mei, Y. Li, Z. Ren, R. Xu, G. Zhao, M. Stevenson, V. Stanford and C. Rochet, *An Automated Acoustic System to Monitor and Classify Birds*, Presentations of the 5th annual meeting of the Bird Strike Committee-USA/Canada, Toronto, Ontario, Canada, August 18-21, 2003

[13] W. Hu, V. N. Tran, N. Bulusu, C. T. Chou, S. K. Jha and A. Taylor, *The Design and Evaluation of a Hybrid Sensor Network for Cane-Toad Monitoring*, In the Proceedings of the Fourth International Symposium on Information Processing in Sensor Networks, IPSN 2005, UCLA, Los Angeles, California, USA, April 25-27, 2005

[14] A. J. Eronen, V. T. Peltonen, J. T. Tuomi, A. P. Klapuri, S. Fagerlund, T. Sorsa, G. Lorho and J. Huopaniemi, *Audio-Based Context Recognition*, IEEE Transactions on Audio, Speech and Language Processing, Vol. 14, No. 1, January 2006

[15] F. M. Ham, N. A. Faour and J. C. Wheeler, *Infrasound Signal Separation Using Independent Component Analysis*, SPIE, Vol. 4055, Orlando, Florida, USA, April 2000

[16] J. Polastre, R. Szewczyk and D. Culler, *Telos: Enabling Ultra-Low Power Wireless Research*, In the Proceedings of The Fourth International Conference on Information Processing in Sensor Networks: Special track on Platform Tools and Design Methods for Network Embedded Sensors (IPSN/SPOTS), Los Angeles, California, USA, April 25-27, 2005

[17] Crossbow Website: *http://www.xbow.com*

[18] National Semiconductor LMC567 Low Power Tone Detector: *http://www.national.com/ds/LM/LMC567.pdf*

[19] S. Kim, D. Culler and J. Demmel, *Structural Health Monitoring Using Wireless Sensor Networks*, Berkeley Deeply Embedded Network System Course Report, 2004

[20] S. Ganeriwal, R. Kumar and M. B. Srivastava, *Timing-sync Protocol for Sensor Networks*, SenSys 2003, Los Angeles, California, USA, November 5-7, 2003

Frequency Converter Integration to Wireless Sensor Network

Reino Virrankoski, Maiwulan Wulayinjiang and Le Manh Linh
University of Vaasa, Department of Computer Science, Communications and Systems Engineering Group
P.O.Box 700, FI-65101, Vaasa, Finland
reino.virrankoski@uva.fi

***Abstract*—Frequency converters are used to control electric motors in many kinds of industrial applications. Since they control the electric motors that run the machines or processes, they must be strongly integrated with the rest of the automation system. If the wireless sensor and actuator networks are used as a part of the same automation system, it would be beneficial to have a direct connection between wireless sensor network and frequency converter instead of routing the information through several interfaces. To enable this, we built a software and hardware integration between frequency converter and wireless sensor network. Developed system was tested by performing several experiments in real industrial environment that consisted of sophisticated mobile machines. In the experiments, several frequency converters and sensor nodes were used in the same network. System performance and reliability was evaluated based on the experiment results, and some directions pointed for the future work.**

Keywords—wireless automation; frequency converter; AC drive; wireless sensor network; industrial internet

I. INTRODUCTION

Many industrial machines and processes are run by one or several electric motors. The control of the motors is an important part of the control of the entire system. Nowadays most of the electric motors are controlled by frequency converters, and they have replaced the gearboxes also in many older systems. Since every electric motor is connected to the electric grid, the power supply for the frequency converter is not an issue. With communication the situation is different. There is often a cabled connection to the frequency converter and it can have its own IP address. However, this is not always the case, especially if the frequency converter is located to moving or difficultly accessible location or in a harsh environment, where the communication cables can easily be damaged. It can also be the case that the measurement information, which is utilized in the control of the frequency converter, is collected by using a wireless sensor network (WSN). If the information collected by WSN must be transmitted to separate gateway and then from the gateway to the frequency converter by using another cabled network, it will cause additional delays and make the communication system architecture more complex.

978-1-4673-8743-9/16/$31.00 ©2016 IEEE

The technology to collect data from the frequency converters of the mobile systems during the operation can provide also additional benefits. The data can be utilized to extract information about the condition of the electric motor and the machine which is run by the electric motor. This sort of condition monitoring enables us to detect problems that require fixing service when they are still in an emerging level. This supports the common target to move from the hourly based service towards more specified service schedule, which is based on the individual needs of each machine. This would be especially beneficial in the case of older machines, which does not have that much sensors or any other embedded electronics, but may still have frequency converter controlled electric motors, whose monitoring data can provide us information about the condition of the machine itself.

In this paper we present frequency converter integration to WSN. We apply Vacon frequency converters [1] and wireless sensor platform the UWASA Node [2]. First a communication between the frequency converter and the UWASA Node is enabled. Then a data logging protocol to collect data from several frequency converters over WSN in real time is designed and implemented. The implementation is then evaluated through several experiments in real industrial environment [3]. Finally the star topology system is extended also for multi-hop WSN deployments [4].

The rest of this paper is organized as follows. Applied hardware architecture is explained in Section II and communication software architecture in Section III. Experiments and their results are presented in Section IV. Final conclusions and discussion about the future work are given in Section V.

II. HARDWARE ARCHITECTURE

A. Frequency Converter

We used frequency converters produced by Vacon Inc., which is an AC drive manufacturer headquartered in Vaasa, Finland [1]. The frequency converter, that was used to control the electric motors are illustrated in Figure 1.

A removable control panel is connected to the frequency converter through RS-232 serial port. We used the same connection to connect the UWASA Node with the frequency

converter [3]. Several parameters, which indicate the state of the electric motor, can be read from the frequency converter through this connection, and commands to the frequency converter can also be sent by using the same connection.

Figure 1. Vacon frequency converter. Same RS-232 serial port that connects the LCD panel is used to connect a wireless sensor node to the frequency converter.

B. *The UWASA Node*

The UWASA Node is a modular and stackable wireless sensor platform, which is developed for wireless automation applications. The modular architecture presented in Figures 2 and 3 makes the node applicable for a wide range of automation applications [2], [5].

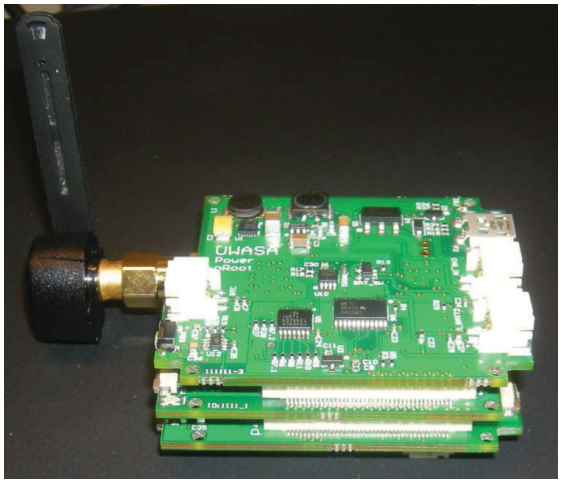


Figure 2. The UWASA Node. Modular hardware and software architecture makes the node feasible for a wide range of automation applications.

The user can select the required sensors based on the particular application needs, and then add them to UWASA Node with minimum software and hardware changes. The basic part of the node consists of main module and power module. Then one can use one or several slave modules based on the particular application needs. The slave modules can be used to add one or several sensors, additional memory, slave processors, additional radios etc. In the third revision of the UWASA Node, the slave module system is developed further so that there is a generic slave module with several types of connectors, that can be used instead of simple slave modules [2], [5]. The hardware model is illustrated in Figure 3.

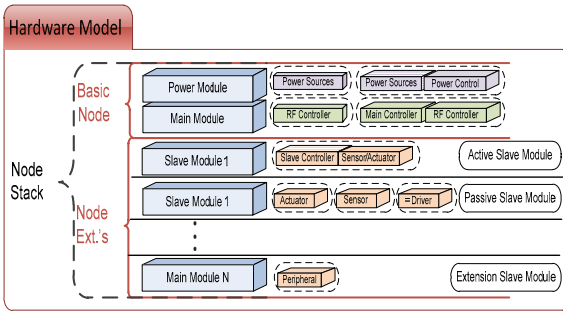


Figure 3. The hardware model of the UWASA Node.

C. *Gateway and Datalogger*

Embedded Linux computer the Raspberry Pi was used as a data logging PC. One UWASA Node was connected into it by USB cable, and it was acting as a gateway between PC and WSN.

III. Software Architecture

A. *Communication Links*

There are three types of communication links in the developed architecture: communication between the frequency converter and UWASA Node through RS-232 serial port, wireless communication between UWASA Nodes and communication between the UWASA Node, which is acting as a gateway, and the data logging PC [3].

Vacon Inc. has developed its own Vacon Human Machine Interface (VHMI) protocol for the interfacing of their frequency converters. This protocol is implemented to the UWASA Node, which is connected to the frequency converter trough RS-232 serial port. Conversion software was also implemented to convert the VHMI packets used in the communication between the frequency converter and the node to the packet form, which is used in the wireless communication between the UWASA Nodes in the WSN. Frequency converter data can be read by index or by ID of the selected parameter. Respective packet structures are presented in Tables 1-3 [3].

Table 1. Data request packet from the UWASA Node to the frequency converter. Data is read by the index.

Start Chars		Command	Object	Read Command Identifier	Index		Number of Variables		Stop Chars		CRC
DLE	STX	Read	Variable Object	Read by Index	Memory Location		In this application always 1		DLE	ETX	
0x10	0x02	0x40	0x0B	0x00	0x00 MSB	0xBC LSB	0x00 MSB	0x01 LSB	0x10	0x03	0x66

Table 2. Data request packet from the UWASA Node to the frequency converter. Data is read by the ID.

Start Chars		Command	Object	Read Command Identifier	ID		Number of Variables		Stop Chars		CRC
DLE	STX	Read	Variable Object	Read by ID	Parameter ID				DLE	ETX	
0x10	0x02	0x40	0x0B	0x01	MSB	LSB	0x00 MSB	0x01 LSB	0x10	0x03	0x66

Table 3. Data response packet from the frequency converter to the UWASA Node.

Start Chars		Command	Object	Data Type	Data		Stop Chars		CRC
DLE	STX	Response	Variable Object	Integer	Memory Contents		DLE	ETX	
0x10	0x02	0xC0	0x0B	0x03	0x00 MSB	0x00 LSB	0x10	0x03	0x52

UWASA Nodes are equipped with ZigBee radios and the communication in the WSN applies the IEEE 802.15.4 communication protocol in 2.4 GHz ISM band. After receiving the data response (Table 3) from the frequency converter, the UWASA Node, converts it to the packet form which is used in wireless communication. The general structure of the packet is presented in Table 4.

Table 4. General structure of the packet used in wireless communication.

Destination Node ID	Command Specification	Data	CRC
2 bytes	1 byte	N byte	1 byte

Communication between the data logging PC and the gateway node is done by using the USB connection. Data, which is wirelessly transmitted by the sensor nodes and collected by the gateway node, is sent to PC over this connection. Commands and requests, which are sent by the network control program executed by the PC, or which are manually sent by using the GUI in PC, are transmitted from the data logging PC to the gateway node by using this connection [3]. The general structure of the packet used in the connection between gateway node and data logging PC is presented in Table 5.

Table 4. General structure of the packet used in communication between the gateway node and the data logging PC.

Start	Length	Node ID	CS	Data	End or CRC
0xAA	1 byte	2 bytes	1 byte	n bytes	0x03

B. Communication Protocol

We applied star topology, where the nodes which are connected to frequency converters can communicate directly with the gateway node, which is connected to data logging PC. All sensor nodes are UWASA Nodes and they use FreeRTOS [6] as their operating system.

The software implementations of the communication protocol are different in the gateway node and the rest of the nodes. The gateway node communicates directly with every other node, and also with the data logging PC. The packet structures in the wireless communication and in the USB communication between the gateway and the data logging PC are different. As a consequence, the gateway must convert the data packets from one form to another [3].

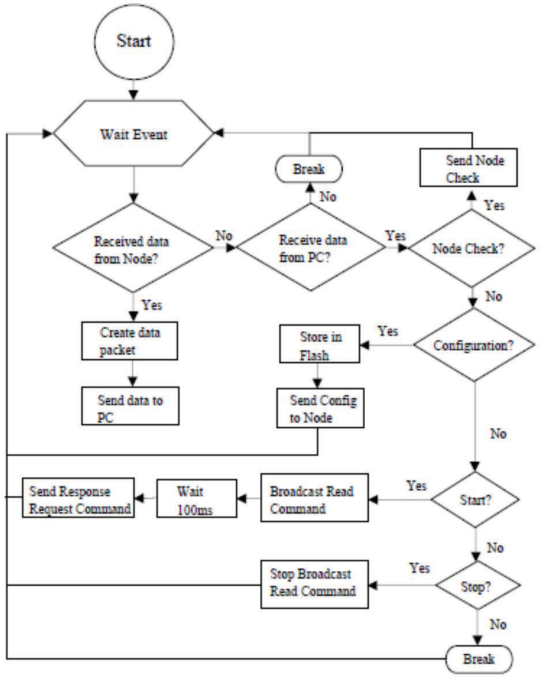


Figure 4. The software flowchart of the gateway node.

The software architecture of the gateway node is illustrated in Figure 4. Gateway node sends commands to other sensor nodes to read the frequency converter data and to transmit it to the gateway. It was discovered that usually it takes 50-90 ms to read the data from the frequency converter to the connected node. Thus, 100 ms timer was set between read and response request commands to provide enough time for the node to read the data and re-package it for the wireless transmission. To avoid data packet collisions, the response requests are sent to sensor nodes in sequence [3].

The software architecture of the sensor nodes that are connected to the frequency converters is illustrated in Figure 5. There are four types of commands that can be received from the gateway node: node check, configuration, read and response request. The operations that follow the respective commands are presented in the flowchart in Figure 5. The node will re-package the data it receives from the frequency

converter for wireless transmission, but it will not send it until it receives the response request command from the gateway node. This mechanism keeps the sequence so that collisions in wireless communication can be avoided [3].

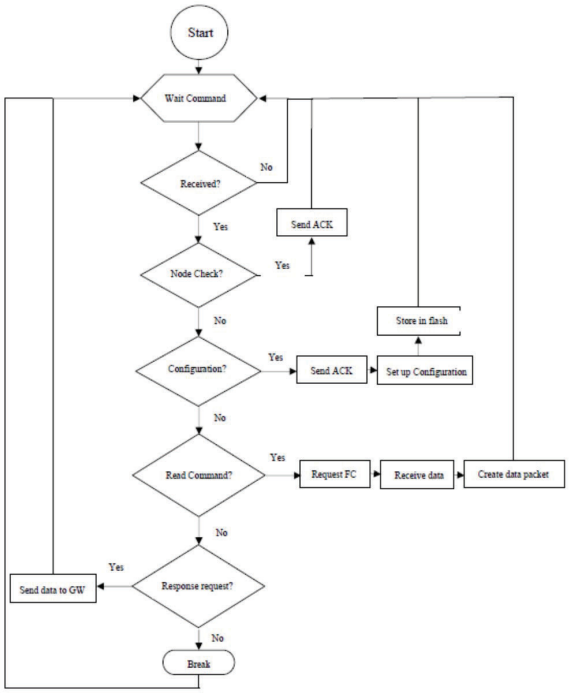


Figure 5. The software flowchart of the sensor nodes that are connected to the frequency converters.

IV. EXPERIMENTS

A. Setup

Experiments were done in real industrial environment. Data was collected over WSN from six frequency converters, which were controlling the electric motors of the mobile machine. Ten parameters were collected from each frequency converter in a sampling interval of 200 ms, which equals five times per second from each frequency converter. The location of the gateway node and the data logging PC was fixed, but the wireless sensor nodes and frequency converters were moving with the operating machine so that their distance to the gateway varied between 3 and 30 meters [3].

B. System Performance and Communication Reliability

The experiments indicated that the system was able to collect data from six frequency converters once in 200 ms, as targeted. Achieved total throughput of the system at the gateway node was 16.8 kbits/s. However, a packet loss was observed in the first experiments and then further analyzed.

The packet loss was tested by performing a set of 30 individual tests each of them having 10 packet loss measurements. Results are presented in Figure 6. Each point indicates the average packet loss percentage in 10 measurements, and the red line indicates the overall average packet loss over 30 sets of measurements. As shown in the figure, the packet loss varied between 9 and 11 percentage, and the average packet loss was 9.92%. Such a level was significant and a problem for the communication reliability.

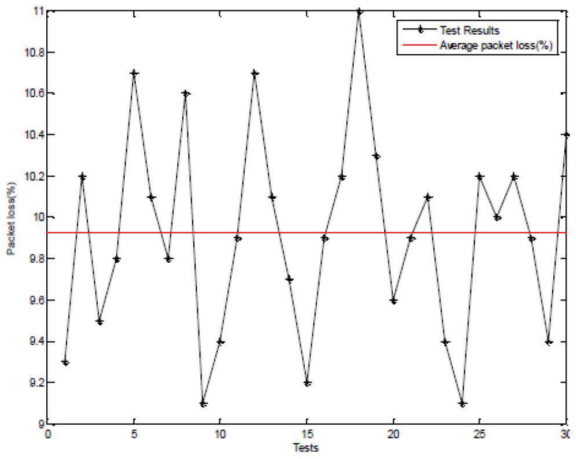


Figure 6. Packet loss before the implementation of the retransmission protocol. Overall average packet loss was 9.92%.

A resend request mechanism was developed and implemented to overcome the packet loss. First, the gateway node broadcasts the read command to all nodes and then after 100 ms, it starts to send response request commands to all six nodes one by one. After sending packet response request to one node, the gateway checks if it can take the response semaphore in 50 ms. If that happens, it indicates that the response from the node is received and the gateway node will send the response request to the next node. The software flowchart of the resend function is illustrated in Figure 7.

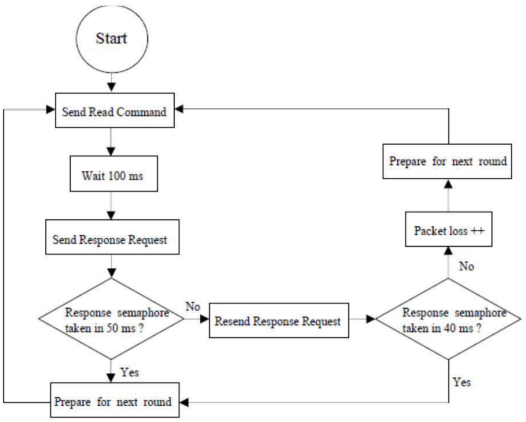


Figure 7. The software flowchart of the resend function.

If the response semaphore is not taken in 50 ms, it indicates that the response is not received and the gateway will re-send the response request for that node. After resubmission, the gateway waits now 40 ms, and if the response is still not received, the packet is counted as lost and the gateway moves to the next node [3].

When the packet loss experiment was repeated after the implementation of the resend request mechanism, it was observed that the packet loss rate was reduced so that it was between 0.5 and 2 percentage in the experiments and the overall average packet loss was 1.22% [3]. Results are presented in Figure 8. If higher communication reliability is still needed, it can be achieved by increasing the number of resend requests. If the receiving of every packet is critical, a handshake mechanism, that requires retransmission as many times as needed to receive the data, can be implemented. However, several retransmission requests will slow down the system so that collecting data from six frequency converters once in 200 ms may not be possible after that.

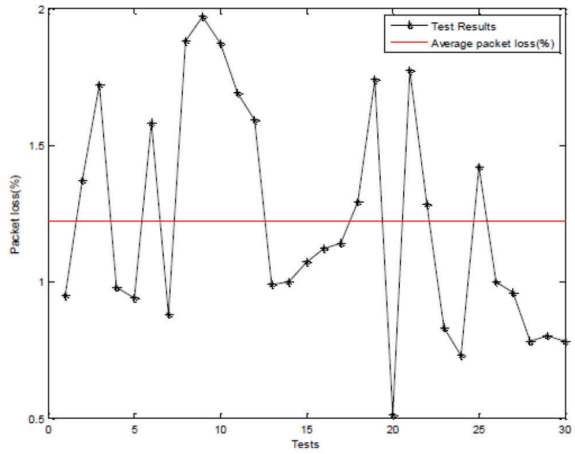


Figure 8. Packet loss after the implementation of the retransmission protocol. Overall average packet loss was reduced from 9.92% to 1.22%.

V. Conclusions and Future Work

In this paper, we presented a frequency converter integration to wireless sensor network. We used Vacon frequency converters and UWASA Node sensor nodes. Communication between the frequency converter and UWASA Node was done through RS-232 serial port by using VHMI protocol developed by Vacon. The UWASA Node uses FreeRTOS operating system. It is equipped with ZigBee radio, and the communication in WSN uses IEEE 802.15.4 protocol in 2.4 GHz ISM band.

Implemented system was tested in operating machine in real industrial environment. It was possible to collect information from six frequency converters once in 200 ms, every time ten parameters from each of them. Achieved system throughput in the gateway was 16.8 kbits/s. Packet loss rate was originally 9.92%, but after the implementation of the retransmission protocol, it was reduced to 1.22%. It can be reduced further by using the option of several retransmissions or by a handshake protocol, but both of them can also slow down the system performance.

The system is extendable. Since the data is collected over WSN, it is also possible to collect other types of sensor measurements and add other types of actuators to the same network. All communication is bidirectional. In addition to collecting data from the frequency converters, it is also possible to send commands to them over the WSN. This makes it possible to control the electric motors (via frequency converters) directly based on the measurements of the sensor network

In the star topology setup it was possible to communicate within the range of 30 meters in the experiments. Based on our experience about the UWASA Node, the communication range can be bigger, but to ensure longer distance communication in challenging environments, multi-hop communication should be applied. We have implemented multi-hop communication and respective routing and communication protocols for the UWASA Node with frequency converters [4]. Some basic testing for that system performance is already done, but more testing and experiments are needed to be able to better characterize its performance.

In the star topology setup the node power supply was not an issue, because the sensor nodes that were connected with the frequency converters got their power supply from them, and the gateway node from the data logging PC. If more sensor nodes and multi-hop deployments are applied, the nodes must be either battery powered or harvest their energy from their environment. In both cases the limitations caused by the power resources must be considered in the system design.

References

[1] Vacon frequency converters, http://www.vacon.com

[2] Yigitler, H., Virrankoski, R. and Elmusrati, M.S., “Stackable Wireless Sensor and Actuator Network Platform for Wireless Automation: the UWASA Node”, Aalto University Workshop on Wireless Sensor Systems, November 19th, 2010, Espoo, Finland

[3] Maiwulan Wulayinjiang, “Wireless Data Logger – a Joint Use of Frequency Converter and Wireless Sensor Network”, Master’s Thesis, University of Vaasa, Department of Computer Science, Communications and Systems Engineering Group, 2014.

[4] Le Manh Linh, “Wireless Sensor Network Integration with Frequency Converter by Using Multi Hop Protocol”, Master's Thesis, University of Vaasa, Department of Computer Science, Communications and Systems Engineering Group, 2014.

[5] Virrankoski, R. (Ed.), Generic Sensor Network Architecture for Wireless Automation (GENSEN), Proceedings of the University of Vaasa, Reports 174, Vaasa 2012.

[6] FreeRTOS, http://www.freertos.org

Solar Energy Harvesting Solution for the Wireless Sensor Platform The UWASA Node

Thomas Höglund[1], Reino Virrankoski[2] and Timo Mantere[2]
[1)]*University of Vaasa, Department of Electrical Engineering and Automation*
[2)]*University of Vaasa, Department of Computer Science, Communications and Systems Engineering Group*
P.O. Box 700, FI-65101, Vaasa, Finland
thomas.hoglund@tfif.fi, {reino.virrankoski, timo.mantere}@uva.fi

Keywords: Energy harvesting, energy management, energy storage, solar power generation, wireless sensor networks

Abstract: This paper presents a solar energy harvester and energy management prototype developed for the UWASA Node wireless sensor platform. The prototype was designed using a modular approach, requiring only minor hardware modifications in order to allow harvesting from different energy sources. The primary sensor network application for which the design was developed is wind turbine monitoring. The energy harvesting prototype and the performance level it enables for the sensor networking are evaluated through experiments, and methods of optimizing energy harvesting and energy management are discussed.

1 INTRODUCTION

Wireless sensor networks enable a range of completely new kinds of monitoring and control applications as a part of the Internet of Things concept. Even though wireless sensor nodes have been developed rapidly during the last decade, their power supply still constitutes a significant bottleneck for their applicability. Having to service a wireless sensor node and change its battery can be prohibitively expensive or difficult due to the location and means of installation of the sensor node. This greatly limits the number of feasible applications in which wireless sensor nodes would otherwise be perfectly suited for monitoring and control. Different types of energy harvesting systems have been developed to overcome this problem. A common challenge related to them is that the energy resources they are able to harvest usually enable a remarkably lower sensor node performance level compared with powering from a battery without energy harvesting. This level might not be enough to fill the requirements of the particular monitoring or control application.

In this paper we present a solar energy harvesting solution for the UWASA Node wireless sensor platform (Yigitler, 2010). The solution was developed as a part of our wireless automation research activities, and it is primarily targeted for wireless sensor network (WSN) applications for wind turbine monitoring (Höglund, 2014a, 2014b). It would be beneficial to collect information about different kinds of forces and vibrations that affect the wind turbine structures. The dimensions of the wind turbines used for industrial-scale electricity generation are so large that energy harvesting capability is a necessity to make the wireless sensor nodes feasible for monitoring and control installations. In addition to solar energy, energy harvesting from vibrations was also considered, and with small modifications, the developed energy harvester could be adapted to harvest vibrational energy.

The rest of this paper is organized as follows: The UWASA Node wireless sensor platform is introduced in Section 2. Methods of energy harvesting are discussed briefly in Section 3 and general requirements of the energy harvester in Section 4. The developed energy harvester prototype is described in Section 5 and the applied solar cell in Section 6. Maximum power point tracking is discussed in Section 7. The implemented energy management and storage is explained in Section 8, and the system performance is evaluated trough experiments presented in Section 9. Finally, Section 10 concludes the paper.

2 THE UWASA NODE

The UWASA Node, shown in Figure 1, is an open source wireless sensor node developed by Aalto University and the University of Vaasa (Yigitler, 2010). It is a modular and stackable platform, the software and hardware design of which allow it to be used for different types of applications with minimal changes to the main architecture. The possibility to stack different slave boards onto the main board allows the creation of custom solutions for any application. In its simplest form, called the basic stack, only the main module and the power module are used. These are sufficient to comprise a wireless sensor node that consists of processors, a wireless communication interface, peripheral interfaces, and power management and distribution (Çuhac, 2012; Virrankoski, 2012).

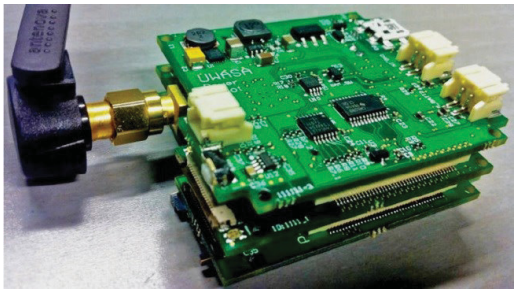


Figure 1: The UWASA Node with power module.

2.1 The Main Module

The main module of the UWASA Node contains two processors: one main controller and one radio frequency controller. The radio frequency controller can handle all computation and communication in simple applications, and then the main controller need not be used. For more demanding applications, the main controller is preferable. The main controller is an LPC2378 ARM7TDMI-S-based high-performance 32-bit RISC microcontroller from NXP Semiconductors.

2.2 Operating System and Software

The modularity of the UWASA Node is realized by both the hardware and the software architectures. The FreeRTOS (Free Real Time Operating System) was chosen for the UWASA Node in order to enable real-time operation and preemptive multitasking. The UWASA Node can thus handle many communication, measurement and control tasks simultaneously.

Middleware has been written for the UWASA Node to provide device drivers and hardware abstractions that are used to establish a uniform programming interface for both the main controller and the radio frequency controller. The same API functions can thus be used for programming both controllers. Automated daemons run in the background, taking care of tasks related to power management, time synchronization and system diagnostics (Çuhac, 2012).

2.3 Auxiliary Hardware

The UWASA Node can be connected to a number of slave modules by using the hardware stack connectors with a total of 160 pins per module. These connectors provide all necessary inter-modular connections for signals and power supplies. The slave modules can be any peripherals such as sensors, actuators and drivers.

2.4 Power Source and Energy Management

The energy management of the UWASA Node is handled by the power module, which is a separate module that can be stacked onto the main module. The power module features dynamic power path management and is capable of choosing the most suitable power source and charging a battery if one is connected and sufficient power is supplied. There is a battery monitoring chip that accurately measures current, voltage and temperature. This can be used for calculating the energy state of the battery and for measuring the power consumption of different applications (Çuhac 2012).

The battery input of the power module is designed for one-cell lithium ion batteries with a nominal voltage of 3.7 V. It accepts voltages between 1.8–4.2 V. A charger input features an undervoltage lock-out (UVLO) that cuts the power when the charger voltage is below 3.3 V. During undervoltage lock-out a very small, but nontrivial current (tens of milliamperes were measured) is drawn from the charger input. Similarly, a very small but nontrivial current flows into the battery input when the battery voltage is below 1.8 V and the charger is in short circuit mode. To eliminate this loss, an external, very low-power UVLO circuit is proposed for energy harvesting applications.

3 METHODS OF ENERGY HARVESTING

For outdoor WSN applications, such as wind turbine monitoring, solar energy harvesting is the most suitable energy harvesting method because of the good availability of sunlight and the proven technology of solar cells. Energy harvesters using sunlight as their energy source can provide power on the order of 10 mW/cm^2 under ideal circumstances (Höglund, 2014a).

The most efficient method of energy harvesting is always case-specific because of the large differences in the availability of energy over time from different sources and locations, and because of the highly varying power consumption of wireless sensor nodes (Höglund, 2014a). There are three methods of energy harvesting that were deemed feasible for supplying the UWASA Node with power in wind turbine monitoring applications: solar energy harvesting using photovoltaic (PV) cells, vibration energy harvesting using a piezoelectric cantilever, and wind energy harvesting using a microscale wind turbine with an electromagnetic generator. These three methods could also be used in parallel in a hybrid energy harvester.

3.1 Solar Energy Harvesting

Using a PV cell as the energy source would be a safe choice, because it is a well-established technology. Solar cells are readily available in all sizes and in many different configurations with conversion efficiencies around 15% (Gilbert and Balouchi, 2008). A suitable number of photovoltaic fingers should be connected in series in the cell to yield an optimum nominal output voltage and more fingers can be connected in parallel to cover the rest of the available area. The generic 92 × 61 mm, 0.45 W solar cell sold by SparkFun Electronics (Niwot, Colorado) is a suitable choice, because its open circuit voltage is approx. 5 V and its size roughly matches that of the UWASA Node. If more energy is required, it is possible to connect more than one such cell in parallel to the energy harvester, while still keeping the maximum power point (MPP) voltage and energy harvesting circuit the same.

PV cells are made for outdoor use and are not damaged by rain or large temperature changes. Energy can reliably be harvested from them whenever the ambient illuminance is sufficiently high. A suitable harvesting schedule can be estimated by analyzing weather data to determine how much energy can be generated on average at a given time of day and time of year. A large fraction of the available energy is lost when the PV-cell is not oriented directly against the sun, but this is typically unavoidable. If possible, the PV-cell should be oriented in the direction of average maximum sunlight. The reflectiveness of the surroundings highly influences the received energy and a heavy cloud cover reduces the available energy by approximately an order of magnitude (Gilbert and Balouchi, 2008). In the worst case, the PV-cell will experience sufficiently bright conditions for so short a time that it cannot harvest enough energy for the load to operate. Seasonal and weather conditions may make it impossible to harvest a sufficient amount of energy for a long time and therefore it is important to store enough energy in the sensor nodes for them to be able to operate during such times.

3.2 A Hybrid Energy Harvester

Several different energy sources can be harvested simultaneously by using a modular energy harvester. Harvesting both solar and wind or vibrational energy would reduce the downtime of the harvester and produce power more evenly. The largest drawback of using several sources is the increased requirement for space. Park and Chou (2006) developed a modular energy harvesting system called AmbiMax. They propose to use a reservoir capacitor array, i.e. a separate supercapacitor for each energy harvester. These supercapacitors need to be able to reach the same voltage in order to power the common voltage rail. If the voltage over one of the capacitors is higher than that of the others, only that one will supply the voltage rail. If more than one capacitor is used like this, diodes may be necessary to prevent backflow from the voltage rail to the capacitors. Diodes should be avoided when possible, because they cause a small voltage drop and power loss. An energy harvester that outputs less power than the other harvesters needs to have a smaller supercapacitor so that it can reach the target voltage quickly enough to be efficient (reaching its maximum power point). The voltage rail can be used to power the wireless sensor node and/or a battery charger.

4 ENERGY OPTIMIZATION

When using an energy harvester to power a wireless sensor node, there are many aspects that must be

considered when seeking optimal performance to harvest as much energy as possible and to store and use the harvested energy as efficiently as possible. If one part of the system is wasteful, it is not very helpful to get another part of the system to operate efficiently. Some of the parts of the system that must be considered when seeking optimal overall system performance are: harvesting location and schedule, size of the electronics, energy conversion, voltage conversion for storage, electrical switching, energy storage, voltage conversion for consumption, energy consumption of the load circuit, sensor node program execution, wireless communication scheme, and transmission power.

There are complex tradeoffs to be considered when selecting components for the energy management and storage connected to an energy harvester. When the values of the voltage and current of an energy source result in a maximum power output, the circuit is said to be operating at the maximum power point (MPP). The MPP voltage varies with ambient conditions. This voltage may not be optimal for the energy harvesting circuit and voltage regulator that drain the source and supply the voltage rail or storage device with a suitable voltage. Thus, significant power may be lost if the source and the harvesting circuit are not well matched.

The energy harvesting circuit is also optimally efficient at a certain output voltage. For example, step-up DC/DC converters are the most efficient when their output voltage is only slightly below the input voltage. When such a converter is charging a battery or a supercapacitor, its output voltage will gradually increase as the load is charged, and the conversion may be efficient only for a short time. For this reason, supercapacitors are commonly used as buffers to allow the output voltage to rapidly climb to a suitable voltage when harvesting, and then battery charging begins when the optimum voltage is exceeded, thus keeping the output at this voltage until the battery is charged to a higher voltage, after which the efficiency again decreases as the voltage closes in on the setpoint.

Voltage regulators supplying the sensor node cause an additional power loss that depends on the regulator type and its input and output voltages. It can be very difficult to optimize the overall system performance when so many components must be considered. As a rule of thumb, in electronics design, the optimum voltages of all components should be as close to each other as possible.

A truly optimized energy harvesting system should take into account the limitations of the energy storage circuit and the draining schedule of the storage over time. The wireless sensor node needs to work intermittently and go into sleep mode at certain times in order to conserve energy for future measurements, data logging, and transmissions. The schedule could also be changed by the sensor node based on measurements of the environment. For example, the system has to take into account that no power is harvested from a photovoltaic harvester at night. Schedules of harvesting and consumption can be simulated using computer models before they are tested in hardware in order to make the best use of the harvested energy.

5 THE ENERGY HARVESTING PROTOTYPE

After measuring the typical power consumption of the UWASA Node and investigating what forms of energy harvesting would be suitable, the energy harvester prototype shown in Figure 2 was designed and built. The design was made with modularity and expandability in mind. The harvester was designed to work with a small solar cell, but other sources can be added in parallel if some modifications are made (Höglund, 2014b).

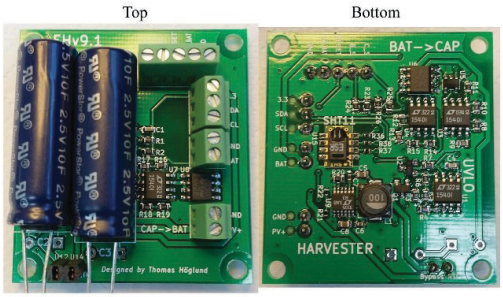


Figure 2: Developed energy harvester prototype.

The chosen implementation is based on the AmbiMax system described by Park and Chou (2006). It is an entirely analog energy harvesting system that was relatively efficient when it was made in 2006, but the power consumption of common, low-power digital controllers has since dropped significantly, making them a viable alternative. The maximum power point tracking was not implemented in the same way in this project as in the AmbiMax. The LTC3105 energy harvesting IC was chosen to perform the harvesting. Since 2013

when this choice was made, some even more efficient energy harvesting ICs have become available (Höglund, 2014b). The energy management was designed with components similar to those of the AmbiMax, but slightly more efficient (Höglund, 2014b).

The architecture of the AmbiMax platform is shown in Figure 3, reproduced from Park and Chou (2006). It consists of a comparator with hysteresis that performs MPPT with the aid of a sensor and controls a boost regulator. The regulator charges a supercapacitor that is connected to the voltage rail. All of these components can be grouped as a subsystem and used in parallel if more than one energy source is used. The supercapacitors are connected to the voltage rail via optional protection circuitry, and the voltage rail powers the sensor node. If the voltage of the voltage rail increases above a certain threshold, the battery is charged from the voltage rail via a current limiter. Conversely, if the voltage of the voltage rail drops to below a certain threshold, the battery feeds the voltage rail as long as its voltage is above another fixed threshold. This, in short, is how the AmbiMax and the developed energy harvester work. Additionally, a low-power undervoltage lock-out circuit and a real-time clock-controlled latch switch were designed to cut off the voltage rail from the sensor node when it drops below a threshold or when the node signals it to shut down for a length of time; these were not part of the AmbiMax.

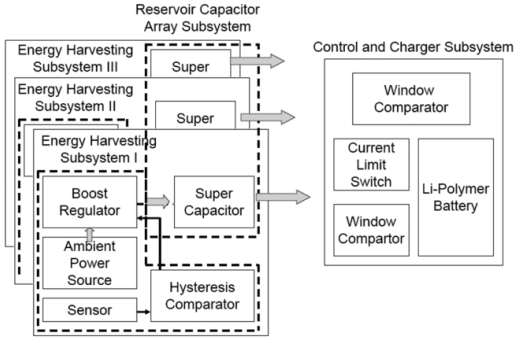


Figure 3: The architecture of the AmbiMax Platform (Park and Chou, 2006).

The LTC3105 energy harvesting IC by Linear Technology (Milpitas, California) was chosen from many alternatives to be the energy harvester used in the prototype of this work. It is listed as a 400 mA step-up DC/DC converter with MPP control and 250 mV start-up voltage. It is capable of supplying up to 5.25 V. The prototype is designed to supply 4.2 V, which is the maximum voltage of a one-cell lithium-ion battery. The very low start-up voltage of the LTC3105 allows it to harvest from a photovoltaic cell that outputs a low voltage due to low ambient illuminance. The low input voltage compatibility can also be useful for other types of energy harvesting sources such as thermoelectric, electromagnetic, or magnetostrictive sources, which output a low voltage.

6 THE SOLAR CELL

A 92 mm × 61 mm solar cell, with a nominal power of 0.45 W, was chosen for the energy harvester prototype, because its open circuit voltage is approximately 5 V, which is suitable for the LTC3105 and the battery, and its size is approximately that of the UWASA Node's. If more energy is needed, it is possible to connect more than one such solar cell in parallel with the other cells to the energy harvester, while still keeping the MPP voltage and energy harvesting circuit the same. Protection diodes could be used to allow operation with solar cells of higher voltages, but the LTC3105 operates most efficiently at input voltages slightly lower than its output voltage, and therefore the MPP voltage of the solar cell should be lower than the desired output voltage.

In order to measure the MPP of the solar cell, it was connected to a potentiometer used as a variable load. It was then placed under a constant illuminance of 2.8 klx and its output current and voltage were measured while varying the load. The output power was calculated, and the result is plotted in Figure 4. The MPP occurs at approximately 3.6 V and 6.3 mW. The MPP varies slightly with the illuminance, but after a few attempts at maximum power point tracking, it was decided that a fixed MPP voltage is sufficient for this application.

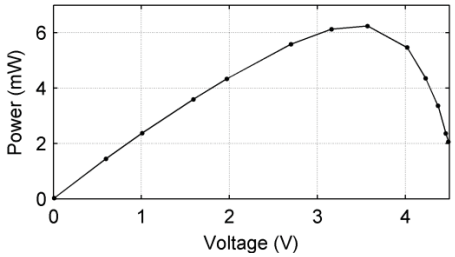


Figure 4: Power vs. voltage for the 0.45 W solar cell at 2.8 klx.

7 MAXIMUM POWER POINT TRACKING

Maximum power point tracking (MPPT) aims to adapt the energy harvesting load to the ambient conditions so that the input voltage of the energy harvester is always equal to the MPP voltage as it varies, in effect performing impedance matching.

In the case of the LTC3105, the MPPT is integrated on the chip and there is a pin named MPPC. The LTC3105 keeps the source voltage the same as the voltage on the MPPC pin, which constantly outputs 10 µA. If MPPT is not necessary, this pin can be connected via a fixed resistor R_{MPPC} to ground in order to set the MPP to a fixed voltage (U_{MPPT}) according to (1). In the prototype, a 360 kΩ resistor was used to achieve a U_{MPPT} of 3.6 V.

$$U_{MPPT}=10\ \mu A^{*}\ R_{MPPC} \quad (1)$$

The datasheet of LTC3105 proposes to use a diode thermally coupled to the solar cell for MPPT, but this is unlikely to work well over the large temperature range of this application; it would also be difficult to achieve thermal coupling.

MPPT could also be performed digitally by a low-power microcontroller, digital signal processor, or field-programmable gate array. It is easier to calculate the MPP digitally and take several factors into account, such as illuminance and temperature, but unless such a digital control system is carefully designed, not much power can be saved.

8 ENERGY MANAGEMENT AND STORAGE

The energy management part of the circuit takes care of routing the power in an optimal way between the energy harvester and sensor node components for maximum performance and optimal schedule of operation. The energy management of the energy harvester prototype consists of supercapacitors, a Li-ion polymer battery, nanopower voltage comparators, a logical AND gate, two MOSFETs and a few current limiters.

The LiPo battery is charged by two supercapacitors connected in series when the supercapacitors reach a voltage threshold. The charge current is limited by a current limiter that also works as a switch. Charge current flows intermittently due to a configured hysteresis, until 4.2 V is reached. The voltage thresholds at which power is transferred in the prototype between the supercapacitors, the battery, and the UWASA Node are governed by LTC1540 nanopower voltage comparators by Linear Technology (Milpitas, California). These comparators feature an ultralow quiescent current of nominally 0.3 µA, a voltage reference, and a hysteresis, both adjustable by resistor voltage dividers.

One comparator is used for activating the current flow from the voltage rail (supercapacitor) to the battery when the rail voltage is more than 3.7 V. Another comparator is used in the undervoltage lock-out (described in Section 8.3), and a pair of comparators with an AND gate is used for activating the current flow from the battery to the voltage rail. All comparators were configured for a hysteresis of approximately 100 mV.

The comparator that activates battery charging and the AND gate that activates battery draining are connected to the enable pins of two separate current limiters that, when enabled, permit a limited current flow through them in one direction. These current limiters were implemented using TPS2030D power distribution switches by Texas Instruments (Dallas, Texas). They allow 300 mA to pass through them when activated.

8.1 Supercapacitors

Supercapacitors can act as a buffer and be used to store the first energy delivered by the energy harvester until there is enough energy to begin charging the battery or supplying the sensor node. The voltage of the supercapacitor can rise quickly to a voltage where the step-up (or step-down) converter operates the most efficiently because of its much lower capacity compared with a battery. Connecting the harvester directly to the battery would cause its voltage to rise very slowly and energy would be harvested less efficiently because of the step-up inefficiency at lower voltages. Supercapacitors can also smooth out the wide dynamic range of energy harvesters and the node load, especially if more than one harvester subsystem is connected in parallel. Another advantage of using supercapacitors is that they can be used to preferentially supply the sensor node before the battery is needed. This keeps the battery voltage more even, which slows down battery aging.

According to Mars (2009), (2) gives an approximation for the necessary capacitance C of the supercapacitor assuming there is a constant load current I_L and that the supercapacitor needs to be able to supply I_L for time t. When current is drawn

from a supercapacitor, there is an instantaneous voltage drop due to its equivalent series resistance R_{ESR}. The load voltage is allowed to decrease from U_{max} to U_{min}. Equation (2) shows that an approximately 12 F supercapacitor is necessary to supply 250 mA for 60 seconds with the voltage limits of the developed prototype. In the actual case, the current would vary significantly over time, but this equation provides a useful indication of how large a capacitor is required.

$$C = \frac{I_L t}{U_{max} - U_{min} - I_L R_{ESR}} = \quad (2)$$

$$= \frac{250*10^{-3}\ \text{A}*60\ \text{s}}{4.2\ \text{V} - 2.9\ \text{V} - 250*10^{-3}\ \text{A}*200^{-3}\ \Omega} = 12\ \text{F}$$

8.2 Switch Controlled by Real-Time Clock

A real-time clock (RTC) was added to the prototype so that the sensor node can cut off its own power supply in order to avoid consuming any energy on the node side while it is in sleep mode. The RTC has an alarm output that can be set to trigger at the point in time when the node should be powered on. The RTC consumes only a few microamperes of current. The alarm output is connected to a latch IC that turns on or off the current flow through two MOSFETs that supply the node with power. The sensor node can request the RTC to activate the latch at a specific time in the future, turning the power supply on at that time, and then use the reset line of the latch to shut itself down. An SHT11 temperature and humidity sensor was also included on the PCB on the same I^2C bus as the RTC because temperature and humidity measurements are needed in wind turbine monitoring.

8.3 Undervoltage Lock-Out Circuit

An undervoltage lock-out (UVLO) circuit was designed to cut off the power from the sensor node when the voltage rail is below 2.9 V. There is a 20 MΩ feedback resistor that creates an extra high hysteresis of 350 mV to allow enough energy for the sensor node to wake up and measure the voltage without allowing the turn-on current surge and any startup tasks to drain the voltage rail below the UVLO threshold again. The switching is done using a 2N7002 small signal N-channel MOSFET and an IRLML6401 P-channel power MOSFET.

9 THE PERFORMANCE OF THE PROTOTYPE

The lowest level of illuminance at which the LTC3105 was able to harvest was a few hundred lux, depending on the voltage of the supercapacitor. The energy harvester prototype was tested in a long-term test that lasted six continuous days. The solar cell was located on a roof where it was not shadowed by any object at any time of the day. There was no load connected to the prototype. A

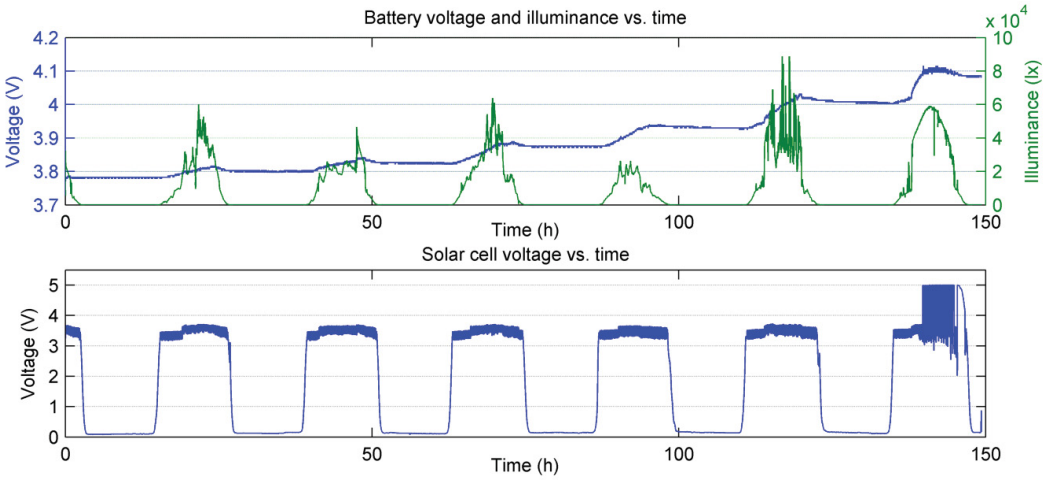


Figure 5: Energy harvester performance over six days.

data logger was connected and used for measuring time, illuminance, and the voltages of the supercapacitor, battery, solar cell, and MPPC pin of the LTC3105. During the test, the temperature was a few degrees C below the freezing point. Figure 5 shows how the prototype performed. On average, the energy harvester was active for 9.0 hours per day (the sunny hours) and harvested at 35.6 mW. On average, 1.16 kJ was harvested per day, or 2.14 J per minute active. In 6 days, the total energy harvested was 6.9 kJ, which corresponds to 51% of the capacity of the 1,000 mAh, 3.7 V LiPo battery. Once the battery was fully charged, the voltage rail reached the set point of the energy harvesting circuit and the solar cell was automatically disconnected, causing a voltage of more than 5 V over the solar cell.

For most applications, one solar cell of the type tested should be sufficient and the 1,000 mAh battery capacity is useful to have to ensure the sensor node can operate during days of low illuminance. The solar cell, battery and energy harvester of the prototype were well-dimensioned.

Regarding the energy consumption of the UWASA Node, experiments showed that the startup and initialization of wireless communication and a few sensors consumes between 0.7 and 1.5 J. Measuring three voltages 10,000 times using the internal ADC consumes approx. 400 mJ (no peripherals turned off). Transmitting 100 bytes of data consumes ~850 mJ. Measuring 3-axis, 10-bit acceleration at a sample rate of 500 Hz for 2 s consumes 1.82 J. A typical program reading several sensors at a high rate will consume approx. 3-10 J for measurements and 50-100 J for transmission of thousands of bytes. If few bytes are transmitted, the node will consume less than 5 J and can thus operate intermittently at an interval of 3-4 minutes on harvested power.

10 CONCLUSIONS

The goal of this work was to build and test a small energy harvester and power management prototype optimized for the UWASA Node for outdoor use in cold weather, primarily for wind turbine monitoring applications. The developed energy harvester was tested using only a solar cell, but the prototype was designed so that more energy harvesting sources can easily be added. Every part of the energy harvester and power management was chosen to operate at voltages optimal for the UWASA Node with power module. The energy measurements presented in Section 9 can be useful for energy harvester developers. The presented prototype is an improvement on the AmbiMax system described by Park and Chou (2006). By integrating the RTC switch on the energy harvesting PCB, the power consumption of any connected sensor node can be eliminated when inactive.

Powering the UWASA Node by energy harvesting is a useful idea, as it makes the node self-sufficient and allows it to operate in places where servicing would be prohibitively expensive or impossible. By using energy harvesters, wireless sensor nodes can potentially operate independently for several years, if the rest of the software and hardware platform is sufficiently robust.

REFERENCES

Çuhac, C., 2012. UWASA Node Reference Manual 3.0.0. [Unpublished, internal document]25 August. Aalto University and University of Vaasa, ComSys group.

Gilbert, J.M. and Balouchi, F., 2008. Comparison of Energy Harvesting Systems for Wireless Sensor Networks. International Journal of Automation and Computing, [online]. University of Hull. Available at: <http://web.eecs.umich.edu/~prabal/teaching/eecs598-w10/readings/GB08.pdf>.

Höglund, T., 2014a. Energy Harvesting for the UWASA Wireless Sensor Node: Applications for Wind Turbine Monitoring. B.Sc. University of Vaasa.

Höglund, T., 2014b. Energy Harvesting Solution for the UWASA Node: Applications for Wind Turbine Monitoring. M.Sc. University of Vaasa.

Mars, P., 2009. Using a Supercapacitor to Manage Your Power. IDTechEx Ltd, [online]14 December. Available at: <http://www.energyharvestingjournal.com/articles/using-a-supercapacitor-to-manage-your-power-00001921.asp>. [Accessed 24 August 2014].

Park, C. and Chou, P.H., 2006. AmbiMax: Autonomous Energy Harvesting Platform for Multi-Supply Wireless Sensor Nodes. *Sensor and Ad Hoc Communications and Networks, 2006. SECON '06. 2006 3rd Annual IEEE Communications Society,* [online]. Available through: IEEE Xplore Digital Library website [Accessed 18 March 2013].

Virrankoski, R. (Ed.), Generic Sensor Network Architecture for Wireless Automation (GENSEN), Proceedings of the University of Vaasa, Reports 174, Vaasa 2012.

Yigitler, H., Virrankoski, R. and Elmusrati, M.S., Stackable Wireless Sensor and Actuator Network Platform for Wireless Automation: the UWASA Node, Aalto University Workshop on Wireless Sensor Systems, November 19th, 2010, Espoo, Finland.