Joakim Flathagen

A routing and cross-layering approach for energy and bandwidth efficiency in Wireless Sensor Networks

Thesis for the degree of Philosophiae Doctor

Trondheim, October 2013

Norwegian University of Science and Technology
Faculty of Information Technology, Mathematics
and Electrical Engineering
Department of Telematics

NTNU – Trondheim
Norwegian University of
Science and Technology

NTNU
Norwegian University of Science and Technology

Thesis for the degree of Philosophiae Doctor

Faculty of Information Technology, Mathematics
and Electrical Engineering
Department of Telematics

ISBN 978-82-471-4659-0 (printed ver.)
ISBN 978-82-471-4660-6 (electronic ver.)
ISSN 1503-8181

Doctoral theses at NTNU, 2013:266

Printed by NTNU-trykk

Abstract

A Wireless Sensor Network (WSN) consists of a large number of small, low-cost and low-power wireless sensing nodes. WSNs can gather information about the environment automatically and unattended and are suitable for many applications.

The typical characteristic of WSNs is that they are *energy and bandwidth* constrained. Hence, routing protocols and algorithms for WSN must aim to conserve these two scarce resources. WSNs are also highly *application-specific*. This mean, firstly, that there is a tight bound between the application layer and the different protocol layers. Secondly, there are some WSN target applications that require certain protocol functionality that is not mandatory for other WSNs. In other words, both the general challenges and the specific application challenges must be addressed.

This thesis aims to address routing in WSNs both from a general and an application specific perspective. Among the general energy-and bandwidth related topics the work in this thesis focuses on aggregation and routing-efficiency. Among the application-related topics the work focuses on localization and interoperability.

The main contributions are:

- A method for letting the routing protocol contribute in node localization.
- A method for increasing the energy and bandwidth utilization with passive clustering.
- A method for increasing the energy and bandwidth utilization using multiple sinks.
- A data-aggregation scheme for WSNs that interoperates with external networks via a standardized interface.
- A hybrid routing mechanism that are able to operate in high-interference scenarios.
- Lessons learned from a real-world test campaign of a surveillance WSN.

Preface

This thesis is submitted to the Norwegian University of Science and Technology (NTNU) as partial fulfillment of the requirements for the degree of philosophiae doctor. The work for this dissertation started in August 2009. The thesis period includes a paternity leave period of six months.

The work was mainly carried out at the Norwegian Defence Research Establishment (FFI), under supervision of Professor Øivind Kure at the Norwegian University of Science and Technology (NTNU) and Professor Paal Einar Engelstad at the University of Oslo and FFI.

The thesis work was financed partially by the Norwegian Defence Research Establishment and partially by University Graduate Centre Kjeller (UNIK). The research has been part of the FFI-projects “1141 - Situational Awareness Sensor Systems (SASS)” and “1196 - Force protection for land based operations”.

Acknowledgments

First and foremost, I would like to thank my supervisors, Professors Øivind Kure and Paal E. Engelstad. You have both been motivating discussion partners, have provided the necessary guidance and have been paramount to realize this thesis.

I would also like to thank my former colleague at FFI, Professor Knut Øvsthus, now at Bergen University College, who persuaded me to pursue a PhD. My project manager at FFI, Rune Lausund, deserves thanks for acknowledging my ambitions and for giving me the opportunity to carry out the work towards a PhD. Rune has allowed me to concentrate exclusively on my research but has also included me in many interesting discussions and work outside my narrow research scope. This has undoubtedly broadened my perspectives.

My gratitude also goes to all my colleagues at the Information Management Division at FFI for discussions, feedback and support. Especially to Erlend Larsen and Vinh Pham for showing much interest in my research topics and Reinert Korsnes for helping me with the first article. I will also thank the rest of the SASS-group for their hard work in hacking sensor networks. My gratitude also goes to my research group at the Protection Division at FFI. Especially I would like to thank Lars Erik Olsen who had to do all the work I did not have the time to do while I was working on my PhD, and to Svein Rollvik who always has had confidence in me.

Furthermore, I would like to thank the management staff at both FFI, UNIK and NTNU for coordinating the financing of the PhD and for sorting things out whenever I forgot a deadline or did not show up where I should have.

Finally, and most importantly, I want to thank my small family, Elin and our daughter Ylva (who was born halfway through the PhD work). Elin has always motivated me and given me highly appreciated encouragement and support during these years.

List of Publications

The author of this thesis has the primary authorship of papers A through F (appended as Part II of the thesis.) The papers B, C, and E are co-authored with the external, technical supervisors, while the papers A, D and F are co-authored with colleagues at FFI. In addition, the author of this thesis has contributed to papers G to J with simulation code, analysis and as a discussion partner. All papers were published through peer-reviewed conferences and journals.

PAPER A: J. Flathagen and R. Korsnes, "Localization in Wireless Sensor Networks based on Ad hoc Routing and Evolutionary Computation", In proceedings of the IEEE Military Communications Conference, MILCOM, San Jose, CA, USA, October 31–November 4, 2010, pp. 1173–1178, ISBN: 978-1-4244-8179-8.

PAPER B: J. Flathagen, O. V. Drugan, P. E. Engelstad and Ø. Kure, "Increasing the Lifetime of Roadside Sensor Networks using Edge-Betweenness Clustering", In proceedings of IEEE International Conference on Communications, ICC 2011, Kyoto, Japan, 5-9 June, 2011, pp. 1-6, ISBN: 978-1-61284-232-5.

PAPER C: J. Flathagen, Ø. Kure, and P. E. Engelstad, "Constrained-based Multiple Sink Placement for Wireless Sensor Networks", Presented in WiSARN-FALL 2011, In workshop proceedings of IEEE International Conference on Mobile Ad-hoc and Sensor Systems, MASS 2011, pp. 783-788, ISBN: 978-1-4577-1345-3.

PAPER D: J. Flathagen and F. T. Johnsen, "Integrating Wireless Sensor Networks in the NATO Network Enabled Capability using Web Services", In proceedings of the IEEE Military Communications Conference, MILCOM, Baltimore, MD, USA, November 7–10, 2011, pp. 828–833, ISBN: 978-1-4673-0080-3.

PAPER E: J. Flathagen, E. Larsen, P. E. Engelstad and Ø. Kure, "O-CTP: Hybrid Opportunistic Collection Tree Protocol for Wireless Sensor Networks", Presented in SenseApp 2012, In workshop proceedings of the IEEE Conference on Local Computer Networks, LCN, 2012, pp. 943 - 951, ISBN:978-1-4673-2129-7

PAPER F: J. Flathagen, R. Korsnes, V. Pham, T. M. Mjelde, J. Sander, "Experiences from deploying a Wireless Sensor Network for Military Base Protection", Submitted to IEEE Sensors Journal, 2013.

Related papers:

PAPER G: F. T. Johnsen, J. Flathagen, T. Hafsøe, "Pervasive Service Discovery across Heterogeneous Tactical Networks", In proceedings of the IEEE Military Communications Conference (MILCOM), October 18-21, 2009, pp. 1–8, ISBN: 978-1-4244-5238-5.

PAPER H: E. Larsen, J. Flathagen, V. Pham and L. Landmark, "iOLSR: OLSR for WSNs Using Dynamically Adaptive Intervals", In proceedings of the International Conference on Sensor Technologies and Applications, SENSORCOMM, Nice, France, August 21, 2011, pp. 18–23, ISBN: 978-1-61208-144-1.

PAPER I: E. S. Boysen, J. Flathagen, "Using SIP for seamless handover in heterogeneous networks", In proceedings of the IEEE Congress on Mobile Computing and Network Technologies, October 5-6, 2011, pp. 1–8, ISBN: 978-1-4577-0682-0.

PAPER J: E. Larsen, J. Flathagen, V. Pham and L. Landmark, "Adapting OLSR for WSNs (iOLSR) Using Locally Increasing Intervals", Sensors & Transducers Journal, Vol.14-1, Special Issue, March 2012, pp.254-268, ISSN 1726- 5479.

Contents

Abstract iii

Preface v

Acknowledgments vii

List of Publications ix

Contents xi

List of Terms and Acronyms xvii

I Introduction 1

1 Introduction 3
- 1.1 Background 3
- 1.2 Motivation and challenges 4
- 1.3 Overview of the work 6
- 1.4 Scope 9
- 1.5 Research Methods 10
  - 1.5.1 Simulations 10
  - 1.5.2 Test-beds 11
  - 1.5.3 Field testing 12
- 1.6 Thesis Outline 12

2 Related work 15
- 2.1 Introduction 15
- 2.2 Routing 16
  - 2.2.1 Metrics 16
  - 2.2.2 Point-to-point routing 17

    - 2.2.3 Multipoint-to-Point routing 20
    - 2.2.4 Opportunistic routing 23
  - 2.3 Cross-layering 24
    - 2.3.1 Introduction 24
    - 2.3.2 Cross-layering for application support 26
    - 2.3.3 Cross-layering to improve energy and bandwidth utilization 26
    - 2.3.4 Implementing cross-layer interfaces 28
  - 2.4 Selected related work 29

3 Contributions **31**
  - 3.1 A summary of the contribution as a whole 31
    - 3.1.1 A thematic overview 31
    - 3.1.2 A chronological overview 32
  - 3.2 Contribution of paper A 34
    - 3.2.1 Related Work 34
    - 3.2.2 Contributions 35
  - 3.3 Contribution of paper B 37
    - 3.3.1 Related Work 38
    - 3.3.2 Contributions 39
  - 3.4 Contribution of paper C 41
    - 3.4.1 Related Work 41
    - 3.4.2 Contributions 42
  - 3.5 Contribution of paper D 44
    - 3.5.1 Related Work 45
    - 3.5.2 Contributions 46
  - 3.6 Contribution of paper E 47
    - 3.6.1 Related Work 48
    - 3.6.2 Contributions 49
  - 3.7 Contribution of paper F 50
    - 3.7.1 Related Work 50
    - 3.7.2 Contributions 50

4 Conclusion **53**
  - 4.1 Summary 53
  - 4.2 Future work 55

Bibliography **57**

II Research papers 69

A Localization in Wireless Sensor Networks based on Ad hoc Routing and Evolutionary Computation 71

B Increasing the Lifetime of Roadside Sensor Networks using Edge Betweenness Clustering 79

C Constrained-based Multiple Sink Placement for Wireless Sensor Networks 87

D Integrating Wireless Sensor Networks in the NATO Network Enabled Capability using Web Services 95

E O-CTP: Hybrid Opportunistic Collection Tree Protocol for Wireless Sensor Networks 103

F Experiences from deploying a Wireless Sensor Network for Military Base Protection 115

List of Figures

1.1 Structure of the thesis work 6

3.1 Evaluation methods used in the thesis 33
3.2 Initialization of a sensor network. 36
3.3 Roadside surveillance with clustered sensors 38
3.4 Four scenarios used in network simulations 44
3.5 Connecting a sensor network to an external infrastructure 45
3.6 A sensor network testbed with an interference source 48

List of Terms and Acronyms

6LoWPAN	IPv6 over Low power Wireless Personal Area Networks
API	Application Programming Interface
AOA	Angle Of Arrival
AODV	Ad hoc On-Demand Distance Vector Routing
CoRE	Constrained RESTful Environments
CTP	Collection Tree Protocol
CTS	Clear To Send
DIO	DODAG Information Object
DODAG	Destination Oriented Directed Acyclic Graph
DYMO	Dynamic MANET On-Demand Protocol
ETX	Expected Transmission Count
ExOR	Extremely Opportunistic Routing
GeRaF	Geographic Random Forwarding
GPS	Global Positioning System
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
ILP	Integrated Layer Processing
IP	Internet Protocol
LLN	Link Layer Notification
LOAD	6LoWPAN Ad Hoc On-Demand Distance Vector Routing

LOADng	The Lightweight On-demand Ad hoc Distance-vector Routing Protocol - Next Generation
LLN	Low power and Lossy Network
LRREQ	Localization Route Request
LRREP	Localization Route Reply
LQI	Link Quality Indicator
MAC	Medium Access Control
MANET	Mobile Ad hoc NETwork
MHC	Minimum Hop Count
MP2P	Multipoint-to-Point
MPR	MultiPoint Relay
MRHOF	Minimum Rank Objective Function with Hysteresis
NACMTC	Norwegian Army Combat Maneuver Training Centre
NNEC	NATO Network Enabled Capability
NS-2	Network Simulator 2
O-CTP	Opportunistic Collection Tree Protocol
OF	Objective Function
OLSR	Optimized Link State Routing
OSI	Open Systems Interconnection
P2P	Point-to-point
P2MP	Point-to-Multipoint
PDR	Packet Delivery Ratio
PIR	Passive Infrared
RERR	Route Error
REST	Representational State Transfer
RoLL	Routing over Low-power and Lossy Networks
RPL	IPv6 Routing Protocol for Low power and Lossy Networks

RREP	Route Reply
RREQ	Route Request
RSSI	Received Signal Strength Indicator
RTS	Request To Send
SOAP	Simple Object Access Protocol
TCP	Transmission Control Protocol
TOSSIM	TinyOS Simulator
TTL	Time To Live
TYMO	DYMO on TinyOS
UGS	Unattended Ground System
WSN	Wireless Sensor Network
XML	Extensible Markup Language

Part I

Introduction

Chapter 1

Introduction

1.1 Background

Recent advances in wireless communication and miniaturized electronics have enabled the development of small, low-cost and low-power wireless sensing nodes. Such tiny sensor nodes with sensing, processing and communication components autonomously form a Wireless Sensor Network (WSN). The WSN concept can gather information from the environment automatically and unattended. WSNs provide significant improvement for many sensing applications in terms of ease of use and their ability to cope with node failures.

The developments within microelectronics have substantially decreased the energy-consumption per bit for both computing and communication. Thus, a system lifetime of months or even years is now achievable. Another benefit is that WSNs can accommodate large network deployments. Due to these characteristics, WSNs have applications within several areas, such as habitat monitoring [1], environmental research [2, 3], volcano monitoring [4], industrial control systems [5] and military surveillance [6,7]. The sensor redundancy, the small physical appearance, and the diminishing maintenance cost make WSN a very attractive technique in these areas.

Although WSN is envisioned a promising technology for a wide range of sensing applications, many challenges remain.

1.2 Motivation and challenges

Wireless Sensor Networks (WSNs) share several features with similar wireless distributed systems such as Mobile Ad hoc NETworks (MANETs). In both a WSN and a MANET, the nodes autonomously collaborate in forming a wireless multihop network without relying on existing infrastructure. For both network types, the routing protocol forms and relays data throughout the network. There are, however, some characteristics of WSNs that make the design of these routing protocols particularly challenging. To start with, most WSNs are highly application-specific. This means that the design requirements of a WSN change with the target application. Furthermore, WSNs are very limited in energy and bandwidth. Hence, the route discovery and relaying must be performed such that the lifetime of the network is maximized. The particular characteristics of WSNs are elaborated below.

Application-specific:

- Most computer network systems are designed to be *application-agnostic*. This means that the underlying network structure and protocols are designed to accommodate a wide range of possible applications. WSNs are in contrast, highly *application-specific* and are essentially built for a special purpose. Although a WSN protocol can be designed generically to facilitate many purposes (or applications), a generic protocol is not necessarily the most efficient one for a certain application. For example; a low-delay military surveillance application may require a different routing scheme than a periodic agricultural monitoring task.
- For many WSN applications there is a need for a precise knowledge of the location from where the data was captured. For example for a surveillance WSN, it is important to determine the location of a possible intruder. For such systems, localization of the sensor nodes is an important feature.

Energy and bandwidth limitations:

- WSNs are energy constrained and have very limited bandwidth. Thus, the routing protocol must limit the total number of packet transmissions to conserve bandwidth and to maximize the system lifetime.
- Many WSNs consist of several hundred nodes. A scalable protocol design is therefore necessary. Furthermore, since some nodes might fail due to energy depletion and may be replaced, the routing protocol must accommodate changes in the number of nodes.
- Depending on the application, there can be several sensor nodes collocated

in the same area. In such cases, sensor data can be redundant. To improve energy and bandwidth utilization, the routing protocol must exploit such redundancy by letting the sensors collaborate, e.g., by using data aggregation.

- The topology of many sensor networks involves a packet flow from multiple sensors to one or more sinks (i.e., a Multipoint-to-Point (MP2P) topology). This is in contrast to other communication networks which usually build upon a Point-to-point (P2P) traffic pattern. Both energy and bandwidth can be conserved by enabling protocol support for MP2P.
- Ultra-low power microcontrollers are necessary to fulfill the lifetime and cost requirements. Such microcontrollers are very limited in computation power and available memory capacity. This means that algorithms and protocols must be computationally efficient and consume a limited amount of the scarce memory capacity.

It is clear from the list above that there are several application requirements and energy and bandwidth constraints that dictate the design of a WSN and impact the routing protocol. Traditional routing protocols known from the wired network domain consumes too much bandwidth and energy to be considered relevant for WSNs. However, some of the principal ideas from these protocols, such as the routing metrics can be applied.

Routing protocols for MANETs on the other hand, can sometimes be adapted to WSNs by reducing the number of control messages and optimizing the size of the packet headers. It is also possible to further reduce the overhead of a MANET protocol by including support for MP2P data collection. A variety of MANET adaptations are discussed in Chapter 2 of this thesis and explored in the papers A, B, C, H and J.

A major challenge with WSN routing protocols is to maximize the lifetime of the network. Since there is a trade-off between the scarce energy and bandwidth resources on one hand, and the Packet Delivery Ratio (PDR) on the other hand, this is a difficult challenge to address. Additional factors that can influence on the lifetime and performance of the network are interference, lossy links and energy-depleted nodes.

The performance of a WSN is also affected by several application and deployment issues, such as the placement of the sensing nodes, the placement and the number of sinks, and how the nodes collaborate. Considering these issues, there are several methods that can contribute to increase the lifetime and the PDR. For example, multiple sinks can be used to increase the scalability and the redundancy. The use of multiple sinks also reduces the energy usage due to shorter paths and increases

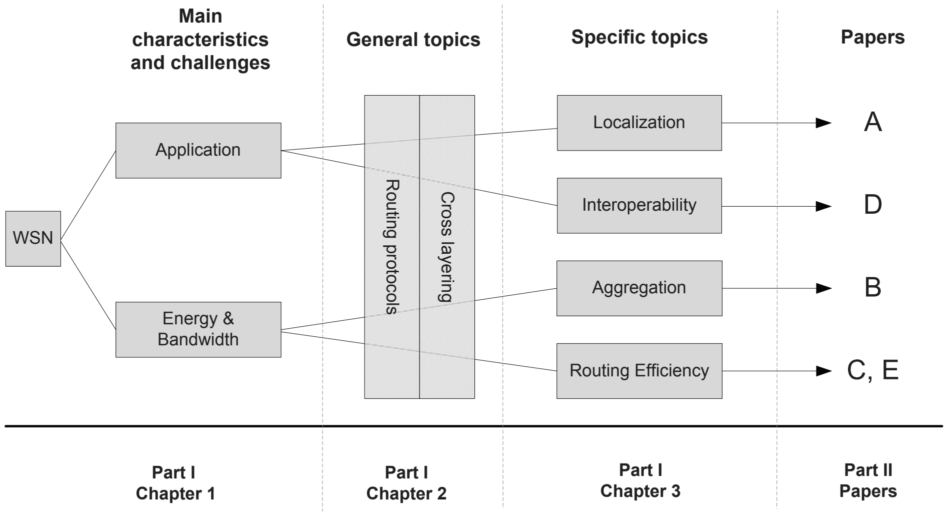


Figure 1.1: Structure of the thesis work

the throughput due to spread of the load. Another method is to use network clusters to separate the network into smaller and more manageable units. The nodes can also collaborate by aggregating sensed data inside the network.

The limited bandwidth, energy and memory capacity in a WSN makes it important to have a holistic approach to protocol development; i.e., protocols and functionality on different layers must be seen in combination and even merged if feasible. Cross-layering is a technique that contributes to increase the performance of the network by exploiting information exchange and dependencies between the protocol layers. A common theme in this thesis is therefore to use a cross-layering approach to address some of the challenges related to energy- and bandwidth efficient routing in WSNs. The cross-layering techniques that are used includes creation of new interfaces between protocol layers and design coupling without new interfaces. In addition to energy- and bandwidth challenges, the thesis also addresses some application-related challenges such as node localization and interoperability.

1.3 Overview of the work

In the previous section we discussed that the main characteristics of a WSN are: *i*) that the design is often application-specific; and *ii*) that the nodes are energy and bandwidth constrained. These characteristics lead to a set of challenges that must be solved. Among these challenges, the work in this thesis focuses on localization and interoperability, which are application-related topics; and aggregation and routing-efficiency, which are topics related to energy and bandwidth. Figure 1.1 show the relation between the main challenges, the specific topics, and the thesis contributions including the research papers. Paper A–E directly address the specific topics shown in the figure, while Paper F addresses a particular surveillance application for WSN.

All topics are addressed from a routing protocol perspective. Routing protocols are therefore the recurring theme through all the work. Furthermore, cross-layering is used as a method to provide performance improvements e.g., by linking the routing protocol with other WSN functions such as localization or aggregation. Cross-layering is thus the second recurring theme in the thesis. These two themes; routing protocols and cross-layering, is described in detail in Chapter 2, whereas the related work specific to each paper is described in Chapter 3 and in each of the papers.

The six research papers have been published in peer-reviewed international conferences or journals. The author of this thesis is the principal contributor and first author of all the papers. An overview of the work will now be given, starting with Paper A.

In many WSN applications, such as battlefield surveillance or environmental monitoring, location awareness is an important function. Paper A investigates how the routing protocol can facilitate localization of the sensor nodes. Since the routing protocol provides the fundamental messaging service in the WSN, this service can be exploited to distribute localization information. The paper investigates a method to provide routing and localization simultaneously. Cross-layering is here used to exchange information between the localization mechanism and the routing protocol. Furthermore, a part of the solution is to improve the efficiency of a common distance-vector routing protocol by augmenting the protocol with support for the MP2P traffic pattern.

Paper B investigates how the routing protocol can contribute in cluster generation. In a WSN used for surveillance, there can be a high probability that an event is detected simultaneously by multiple nodes. By using data-aggregation within clusters, such events can be combined into one report. This will improve the utilization of the scarce energy and the limited bandwidth resources. The work builds on the routing protocol (supporting MP2P traffic pattern) developed in Paper A. Furthermore, Paper B studies how virtual network clusters can be determined based on fetching topology information from the routing protocol via cross-layering. Both centralized and distributed clustering methods are compared in a surveillance context.

Paper C investigates multi-sink WSNs. The use of multiple sinks in a WSN extends the MP2P traffic pattern with anycast and improves the network lifetime, increases the scalability and gives redundancy. The method also reduces the average path length in the network. In the paper we investigate algorithms for finding optimal sink locations for a given network topology and coverage. The sink placement method utilizes the route establishment phase of the routing protocol to obtain information about the current network topology. The scheme can then propose locations for a given number of sinks that will lead to the assumed highest performance and system lifetime.

While a few WSNs operate isolated and on their own, most WSNs are connected to an external network. The integration between a low-power, application-specific WSN and an external network system must be performed by taking into account the limitations of the WSN and particularly the routing protocol. Paper D addresses this challenge and provides a data-aggregation scheme for WSNs that interoperates with the outside world (external networks) via a standardized interface. The architecture in Paper D provides a wrapper for Web services. The wrapper enables external systems to interoperate with the sensor network using standardized Extensible Markup Language (XML) and Web services. The routing and data-aggregation schemes inside the WSN on the other hand, use traditional routing mechanisms that can adapt according to the external system queries.

Paper E focuses on routing mechanisms that are able to operate in high-interference scenarios. The paper investigates the trade off between traditional routing, which has its clear advantage in stable networks, and opportunistic routing, which has its advantage when the network is unreliable. The paper presents a hybrid method that performs automatic switching between traditional routing and opportunistic routing based on the underlying network characteristics. The approach presented in the paper is compared to five other routing protocols and reveals that the hybrid protocol gives the overall best balance between PDR and overhead.

Paper F is addresses the holistic sensor network vision for military surveillance. The main goal of this work is to gain better knowledge about how to build an entire sensor network system. The work includes data collection, configuration, design and the development of the hardware. It also includes a simple method for node localization. Paper F reports on several experiments with sensor networks deployed in a military training facility. The experimental results quantify the operative effect of using a military surveillance WSN. The performance of the network protocols in the field is also evaluated and compared to test-bed results. Furthermore, the paper summarizes our experience and lessons learned in building sensor nodes and conducting realistic field system trials of WSNs.

1.4 Scope

There are different approaches that can be taken when designing an application specific sensor network. One approach is to create a vertically optimized system without defined communication layers. Such a bespoke system will often give the best performance and system lifetime. The drawback with this approach is that the system can be inflexible and replacing one of the protocols can be difficult without a complete redesign. The approach taken in this thesis is therefore different. We use standardized protocols and layers and adapt these for a given application. The advantage with this approach is that the protocols can be replaced. Furthermore, the solutions that we have used can be adapted to a wider range of applications.

Even if some of the approaches and considerations in this thesis are strongly focused towards military surveillance applications, this does not imply that the proposed solutions are applicable for military WSNs only. There are many civilian security and monitoring applications that are similar to military application and can use our proposed solutions directly.

Security is of utmost importance for all wireless networks and particularly for military WSNs. Thus, we emphasize that a security architecture must be part of a final system although we do not directly address security aspects of WSNs in this thesis. It is well-known that any security solution can affect the performance of some parts of the system, such as the routing protocol. However, a range of lightweight security mechanisms can be applied alongside with the WSN protocols and solutions presented in this thesis. For example, both TinySec [8] and IEEE 802.15.4 [9] provide link-encryption that can be used with minor increase in both overhead and energy consumption. Furthermore, data-integrity and authentication can be included in the routing protocol as shown in the work by Pecho et al. [10].

A subject that is closely related to routing protocol development, is that of Medium Access Control (MAC) protocols. Routing protocols can perform differently depending on the chosen MAC protocol, and the study of optimizations herein is an active research area. The subject of MAC protocols is, however, not extensively studied in this thesis. We ensure that our solutions are applicable for a wide range of target applications by basing the work in the thesis on the de-facto standard 802.15.4 MAC protocol developed by the Institute of Electrical and Electronics Engineers (IEEE).

1.5 Research Methods

The four well-known techniques for evaluating Wireless Sensor Network protocols are: analytical models, simulation, emulation and real-world experiments. Real-world experimentations can further be divided in test-bed experimentation and field experiments (i.e., full implementations).

Analytical models are used to evaluate certain protocol properties and can provide the fundamental knowledge necessary for directing future research. A distributed wireless sensor network is, however, extremely complex. Thus, it is not feasible to analytically model the dynamics across several protocol layers and among distributed sensor nodes simultaneously without making abstractions of the physical world. For these reasons, analytical models are not used in the research in this thesis. Simulations are generally better suited than analytical models to investigate the complex dynamics and distributed interactions in WSNs. A limitation with simulations is however, that the simulator must simplify the properties of the physical world. Real-world experiments are therefore required to prove that algorithms and protocols work as expected when deployed in an operational setting. Despite providing the necessary realism, conducting effective real-world experiments involve considerable complexity, cost and man-hours.

Paper	A	B	C	D	E	F
Simulation	X	X	X			
Test-bed				X	X	X
Field test						X

Table 1.1: Research methods used in the papers

To bridge the gap between simulations and real-world experiments, emulations can be used. Emulations can reduce cost and experimentation time and at the same time provide better repeatability than real-world experiments. In spite of the benefits, network emulations are not used in the research in this thesis. Instead, we use a combination of simulations, a test-bed with real sensing nodes–which serves the same purpose as an emulator, and field experiments. Table 1.1 shows the relations between the evaluation methods and the papers. We elaborate the different methods below.

1.5.1 Simulations

Simulators such as Network Simulator 2 (NS-2) and NS-3 [11], TinyOS Simulator (TOSSIM) [12], Castalia [13] and OMNeT [14] all come with built-in support for a wide range of sensor network protocols. NS-2 is among the most well-tested and reliable simulators and is widely used for performance evaluation of both wired and wireless networks. The papers A, B and C of this thesis all rely on performance evaluation using the NS-2 simulator.

For the works in Paper A and Paper B, we implemented the DYMO-low routing protocol for NS-2 according to the draft standard [15] and adapted it to include our ideas for localization and clustering respectively. A part of the experiments in Paper B was based on the Optimized Link State Routing (OLSR) code from the University of Murcia [16], albeit we altered the code to increase the topology awareness. For the work in Paper C, we implemented 6LoWPAN Ad Hoc On-Demand Distance Vector Routing (LOAD) [17] as a basis for our multiple-sink solution.

An important factor in wireless network simulations is the choice of propagation model. For Paper A and Paper B, we used the simple two-ray ground propagation model in NS-2, which gives a high level of abstraction of the physical layer. Although this abstraction means that the results are not directly transferable to the real world, the trends that are produced should still be valid. In Paper C, the complexity of the scheme required a more detailed propagation model. For this purpose we used the ShadowingVis propagation model, which is one of the most realistic propagation models available for NS-2.

Besides the challenges involved in using unrealistic physical models, there are also other limitations with network simulators. Some common shortfalls in conducting MANET research was identified by Kurkowski et al. in the work [18]. These shortfalls, such as improper simulation setup and the use of unrealistic scenarios, also apply for WSNs. In spite of these, simulations are virtually inevitable in order to validate the scaling characteristic of a protocol design. It is therefore a very valuable and trustworthy tool when realistic scenarios are used. In Paper C for example, we used a wide range of scenarios and repeated experiments to avoid biased results.

1.5.2 Test-beds

Simulations can only be the first step towards a deployed sensor network. Inspired by other test-beds, such as Emstar [19], MoteLab [20] and WiseBed [21], we decided to build our own test-bed consisting of 20 TelosB [22] sensing nodes with TinyOS 2.x [23]. The nodes were mounted on a wall covering an area of 2.5 m x 2.5 m. An output power of -25dBm, gave a multihop network with an average node degree of 6. The nodes were connected to a standard laptop using a combination of USB cables and hubs. This USB backbone was used for reprogramming and debugging. The test-bed enabled testing of real hardware and code in a controlled environment and was essential for studying data aggregation and interoperability in Paper D.

Small test-beds often underestimate the network dynamics found in real deployments. Hence, we developed our test-bed further by introducing a software controlled interference source for the research in Paper E. Moreover, we developed a set of tools which let us run automated and repeated experiments with a variety of routing protocols as well as different interference levels. In this way, the test-bed combined the benefits of simulations and real-world experiments.

1.5.3 Field testing

Even though the term Wireless Sensor Network usually refers to a general concept, most WSNs are (as discussed in Section 1.2) constructed with a certain purpose in mind and is therefore targeted a specific environment and application. Simulations, and even detailed testbeds, cannot precisely determine how the WSN behave in the desired environment. Thus, the only way to make sure that a WSN system work as expected is to carry out real-world experiments in the field. In Paper E we conducted experiments with a 50-node surveillance sensor network in an operational setting. An important part of the research method was to compare our surveillance WSN both with a state-of-the-art commercial Unattended Ground System (UGS) and with using human soldiers as sensors. This method let us quantify the operative effect of using a WSN as a surveillance system as well as evaluating the technical properties of the sensing nodes.

1.6 Thesis Outline

The thesis is organized in two parts. Part I is an introduction and discussion of the areas where the thesis contributes whereas Part II consists of a set of published articles. Figure 1.1 show the main structure of the thesis work.

The list of figures and the list of terms and acronyms given in the beginning of the thesis are restricted to Part I. Likewise, since each article includes a reference list, the reference list found at the end of Part I is exclusive to this part of the thesis.

Part I begins with a brief introduction in Chapter 1 describing the background, motivation and outline of the thesis. The employed research methods are also described here. Chapter 2 describes the related works in the areas of routing in

WSNs and cross-layering. These two subjects are important throughout the whole thesis. The related works that are specific to each of the thesis contributions are discussed in Chapter 3. This chapter also describes the motivation and the main results for each of the papers. The thesis is concluded in Chapter 4, summarizing the contributions and future work.

Part II consists of the following six research papers, in chronological order:

PAPER A: Localization in Wireless Sensor Networks based on Ad hoc Routing and Evolutionary Computation

PAPER B: Increasing the Lifetime of Roadside Sensor Networks using Edge-Betweenness Clustering

PAPER C: Constrained-based Multiple Sink Placement for Wireless Sensor Networks

PAPER D: Integrating Wireless Sensor Networks in the NATO Network Enabled Capability using Web Services

PAPER E: O-CTP: Hybrid Opportunistic Collection Tree Protocol for Wireless Sensor Networks

PAPER F: Experiences from deploying a Wireless Sensor Network for Military Base Protection

Chapter 2

Related work

2.1 Introduction

There are two general topics that are important throughout the whole thesis (cf. Figure 1.1). The first topic deals with performance optimizations of the WSN routing protocols themselves. This chapter therefore begins with an introduction to the most prominent routing schemes for WSNs (Section 2.2). The second recurrent thesis topic addresses cross-layer interoperation between the routing protocol and other protocols. Hence, Section 2.3 discuss such cross-layer optimizations in WSNs and the challenges lying therein.

The purpose with these two sections is not to present an in-depth study of all areas of research, but to provide the necessary background information and describe how the related work is used as a basis for our research. In order to be comprehensible, we do not describe all functionality in the different protocols, but rather simplify it down to the essence needed to understand the fundamentals behind our research. The reader should consult the appropriate protocol documents for a more complete description.

After describing the related works in the two main categories, we conclude the chapter in Section 2.4 by presenting an overview of some important related works that—although they are not necessarily directly related to either routing or cross-layering as such—have been inspirational and have yielded important insight to complete the thesis work, and thus, should be acknowledged.

2.2 Routing

Various target applications may require radically different routing protocols. To accommodate for the variety of possible WSN-applications, a wide range of routing protocols have been proposed. The work in [24] surveys numerous early works in this category. In this chapter we focus mainly on protocols that are implemented on real hardware and protocols that are standardized by the Internet Engineering Task Force (IETF). According to the IETF, WSNs differ mainly in the traffic patterns the protocols support. We name the traffic patterns as follows (borrowed from [25] and [26]):

- Point-to-point (P2P) refers to traffic exchanged between any two nodes in the network.
- Point-to-Multipoint (P2MP) refers to traffic between one node and a set of nodes. A common WSN use case involves P2MP flows from or through a sink node outward towards other nodes contained in the network.
- Multipoint-to-Point (MP2P) is a common WSN use case in which packets collecting information from many nodes in the network flow inwards towards the sink node(s).

Regardless of the traffic pattern, routing protocols for WSNs also differ in the employed objective function and the set of metrics and constraints that are used in the route selection process. We begin this section with an overview of the central WSN routing metrics. Then we discuss the most relevant routing protocols within three categories; P2P routing protocols, MP2P protocols, and finally opportunistic routing protocols. The latter category can potentially support all of the three traffic patterns listed above.

2.2.1 Metrics

Most routing protocols use hop count as a simple cost metric. The Minimum Hop Count (MHC) route metric is simple and intuitive and lets the routing protocol select the path with the minimum number of hops between the source and the destination. One problem with MHC is that it maximizes the distance traveled by each hop. Since physical links degrade with distance, MHC is prone to favor low signal strength links. Furthermore, as the metric does not consider the quality of the links, it implicitly assumes that the link either works perfectly or not at all. In practice however, the link might deliver only a small percentage of the packets on average. Hence, the underlying MAC protocol must retransmit the packet several times to achieve success. For these reasons, the minimum-hop-path is seldom the most efficient in terms of energy consumption and bandwidth utilization.

It is clear that in many circumstances it might be beneficial to use a different metric than MHC. Alternative routing metrics have therefore been proposed. One of these is the Expected Transmission Count (ETX) metric [27]. ETX finds paths with the fewest expected number of transmissions (including retransmissions) required to deliver a packet all the way to its destination. By measuring the bidirectional packet loss ratios on each link, the metric predicts the number of retransmissions required. Since the ETX information lets the routing protocol minimize the total number of transmissions, the energy consumption is also minimized.

Some radio transceivers, such as the Chipcon CC2420 commonly used in WSNs, produce a Link Quality Indicator (LQI) value upon packet reception [28]. The LQI value can be used by the routing protocol via cross-layer interfaces. One example of a routing protocol using such information from the physical layer is MultihopLQI [29]. MultihopLQI uses LQI to build its routing tree. LQI can also be used in combination with MHC as in LOAD [17].

An issue with the different metrics presented above is that none of them consider balancing the energy usage amongst the different nodes. Since MHC, ETX and LQI all prefer certain paths in the network, the nodes that constitute these paths can run out of energy much faster than the other nodes. To achieve better balance in the energy consumption, some routing protocols use the residual energy on the nodes as part of the cost metric. As an example, Chang et al. [30] proposed a routing protocol that selects the path with the largest residual energy nodes. With this approach, the nodes in this path will not deplete their energy completely, since a better path (more residual energy) will be chosen at an earlier stage.

2.2.2 Point-to-point routing

A Point-to-point (P2P) protocol can enable data transmission between any two nodes in the network. The P2P traffic pattern is essential in MANETs, and several P2P routing protocols are developed for such networks. P2P protocols can also be beneficial for WSNs, since they provide more flexibility in the data exchange compared to a mere MP2P protocol. Moreover, P2P protocols are well-understood and implemented in several simulators and operating systems. Since the IETF IPv6 over Low power Wireless Personal Area Networks (6LoWPAN) working group [31] has made great efforts to bring the Internet Protocol (IP) to WSNs and other low power wireless networks, the process of translating MANET protocols to such networks has become simpler.

In the following, we will explain two different point-to-point routing protocols: AODV (and its derivatives) and OLSR. Both protocols have been studied and applied for WSNs during the work of this thesis.

AODV

One of the most successful MANET protocols is Ad hoc On-Demand Distance Vector Routing (AODV) [32]. AODV is a reactive routing protocol that aims to obtain routes on-demand, i.e., when an upper layer communication packet is destined to a node not known in the routing table. The reactive nature of AODV is also attractive for WSNs, especially for WSNs with few concurrent traffic flows and little overall traffic. Dynamic MANET On-Demand Protocol (DYMO) [33] is proposed as the successor of AODV, but has not achieved this status yet. Since AODV and DYMO share the salient features, we describe only AODV in the following.

AODV is based on flooding control traffic before data transmission. Routes are discovered by a node by broadcasting Route Requests (RREQs) which are flooded throughout the network in search for the destination. The destination may receive multiple RREQs originated from the original request. From this pool, it responds to the request that has traversed over the assumed shortest path from the source to the destination. The respond message is called a Route Reply (RREP) and follows a reverse path towards the source.

To improve the basic request-reply phase, some additional techniques are included in AODV: First, any node that has a fresh route to the destination may respond to the RREQ with a Gratuitous RREP. Second, to limit the overhead caused by the number of rebroadcasted RREQs, AODV can utilize an expanding ring technique by gradually increasing the Time To Live (TTL) for each route request. Third, each node maintains the routes via Hello-messages and stores the neighbors that use the node as a router in its precursor list. Fourth, upon data transmission, an intermediate node may detect a link break. This node will then generate a Route Error (RERR) message that is sent to the nodes in the precursor list. When the originator node receives the RERR, it initiates a new route discovery for the given destination. The node that detects the link break may optionally employ local repair in an attempt to re-discover the broken route.

Several efforts have been made to optimize AODV to make it more suitable for low-power and bandwidth-constrained networks. NST-AODV [34] reduces some of the overhead of AODV by tailoring the protocol headers for IEEE 802.15.4 devices. The complexity is reduced by removing the expanding ring search technique and relying on Link-layer notifications to detect lost packets instead of using hello-messages. LOAD [17] shares the same features but also removes the Gratuitous RREP technique. DYMO-low [15], MSRP [35] and DYMO on TinyOS (TYMO) [36] take a step further and also remove the local repair technique. The most simplified versions of AODV are AODVjr [37] and TinyHop [38]. In these, even the RERR messages are omitted.

LOAD and DYMO-low was initially proposed by the IETF but was suspended by the 6LowPAN working group pending the results from RPL [25, 39]. The vast interest in WSNs however, has triggered a renewed interest of AODV derivatives within the IETF. The ideas of LOAD and DYMO-low are now being standardized in LOADng[1] [39]. LOADng shares the salient features of LOAD, but also introduces optimized flooding, reducing the overhead incurred by RREQ generation and flooding.

Three of the AODV-derivatives mentioned above have been used in the thesis. DYMO-low was used in Paper A and Paper B. To reduce the number of RREQs, the protocol was modified to establish a routing tree rooted in the sink (i.e., a MP2P pattern) rather than letting all nodes establish independent point-to-point routes towards the sink. For the work in Paper C, LOAD was used instead of DYMO-low, since it provides LQI based routing in addition to MHC. In this paper, LQI was used as an input to the sink positioning algorithm. In the experiments in Paper E, TYMO [36] served as an example of a P2P protocol in the experiments. TYMO was chosen among the different AODV-derivatives since this protocol is readily available in TinyOS 2.x.

OLSR

Optimized Link State Routing (OLSR) [40] is a proactive link state MANET routing protocol. It sets up and maintains routes regardless of the application layer communication demands. The nodes discover their neighbors by exchanging HELLO messages. The novelty of OLSR is to employ MultiPoint Relays (MPRs) to minimize the number of control messages flooded in the network. Each node chooses a subset of its neighbors as MPRs so that these MPRs cover all two-hop neighbors. Routes are discovered and maintained based on the regular transmission of control traffic between nodes and the designated MPRs. Control messages are only flooded through these MPRs, and not to all nodes.

Despite various optimizations in OLSR, the proactive nature of the protocol leads

---

[1] The Lightweight On-demand Ad hoc Distance-vector Routing Protocol - Next Generation (LOADng)

to a high number of control packet transmissions. Thus, the protocol is less suitable for energy-constrained WSNs. On the other hand, OLSR offers several advantages that are not as easily available with reactive protocols such as AODV. For example, it can provide quick rerouting in case of topology changes and spanning trees for information distribution. It can also contribute to node cooperation and node localization.

Although OLSR is seldom considered viable for the smallest flavor of WSN nodes, it is an excellent candidate for routing between more advanced sensor nodes. OLSR is for example used in the 100-node CitySense sensor network [41]. By taking the residual energy into account in MPR and route selection, the protocol can also be applied to a broader spectrum of WSNs [42]. OLSR is especially attractive when interoperation between the WSN and mobile nodes is required [43].

OLSR provides the attractive feature that each node keeps an updated view of the network topology. In Paper B of this thesis, topology information from OLSR was used as input for our clustering protocol. Here, updated information about the network topology is used to create network clusters for data-aggregation. In Paper H and Paper J, we present methods that adapt OLSR to work better in a wireless sensor network environment. This is done by sending control messages with a low frequency when the network is stable and more often when topology changes occur.

2.2.3 Multipoint-to-Point routing

As opposed to most other distributed systems, WSNs deals with distributed data collection characterized by the Multipoint-to-Point (MP2P) communication pattern. The routing control traffic can therefore be severely reduced by letting the routing protocol accommodate this traffic pattern. The main limitation of Point-to-point (P2P) protocols, such as those described above, is that they do not exploit the fact that most traffic in a WSN is destined to one node (i.e., the sink). MP2P (or convergecast) routing protocols on the other hand, assume that all data produced by the sensors are destined to a sink node. This protocol category is therefore more efficient in data collection applications.

For the works in Paper A and B of this thesis, we identified a performance limitation of the P2P protocol DYMO-low, and extended the protocol to allow MP2P communication. For the work in Paper C, we modified LOAD similarly. It is worth noting that the same ideas have recently been explored further for LOADng by Yi et al. [44].

In the following, we describe three other MP2P protocols that have gained significant attention in the research community and have been used during the work of this thesis.

MultihopLQI

MultihopLQI [29] takes advantage of the LQI from the physical layer to additively build a routing tree rooted at the sink. The LQI is related to the strength of the received signal and is measured when receiving beacons. The beacons, which are transmitted with a fixed interval, is also used to detect changes in the network topology. The protocol avoids using routing tables by only keeping the state of the current best parent. This approach reduces the memory usage and the control overhead considerably compared to e.g., LOAD. In MultihopLQI the chosen parent is the parent that advertises the best route (regarding accumulated link cost) towards the sink. A node selects a new parent if another node advertises a lower cost to the sink than the parent currently used. Since the LQI is a radio-specific feature, MultihopLQI cannot be used on all hardware platforms. In fact, the Chipcon CC2420 was the target radio platform for the protocol, and it might perform differently when ported to other radio circuits.

In Paper E of this thesis, the TinyOS 2.x implementation of MultihopLQI was used as a part of the comparison.

CTP

The Collection Tree Protocol (CTP) [45] is the de-facto collection protocol in TinyOS-based WSNs and is used in numerous test-beds and real-world implementations. CTP builds on MultihopLQI but is significantly different in two central features. First, it uses the ETX as its routing metric as opposed to LQI. Starting with an ETX of 0 at the sink, each node calculates its own ETX as the ETX reported by the parent plus the ETX of its own link to the parent. The ETX of the link to the parent is calculated based on the number of successful transmissions to this node and the number of beacon messages received. Since the protocol uses ETX rather than LQI, it is independent on the underlying radio system. Second, CTP uses adaptive beaconing (as opposed to fixed beaconing in MultihopLQI) by extending the Trickle algorithm [46]. Adaptive beaconing reduces the route repair latency and contributes to sending fewer beacons when the network is stable. To adapt quickly to topology changes, the trickle timer interval is reset to a low value whenever a routing loop is detected or the routing cost decreases significantly.

The TinyOS 2.x implementation of CTP was used in Paper D, Paper E and Paper F of this thesis. In Paper D, the protocol was modified to provide tree-based data aggregation. In Paper E, the protocol served as a basis for a novel hybrid protocol proposal, Opportunistic Collection Tree Protocol (O-CTP).

RPL

The goal of the IETF Routing over Low-power and Lossy Networks (RoLL) group[2] is to standardize a routing protocol for Low power and Lossy Networks (LLNs). The term LLN includes all networks composed of embedded devices with limited power, memory, and processing resources interconnected by a variety of links. In 2010, the group introduced the IPv6 Routing Protocol for Low power and Lossy Networks (RPL) [25]. RPL is based on the efforts behind MultihopLQI and CTP. The main difference between RPL and CTP is that RPL provides the support for three traffic patterns, whereas CTP only supports one traffic pattern (MP2P). RPL supports Multipoint-to-Point (convergecast), Point-to-Point and Point-to-Multipoint traffic. In other words, RPL brings together the benefits from a traditional MP2P collection protocol (e.g., CTP), with the ability to route between arbitrary nodes in the network (as in e.g., LOAD), with the ability to disseminate information from the sink to the entire network as in a dedicated dissemination protocol such as Drip [47].

The routing tree in RPL is called a Destination Oriented Directed Acyclic Graph (DODAG). The pre-assigned root (i.e., the sink) starts the construction of the DODAG by sending DODAG Information Object (DIO) packets (analogue to the beacon packets in CTP). Once a node connects to the DODAG, it propagates its own DIO further down the network. To reduce the cost of propagating the routing state, RPL uses the Trickle timer in the same way as in CTP. In addition to including information about the current rank in the routing tree, the DIO message also contains the Objective Functions (OFs), which define the details on the computation of the routing metric. Two of the current OFs are *OF0* [48], which gives Minimum Hop Count (MHC) routing, and Minimum Rank Objective Function with Hysteresis (MRHOF) [49], which gives ETX routing. RPL provides huge flexibility, since the OF can be chosen depending on the metric the network designer decides to use.

Besides RPL's ability to provide bidirectional routes, CTP and RPL share most salient features. It is shown that the performance of CTP and RPL with MRHOF is comparable [50]. RPL has only been used for test purposes during the work of

---

[2] https://tools.ietf.org/wg/roll/

this thesis. However, due to the similar approach taken by the two protocols, the research that is done on CTP in the papers D, E and F of this thesis can also be applied to RPL.

2.2.4 Opportunistic routing

The purpose of both P2P and MP2P routing protocols is to find the optimal paths throughout a network by daisy-chaining the links with the presumed best qualities. This approach is identical to the method used in fixed infrastructure and is ideal when the network dynamics are minimal. However, the metric calculations can have difficulties in coping with the rapid fluctuations that can occur in the wireless domain. Consequently, the routing decisions may be based on historic and outdated metrics. Opportunistic routing takes a different approach. It exploits, rather than attempts to hide, the broadcast nature of the wireless medium. With opportunistic routing, a node does not preselect a preferred forwarder according to a set of (possibly outdated) metrics. Instead, opportunistic routing takes advantage of the fact that there might be many potential forwarders in a node‘s vicinity that are able to receive and forward the broadcast packet.

Various opportunistic routing protocols differ mainly in the way the protocol decides on which of the available forwarding nodes that should retransmit the packet. In the seminal opportunistic routing protocol Extremely Opportunistic Routing (ExOR) [51], the sender chooses a candidate subset of all its neighboring nodes that could bring the packet closer to its destination. This subset is prioritized according to ETX and is listed in the packet header. Each packet recipient then delays a certain time depending on its priority in the list before forwarding the packet.

In Geographic Random Forwarding (GeRaF) [52, 53] each packet carries information about the location of the sender and the destination. Prioritization of the forwarding nodes is based on location information. In other words, the forwarding node that brings the packet geographically closest to the destination is preferred. GeRaF has the possibility to use RTS/CTS[3] handshakes for collision avoidance. Furthermore, the protocol supports duty-cycled wireless sensor networks by using a busy tone protocol to wake up a sleeping node. An element of complexity of GeRaF is the need for exact position information. The geographic routing can also fail when no node is able to provide positive advancement (i.e., a dead-end).

ORW [54] is tailored for duty-cycled WSNs. In ORW, packets are addressed to sets of potential receivers and are forwarded by the neighbor that wakes up first and

---

[3] Request To Send (RTS) and Clear To Send (CTS)

successfully receives the packet. In contrast to the forwarder selection in ExOR, ORW focuses on energy efficiency and delay instead of network throughput. ORW also exploits spatial and temporal link-diversity by ensuring that many potential forwarders can overhear a packet in a single wake-up period.

Both ExOR and GeRaF supports the P2P pattern, since packets can be transmitted between any two nodes in the network. ORW, on the other hand, is an anycast MP2P protocol and is tailored for data collection in WSNs. Given that broadcast is the fundamental building block in opportunistic routing protocols, most of these routing protocols can be extended to support the P2MP pattern.

A disadvantage with opportunistic routing is that the use of multiple forwarders often leads to duplicate packets. This leads to unnecessary energy and bandwidth usage. In Paper E of this thesis, we therefore propose to use opportunistic routing only when traditional routing fails, e.g., when the network is subjected to interference and opportunistic forwarding is superior. In Paper E, we implemented an opportunistic routing protocol (GEOPP) inspired by GeRaF. As in GeRaF, our protocol uses geographical information about the nodes for the forwarding selection. GEOPP is, however, simplified and does not use RTS/CTS handshakes before transmitting. This protocol is combined with CTP to create a hybrid protocol, Opportunistic Collection Tree Protocol (O-CTP).

2.3 Cross-layering

2.3.1 Introduction

The 7-layered Open Systems Interconnection (OSI) model and the current TCP/IP Internet architecture are examples of common layered models for networking. The layered thinking has provided several advantages for network designers. First, by working in layers, the implementation and design effort can be parallelized. Thus, designers can independently focus on particular layers with the assurance that the final system will interoperate. This makes it possible to upgrade individual modules (e.g., the routing protocol) without necessitating a redesign of the complete system. Second, by defining one layer in the model as the “narrow waist”, interconnection between different networks is possible. In the TCP/IP model for example, the network layer (IP) constitutes this narrow waist. Third, layered models provide natural abstractions to deal with. This increases the synergy between research efforts and facilitates the progress towards working systems.

Although strict boundaries between the layers have several advantages, there is always a temptation to take architectural shortcuts to increase the performance. One of the first attempts to combine functions from several layers is Integrated Layer Processing (ILP) [55]. The purpose of ILP was to combine the data manipulation functions from several protocol layers in a single processing loop. Now, the use of architectural shortcuts, actively exploiting the dependence between protocol layers for performance gains, is commonly referred to as *cross-layer interactions*. According to Srivastava and Motani [56], different cross-layer proposals can be categorized in the following four categories:

1. Creation of new interfaces.
2. Merging of adjacent layers.
3. Design coupling without new interfaces.
4. Vertical calibration across layers.

The first category is the most common in WSNs. New interfaces can be created both downward from a higher layer to a lower layer and upward from a lower layer to a higher layer. One example of downward communication is a routing protocol that dictates the radio transmit power, as in the work [57]. An example of upward communication could be an application layer protocol taking advantage of the Received Signal Strength Indicator (RSSI) from the physical layer as in Paper A of this thesis.

The second type of cross-layer designs is to merge adjacent layers. For WSNs, there are several proposed schemes that involves a complete integration of the MAC protocol and the routing protocol. One example of such integration is the work in [58].

The third category involves coupling of two or more layers without any extra interfaces for information sharing. Here, mechanisms in one layer implies that another layer is capable of performing certain operations. Hence, it may not be possible to replace one of these layers without changing the other layer. Cross-layering methods of this category is used in some of the solutions in this thesis.

The final category involves setting parameters across several layers. For example a QoS-aware application layer can dictate certain operations at the routing layer, which in turn dictates the preferred modulation at the physical layer.

The use of cross-layer interactions for WSNs can have a wide range of motivating factors. In the introduction of the thesis, we identified that the main characteristics of WSNs are that they are highly application-specific and that they are very limited in energy and bandwidth. For the sake of brevity, we use these two groups to characterize the different cross-layer approaches for WSNs. In other words, cross-layer interactions can be used to:

1. facilitate application layer protocols (e.g., a node localization scheme),
2. improve the energy and bandwidth consumption.

In the following subsections we describe cross-layer interactions belonging to these categories and relate them to the work of this thesis.

2.3.2 Cross-layering for application support

Several application-layer schemes can benefit from exploiting lower-layer functionality. In many circumstances there is a need to coordinate some data amongst several WSN nodes. It is difficult to perform such coordination in an efficient way using only the mechanisms available on the application layer. A more efficient approach is to create an interface between the application layer and the network layer. Via such an interface one can exploit that the routing protocol has the capability to perform efficient distribution. For example, in Paper B of this thesis, we propose to coordinate node-to-clusterhead memberships using the existing MPR flooding mechanism in OLSR.

A more extensive use of cross-layering for application support is the work in Paper A of this thesis. In this work, the localization method at the application layer uses signal strength information from the physical layer and link information from the network layer. This information is fed to an application layer protocol, which combines the information and feeds it back to the routing protocol. The routing protocol then forwards the information to the sink for further processing. In other words, the scheme in Paper A use back-and forth cross-layer interaction involving three protocol layers.

2.3.3 Cross-layering to improve energy and bandwidth utilization

Most prevalent WSN routing protocols employ cross-layer mechanisms to a greater or lesser degree to increase the efficiency and to improve the energy and bandwidth utilization. Routing protocols typically use downward or upward interfaces to adjacent layers for information sharing. For example, a routing protocol can perform topology control by transmitting notifications to the physical layer about a preferred radio transmit power as in the work by Chipara et al [57]. A routing protocol can also rely on hints from lower layers, such as information about the residual energy on the node or information about signal strength or link quality from the physical layer. Several routing protocols in the latter category are relevant for the work of this thesis.

The central metric in the previously mentioned MultihopLQI protocol [29], relies on the Link Quality Indicator (LQI) from the physical layer. LQI gives information about the quality of the decoding of an incoming packet and is provided by the Chipcon CC2420 radio chip. A limitation with MultihopLQI is that the metric calculation is tied to this radio chip only. Since it is readily available in TinyOS 2.x, MultihopLQI was used for the experiments in Paper E of this thesis.

LOAD [17] is another protocol that uses link information from the physical layer in addition to the hop distance in the routing decision. The quality of a link is measured upon RREQ reception and could for example be based on LQI. However, while MultihopLQI is tailored for one particular radio chip, LOAD can be used on top of all radio chips that can provide some kind of simple link quality measurements. The cross-layer interaction here is therefore simpler and more flexible. The basic mechanism is as follows: If the quality value measured in LOAD is below a certain threshold value, the link is considered *weak*. The route cost then becomes a combination of the number of hops and the number of weak links. If the radio chip does not support link measurements or a cross-layer interface is unavailable, the protocol operation is identical to Minimum Hop Count (MHC). In Paper C of this thesis, LOAD was extended to provide anycast MP2P routing in a multisink setting.

The idea behind LOAD is to exploit cross-layer interactions between the routing layer and the physical layer without being bound to one particular radio circuit. The four-bit wireless link estimation module [59], which is used by CTP, follows the idea of such a general link estimation method a step further. It combines information from the network, link, and physical layers when estimating the link quality. The scheme provides simple interfaces between these layers. Despite the fact that the method definitively is cross-layer, the interfaces the method provides enables a generalized link estimation method that can be applied for a variety of routing protocols and physical layers. This link estimation module was used with CTP for the papers D, E and F of this thesis. In Paper E, it was revealed that the link estimation has difficulty coping with radio interference. This motivated for using a hybrid opportunistic protocol, which proved to perform better than CTP in interfered environments.

As we have now seen, one method to improve the energy and bandwidth utilization of the WSN is to use cross-layering to increase the routing efficiency. Two other approaches to improve the utilization of these scarce resources is to employ in-network data-aggregation and to dutycycle the radio circuit. An example of a scheme that use both approaches is the distributed cross-layer scheduling protocol for data aggregation proposed by Wu et al. in [60]. In this scheme, each node employs its MAC, routing, and query layers in a cross-layer fashion to negotiate with its parent the timing of transmission. The nodes distributively construct schedules that dictate the query processing, computation, communication, and sleep. In addition, data aggregation is performed using cross-layer interfaces with the application layer.

Paper D of this thesis presents a scheme that, similarly to the work by Wu et al., is motivated by the need for query processing and in-network data aggregation. The aggregation is here performed along the routing tree by extending the CTP routing protocol [45]. Paper B of this thesis is also concerned with in-network data-aggregation. Here, data aggregation is performed at the cluster-heads. The clusters are created based on link information from the routing protocol. This requires an upward interface from the network layer to the application layer. An alternative method to create clusters, not involving cross-layering, is to base the cluster structure on the geographical location of the nodes as in [61]. Paper B reveals that this approach is not as effective as clustering based on the network topology via cross-layering.

Paper C use a similar interfacing method as in Paper B. Here, the application goal is to determine the best locations to put multiple sinks. These locations are determined based on the LQI from the physical layer as well as link information from the networking layer. Information from these layers are collected by the application via cross-layer interfaces.

2.3.4 Implementing cross-layer interfaces

Any cross-layer method can have undesirable consequences on system performance if done without care [62]. A variety of architectural frameworks are therefore proposed to address some of the common challenges in introducing cross-layer optimizations [63–68]. These frameworks enable the use of cross-layer optimizations without violating the architecture or creating dependencies that hinders future system extensions. However, these frameworks are not free of costs. First, the desire for generic architectures and frameworks has the drawback of added complexity. Second, the memory footprint and the extra processing required for these frameworks can sacrifice performance for the energy and memory constrained WSN platforms. One of the lessons learned in Paper F of this thesis was that the program memory on typical WSN nodes quickly fills up. It is therefore paramount both to avoid duplicated functionality across the protocol layers (i.e., use cross-layering when feasible), and to avoid implementing unnecessary code.

One of the common arguments against using cross-layering, particularly for MANETs, is that it can lead to spaghetti-like code that is hard to maintain [62]. Passing information between protocol layers on a MANET-PC can indeed be tedious, and intrinsically difficult without a proper Application Programming Interface (API). WSNs are somewhat different, since the entire operating system (e.g., TinyOS [23] or Contiki [69]) and all protocol layers are implemented using the same programming language and often the same code base. Thus, passing information between the protocol layers is reduced to simple function calls. This makes it easier to implement and test cross-layer interactions without the danger of destroying the code structure or introducing bugs in the system.

2.4 Selected related work

The two previous sections have discussed previous work related to routing and cross-layering respectively. This thesis relies, however, on a wide range of early work. Some of the previous works, as the ones presented above, are used directly in the research. There are also other works from adjacent disciplines that have inspired the research in this thesis. Some works have described a highly relevant research method and some of the related works have been important to see the research field in a larger, historical context. These works also deserve recognition, and a selection of these is presented in this section.

Let us first take a step back in the history of wireless sensing. It is impossible to date the specific point in time when Wireless Sensor Networks was established as a research field. The US Army established the pioneering age of ambient battlefield intelligence by their work in the late 1960s as reported in [70]. This excellent and previously classified work reminds us that although wireless sensing is considered a new concept, the ideas stems from decades-old visions. A work from more recent times, but equally important, is the report from the Smart Dust project, which started in 1997 [71]. This project was a breakthrough in miniaturizing the wireless sensor concept, and paved the way for a new paradigm with respect to wireless sensors. Their work still serves as an inspiration for beginning researchers in the field.

The research method in the field of wireless sensing has gradually evolved from conceptual architectural thinking, through simple network simulations, to in-depth and thorough system testing and evaluation. Protocol testing and evaluation for Wireless Sensor Networks is a tedious task that requires many man-hours. The works by Gnawali et al. [45] and Mottola et al. [72] are excellent textbook examples of how to perform such testing carefully. The approaches and the thoroughness presented in these papers are truly inspiring.

The transition from ideas and simulations to full field implementations is far from trivial. Barrenetxea et al. [3] has provided a “Hitchhikers guide” for WSN developers planning to perform experiments in the field. This article is a must-read for all WSN researchers with the ambition to conduct real-life system testing. This work was one of the inspirations for conducting the field experiments presented in Paper F. Another article aimed to guide fellow researchers is the thought-provoking article by Raman and Chebrolu [73]. Their work encourages to employ a bottom-up approach in protocol development and advocates for simple solutions in WSNs.

Finally, the TinyOS developers have made great efforts in creating the de-facto operating system for WSN researchers. TinyOS has undoubtedly contributed to increase the understanding and prevalence of WSNs. Furthermore, the ability of the TinyOS team to see their 10-years endeavor in retrospect and share their lessons learned in the paper [74] is an example to follow. However, other OS alternatives are emerging. ConTiki [69] and Arduino [75] are both considered easier to learn and are excellent candidates for building sensing applications.

Chapter 3

Contributions

3.1 A summary of the contribution as a whole

The work in this thesis address a broad range of issues regarding Wireless Sensor Networks (WSNs). To give an overview of the contributions, we will first present the works thematically to illustrate how the contributions are related to certain challenges within WSNs. After that, the work will be presented chronologically. This will give the logical relationship between the papers and reveal how the research methodology has evolved through the work of the thesis.

3.1.1 A thematic overview

As described in the introduction of this thesis, there are two general characteristics of WSNs that define the spectrum of challenges. To begin with, WSNs are highly *application-specific*. This means that all mechanisms in the WSN are tailored towards a specific application and that different target applications (e.g., surveillance vs. agricultural monitoring) may require different protocols.

- *Localization* is one of the application-related challenges addressed in this thesis. In many applications for distributed sensing, it is important to know the location of the sensing nodes. Two different methods are studied in Paper A and Paper F respectively.
- *Interoperability* between the WSN and an external system is another application-related challenge. This challenge is addressed in Paper D.

The second subject that characterizes all WSNs is that they are very limited in *energy and bandwidth*.

- *Data-aggregation* is one method to conserve both energy and bandwidth. In Paper B, data-aggregation is implemented on top of a clustered network. In contrast, Paper D describes data-aggregation implemented on top of the collection tree. Again, the target application defines which of these two alternatives that is most efficient.
- A second method to conserve energy and bandwidth is to improve the *Routing Efficiency*. The scheme presented in Paper C, falls into this category since the use of multiple sinks improves both the redundancy and gives energy fairness. Multiple sinks also contributes to shorten the paths in the network. The work in Paper E more directly address routing efficiency by introducing a routing scheme that switches between traditional routing and opportunistic routing based on the underlying network characteristics. One additional method to improve the routing efficiency in a data-collection WSN is to incorporate support for a MP2P traffic pattern. This was performed in Paper A and B.

As shown in Figure 1.1, the common denominator and underlying theme of all the works is *routing*. The second common theme is *cross-layering*, which is used as a method in many of the papers to achieve increased application support and to improve the routing efficiency.

3.1.2 A chronological overview

The thesis consists of six research papers, A–F in chronological order.

The main motivation behind Paper A was to create an effective and precise localization system for WSNs. In order to take and collect the necessary ranging measurements for the localization mechanism, it was necessary to apply some changes to the routing protocol. To support collection in an efficient way, the MP2P traffic pattern was included to the routing protocol.

The routing protocol created in Paper A has the ability to collect all the topology information in the network. Paper B was motivated by using this information to create network clusters. Such clusters can contribute to separate the network into smaller and more manageable units. However, while a localization algorithm may well be centralized (as in Paper A), a clustering mechanism should be distributed. Hence, Paper B presents both centralized and distributed clustering. It is also demonstrated how the clusters can be used as a basis for data-aggregation.

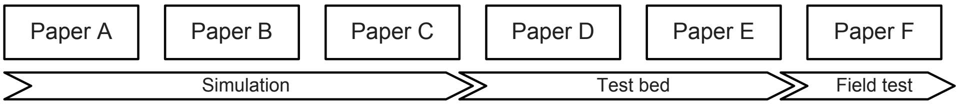


Figure 3.1: The evaluation methods used in the papers

The sink represents a single-point of failure and, since only one sink is used, this can give a skewed distribution of the energy consumption in the network. A natural extension of the work in Paper B is to expand the network to include more sinks. The specific challenge that was addressed in Paper C was to find the locations for a given number of sink that contributes to maximize the system lifetime.

The first three papers all dealt with the gathering of information from the sensors to one or more sinks. A natural next step in the research was thus to address how this information could be forwarded from the WSN to an external network. The motivation behind Paper D was to create this interoperability using a tight integration between the routing protocol in the WSN and standardized protocols in the external network.

Paper D involved many experiments with the CTP routing protocol. One of the lessons was that CTP is inefficient in certain conditions, particularly with interference in the network. Consequently, the motivation for Paper E was to create a new protocol that had better performance in such conditions.

One application that has been important throughout this thesis is surveillance. In Paper B and Paper D, we investigated surveillance WSNs using simulation and test-beds respectively. The motivation behind Paper F was to extend this research to a real life scenario, and establish a large surveillance WSN in a realistic setting.

Just as the ideas have evolved during the work of the thesis, the evaluation method has also evolved (cf. Figure 3.1). In the first work, presented in Paper A, the method included algorithmically implementation and evaluation using a network simulator. For Paper B and Paper C, the nature and complexity of the schemes required more complex simulations. During the work, we wanted to gain experience with real sensor nodes and provide more realistic results than simulations can offer. Hence, for the Papers D and E, the method shifted towards test-bed implementation and evaluation with real sensing nodes. Finally, in Paper F, results and fruitful experience was obtained using real implementations of large military surveillance WSNs.

3.2 Contribution of paper A: Localization in Wireless Sensor Networks based on Ad hoc Routing and Evolutionary Computation

In many applications there is a need for sensor node localization. For example in a surveillance application, the precise position of an intruder can only be derived by sensors with known positions. Another example is a sensor network for environmental monitoring. Here, the sensed data is of little value without knowing the position from where the data is obtained.

Location information is also crucial for other purposes such as geographic routing, data fusion, and data filtering [24, 76–78]. Although satellite navigation systems (e.g., Global Positioning System (GPS) or Galileo) may be valid methods for self-localization in some outdoor WSNs, they also increase the cost, consumes energy, and can be imprecise in woodland and indoor scenarios. The work in paper A investigates how to perform localization in WSNs without the need for any extra hardware. Instead, the scheme exploits the information inherent in the routing mechanism and the layers below using cross-layer interactions.

3.2.1 Related Work

A simple solution to provide self-localization in WSNs is proposed in [79], allowing self-localization using mere connectivity information. In other words, the scheme determines node positions without using any information about the distance between the nodes. Since the scheme does not require any special hardware, it can be implemented on low cost wireless sensing devices. However, the lack of precise ranging causes the position estimates to be imprecise, particularly for sparse networks.

Niculescu and Nath [80] propose an architecture deriving positioning information using Angle Of Arrival (AOA) antennas. The proposal therefore has a potential for precise localization also in sparse networks. Another example using specialized ranging hardware is the work [81], using ultrasound ranging sensors. For both these schemes, the precision comes at a cost, since such additional hardware increases both the price-tag and the complexity of the sensor network.

The works [82, 83] propose to use mobile nodes to aid localization of the sensor nodes. In such cases a vehicle, robot, or soldier enters the sensor field to assist the localization scheme. These schemes can provide both an efficient and low-cost solution to the localization problem. In many manually deployed networks, this is a sufficient solution. For example in Paper F of this thesis, this approach is taken. However, the precision obtained by this method, highly depends on the movement of the mobile node. In some deployment cases, like with sensor drop from airplanes, a different approach is needed, since there might be impractical to deploy a mobile node in the area.

Tam *et al.* [84] attempt to estimate node positions in randomly deployed sensor networks using a two-phase process. First, the positions are estimated using triangulation based on distance measurements. Then, a genetic algorithm is applied to refine the result from the first phase.

3.2.2 Contributions

Our contribution is inspired by the work in [84]. Both their and our proposal use central evolutionary computation to derive the node positions. Our work has two central contributions:

- We provide a holistic view of the complete localization and routing system including the ranging measurements, the measurement data gathering and the localization algorithm itself;
- the localization scheme uses evolutionary computation directly on the measured ranging information rather than using a two-step process as in [84].

In the following we describe the measurement gathering protocol first, and the localization algorithm is described in the subsequent paragraphs.

Implementing a localization algorithm in a WSN requires extracting information from the network, which consumes energy. However, in neither of the related works there are explanations of how to collect the necessary measurements to perform localization. Such functionality is often assumed to be accomplished by a separate protocol. We argue that it is inefficient to add separate measurement and collection mechanisms to the application layer as this will increase the energy and bandwidth consumption. Instead, the network designer should exploit that the routing protocol is already performing similar services (i.e., a cross-layer method is necessary).

The approach in Paper A performs ranging measurements and transports the measurements *concurrently* with routing. This is performed by exploiting the route request and route reply process in the reactive distance vector routing protocol DYMO-low [15]. DYMO-low is intended for use on IEEE 802.15.4 devices and is

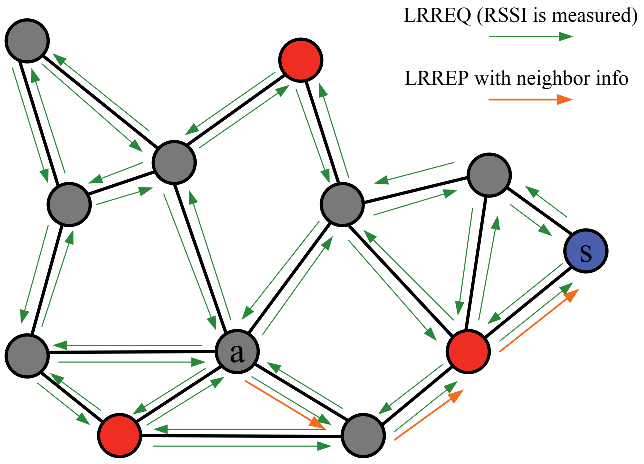


Figure 3.2: Initialization of a sensor network.

based on the principle of flooding Route Requests (RREQs) and unicasting Route Replies (RREPs) as known from AODV [32] and DYMO [33].

Our extension introduces two new messages to the protocol, Localization Route Request (LRREQ) and Localization Route Reply (LRREP) that collects ranging information and establishes a MP2P routing tree at the same time. Fig. 3.2 illustrates the protocol operation consisting of a request phase and a reply phase. The sink initiates the request phase by sending a LRREQ. All nodes collect signal strength (RSSI) measurements from their neighbors while the LRREQ disseminates. They report their measurements as attachments to their individual route reply back to the sink. This routing scheme builds up a collection tree towards the sink. While our routing process shares resemblance with the principle of CTP [45], our protocol has two added benefits: The first one is described already, i.e., the collection of the measurements concurrently with the route establishment. Secondly, whereas CTP only allows routes to the sink (Multipoint-to-Point (MP2P) routing), our protocol also allows routing between two arbitrary nodes in the network (Point-to-point (P2P) routing).

The localization algorithm is employed centrally at the sink. Here a genetic algorithm seeks to estimate the node positions based on the distance estimates from the RSSI measurements. The algorithm is based on an evolutionary approach working directly on the measured data. The algorithm initially proposes a population $\mathcal{P}$ of random node positions. A minimum of three anchor nodes with known positions (the red nodes in Figure 3.2) are necessary to create physical coordinates of the unknown nodes. The evolution is performed by comparing the distances between the proposed node positions $s_{i,j}$ with the known (observed) distances $o_{i,j}$, and building new generations on the best individuals from the populations. A fitness measure defines the rank of the individuals in the population and is used as a basis to build new generations.

The results in the paper show that the scheme was able to perform accurate localization even with measurement errors of 10-50$\%$. However, the performance of the evolutionary algorithm is sensitive to reduction of the transmission range or the network degree and the spatial distribution of the anchor nodes. An additional result is that our proposed changes to DYMO-low can reduce the number of routing messages with a factor of n compared to standard DYMO-low (where n represents the number of nodes in the network). The idea to provide MP2P routing for an AODV-derivative protocol is recently explored further for LOADng by Yi et al. [44].

The proposed extension to the DYMO-low protocol can potentially be used to facilitate other centralized localization algorithms than the evolutionary computation algorithm proposed in the paper. Likewise, the evolutionary algorithm can take advantage of information gathered using a link state routing protocol, such as OLSR [40]. Moreover, the evolutionary algorithm can benefit from more precise ranging methods such as acoustic ranging. Hence, rather than being a complete alternative to previous works, our techniques complement them.

3.3 Contribution of paper B: Increasing the Lifetime of Roadside Sensor Networks using Edge-Betweenness Clustering

In a surveillance WSN, the sensor nodes collaborate in detecting movement or certain behavior of objects in the sensed area. Multiple nodes are here likely to sense the same event simultaneously. Conventional routing treats these sensor readings individually and ignores the redundant and highly correlated nature of the data. This leads to ineffective use of the scarce energy and limited bandwidth. By employing data aggregation, designated aggregation nodes can wait for multiple reports, either from the same node (temporal redundancy), or from neighboring nodes (spatial redundancy), before reporting about the event to the sink.

Furthermore, most sensor devices (e.g., Passive Infrared (PIR) or radar) are likely to be inaccurate and have a small probability of falsely reporting events that are not actually present. Hence, using an alarm aggregation and combination strategy

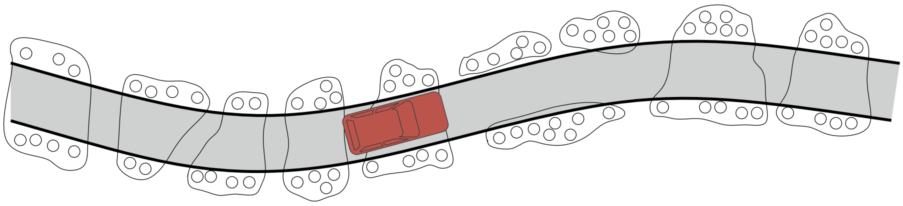


Figure 3.3: Roadside surveillance with a sensor network. The sensors are clustered using topology information from the routing protocol.

is not merely a method to conserve energy and bandwidth. It also reduces the probability of false alarms.

The work in paper B focuses on the problem of data aggregation in surveillance WSNs, and investigates how the routing protocol can contribute in cluster generation. The scenario we use is roadside surveillance as shown in Figure 3.3. Specifically we answer the following questions:

- Can Edge-Betweenness Clustering combined with data aggregation improve network lifetime?
- Is centralized clustering feasible?
- Is it possible to create clusters in a distributed fashion using a link state routing protocol?

3.3.1 Related Work

Different data-aggregation alternatives can be categorized based on the network architecture involved in the aggregation. The architecture can be structured either as a chain, a tree, or by clusters.

In a chain-based aggregation scheme such as PEGASIS [85], a linear chain is created for the data-aggregation. Each node in the chain only transmits to its closest neighbor, which fuses the data with its own measurements, and retransmits along the chain. Directed diffusion [86] instead organizes the nodes in an aggregation tree rooted at the sink. Since the aggregation tree is better than the chain for mere packet routing (shorter paths are created), tree based techniques often provide better performance than the chain-based counterparts—especially when only a subset of the nodes in the network are sensing nodes. The drawback with both schemes is that the aggregation delay perceived by a node is based on its position in the aggregation tree (or chain). As a consequence, the overall aggregation delay increases drastically with the number of nodes in the network [87]. The challenge is to balance the trade-off between energy efficiency and the delay posed by the aggregation. Both the tree-based and the chain-based aggregation techniques are best suited for scenarios where *all* nodes in the network produce relevant information *periodically*. In surveillance-based scenarios, these proposals are inadequate, since the long aggregation delay makes it difficult to uniquely distinguish separate events.

The seminal work LEACH [88], presented a cluster-based routing scheme. Cluster-based schemes organize the sensor nodes into virtual groups and perform aggregation only at designated cluster-heads (CHs). This approach drastically reduces the aggregation delay compared to the chain and tree architectures, at the cost of possibly longer routing paths. LEACH suffer from only supporting single-hop transmission between each cluster-head and the sink, but has later been extended to multihop by Lai et al. [89]. Gong et al. [61] takes a different approach and propose to use modified K-means clustering, and derive the clusters centrally assuming that the geographical positions of the nodes are known.

3.3.2 Contributions

The main contributions in paper B are:

- Our scheme has the ability to passively exploit the underlying routing protocol in a cross-layer fashion to gain topology knowledge. This is in contrast to the works [61, 88, 89], which use explicit control messages to initiate the clusters.
- Our approach allows some traffic classes to take an optional (shortest-path) route towards the sink, while the related works require that all traffic must pass through the cluster-head. The latter alternative prolongs the routing paths and increases the delay compared to our approach.

Our work is inspired by the work by Gong et al. [61]. The K-means clustering approach in their proposal does, however, suffer from two deficiencies that limit the practical use of the scheme. First, it requires that a localization scheme is present in the network. Second, K-means assumes that geographically adjacent nodes also are 1-hop neighbors. This is not always the case, since terrain or obstacles can prohibit communication even between nodes that are very close to each other. This assumption can therefore lead to suboptimal clusters and excessive paths between cluster members and the cluster-head. Our paper solves these two issues by proposing a radically different clustering algorithm.

Edge-betweenness community detection is a method proposed by Newman and Girvan [90]. Community detection algorithms are known from physics literature, (i.e., a community is a region of the network with dense connections) and have been successfully used to capture interactions in ad hoc networks [91]. Betweenness is a centrality measure of a link between two nodes in the network. This measure can be seen as an importance value that increases with the number of shortest routing paths that goes through the link. The outcome of the algorithm (i.e., the proposed network clusters) is a network where these important links are removed. In other words, the individual clusters include the nodes that have the most similar interconnections and neighbors. As opposed to most existing clustering methods, Edge-betweenness clustering does not put any a priori constraints on the cluster structures (e.g., cluster diameter, number of nodes in a cluster or number of clusters).

The prerequisite for the clustering algorithm is to have an updated view of the network topology. We propose three different methods to accomplish this. All methods exploit topology knowledge gathered from the underlying routing protocol via cross-layer interfaces:

1. Distributed clustering using OLSR.
2. Centralized clustering using OLSR.
3. Centralized clustering using DYMO-low.

Distributed clustering gives the unique advantage that all nodes can determine their cluster memberships without communicating any explicit messages to other nodes. The method can also be described as passive clustering. Since OLSR provides the attractive feature that each node keeps an updated view of the network topology, this information can be used directly in the Edge–Betweenness algorithm to determine the clusters. To ensure full consistency (i.e., that all the nodes determine exactly the same clusters), each node needs to obtain accurate topology information. However, if default OLSR settings are used, only partial link-state can be obtained. Inspired by previous works on extending the network topology knowledge in OLSR [92] and [93], we present a scheme that can establish consistent and distributed cluster generation in the WSN.

Although distributed architectures are desirable in most wireless networks, many WSNs already have a central entity, namely the sink. This means that a central protocol design is feasible and can in fact simplify the network design. In the paper we therefore study cluster generation centrally using the routing protocol to provide topology information. We observed that it is feasible to run the cluster protocol centrally even with the partial link state that standard OLSR provides. We also present a scheme using DYMO-low instead of OLSR. Based on the protocol extension presented in Paper A, we are able to lower the overhead considerably, while increasing the quality of the clusters generated, compared to the OLSR scheme.

For the evaluation part of the paper, we simulate data aggregation in a roadside sensor network (cf. Figure 3.3). Our Edge-betweenness scheme is compared with the K-means approach from [61]. The results show that our scheme (which does not requires geographical information for all sensor nodes) always provide identical or better performance than K-means (which requires such information).

3.4 Contribution of paper C: Constrained-based Multiple Sink Placement for Wireless Sensor Networks

Routing in Wireless Sensor Networks is commonly performed by a collection protocol building a routing tree routed at one sink. However, in large networks, the network lifetime and the scalability can be improved by deploying *multiple sinks*. The use of multiple sinks improves the energy fairness by load balancing and gives redundancy if one of the sink-nodes should fail due to energy shortage, or if it is vandalized or stolen. This approach also reduces the average path length between the sensing nodes and the corresponding data sink.

The work in Paper C focuses on the problem of determining the optimal *placement* of the sink nodes. The algorithms described in the paper find the optimal sink locations for a given network topology and coverage.

3.4.1 Related Work

Oyman et al. [94] propose to find the optimal placement of multiple sinks using the well-known K-means clustering. In their method, the cluster centroids for the k clusters are chosen as the optimal placement for the sinks. As already discussed, the limitation by the K-means approach is that the algorithm requires global location information to find the optimal sink placements. Vincze et al. [95] aim to relax this requirement by approximating the location of nodes with unknown positions. The system is, however, based on a geographical routing protocol, which requires a functional location system in the WSN. The main limitation with both approaches is that they study *unconstrained* sink placement. This means that they are based on the assumption that there are no physical boundaries constraining the proposed placement of the sinks. The presumed optimal sink locations found by the algorithms are therefore not necessarily viable in practice due to physical constraints in the scene. A proposed sink location may even be outside radio-range of the surrounding sensor nodes and turn out to be useless.

The work by Dai et al. [96] aims to solve this problem by only proposing sink positions at locations that are known to be in communication range with at least a subset of the network. To accomplish this, they restrict sink placement only to locations already occupied by sensing nodes. However, since their network model is restricted to Manhattan grid layouts and assumes uniform link lengths and link weights, the approach is not useful for other deployments. The works [97] and [98] are therefore considered more flexible. Deployment constraints are here used to limit relay node placements at some pre-specified candidate locations only, meaning that the proposed locations are not restricted to known sensor node locations as in [96]. Although their methods are more flexible and practical in a real setting, they require that the deployment algorithm a priori knows the deployment constraints. This requirement cannot always be fulfilled.

In Paper C, we study *constrained* sink node placement, meaning that the sinks can only be placed in a subset of the WSN scene. Another benefit with our approach is that the deployment strategies presented are not bound to particular network layouts. Furthermore, sink deployment constraints are not input parameters to the algorithms but are instead learned by inspecting the link information by interacting with the routing protocol in a cross-layer fashion.

3.4.2 Contributions

To effectively determine the optimal placement for multiple sinks, network information must be gathered globally or estimated. The different sink placement schemes from the related works can be placed in two categories based on the network information they require: (i) those that require knowledge about the geographical positions of all sensor nodes (geo-aware); and (ii), those that rely on the network topology (topology-aware). Paper C presents four different sink deployment strategies, two in each category. The first method we consider is based on K-means, and is similar to the work [94]. Thus it serves as a baseline for the comparison.

The second geo-aware method presented is based on the well-known K-medoid clustering [99]. K-medoid builds on the concept of *medoids* instead of using cluster centroids as in K-means. A medoid is defined as the most central object in a cluster. We propose to use this method for sink placement. The output of the algorithm (i.e., the medoids), will thus represent the k nodes that are most central. These locations are then proposed as the placement for the sinks. By constraining the algorithm to known locations, rather than proposing new locations, the network designer can be certain that sinks placed in these locations are in an area of sufficient network coverage.

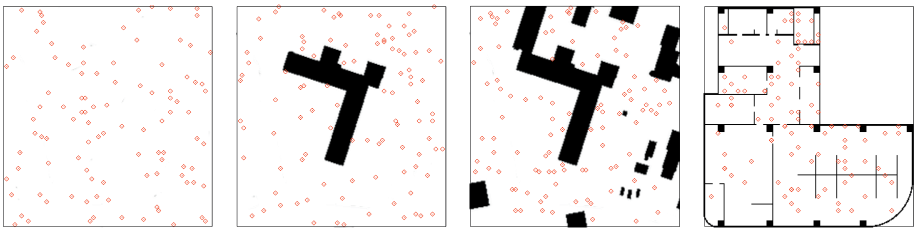
All geo-aware methods suffer from known shortcomings:

1. The geographical positions of the sensor nodes must be known. To obtain the individual node positions, a localization and collection scheme must be present in the network.
2. Since the methods are based on Euclidean distance, the algorithms inherently assume that all sensor nodes share the same transmission range and that geographically adjacent nodes also are 1-hop neighbors. This is not always true in obstructed environments.

To overcome both these limitations, Paper C proposes two new sink placement schemes that rely on the network topology instead requiring geographical positions. The first of these methods simply computes an adjacency matrix using the link information from the routing protocol, and makes an all pairs shortest path matrix from the adjacency matrix using Dijkstras algorithm [100]. The distance measure computed can effectively replace the Euclidean distance measure used in the K-Medoid algorithm. The algorithm thus finds k nodes (sinks) in the network that minimizes the average number of hops in respect to the remaining nodes in the network. The final method proposed in the paper is based on the Shortest Path scheme, but also takes routing metrics (LQI) into account.

The four sink deployment methods are tested extensively using simulations. To ensure that the results are not biased by the selection of a particular network layout, we consider four different network scenarios as shown in Figure 3.4. The first scenario represents an open area with no obstructions. The second scenario represents the same area but with a large obstruction (building). More buildings are added in the third scenario. The fourth scenario is an indoor office area.

From the results we observe that the chosen scenario significantly affects the relative difference in performance between the schemes. Even the simplest deployment mechanisms perform well under open-field and ideal conditions such as S1, while they perform poorly in obstructed environments. In fact, for simple unobstructed scenarios, all schemes have comparable results. However, in complex environments with obstructions, the topology-aware method proposed in the paper gives the longest lifetime, the highest number of packets received, and the lowest number of isolated nodes. This result leads to the conclusion that previous sink



(a) S1 (b) S2 (c) S3 (d) S4

Figure 3.4: The four scenarios used in the simulations

deployment mechanisms only validated in simple simulation scenarios may be of little use in real world implementations.

Paper C show that there are two circumstances in which the choice of deployment strategy is irrelevant. The first is in a very simple scenario (as in S1), which is very unlikely to occur in a real deployment. The second is when a large number of sinks are available. Obviously, when a high percentage of the deployed nodes are sinks, the choice of deployment strategy eventually becomes irrelevant, regardless of the scenario. Since these extremes are very unlikely, a more sophisticated method must be considered. Our results show that a constraint-based deployment algorithm that takes the topology into account is paramount.

3.5 Contribution of paper D: Integrating Wireless Sensor Networks in the NATO Network Enabled Capability using Web Services

The work in paper D is concerned about the integration of a surveillance sensor network with an external communication infrastructure. The infrastructure can consist of a variety of end-users with different requirements regarding the data produced by the WSN. For this reason it is important that the WSN provides a clean and flexible interface for communicating queries to the WSN and for returning reports from the WSN. But on the other hand, as the WSN must at all time aim to fulfill its energy efficiency goals, the algorithms and protocols within the WSN must be adaptable based on the queries.

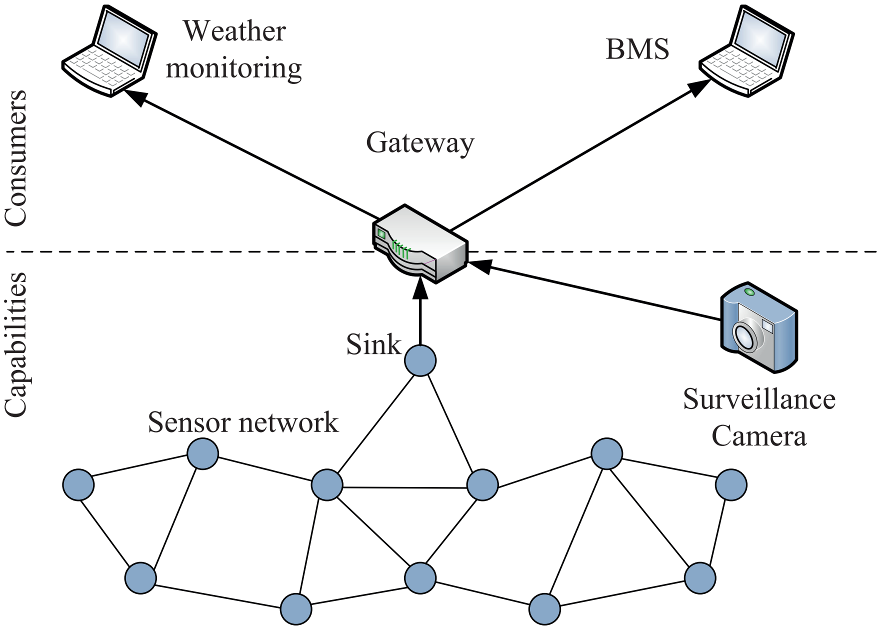


Figure 3.5: Sensor network enabled as a service providing capabilities to different consumers. The gateway may invoke additional services to provide a composite service.

3.5.1 Related Work

Directed diffusion [86] was one of the first initiatives to create a combined routing and querying system for WSNs. In Directed Diffusion, the queries are formatted as *interest* messages which are disseminated to all sensing nodes. Gradients from each sensing node back to the base station are set up during the interest dissemination. Directed diffusion supports in-network data processing and aggregation. However, the protocol is based on a query-driven on demand data model, and is not efficient for event-initiated alarm scenarios, such as e.g., tactical surveillance. The proprietary interest format used in Directed Diffusion is not convenient when used in a multi-consumer WSN such as the one in Figure 3.5. Query processing systems such as TinyDB [101], on the other hand, allows queries to be formulated by multiple consumers simultaneously.

Web services provides a simple method to provide interoperability between computer systems. A special motivation for using Web services in a military setting is that the method is considered the key enabling technology for NATO Network Enabled Capability (NNEC). A Web service based WSN can be realized either by service-enabling each and every sensing node or by providing a Web service gateway that hides the inner WSN protocols. The work by Delicato et al. [102] was an early architecture work belonging to the first category proposing to integrate full Simple Object Access Protocol (SOAP) support in the WSN sensing nodes. The full SOAP is, however, not feasible in a WSN due to the memory, processing and bandwidth limitations.

Although compression can reduce the overhead of XML significantly, and binary coding such as Efficient XML can enable XML to be used even at the tactical edge [103], previous research [104] has shown that the overhead associated with compression libraries make them unsuitable for use on severely limited devices. Thus, in contrast to other WSN implementations, such as [105], we do not attempt to employ XML compression in our WSN in Paper D.

An alternative method to reduce the overhead is to convert the XML messages to a more optimized format at a gateway before relaying them to the WSN devices. The authors of [106] for example, propose WSN-SOA to reduce XML formats to a size applicable for 802.15.4 devices, while Bressan et al. [107] rely on the Constrained RESTful Environments (CoRE) based on REST[1].

3.5.2 Contributions

In paper D, we explore enabling WSNs as a capability using Web services. There are several ways of realizing a capability as a service. For example, a service may be created from scratch, it may function as a front-end to a legacy system, or it may be a combination of existing services. Due to the scarce resources in WSNs it does not make sense to attempt to service-enable each and every sensing node. Instead, we use the wrapping approach, thus allowing existing mechanisms to be used within the WSN, while nodes external to the WSN may configure and receive information from the network using Web services.

The first contribution of the proposed architecture is to provide a Web services wrapper that enables external consumers to interoperate with the sensor network using XML and Web services. The interaction operates in both directions. The wrapper provides an interface (a front-end) to the WSN using established Web services standards (cf. Listing 3.1). The wrapper interfaces with a back-end where the standardized requests are transformed to a much more resource efficient and compact representation of the sensor queries.

The second contribution of the architecture is query dissemination and collection

---

[1] Representational State Transfer (REST)

formation that is adaptive and based on the requests posed by the Web service consumers. The scheme uses a flexible data aggregation scheme running on the WSN nodes. The aggregation scheme is build on extending the forwarding engine in CTP to intercept and process the packets relayed. The processing changes based on the requirements posed by the Web services consumers.

Listing 3.1: XML Query requesting alarm reports when at least four IR detectors are trigged. Only intruders that enters the area during night-time (<1lux) should be reported

```
<GetIntruder>
    <MinPIRDetections>4</MinPIRDetections>
    <LightMaxThreshold>1lux</LightMaxThreshold>
    <Duration>30d</Duration>
    <IncidentReport>http://10.0.0.2/</IncidentReport>
</GetIntruder>
```

The results from running the scheme in a real-world testbed show that the Web services based architecture is feasible and that the WSN can take advantage of the attribute information in Web services queries. It is also possible reduce the energy and bandwidth utilization by employing in-network data-aggregation.

The results from the experiments also show that that there is no point in extending Web services to every sensing node. In contrast, the WSN should (from the Web services perspective) be seen as one single sensing unit, providing filtered and aggregated sensed data to one or more consumers. Therefore, a gateway should be responsible for interacting with the WSN nodes on the back-end side, and the consumers on the front-end side. This approach lets the WSN designers focus on energy-efficient protocols inside the WSN, thus limiting the need for implementing computationally intensive standards to the gateway.

3.6 Contribution of paper E: O-CTP: Hybrid Opportunistic Collection Tree Protocol for Wireless Sensor Networks

Paper E investigates routing mechanisms that are able to operate in high-interference scenarios. Radio interference or deliberate jamming attacks can cause highly unpredictable communication in Wireless Sensor Networks (WSNs). Results from test campaigns show that typical Packet Delivery Ratio (PDR) in deployed networks are between 70 and 99% [45, 108, 109], but could even go as low as 20-40% [110–112]. In paper E we investigate how different routing protocols behave when the sensor network is affected by temporal interference. Furthermore, we propose mechanisms to maximize the delivery rate under these conditions.

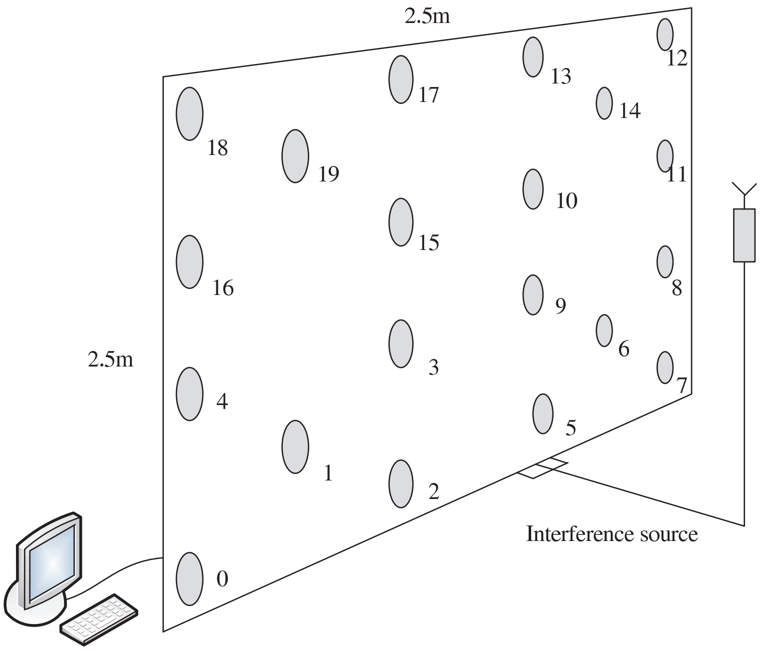


Figure 3.6: The testbed consists of 20 TelosB sensing nodes and a 2.4GHz software controlled interference source.

3.6.1 Related Work

As discussed in Section 2.2.1, traditional routing protocols for WSNs attempts to deal with the dynamics in the underlying network structure by using various metrics. These metrics can for example be the number of hops (MHC) as in TYMO [36], the radio link quality as in MultihopLQI [29] or the Expected Transmission Count (ETX) as in CTP [45]. The metric calculations used in these protocols have, however, difficulties in coping with the rapid changes in the unreliable wireless medium. Consequently, the routing decisions may be based on historic and outdated metrics.

This observation has led to the development of opportunistic routing, which was discussed in detail in Section 2.2.4. The seminal opportunistic routing protocol ExOR [51] serves as a typical example of how an opportunistic routing protocols works. Here, the sender chooses a candidate subset of all its neighboring nodes that could bring the packet closer to its destination. This list is prioritized according to distance and put in the packet header. Each recipient delays a certain time depending on its position in the list before forwarding the packet. These mechanisms are proven to be very effective in error-prone wireless networks. On the other hand, since there are multiple packet forwarders, duplicate packets are bound to occur.

3.6.2 Contributions

Of the related work, none has analyzed or proposed a method to perform automatic switching between traditional routing and opportunistic routing based on the underlying network characteristics. Since both routing alternatives has clear advantages and disadvantages, our contribution aims to investigate the advantages of such switching between the two different routing paradigms.

The main contributions in paper E are:

- Showing that there is a trade-off between traditional routing and opportunistic routing regarding overhead and packet delivery ratio. This trade-off depends on the interference in the network.
- A presentation of a new protocol: O-CTP, which uses traditional routing based on CTP [45] when the network is stable and has reasonably little packet loss. However, if the network is subjected to interference or jamming the protocol switches to opportunistic forwarding. The opportunistic part of the protocol is a simplified version of GeRaF [52, 53].
- An empirical comparison of six routing protocols in an environment with interference. The comparison is conducted using a testbed of 20 TelosB sensing nodes (cf. Figure 3.6) is used. We employ four different interference patterns.
- We show that O-CTP gives the overall best balance between packet delivery ratio and overhead among the protocols studied.

Paper E also study different triggers. The triggers are responsible for switching the protocol operation between traditional routing and opportunistic routing. To avoid breaking the hardware independence of CTP, the triggers are built on monitoring the network status within the forwarding engine of CTP. However, we believe that the protocol can perform better by taking advantage of measurements directly from the physical layer in a cross-layer fashion.

3.7 Contribution of paper F: Experiences from deploying a Wireless Sensor Network for Military Base Protection

Paper F reports on the implementation and experimentation with a WSN for surveillance. Although a military scenario is targeted in the paper, the research methods and the WSN that was produced can be applied for civilian surveillance purposes as well.

3.7.1 Related Work

As reported in [70], different sensors, and even wireless so, has been of use in the tactical arena for about half a century. One of the earliest examples of a modern deployed wireless sensor network for tactical purposes is, however, the work by Arora et al. [113]. They present an experimental system for intrusion detection and target classification using 90 wireless sensing nodes equipped with radar and magnetic sensors.

VigilNet [114] is another WSN surveillance system comprising 70 sensing nodes with magnetic sensors. By employing alarm aggregation among sensing nodes, their system has very low probability of reporting false alarms.

More recently, Rothenpleier et al. have presented FlegSens [115], an experimental surveillance WSN based on PIR for detecting trespassers. Their prototype implementation consists of 16 nodes.

Our system is inspired by the related works, but distinguishes from them in the combination of sensors used and the specific scenario we attack.

3.7.2 Contributions

From the related works, few have evaluated the operative effect of using a military WSN for surveillance. To achieve a positive operative effect, a multitude of components must work as expected: The system must be easy to deploy and use; the sensors must be effective in detecting the target, but at the same time they should not produce false alarms; and the routing protocol must transmit alarm messages towards the sink in a reliable and efficient way.

The contributions in Paper F are threefold. The paper provides:

- A description of a complete implementation of a military Wireless Sensor Network.
- A test methodology and results for two different tactical purposes. The first scenario is a deployment for area surveillance. The setup is similar to the one presented in paper D, albeit in a tactical area. The second scenario is for roadside surveillance and is similar to the scenario used in Paper B (cf. Figure 3.3).
- A detailed list of the lessons learned from the test campaigns. These results yield insights which are relevant also for non-military WSN applications.

The experiments in the paper use a network of 50 TelosB-based sensor nodes. Each node is equipped with three sensors for target detection: A Passive Infrared (PIR) detector, a microphone and a pulsed Doppler radar. Each of the detectors use an algorithm with configurable threshold values for data filtering and target detection. Alarms are transmitted to the sink using CTP.

To obtain qualitative results on the operative effect, the experiments were conducted at the Norwegian Army Combat Maneuver Training Centre (NACMTC). This facility enabled realistic testing of the system in an operative environment. For example, insurgents, which were tasked to pass through the monitored area, were equipped with combat training equipment allowing their movements to be evaluated after the exercise by incorporating GPS position tracking. Hence, the investigation of positive detections, missing detections and false alarms was possible.

The sensor system produced in Paper E is compared with a base line, i.e., using sentry soldiers as sensors, and with using a state-of-the art Unattended Ground System (UGS). The results reveal that our WSN has the best performance of the three alternatives regarding both the detection time and the detection precision.

Chapter 4

Conclusion

4.1 Summary

Wireless Sensor Networks (WSNs) are highly *application-specific*. This means that a generic routing protocol is not necessarily efficient for all WSN applications. For example; a 200-node air-dropped surveillance WSN can require a different routing protocol than a 20-node WSN for monitoring temperature and pressure in the processing industry.

Even though WSNs are more application-specific than other networks, there are two characteristics that apply to virtually all WSNs, regardless of the target application. These two characteristics are the *energy and bandwidth* limitations of the WSN nodes. These limitations must be taken into account when designing protocols for WSNs. To achieve the best performance, the protocols must also be designed with the specific target application in mind.

All the topics studied in this thesis are related to routing protocol issues. The topics are divided in two groups, based on whether they address application-related issues or whether they address more generic issues (i.e., energy-and bandwidth limitations). Among the application-specific topics, the research in the thesis has addressed: *Localization* of sensor nodes and *Interoperability* between the sensor network and external networks. Among the topics related to energy and bandwidth, the thesis has addressed: *Aggregation* of the sensor data and *Routing Efficiency*.

The key findings from the study of these topics can be summarized as follows:

- *Localization* of the sensor nodes can be performed centrally by letting the routing protocol collect ranging estimates. This can be done with cross-layer interactions between a MP2P routing protocol and the adjacent protocol layers.

- *Interoperability* between the WSN and an external network is best performed via a gateway node. This node communicates with the external network via standardized queries and reports. The interface to the WSN must be created by taking into account the limitations of the WSN. The routing protocol can also aggregate data according to the requirements derived from external queries.
- *Aggregation* of data can conserve both energy and bandwidth. Tree-based aggregation can be implemented with minor modifications to a MP2P routing protocol. Aggregation can also be performed within network clusters. Another conclusion is that clustering methods that passively exploit the network topology have less overhead than alternative clustering methods that are based on location information.
- The *routing efficiency* depends on the underlying radio characteristics. For example, when the channel is subjected to interference, the routing protocol has difficulty balancing the PDR and the energy consumption. A hybrid protocol, that switches between traditional routing and opportunistic routing based on the underlying network characteristics, improves this balance and is attractive in a wider range of network conditions. In a data collection WSN, the routing efficiency can be improved by adding support for MP2P. The use of multiple sinks can prolong the lifetime of the WSN further. However, the placement of these sinks is important. The best performance is achieved when the sinks are placed based on link quality measurements.

The research method used to address the research topics has been combination of algorithmically evaluation, simulations, and performance evaluations in test-bed and real-life implementations. The research methods have been used to gradually gain understanding of WSNs and aimed towards real implementations. For example, Paper B simulates a WSN for surveillance. Paper D extends the setup of the surveillance WSN to a test-bed with real sensing nodes. Finally, in Paper F, a real-world surveillance WSN with 50 nodes was constructed and tested in the field.

4.2 Future work

Some of the contributions in the published articles have a potential for further exploration and optimization. In addition, some of the solutions that are presented in the individual papers can be successfully combined.

Paper A investigates how the routing protocol can contribute in localizing the sensor nodes. To achieve high precision, the localization method relies on adequate correlation between the Received Signal Strength Indicator (RSSI) and distance. In most radio circuits, this correlation is only superficial and a better distance measurement method could be required. However, the routing scheme and localization algorithm is flexible and can be adapted to any such method.

Paper B studies different methods to let link-information from the routing protocol contribute to improve cluster generation. Although the paper presents a flexible and well-performing scheme, there are some issues that are left unresolved: For example, the protocol does not implement the distribution of node-to-clusterhead memberships. In a future version, this can be done using a simple cross-layer plugin. Another issues to pursue is reliability and energy distribution.

Paper C presents methods to determine the optimal placement of one or more sinks based on topology information from the routing protocol. One limitation with the scheme is that it does not consider the residual energy in the sensor nodes when proposing better locations for the sinks. Thus, the methods can only be used for deployment of a fresh sensor network and not for relocating sinks in an established and long-running network. Another issue to pursue is to let the algorithm consider only stable and long-lasting links in the calculations rather than using a snap-shot of the current situation.

Paper D proposes a method to use an energy-efficient routing protocol inside the WSN, but at the same time provides a standardized interface to the outside world. An important part of the solution in the paper is an adaptive tree-based aggregation method. The method proved to work well in practice. However, since the method bases its aggregation decisions on the current network structure, packets can be delayed, or even lost, if the underlying network is unstable, since such instability leads to frequent parent switching in CTP. This is an issue that should be investigated. Moreover, in most WSNs, such as the target application studied in the paper, there is a need for both dissemination (P2MP) and collection (MP2P). Since RPL can potentially cover both patterns, providing a similar aggregation service for RPL, could be another issue to pursue.

Paper E investigates a challenge that is of particular interest for military WSNs, but also for other lossy networks: How can the routing protocol be resource-efficient when the network is stable, yet provide sufficient delivery rate when the network is subject to interference or jamming? In the paper we propose a novel routing protocol (O-CTP) that performs automatic switching between traditional routing (CTP) and opportunistic routing based on the current conditions of the underlying network. O-CTP shows promising results in the experiments conducted. There is, however, still a potential for improvement of O-CTP. First, the triggers that decide when to switch from traditional routing to opportunistic routing could be based on cross-layer mechanisms, exploiting channel information from the physical-layer. Second, the code size of the routing protocol should be optimized to make the solution more applicable to memory restricted WSN nodes. Finally, the performance of CTP itself, which O-CTP is based on, suffers heavily from routing loops when the network is subject to interference. This issue deserves further exploration.

Paper F describes the results from implementing and experimenting with a 50-node military WSN for surveillance and reconnaissance. The system was very successful in achieving its design targets. An important lesson learned from the implementation and test campaign was that memory size quickly becomes a limiting factor. For the system described in the paper, this means that several nice-to-have components, such as an advanced low-power listening MAC, in-network programming protocol, and the O-CTP, had to be left out of the implementation. This observation motivates for more optimization in a future version, for example using cross-layering. There is simply not enough space on the nodes to allow duplicated functionality and code on adjacent protocol layers. Microcontrollers with substantially higher memory capacity and the same energy consumption is probably not available in the near future [74].

Finally, many of the mechanisms presented in the papers can be combined. For example, for the test-campaigns in Paper F, the intention was originally to integrate the aggregation method from Paper D and the O-CTP routing protocol from Paper E. However, as mentioned above, the very limited memory space on the TelosB sensing nodes made this impossible without leaving out essential code such as the sensor drivers. A combination that is, however, highly relevant, is to use multiple sinks together with O-CTP. CTP is already an anycast protocol and the use of multiple sinks in O-CTP could be achievable with minor modifications. This would improve both the scalability and the lifetime of an O-CTP-based WSN. Another possibility that was considered during the work of the thesis was to extend CTP to include the localization methods from Paper A. This would provide a simple localization service for any CTP-network.

Bibliography

[1] R. Szewczyk, J. Polastre, A. Mainwaring, and D. Culler, "Lessons from a sensor network expedition," *Wireless Sensor Networks*, pp. 307–322, 2004.

[2] K. Langendoen, A. Baggio, and O. Visser, "Murphy loves potatoes: Experiences from a pilot sensor network deployment in precision agriculture," in *Parallel and Distributed Processing Symposium, 2006. IPDPS 2006. 20th International*. IEEE, 2006, pp. 8–pp.

[3] G. Barrenetxea, F. Ingelrest, G. Schaefer, and M. Vetterli, "The hitchhiker's guide to successful wireless sensor network deployments," in *Proceedings of the 6th ACM conference on Embedded network sensor systems*. ACM, 2008, pp. 43–56.

[4] G. Werner-Allen, K. Lorincz, M. Ruiz, O. Marcillo, J. Johnson, J. Lees, and M. Welsh, "Deploying a wireless sensor network on an active volcano," *Internet Computing, IEEE*, vol. 10, no. 2, pp. 18–25, 2006.

[5] K.-S. Low, W. Win, and M.-J. Er, "Wireless Sensor Networks for Industrial Environments," in *Computational Intelligence for Modelling, Control and Automation, 2005 and International Conference on Intelligent Agents, Web Technologies and Internet Commerce, International Conference on*, vol. 2, 2005, pp. 271–276.

[6] M. Winkler, K. Tuchs, K. Hughes, and G. Barclay, "Theoretical and practical aspects of military wireless sensor networks," *Journal of Telecommunications and Information Technology*, vol. 2, pp. 37–45, 2008.

[7] J. Altmann, H. Fisher, and H. van der Graaf, "Sensors for Peace: Applications, Systems and Legal Requirements for Monitoring in Peace Operations," UNIDIR, Tech. Rep., 1998.

[8] C. Karlof, N. Sastry, and D. Wagner, "TinySec: a link layer security architecture for wireless sensor networks," in *Proceedings of the 2nd interna-*

tional conference on Embedded networked sensor systems. ACM, 2004, pp. 162–175.

[9] N. Sastry and D. Wagner, “Security considerations for IEEE 802.15. 4 networks,” in *Proceedings of the 3rd ACM workshop on Wireless security*. ACM, 2004, pp. 32–42.

[10] P. Pecho, J. Nagy, P. Hanáček, and M. Drahanskỳ, “Secure collection tree protocol for tamper-resistant wireless sensors,” *Security Technology*, pp. 217–224, 2009.

[11] The ns Team, “The ns network simulator,” Accessed 2012. [Online]. Available: http://www.nsnam.org

[12] P. Levis, N. Lee, M. Welsh, and D. Culler, “TOSSIM: Accurate and scalable simulation of entire TinyOS applications,” in *Proceedings of the 1st international conference on Embedded networked sensor systems*. ACM, 2003, pp. 126–137.

[13] A. Boulis, “Castalia: revealing pitfalls in designing distributed algorithms in WSN,” in *Proceedings of the 5th international conference on Embedded networked sensor systems*. ACM, 2007, pp. 407–408.

[14] OMNet++ Team, “The OMNet++ network simulator,” Accessed 2012. [Online]. Available: http://www.omnetpp.org/

[15] K. Kim, G. Montenegro, S. Park, I. Chakeres, and C. Perkins, “Dynamic MANET On-demand for 6LoWPAN (DYMO-low) Routing,” Internet Engineering Task Force, Internet-Draft, Jun. 2007, Expired.

[16] University of Murcia, “UM-OLSR,” no. http://masimum.dif.um.es/, Accessed 2010.

[17] K. Kim, S. Park, G. Montenegro, S. Yoo, and K. Kushalnagar, “6LoWPAN Ad Hoc On-Demand Distance Vector Routing (LOAD),” Internet Engineering Task Force, Internet-Draft, Jun. 2007, work in progress. [Online]. Available: http://www.ietf.org/internet-drafts/draft-daniel-6lowpan-load-adhoc-routing-03.txt

[18] S. Kurkowski, T. Camp, and M. Colagrosso, “MANET simulation studies: the incredibles,” in *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 9, no. 4. ACM, 2005, pp. 50–61.

[19] L. Girod, J. Elson, A. Cerpa, T. Stathopoulos, N. Ramanathan, and D. Estrin, “EmStar: a software environment for developing and deploying wire-

less sensor networks," in *Proceedings of the annual conference on USENIX Annual Technical Conference*. USENIX Association, 2004, pp. 24–24.

[20] G. Werner-Allen, P. Swieskowski, and M. Welsh, "Motelab: A wireless sensor network testbed," in *Proceedings of the 4th international symposium on Information processing in sensor networks*. IEEE Press, 2005, p. 68.

[21] I. Chatzigiannakis, S. Fischer, C. Koninis, G. Mylonas, and D. Pfisterer, "WISEBED: An Open Large-Scale Wireless Sensor Network Testbed," in *Sensor Applications, Experimentation, and Logistics*, ser. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, N. Komninos, Ed. Springer Berlin Heidelberg, 2010, vol. 29, pp. 68–87. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-11870-8_6

[22] J. Polastre, R. Szewczyk, and D. Culler, "Telos: enabling ultra-low power wireless research," in *IPSN '05: Proceedings of the 4th international symposium on Information processing in sensor networks*. Piscataway, NJ, USA: IEEE Press, 2005.

[23] TinyOS Team, "TinyOS homepage," Accessed 2012. [Online]. Available: http://www.tinyos.net/

[24] K. Akkaya and M. Younis, "A survey on routing protocols for wireless sensor networks," *Ad hoc networks*, vol. 3, no. 3, pp. 325–349, 2003.

[25] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur, and R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks," RFC 6550 (Proposed Standard), Internet Engineering Task Force, Mar. 2012. [Online]. Available: http://www.ietf.org/rfc/rfc6550.txt

[26] T. Watteyne, A. Molinaro, M. Richichi, and M. Dohler, "From MANET to IETF ROLL standardization: A paradigm shift in WSN routing protocols," *Communications Surveys & Tutorials, IEEE*, vol. 13, no. 4, pp. 688–707, 2011.

[27] D. Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," *Wireless Networks*, vol. 11, no. 4, pp. 419–434, 2005.

[28] T. Instruments. (2007) Chipcon cc2420 datasheet. [Online]. Available: http://focus.ti.com/lit/ds/symlink/cc2420.pdf

[29] G. Tolle. MultihopLQI source code. [Online]. Available: http://www.tinyos.net/tinyos-2.x/tos/lib/net/lqi/

[30] J. Chang and L. Tassiulas, “Maximum lifetime routing in wireless sensor networks,” *IEEE/ACM Transactions on Networking (TON)*, vol. 12, no. 4, pp. 609–619, 2004.

[31] 6LoWPAN Team, “6LoWPAN homepage,” Accessed 2012. [Online]. Available: http://datatracker.ietf.org/wg/6lowpan/charter/

[32] C. Perkins, E. Belding-Royer, and S. Das, “Ad hoc On-Demand Distance Vector (AODV) Routing,” RFC 3561 (Experimental), Internet Engineering Task Force, Jul. 2003. [Online]. Available: http://www.ietf.org/rfc/rfc3561.txt

[33] I. Chakeres and C. Perkins, “Dynamic MANET On-demand (DYMO) Routing,” Internet Engineering Task Force, Internet-Draft draft-ietf-manet-dymo-19, Sep. 2009, work in progress. [Online]. Available: http://www.ietf.org/internet-drafts/draft-ietf-manet-dymo-19.txt

[34] C. Gomez, P. Salvatella, O. Alonso, and J. Paradells, “Adapting AODV for IEEE 802.15.4 Mesh Sensor Networks: Theoretical Discussion and Performance Evaluation in a Real Environment,” in *WOWMOM ’06*, 2006, pp. 159–170.

[35] D. Gao, Y. Niu, and H. Zhang, “Micro Sensor Routing Protocol in IPv6 wireless sensor network,” *Networking, Sensing and Control, 2009. ICNSC ’09. International Conference on*, pp. 55 –59, mar. 2009.

[36] R. Thouvenin. TYMO source code. [Online]. Available: http://www.tinyos.net/tinyos-2.x/tos/lib/net/tymo/

[37] I. D. Chakeres and L. Klein-Berndt, “AODVjr, AODV simplified,” *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 6, no. 3, pp. 100–101, 2002.

[38] R. Simón Carbajo, M. Huggard, and C. McGoldrick, “An end-to-end routing protocol for peer-to-peer communication in wireless sensor networks,” in *MiNEMA ’08: Proceedings of the 6th workshop on Middleware for network eccentric and mobile applications*. New York, NY, USA: ACM, 2008, pp. 5–9.

[39] T. Clausen, J. Yi, and A. de Verdiere, “LOADng: Towards AODV version 2,” in *Vehicular Technology Conference (VTC Fall), 2012 IEEE*. IEEE, 2012, pp. 1–5.

[40] T. Clausen and P. Jacquet, “Optimized Link State Routing Protocol (OLSR),” RFC 3626 (Experimental), Internet Engineering Task Force, Oct. 2003. [Online]. Available: http://www.ietf.org/rfc/rfc3626.txt

[41] R. Murty, G. Mainland, I. Rose, A. Chowdhury, A. Gosain, J. Bers, and M. Welsh, “CitySense: An urban-scale wireless sensor network and testbed,” in *Technologies for Homeland Security, 2008 IEEE Conference on*. IEEE, 2008, pp. 583–588.

[42] S. Mahfoudh and P. Minet, “An energy efficient routing based on OLSR in wireless ad hoc and sensor networks,” in *Advanced Information Networking and Applications-Workshops, 2008. AINAW 2008. 22nd International Conference on*. IEEE, 2008, pp. 1253–1259.

[43] J. Davis Sr, “An analysis of network and sensor performance within IEEE 802. x wireless MESH networks in the Tactical Network Topology (TNT),” DTIC Document, Tech. Rep., 2005.

[44] J. Yi, T. Clausen, and A. de Verdiere, “Efficient Data Acquisition in Sensor Networks: Introducing (the) LOADng Collection Tree Protocol,” in *Proceedings of the 8th IEEE International Conference on Wireless Communications, Networking and Mobile Computing - WiCom-2012*. IEEE, 2012.

[45] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis, “Collection tree protocol,” in *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, ser. SenSys ’09. New York, NY, USA: ACM, 2009, pp. 1–14.

[46] P. Levis, N. Patel, D. Culler, and S. Shenker, “Trickle: A self-regulating algorithm for code propagation and maintenance in wireless sensor networks,” in *Proc. of the USENIX NSDI Conf.*, 2004, pp. 2–2.

[47] G. Tolle and D. Culler, “Design of an application-cooperative management system for wireless sensor networks,” in *Wireless Sensor Networks, 2005. Proceeedings of the Second European Workshop on*, Feb. 2005, pp. 121–132. [Online]. Available: http://dx.doi.org/10.1109/EWSN.2005.1462004

[48] P. Thubert, “Objective Function Zero for the Routing Protocol for Low-Power and Lossy Networks (RPL),” RFC 6552 (Proposed Standard), Internet Engineering Task Force, Mar. 2012. [Online]. Available: http://www.ietf.org/rfc/rfc6552.txt

[49] O. Gnawali and P. Levis, “The Minimum Rank with Hysteresis Objective Function,” RFC 6719 (Proposed Standard), Internet Engineering Task Force, Sep. 2012. [Online]. Available: http://www.ietf.org/rfc/rfc6719.txt

[50] J. Ko, O. Gnawali, D. Culler, and A. Terzis, “Evaluating the Performance of RPL and 6LoWPAN in TinyOS,” in *Proceedings of the Workshop on Extending the Internet to Low Power and Lossy Networks (IPSN 2011)*, 2011.

[51] S. Biswas and R. Morris, “Opportunistic routing in multi-hop wireless networks,” *ACM SIGCOMM’04*, vol. 34, no. 1, pp. 69–74, 2004.

[52] M. Zorzi and R. Rao, “Geographic random forwarding (GeRaF) for ad hoc and sensor networks: energy and latency performance,” *Mobile Computing, IEEE Transactions on*, vol. 2, no. 4, pp. 349–365, 2003.

[53] ——, “Geographic random forwarding (GeRaF) for ad hoc and sensor networks: multihop performance,” *IEEE transactions on Mobile Computing*, pp. 337–348, 2003.

[54] O. Landsiedel, E. Ghadimi, S. Duquennoy, and M. Johansson, “Low power, low delay: opportunistic routing meets duty cycling,” in *Proceedings of the 11th international conference on Information Processing in Sensor Networks*. ACM, 2012, pp. 185–196.

[55] D. Clark and D. Tennenhouse, “Architectural considerations for a new generation of protocols,” in *ACM SIGCOMM Computer Communication Review*, vol. 20, no. 4. ACM, 1990, pp. 200–208.

[56] V. Srivastava and M. Motani, “Cross-layer design: a survey and the road ahead,” *Communications Magazine, IEEE*, vol. 43, no. 12, pp. 112–119, 2005.

[57] O. Chipara, Z. He, G. Xing, Q. Chen, X. Wang, C. Lu, J. Stankovic, and T. Abdelzaher, “Real-time power-aware routing in sensor networks,” in *Quality of Service, 2006. IWQoS 2006. 14th IEEE International Workshop on*. IEEE, 2006, pp. 83–92.

[58] A. Ruzzelli, G. O’Hare, and R. Jurdak, “MERLIN: Cross-layer integration of MAC and routing for low duty-cycle sensor networks,” *Ad Hoc Networks*, vol. 6, no. 8, pp. 1238–1257, 2008.

[59] R. Fonseca, O. Gnawali, K. Jamieson, and P. Levis, “Four-bit wireless link estimation,” in *Proceedings of the Sixth Workshop on Hot Topics in Networks (HotNets VI)*, vol. 2007, 2007.

[60] H. Wu, Q. Luo, and W. Xue, “Distributed cross-layer scheduling for in-network sensor query processing,” in *Pervasive Computing and Communications, 2006. PerCom 2006. Fourth Annual IEEE International Conference on*. IEEE, 2006, pp. 10–pp.

[61] Y. Gong, G. Chen, and L. Tan, “A Balanced Serial K-Means Based Clustering Protocol for Wireless Sensor Networks,” *WiCOM ’08. Proceeding*, pp. 1 –6, oct. 2008.

[62] V. Kawadia and P. Kumar, “A cautionary perspective on cross-layer design,” *Wireless Communications, IEEE*, vol. 12, no. 1, pp. 3–11, 2005.

[63] D. Culler, P. Dutta, C. Ee, R. Fonseca, J. Hui, P. Levis, J. Polastre, S. Shenker, I. Stoica, G. Tolle *et al.*, “Towards a sensor network architecture: Lowering the waistline,” in *Proceedings of the 10th conference on Hot Topics in Operating Systems*, vol. 10, 2005, pp. 24–24.

[64] J. Polastre, J. Hui, P. Levis, J. Zhao, D. Culler, S. Shenker, and I. Stoica, “A unifying link abstraction for wireless sensor networks,” in *Proceedings of the 3rd international conference on Embedded networked sensor systems*. ACM, 2005, pp. 76–89.

[65] P. Marrón, A. Lachenmann, D. Minder, J. Hahner, R. Sauter, and K. Rothermel, “TinyCubus: a flexible and adaptive framework sensor networks,” in *Wireless Sensor Networks, 2005. Proceeedings of the Second European Workshop on*. IEEE, 2005, pp. 278–289.

[66] A. Dunkels, F. Österlind, and Z. He, “An adaptive communication architecture for wireless sensor networks,” in *Proceedings of the 5th international conference on Embedded networked sensor systems*. ACM, 2007, pp. 335–349.

[67] I. Akyildiz, M. Vuran, and O. Akan, “A cross-layer protocol for wireless sensor networks,” in *Information Sciences and Systems, 2006 40th Annual Conference on*. IEEE, 2006, pp. 1102–1107.

[68] S. Gajjar, S. Pradhan, and K. Dasgupta, “Cross layer architectural approaches for Wireless Sensor Networks,” in *Recent Advances in Intelligent Computational Systems (RAICS), 2011 IEEE*. IEEE, 2011, pp. 557–562.

[69] A. Dunkels, B. Gronvall, and T. Voigt, “Contiki - a lightweight and flexible operating system for tiny networked sensors,” in *Local Computer Networks, 2004. 29th Annual IEEE International Conference on*. IEEE, 2004, pp. 455–462.

[70] J. Gatlin, “Project CHECO Southeast Asia Report. Igloo White (Initial Phase),” DTIC Document, Tech. Rep., 1968.

[71] J. Kahn, R. Katz, and K. Pister, “Next century challenges: mobile networking for Smart Dust,” in *Proceedings of the 5th annual ACM/IEEE interna-*

tional conference on Mobile computing and networking. ACM, 1999, pp. 271–278.

[72] L. Mottola and G. Picco, “MUSTER: Adaptive energy-aware multisink routing in wireless sensor networks,” *Mobile Computing, IEEE Transactions on*, vol. 10, no. 12, pp. 1694–1709, 2011.

[73] B. Raman and K. Chebrolu, “Censor networks: a critique of "sensor networks" from a systems perspective,” *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 3, pp. 75–78, 2008.

[74] P. Levis, “Experiences from a Decade of TinyOS Development,” in *10th USENIX Symposium on Operating Systems Design and Implementation (OSDI ’12)*. ACM SIGOPS, 2012, pp. 207–220.

[75] Arduino Team, “Arduino Home Page,” Accessed 2012. [Online]. Available: http://www.arduino.cc/

[76] S. Chen, G. Fan, and J.-H. Cui, “Avoid ’void’ in geographic routing for data aggregation in sensor networks,” *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 1, no. 4, pp. 169–178, 2006.

[77] D. Espes and Z. Mammeri, “Backbone-based location-aided routing algorithm to reduce control packets of AODV protocol,” in *Mobility ’06: Proceedings of the 3rd international conference on Mobile technology, applications systems*. New York, NY, USA: ACM, 2006, p. 15.

[78] S. Ratnasamy, B. Karp, L. Yin, F. Yu, D. Estrin, R. Govindan, and S. Shenker, “GHT: a geographic hash table for data-centric storage,” in *WSNA ’02: Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*. New York, NY, USA: ACM, 2002, pp. 78–87.

[79] D. A. Tran and T. Nguyen, “Localization In Wireless Sensor Networks Based on Support Vector Machines,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 19, no. 7, pp. 981–994, 2008.

[80] D. Niculescu and B. Nath, “Ad hoc positioning system (APS) using AOA,” *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, vol. 3, pp. 1734–1743 vol.3, April 2003.

[81] L. Girod and D. Estrin, “Robust range estimation using acoustic and multimodal sensing,” *Intelligent Robots and Systems, 2001. Proceedings. 2001*

IEEE/RSJ International Conference on, vol. 3, pp. 1312–1320 vol.3, August 2002.

[82] M. L. Sichitiu and V. Ramadurai, "Localization of wireless sensor networks with a mobile beacon," *Mobile Ad-hoc and Sensor Systems, 2004 IEEE International Conference on*, pp. 174–183, October 2004.

[83] S. Lee, E. Kim, C. Kim, and K. Kim, "Localization with a mobile beacon based on geometric constraints in wireless sensor networks," *Wireless Communications, IEEE Transactions on*, vol. 8, no. 12, pp. 5801–5805, December 2009. [Online]. Available: http://dx.doi.org/10.1109/TWC.2009.12.090319

[84] V. Tam, K. Cheng, and K. Lui, "Using micro-genetic algorithms to improve localization in wireless sensor networks," *Journal of Communications*, vol. 1, no. 4, pp. 1–10, 2006.

[85] S. Lindsey, C. Raghavendra, and K. Sivalingam, "Data gathering algorithms in sensor networks using energy metrics," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 13, no. 9, pp. 924 – 935, sep. 2002.

[86] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: a scalable and robust communication paradigm for sensor networks," in *Proceedings of the 6th annual international conference on Mobile computing and networking*, ser. MobiCom '00. New York, NY, USA: ACM, 2000, pp. 56–67.

[87] I. Solis and K. Obraczka, "The impact of timing in data aggregation for sensor networks," in *2004 IEEE International Conference on Communications (IEEE Cat. No.04CH37577)*. IEEE, 2004, pp. 3640–3645 Vol.6.

[88] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks," in *System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on*. IEEE, 2000, pp. 10–pp.

[89] W. K. Lai, C.-S. Shieh, and Y.-T. Lee, "A cluster-based routing protocol for wireless sensor networks with adjustable cluster size," in *ChinaCOM 2009*, Aug 2009, pp. 1 –5.

[90] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical Review E*, vol. 69, no. 2, pp. 026 113+, Feb 2004.

[91] O. Drugan, T. Plagemann, and E. Munthe-Kaas, “Detecting communities in sparse MANETs,” *IEEE/ACM Transactions on Networking (TON)*, vol. 19, no. 5, pp. 1434–1447, 2011.

[92] T. Clausen, P. Jacquet, and L. Viennot, “Investigating the impact of partial topology in proactive MANET routing protocols,” in *Wireless Personal Multimedia Communications, 2002. The 5th International Symposium on*, vol. 3. IEEE, 2002, pp. 1374–1378.

[93] P. E. Villanueva-Pena, T. Kunz, and P. Dhakal, “Extending network knowledge: making OLSR a quality of service conducive protocol,” in *IWCMC ’06*. New York, NY, USA: ACM, 2006, pp. 103–108.

[94] E. I. Oyman and C. Ersoy, “Multiple Sink Network Design Problem in Large Scale Wireless Sensor Networks,” in *In: Proc. of 1 st Int. Conf. on Communications (ICC 2004)*, 2004.

[95] Z. Vincze, R. Vida, and A. Vidacs, “Deploying Multiple Sinks in Multi-hop Wireless Sensor Networks,” in *In: Pervasive Services, IEEE International Conference on*, July 2007, pp. 55–63.

[96] S. Dai, C. Tang, S. Qiao, K. Xu, H. Li, and J. Zhu, “Optimal Multiple Sink Nodes Deployment in Wireless Sensor Networks Based on Gene Expression Programming,” in *Communication Software and Networks, 2010. ICCSN’10. Second International Conference on*. IEEE, 2010, pp. 355–359.

[97] S. Misra, S. D. Hong, G. Xue, and J. Tang, “Constrained Relay Node Placement in Wireless Sensor Networks to Meet Connectivity and Survivability Requirements,” in *In: IEEE INFOCOM 2008 Proceedings*, April 2008, pp. 281–285.

[98] D. Yang, S. Misra, X. Fang, G. Xue, and J. Zhang, “Two-Tiered Constrained Relay Node Placement in Wireless Sensor Networks: Efficient Approximations,” in *In: IEEE Secon 2010 Proceedings*, June 2010, pp. 1–9.

[99] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley-Interscience, March 1990.

[100] E. W. Dijkstra, “A note on two problems in connexion with graphs,” *Numerische mathematik*, vol. 1, no. 1, pp. 269–271, Dec. 1959.

[101] S. R. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, “TinyDB: an acquisitional query processing system for sensor networks,” *ACM Trans. Database Syst.*, vol. 30, pp. 122–173, March 2005.

[102] F. C. Delicato, P. F. Pires, L. Pirmez, and L. F. R. d. C. Carmo, "A Flexible Web Service Based Architecture for Wireless Sensor Networks," in *Proceedings of the 23rd International Conference on Distributed Computing Systems*, ser. ICDCSW '03. Washington, DC, USA: IEEE Computer Society, 2003, pp. 730–.

[103] J. Schneider, "Efficient XML: Taking Net-Centric Operations to the Edge," in *13th International Command and Control Research and Technology Symposium (ICCRTS)*, June 2008.

[104] D. Hadzic *et al.*, "Web services i nettverk med begrenset datarate (in Norwegian)," in *FFI report 2006/03886*, 2006.

[105] N. Glombitza, D. Pfisterer, and S. Fischer, "Integrating wireless sensor networks into web service-based business processes," in *Proceedings of the 4th International Workshop on Middleware Tools, Services and Run-Time Support for Sensor Networks*, ser. MidSens '09. New York, NY, USA: ACM, 2009, pp. 25–30.

[106] J. Leguay, M. Lopez-Ramos, K. Jean-Marie, and V. Conan, "An efficient service oriented architecture for heterogeneous and dynamic wireless sensor networks," in *Local Computer Networks, 2008. LCN 2008. 33rd IEEE Conference on*, October 2008, pp. 740–747.

[107] N. Bressan, L. Bazzaco, N. Bui, P. Casari, L. Vangelista, and M. Zorzi, "The deployment of a smart monitoring system using wireless sensor and actuator networks," in *Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on*. IEEE, 2010, pp. 49–54.

[108] S. Lin, G. Zhou, K. Whitehouse, Y. Wu, J. Stankovic, and T. He, "Towards stable network performance in wireless sensor networks," in *RTSS'09*. IEEE, 2009, pp. 227–237.

[109] T. Liu, A. Kamthe, L. Jiang, and A. Cerpa, "Performance Evaluation of Link Quality Estimation Metrics for Static Multihop Wireless Sensor Networks," in *IEEE SECON '09*, june 2009, pp. 1 –9.

[110] V. Shnayder, B. Chen, K. Lorincz, T. Fulford-Jones, and M. Welsh, "Sensor networks for medical care," in *Conference On Embedded Networked Sensor Systems: Proceedings of the 3 rd international conference on Embedded networked sensor systems*, vol. 2, no. 04, 2005, pp. 314–314.

[111] A. Wood, J. Stankovic, and G. Zhou, "DEEJAM: Defeating energy-efficient jamming in IEEE 802.15. 4-based wireless networks," in *SECON'07*. IEEE, 2007, pp. 60–69.

[112] J. Vanhie-Van Gerwen, E. De Poorter, B. Latré, I. Moerman, and P. Demeester, "Real-life performance of protocol combinations for wireless sensor networks," in *2010 IEEE SUTC*, 2010, pp. 189–196.

[113] A. Arora, P. Dutta, S. Bapat, V. Kulathumani, H. Zhang, V. Naik, V. Mittal, H. Cao, M. Demirbas, M. Gouda *et al.*, "A line in the sand: a wireless sensor network for target detection, classification, and tracking," *Computer Networks*, vol. 46, no. 5, pp. 605–634, 2004.

[114] T. He, S. Krishnamurthy, L. Luo, T. Yan, L. Gu, R. Stoleru, G. Zhou, Q. Cao, P. Vicaire, J. Stankovic *et al.*, "VigilNet: An integrated sensor network system for energy-efficient surveillance," *ACM Transactions on Sensor Networks (TOSN)*, vol. 2, no. 1, pp. 1–38, 2006.

[115] P. Rothenpieler, D. Krüger, D. Pfisterer, S. Fischer, D. Dudek, C. Haas, A. Kuntz, and M. Zitterbart, "Flegsens - secure area monitoring using wireless sensor networks," *Proceedings of the 4th Safety and Security Systems in Europe*, 2009.

Part II

Research papers

Paper A :

Localization in Wireless Sensor Networks based on Ad hoc Routing and Evolutionary Computation

J. Flathagen and R. Korsnes

In proceedings of the IEEE Military Communications Conference, MILCOM, San Jose, CA, USA, October 31–November 4, 2010, pp. 1173–1178, ISBN: 978-1-4244-8179-8.

Localization in Wireless Sensor Networks based on Ad hoc Routing and Evolutionary Computation

Joakim Flathagen *‡, Reinert Korsnes *
* Norwegian Defence Research Establishment
‡ University Graduate Center Kjeller/NTNU
Email: {joakim.flathagen, reinert.korsnes}@ffi.no

***Abstract*—We propose evolutionary computation to estimate positions of nodes within a sensor network. The approach uses signal strength measurements between nodes and given positions for a subset of these nodes (anchor nodes). The signal strength measurements and routing requests take place simultaneously. A data collecting unit (sink node) receives distance estimates which are input to the evolutionary algorithm projecting node positions. This evolutionary approach can sort out data outliers and hence produce robust estimates of node positions. The present work contributes to decrease the cost and complexity of applying sensor networks. The approach also provides redundancy for the node positioning where alternative methods fail. The present simulations show examples of network generation and routing combined with estimation of node positions.**

***Index Terms*—Sensor Networks, Localization, Evolutionary Algorithm, DYMO-low**

I. INTRODUCTION

This work argues for evolutionary computation for localization within wireless sensor networks (WSNs). The approach provides low cost and robust localization utilizing signal strength measurements attached to routing control packets. A genetic algorithm [1] here searches for possible sets of node positions explaining these measurements. This search resembles the process of natural evolution.

Wireless sensor networks can consist of hundreds or even thousands of small sensing devices. Location awareness is crucial for many WSN applications such as environment monitoring and military surveillance. Sensor networks can also utilize location information for routing, cooperative computation, data fusion and location dependent sensor data requests [2], [3], [4], [5]. GPS positioning for every sensor node is not a general solution to localize nodes in sensor networks. It may be costly and impractical and sometimes irrelevant. Development of a precise, low-cost, reliable and fast converging localization scheme is therefore essential for the function of many sensor networks.

A. Related work

Localization schemes for sensor networks can be categorized depending on ranging, hardware, mobility, centralization and deployment restrictions. These will be briefly discussed here. *Range-independent* localization schemes [6], [7] determine node positions without using any special measurements. Localization is in this case a result of connectivity information. Even if *low cost hardware* can provide this capability, the position estimates are imprecise, especially for sparse networks. Additional information can improve the position estimates. Such additional information can be from measurements of distance or direction to known reference positions. These estimates are typically from measurements of time of arrival (TOA), time difference of arrival (TDOA), angle of arrival (AOA) or received signal strength indicator (RSSI). Examples of hardware for this type of measurements are ultrasound devices [8], angle-of-arrival antenna arrays [9] or laser [10]. However, introducing such additional hardware increases cost and complexity of sensor network systems.

The *deployment method* of sensor networks often determines the choice of its localization scheme. *Mobile* nodes can sometimes aid localization of individual sensor nodes [11], [12]. In such cases a vehicle, robot, or soldier enters the sensor field to assist the localization scheme. Sensor drop from airplanes, on the contrary, requires autonomous localization. *Centralized* methods may then be the only viable approach. Previous assumptions indicate that centralized methods are impractical due to high communication costs. Our proposal argues against this conclusion.

There are many previous attempts to estimate node positions in sensor networks using centralized search techniques. Kannan *et al.* [13], for example, creates an initial estimate of positions by applying simulated annealing and attempts to correct possible misplaced nodes thereafter. Tam *et al.* [14] apply evolutionary optimization to improve position estimates after initial triangulation, while Zhang *et al.* [15] more directly apply evolutionary computing for localization.

B. Our contribution

Most existing sensor network platforms can use signal strength measurements without additional hardware by employing RSSI from the IEEE 802.15.4 chipset. Our proposal takes advantage of such low-cost measurements to estimate node positions. However, other and more precise measurements (such as acoustic ranging) can be utilized if this is supported by the hardware. The proposal includes a prevalent WSN ad hoc routing protocol, DYMO-low [16] that is exploited to fetch and distribute RSSI values. By combining route establishment and localization our approach contributes in reducing the effort and complexity of sensor network deployments. The sink employs an *evolutionary approach* to provide estimation of node positions using the information gathered. This gives a reasonable robust solution even for poor signal strength measurements. Our implementation of an evolutionary algorithm is simple and intuitive and relaxes the search space as compared to similar work [14], [15]. Section III below clarifies this relaxation via variation of fitness measures and allowing for data outliers.

The remainder of this paper is organized as follows. Section II describes collection and distribution of RSSI measurements using DYMO-low. Section III elaborates the evolutionary localization algorithm. Section IV presents simulations results and analysis. Section V gives concluding remarks.

II. SIMULTANEOUS ROUTING AND RSSI MEASUREMENT

The support of RSSI measurements is common in prevalent IEEE 802.15.4 implementations. But signal strength measurements, especially indoors, may provide imprecise distance estimates due to multipath propagation, reflection and channel fading. RSSI measurements are therefore mainly ignored for node positioning within sensor networks. However, recent research by Holland *et al.* [17] shows that RSSI measurements on sensor nodes strongly correlate with distance. RSSI measurements on a link are also symmetric [18] and localization schemes for small-scale networks can utilize this property [19].

Our scheme extends this work by using RSSI to aid localization in medium to large-scale multihop networks. We have chosen a centralized approach to localization, meaning that only the sink is involved in computation of the node positions. It is worth noting that distributed protocol designs are traditionally preferred before centralized designs in networking systems due to the fault tolerance and lack of scalability of the latter approach. However, we argue that in most WSNs, the sink node is already a single point of failure, and the fault tolerance is not increased by centralizing the localization algorithm. In fact, it simplifies the protocol design and its implementation. Further, the scalability of the centralized algorithm is not a big concern compared to a distributed design, as the sink node can be equipped with several orders of magnitude more memory and CPU than the sensor nodes.

The approach in this paper exploits route establishment to perform measurements and transport RSSI values to the sink by extending the reactive distance vector routing protocol DYMO-low [16]. DYMO-low is intended for use on IEEE 802.15.4 devices and is based on the principle of flooding route requests (RREQ) and unicasting route replies (RREP) as known from AODV [20] and DYMO [21]. Our extension introduces two new messages to the protocol, Localization Route Request (LRREQ) and Localization Route Reply (LRREP). Fig. 1 illustrates the protocol operation consisting of a request phase and a reply phase.

A. Request phase

The sink initiates the network by announcing its address via a Localization Route Request (LRREQ). This message can be seen as a proactive route request destined to *all* nodes in the network. The LRREQ is flooded similarly as a regular DYMO-low routing request (RREQ). The nodes which receive the LRREQ, retransmit the packet only once. This means that all nodes will receive a copy of the LRREQ from each of its neighbors (cf Fig. 1). When a node receives a LRREQ packet, it performs a RSSI measurement and subsequently stores its value and the address of the sender. As the LRREQ disseminates from the sink to the entire network, all nodes will eventually obtain a distance estimate to each of its one-hop neighbors with no more cost than a regular Route Request.

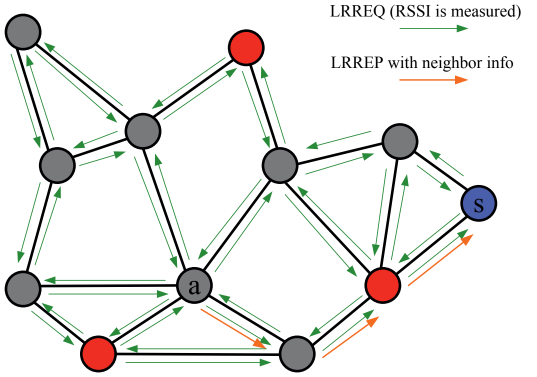


Fig. 1. Initialization of sensor network with anchor nodes (red). The sink S (blue) starts route discovery. All nodes collect RSSI measurements from their neighbors while the LRREQ disseminates. They report their measurements as attachments to their individual route reply back to the sink. Anchor nodes also report their positions. The sink then estimates node positions using evolutionary computation.

B. Reply phase

A node will, after receiving a packet, respond back to the sink using a LRREP (Localization Route Reply). This transmission takes place after a random time delay to avoid network congestion and collisions. Fig. 1 illustrates this process. The response message extends the regular Route Reply defined in DYMO-low, with a list of the one-hop neighbors and their correspondent RSSI measurement values. Anchor nodes will also add their own present position that can be from a GPS receiver. The sink will eventually receive LRREPs from all the sensor nodes in the network. It will then use this information to estimate the individual locations.

Note that the sink receives two RSSI measurement values for each link in the network (one from each link end). The distance estimation applies the mean of the values from each link. The duplicated information also enables reconstruction of missing LRREP information. If the LRREP from for example node a in Fig. 1 is lost on its way to the sink due to congestion or collision, the sink can use the LRREPs from the surrounding neighbors of a to estimate its location. This gives a minimum level of redundancy.

C. Features and considerations

The above approach provides two important additional features.

i) After a complete request/reply phase, all nodes in the network has a valid route to the sink, making them ready to perform their sensing task immediately. Notice that if standard DYMO-low is used, route requests must be initiated from each node in the network to accomplish this. This could cause tremendous overhead due to the flooded route requests. Our approach, on the contrary, limits this to just *one* sink-initiated route request and considerably reduces the number of messages flooded in the network.

ii) By using the information provided in the LRREPs, the network operator at the sink will know which of the nodes in the network are fully functional, within range and operating. It could later be useful to add other sensor information to the LRREP message in order to inform the sink that the individual sensors on each node are operating satisfactorily after deployment.

The size of the LRREP may end up being too large for a IEEE 802.15.4 frame if i) a node has a very large number of one-hop neighbors, or ii) the LRREP includes much status information from the sensors on the node, or iii) a combination of the two. The LRREP will still be transmitted, as the 6LoWPAN sublayer [22] elegantly fragments and reassembles datagrams being larger than a MAC-frame. However, in sparse and medium density networks fragmentation of the LRREP are not likely to occur.

III. LOCALIZATION THROUGH EVOLUTIONARY COMPUTATION

This section describes our centralized evolutionary computation method to estimate node positions within the sensor network. Evolutionary approaches generally provide capacities for searching through large sets of possible explanations of given data. For our purpose, such techniques are therefore particularly interesting for estimation of node positions from error-prone data, such as signal strength measurements.

Parameter estimation is often equivalent to model identification from data. A set of parameter values then typically defines a model within for example a physical setting. The actual set of parameters is in our case the (unknown) set of sensor node positions, and the data is the distance estimates from the RSSI measurements.

Given a set $\boldsymbol{I} = \{\boldsymbol{N}_1, \boldsymbol{N}_2, \ldots, \boldsymbol{N}_K\}$ of K nodes and estimates (measurements) of the distance between them. If the distance between two nodes is within a common detection range r, the estimate is assumed to be a result from measurements (explicit detection). Otherwise, the estimate only tells that the distance between them is larger than r (i.e. missing data defaults to an implicit imprecise distance estimate larger than r). Our algorithm utilizes information inherent in missing or negative observations.

For $i = 1, \ldots, K$, let the real position vector $\boldsymbol{r}_i$ denote an initially proposed location for the node $\boldsymbol{N}_i$. These positions can be restricted to for example a rectangular (test) area covering the whole sensor network. The present simulations are for a rectangular test area of size $(100 \times 50\,\mathrm{m}^2)$. Some nodes acting as anchor points have known location, while the position vectors for the other nodes are random.

Anchor nodes with known positions give the possibility to estimate all node positions from internode distance data. The positions $\boldsymbol{r}_i, i = 1, \ldots, K$, constitute a proposal explaining the inter-node signal strength data. They also constitute ”genes” in the present setting. The algorithm generates a population $\mathcal{P}$ of $L = 1000$ such proposals for node positions. A fitness measure $\boldsymbol{m_f}$ quantifies how well each proposal (individual) in $\mathcal{P}$ fits to the measurement data. The fitness measure provides a linear ordering in the population defining for each individual its probability for producing offspring.

The algorithm generates an offspring by randomly selecting two individuals (parents) in the population $\mathcal{P}$. The position $\boldsymbol{r}_i$ of node $\boldsymbol{N}_i$ for the offspring is then a copy of the corresponding position for one of the parents with equal (50-50) probability. A random mutation with probability 0.02 takes place as a random displacement $\Delta\boldsymbol{r}$ relative to this position. A mutation may be small or large. In our example simulations we apply three types of mutations in alternating sequence during the generations of the evolutionary process. One type of mutations are large mutations $\Delta\boldsymbol{r}$ with a uniform distribution over set $[-100, 100] \times [-50, 50]\,\mathrm{m}^2$ (i.e. $\Delta\boldsymbol{r} \in [-100, 100] \times [-50, 50]$). The two other types of mutations are similarly for $\Delta\boldsymbol{r} \in [-10, 10] \times [-5, 5]$) and $\Delta\boldsymbol{r} \in [-1, 1] \times [-0.5, 0.5]$). Mutations only take place if they result in a new position inside the rectangular test area in which the sensor network is known to be.

The evolutionary process takes place in cycles where a number of $S = 100$ of the best fit individuals survive through *elitism* and the remaining $L - S = 900$ exits the population. For each cycle, 900 new individuals are created. The new population of $L = 1000$ proposals constitute the new generation.

Let $o_{i,j}$ denote the estimate of the distance between the nodes $\boldsymbol{N}_i$ and $\boldsymbol{N}_j$ $(i, j = 1, 2, \ldots, K)$. The statement $o_{i,j} > r$ is equivalent to no available data despite of good attempts to detect. Assume

$$s_{i,j} \overset{def}{=} |\boldsymbol{r}_i - \boldsymbol{r}_j| \tag{1}$$

is the distance derived from the model sensor network $\boldsymbol{I} \in \mathcal{P}$. A possible fitness measure $\boldsymbol{m}^p_{\boldsymbol{f}}(\boldsymbol{I})$ for an individual $\boldsymbol{I} \in \mathcal{P}$ can here be a power sum of the differences between observed distances and the distances according to $\boldsymbol{I}$:

$$\boldsymbol{m}^p_{\boldsymbol{f}} \overset{def}{=} \sum_{i,j=1}^{K} d^p_{i,j} \tag{2}$$

where

$$d^p_{i,j} = \begin{cases} |s_{i,j} - o_{i,j}|^p & \text{if } s_{i,j} \le r \text{ and } o_{i,j} \le r \ ; \\ 0 & \text{if } s_{i,j} > r \text{ and } o_{i,j} > r \ ; \\ |2r|^p & \text{otherwise} \end{cases} \tag{3}$$

The algorithm applies the fitness measure $\boldsymbol{m}^p_{\boldsymbol{f}}$ for $p = 1, 2$. Variation of the fitness measure $\boldsymbol{m_f}$ during the evolutionary process extends the search space making the evolution less likely to stagnate at local optima.

Note that the evolutionary method above extends the search space proposed by Zhang *et al.* [15] which restricts possible node positions to conform to observed neighborhoods.

Evolutionary computation may function in a setting with frequent occurrence of data outliers. The above approach is directly extendable so it can perform combined analysis of spatially related sensor data and sensor positioning. This can extend the application space to for example data transmission ranges shorter than otherwise applicable for sensor node positioning. Note that an evolutionary approach also has possible parallel implementations. This gives the opportunity to distribute computational load available in the sensor network.

IV. TEST AND EVALUATION

The most challenging scenario for a localization scheme is when the nodes are randomly deployed, such as during an airdrop. The example scenarios below are therefore for such situations. In randomly deployed networks, the network degree defines whether the nodes are uniquely localizable or not. The routing scheme is also sensitive to the degree of network connectivity. These aspects are studied in the following simulations.

Two randomly deployed scenarios illustrate the evolutionary localization algorithm. The first scenario is for ideal RSSI measurement conditions (zero measurement error). The second scenario is for more realistic RSSI measurement errors and inaccuracies found using present implementations of IEEE 802.15.4.

A. *Network impact*

As no simulator implementation of DYMO-low was public available, we implemented the Internet Draft in the NS-2.34 network simulator [23]. Then the proposed extensions to the protocol were added to enable measurement and distribution of signal strength.

In the simulations, the packet overhead involved in performing a complete routing and localization process was studied. The routing scheme was evaluated under the effect of network density and node population. The setup used the IEEE 802.15.4 MAC layer and 20 different random simulation topologies were run for each setup.

1) Localizable nodes: The fraction of localizable nodes was examined for different network densities. Given the area A, the number of nodes K and the radio range r, then the average number of possible neighbors d is defined by the average number of nodes within the area $((\pi r^2 K)/A)$. This measure does not account for area edge effects. A node at a corner of a rectangular area will only have an average of $d/4$ neighbors. A density d of 5 here represents a sparse network and a density of 20 a dense network. For each density simulated, the number of nodes K was varied between 50 and 200. The results are shown in Fig. 2.

In a sparse network ($d = 5$) only 70% of the nodes could be localized by the sink, meaning that 30% of the network was partitioned. When $d = 7$, more than 90% of the nodes could be localized. This increased to about 100% when $d = 20$. A small number of route replies (LRREPs) was lost due to collisions or congestion in the network. This caused the actual number of localizable nodes to be lower than the theoretic, as shown in the green bars. The protocol was however, able to reconstruct 60–70% of this lost information thanks to the redundant information in other LRREPs.

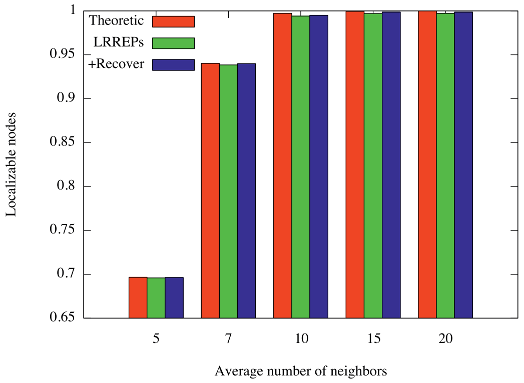


Fig. 2. Fraction of nodes localizable for different network densities. Red bars show theoretical fraction of nodes in the network reachable by the sink node. Green bars show fraction of nodes reachable using incoming LRREPs. Blue bars show fraction of nodes reachable by reconstructing missing LRREPs.

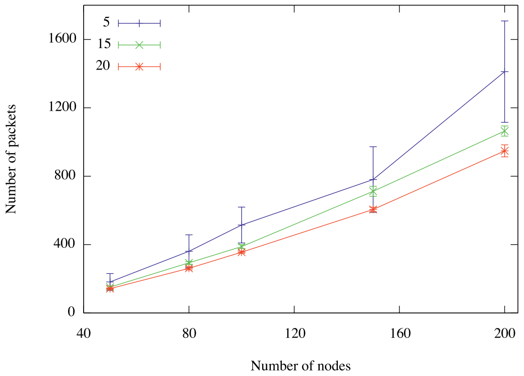


Fig. 3. Number of transmitted packets as a function of network size (number of nodes) for complete route discovery including RSSI-measurement. The average number of neighbors varies between 5–20. The 95% confidence interval included.

2) Network overhead: Fig. 3 shows the total number of packets transmitted to obtain RSSI measurements and route discovery in randomly deployed networks. The number of packets increased with increasing number of nodes. It also increased with lower density due to more hops between an arbitrary node and the sink.

Fig. 4 represents the same network topologies as for Fig. 3 while quantifying data transport in terms of number of bytes instead of number of packets. The total number of bytes

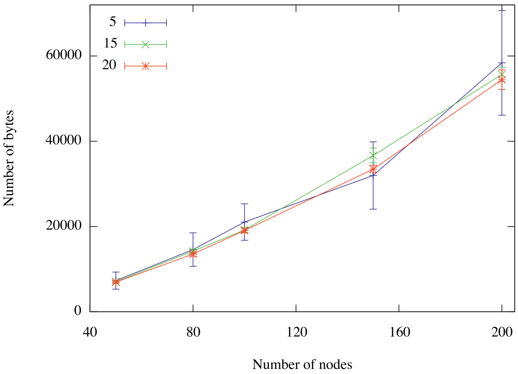


Fig. 4. Number of transmitted bytes as a function of network size (number of nodes) for complete route discovery including RSSI-measurement. The average number of neighbors varies between 5–20. The 95% confidence interval included.

transmitted was approximately constant for a given number of nodes regardless of the density. There is in this way a balance between a tendency for increased traffic due to decreased number of hops and an increase due to larger data packets caused by more one-hop neighbors.

The scheme seems to scale well and we state that the data requirement to run the scheme is within the limits of IEEE 802.15.4.

B. *Localization performance and accuracy*

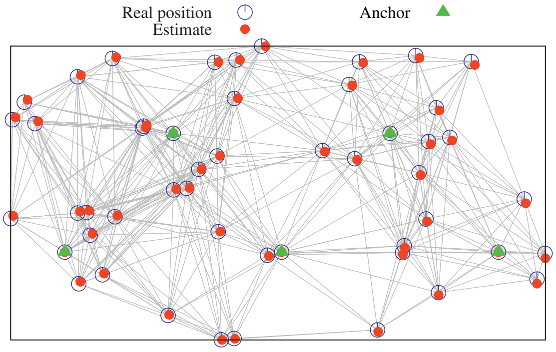


Fig. 5. Result from numerical experiment with 50 nodes including 5 anchor nodes and no measurement errors.

A separate and simple Ada program implemented the proposed evolutionary algorithm. The algorithm was evaluated under the effect of RSSI measurement quality.

As identified in Fig. 2, randomly deployed nodes require a high node density to avoid network partitioning. Therefore we have considered an initial setup consisting of 50 nodes within a $100 \times 50\,\mathrm{m}^2$ rectangular area and with transmission range $r = 30\,\mathrm{m}$. Fig. 5 is for a simulation with ideal measurements (no measurement errors). The average position error is in this case within 0.5 m. However, this error is an artificial effect by the model since candidate solutions (individuals) in the evolutionary process can be subject to fine tuning with arbitrary small mutations. Note that the scenario cannot be considered realistic unless the conditions are ideal or a more exact measure than RSSI is employed. Fig. 6 illustrates a typical generational development of the fitness $\boldsymbol{m}_{\boldsymbol{f}}^{p}(\boldsymbol{I})$ for the most fit individual $\boldsymbol{I}$ in the population in this scenario. The fitness measure $\boldsymbol{m}_{\boldsymbol{f}}^{p}$ did here drive the evolutionary process where the value of p changed between 1 and 2 for each 5 generation.

Fig. 7 shows results from a simulation with significant measurement errors possessing a uniform distribution around the real distance ± 10 percent. Solid black lines here illustrate data outliers which are 50 percent less than the real distance. Such measurement errors are here similar to real sensor nodes [17]. The final position estimates show small errors (average less than 1 m).

The performance of the evolutionary algorithm is sensitive to reduction of the transmission range r or the network degree (or node density) and the spatial distribution of the anchor nodes.

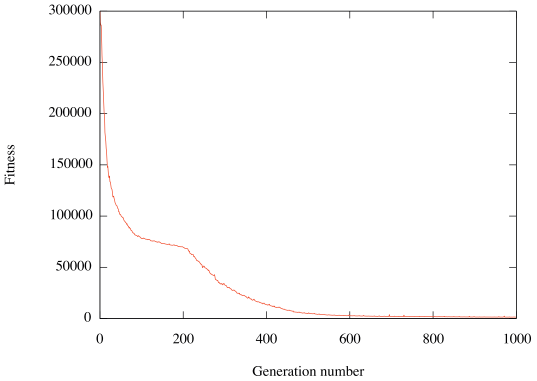


Fig. 6. A typical generational development of fitness by the evolutionary algorithm.

C. *Summary*

We have contributed to the discussion of applying evolutionary computation to estimate positions of nodes. Evolutionary computation seems to provide simple solutions to complex data fusion tasks. Our example simulations indicate that current hardware and standards may provide possible pioneering attempts in this direction. The provided simulation results also show that the data requirement to run the localization scheme is well within the limits of IEEE 802.15.4, meaning that centralized localization is feasible.

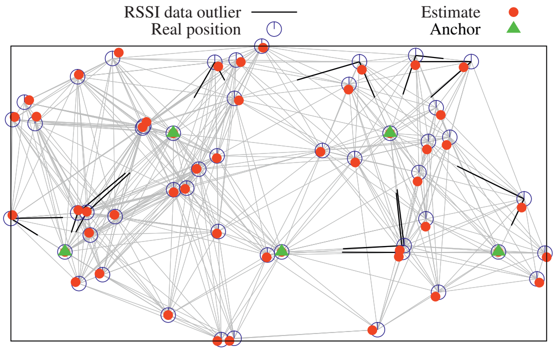


Fig. 7. Result from numerical experiment with 50 nodes including 5 anchor nodes. The error has in this case an uniform distribution around the distance ±10 percent. Solid black lines illustrate data outliers which are 50 percent less than the real distance.

V. CONCLUSION

We argue that both the ranging measurements, the measurement data gathering and the localization algorithm are essential in providing a complete localization system solution. In this paper a scheme including all those components is presented. The proposed localization scheme is based on centralized evolutionary computing and employs the route establishment phase of DYMO-low to fetch and distribute signal strength values.

We conclude by emphasizing the flexibility in the scheme presented in this paper. The proposed extension to the DYMO-low protocol can potentially be used to facilitate other centralized localization algorithms than the evolutionary computation algorithm proposed here. Likewise, the evolutionary algorithm can take advantage of information gathered using a link state routing protocol, such as OLSR [24]. Further, the evolutionary algorithm can benefit from more precise ranging methods such as acoustic ranging. This makes our contributions versatile and attractive to a wide range of WSN applications.

REFERENCES

[1] W. Banzhaf, P. Nordin, R. E. Keller, and F. D. Francone, *Genetic Programming – An Introduction; On the Automatic Evolution of Computer Programs and its Applications*. San Francisco, CA, USA: Morgan Kaufmann, Jan. 1998.

[2] K. Akkaya and M. Younis, “A survey on routing protocols for wireless sensor networks,” *Ad Hoc Networks*, vol. 3, no. 3, pp. 325 – 349, 2005. [Online]. Available: http://www.sciencedirect.com/science/article/B7576-4B3JKH2-3/2/0baf704778ae8cf6e3651eb9f329e4c0

[3] S. Chen, G. Fan, and J.-H. Cui, “Avoid ’void’ in geographic routing for data aggregation in sensor networks,” *Int. J. Ad Hoc Ubiquitous Comput.*, vol. 1, no. 4, pp. 169–178, 2006.

[4] D. Espes and Z. Mammeri, “Backbone-based location-aided routing algorithm to reduce control packets of AODV protocol,” in *Mobility ’06: Proceedings of the 3rd international conference on Mobile technology, applications systems*. New York, NY, USA: ACM, 2006, p. 15.

[5] S. Ratnasamy, B. Karp, L. Yin, F. Yu, D. Estrin, R. Govindan, and S. Shenker, “GHT: a geographic hash table for data-centric storage,” in *WSNA ’02: Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*. New York, NY, USA: ACM, 2002, pp. 78–87.

[6] Y. Shang, W. Ruml, Y. Zhang, and M. P. J. Fromherz, “Localization from mere connectivity,” in *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*. ACM Press New York, NY, USA, 2003, pp. 201–212.

[7] D. A. Tran and T. Nguyen, “Localization In Wireless Sensor Networks Based on Support Vector Machines,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 19, no. 7, pp. 981–994, 2008.

[8] L. Girod and D. Estrin, “Robust range estimation using acoustic and multimodal sensing,” *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, vol. 3, pp. 1312–1320 vol.3, August 2002. [Online]. Available: http://dx.doi.org/10.1109/IROS.2001.977164

[9] D. Niculescu and B. Nath, “Ad hoc positioning system (APS) using AOA,” *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, vol. 3, pp. 1734–1743 vol.3, April 2003.

[10] R. Farrell, R. Garcia, D. Lucarelli, A. Terzis, and I.-J. Wang, “Localization in Multi-Modal Sensor Networks,” *Intelligent Sensors, Sensor Networks and Information, 2007. ISSNIP 2007. 3rd International Conference on*, pp. 37–42, December 2007. [Online]. Available: http://dx.doi.org/10.1109/ISSNIP.2007.4496816

[11] M. L. Sichitiu and V. Ramadurai, “Localization of wireless sensor networks with a mobile beacon,” *Mobile Ad-hoc and Sensor Systems, 2004 IEEE International Conference on*, pp. 174–183, October 2004.

[12] S. Lee, E. Kim, C. Kim, and K. Kim, “Localization with a mobile beacon based on geometric constraints in wireless sensor networks,” *Wireless Communications, IEEE Transactions on*, vol. 8, no. 12, pp. 5801–5805, December 2009. [Online]. Available: http://dx.doi.org/10.1109/TWC.2009.12.090319

[13] A. Kannan, G. Mao, and B. Vucetic, “Simulated Annealing based Wireless Sensor Network Localization,” *Journal of Computers*, vol. 1, no. 2, 2006.

[14] V. Tam, K. Cheng, and K. Lui, “Using micro-genetic algorithms to improve localization in wireless sensor networks,” *J. of Comm., Academy Publisher*, vol. 1, 2006.

[15] Q. Zhang, J. Wang, C. Jin, J. Ye, C. Ma, and W. Zhang, “Genetic Algorithm Based Wireless Sensor Network Localization,” *Natural Computation, 2008. ICNC ’08. Fourth International Conference on*, vol. 1, pp. 608 –613, oct. 2008.

[16] K. Kim, G. Montenegro, S. Park, I. Chakeres, and C. Perkins, “Dynamic MANET On-demand for 6LoWPAN (DYMO-low) Routing,” Internet Engineering Task Force, Internet-Draft, Jun. 2007, work in progress. [Online]. Available: http://www.ietf.org/internet-drafts/draft-montenegro-6lowpan-dymo-low-routing-03.txt

[17] M. Holland, R. Aures, and W. Heinzelman, “Experimental investigation of radio performance in wireless sensor networks,” *Wireless Mesh Networks, 2006. WiMesh 2006. 2nd IEEE Workshop on*, pp. 140 –150, sept. 2006.

[18] K. Srinivasan and P. Levis, “RSSI is Under Appreciated,” in *In Proceedings of the Third Workshop on Embedded Networked Sensors (EmNets*, 2006.

[19] A. Awad, T. Frunzke, and F. Dressier, “Adaptive Distance Estimation and Localization in WSN using RSSI Measures,” *Digital System Design Architectures, Methods and Tools, 2007. DSD 2007. 10th Euromicro Conference on*, pp. 471 –478, aug. 2007.

[20] C. Perkins, E. Belding-Royer, and S. Das, “Ad hoc On-Demand Distance Vector (AODV) Routing,” RFC 3561 (Experimental), Jul. 2003. [Online]. Available: http://www.ietf.org/rfc/rfc3561.txt

[21] I. Chakeres and C. Perkins, “Dynamic MANET On-demand (DYMO) Routing,” Internet Engineering Task Force, Internet-Draft draft-ietf-manet-dymo-19, Sep. 2009, work in progress. [Online]. Available: http://www.ietf.org/internet-drafts/draft-ietf-manet-dymo-19.txt

[22] G. Mulligan, “The 6LoWPAN architecture,” in *EmNets ’07: Proceedings of the 4th workshop on Embedded networked sensors*. New York, NY, USA: ACM, 2007, pp. 78–82.

[23] Software, “The ns-2 network simulator,” Accessed 2010. [Online]. Available: http://www.nsnam.org

[24] T. Clausen and P. Jacquet, “Optimized Link State Routing Protocol (OLSR),” RFC 3626 (Experimental), Oct. 2003. [Online]. Available: http://www.ietf.org/rfc/rfc3626.txt

Paper B :

Increasing the Lifetime of Roadside Sensor Networks using Edge Betweenness Clustering

J. Flathagen, O. V. Drugan, P. E. Engelstad and Ø. Kure

In proceedings of IEEE International Conference on Communications, ICC 2011,
Kyoto, Japan, 5-9 June, 2011, pp. 1-6, ISBN: 978-1-61284-232-5.

Increasing the Lifetime of Roadside Sensor Networks using Edge–Betweenness Clustering

Joakim Flathagen*‡, Ovidiu Valentin Drugan§, Paal E. Engelstad§ and Øivind Kure‡
*Norwegian Defence Research Establishment (FFI), ‡Q2S NTNU, §University of Oslo
Email: joakim.flathagen@ffi.no, ovidiu@ifi.uio.no, {paalee, okure}@unik.no

***Abstract*—Wireless Sensor Networks are proven highly successful in many areas, including military and security monitoring. In this paper, we propose a method to use the edge–betweenness community detection algorithm to determine clusters and to facilitate in-network data aggregation for these applications. To minimize the cost of determining the clusters, the approach is based on exploiting the topology information from the ad hoc routing protocol. Three different schemes are proposed. (1) A distributed clustering scheme using the OLSR routing protocol. (2) A centralized scheme using OLSR. (3) A centralized scheme using an extension to the DYMO-low routing protocol. All schemes support sensor heterogeneity allowing that different data content can use different routing paths. The paper presents simulation results and an analysis of the cluster generation for each of the schemes. The results show that our method is a simple and effective method to improve scalability and lifetime of roadside sensor networks.**

***Index Terms*—Clustering, Data Aggregation, Wireless Sensor Networks**

I. INTRODUCTION

Wireless Sensor Networks (WSNs) are proven effective in the fields of perimeter security and military surveillance [1]. In these areas, great benefit can be achieved by using covert miniaturized sensors, as they are difficult to avoid by a possible intruder and less subject to vandalism or theft compared to traditional sensor systems. Further, the redundancy given by ad hoc network protocols improves reliability compared to previous systems. However, WSNs face two basic challenges; *energy efficiency*, due to the battery powered sensors, and *scalability*, due to a potential high number of devices needing to interoperate. The goal of this paper is to provide a method to solve these two issues by the means of in-network data aggregation.

Data aggregation is particularly interesting for roadside surveillance systems. In such systems, the sensor nodes collaborate in detecting events such as movement and particular behavior of objects along the road. Multiple nodes are here likely to sense the same event simultaneously. Conventional routing treats these sensor readings individually and ignores the redundant and highly correlated nature of the data. This leads to ineffective use of the scarce energy and limited channel resources. By employing data aggregation, designated aggregation nodes can wait for multiple reports, either from the same node (temporal redundancy), or from neighboring nodes (spatial redundancy), before reporting about the event to the sink. This strategy not only reduces the traffic considerably, but also reduces the probability of false alarms, as most sensors are likely to be inaccurate and have a small probability of falsely reporting events that are not actually present.

The contributions of this paper include: (1) A data aggregation scheme based on edge–betweenness community detection, (2) three different routing protocol schemes supporting both centralized and distributed clustering, (3) modification and improvement of the DYMO-low routing protocol, and (4), a quantification of the trade-off between cluster-aggregation and traditional routing, and a comparison of our schemes with the well-known K-means clustering. Although we mainly focus on roadside surveillance networks, our protocols, recommendations and results are also viable to other classes of sensor networks that are topologically similar to our scenario.

Before presenting our own scheme and results, it is worth reviewing some of the preceding work regarding data aggregation in WSN.

II. RELATED WORK

Different data-aggregation alternatives can be categorized based on the network architecture involved in the aggregation, which can be structured either as a chain, a tree, or by clusters.

Chain-based aggregation schemes create linear chains for data-aggregation. Each node in the chain only transmits to its closest neighbor, which fuses the data with its own measurements, and retransmits along the chain. In PEGASIS [2], the chains can be made either centrally or distributed. *Tree-based* data aggregation on the other hand, organizes the nodes in an aggregation tree rooted at the sink. Directed Diffusion [3] is one such example. If only a subset of the nodes in the network are sensing nodes, tree based techniques provide better performance than chain-based since the aggregation tree is better than the chain for mere packet routing. For both strategies, the aggregation delay perceived by a node is based on its position in the aggregation tree (or chain). The overall aggregation delay therefore increases drastically with the number of nodes in the network [4]. The challenge is to balance the trade-off between energy efficiency and the delay posed by the aggregation. Both tree-based and chain-based aggregations are best suited for scenarios where *all* nodes in the network produce relevant information *periodically*. For our event-initiated scenario, these proposals are inadequate since the long aggregation delay makes it difficult to uniquely distinguish separate events.

Cluster-based schemes organize the sensor nodes into virtual groups and perform aggregation only at designated cluster-

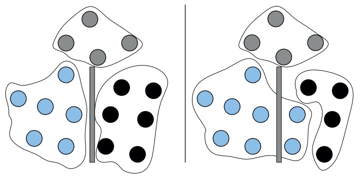


Fig. 1. Edge–betweenness clustering (left) takes the topology into account while K-means (right) use mere geographical positions for clustering.

heads (CHs). This approach drastically reduces the aggregation delay compared to the chain and tree architectures, at the cost of possibly longer routing paths. Notice that cluster schemes are not limited to aggregation only. LEACH [5] for example, uses clustering both as a tool to aid data aggregation and to coordinate access of the wireless channel within the cluster. LEACH only supports single-hop transmission between each cluster-head and the sink, making the approach invalid for our purpose. Lai et al. have recently extended LEACH by allowing multihop transmissions and by better balancing the energy consumption [6]. Gong et al. [7] takes a different approach and propose to use modified K-means clustering, and determines the clusters centrally assuming that the geographical positions of the nodes are known. We describe this method and compare it to ours in the subsequent sections.

While [5]–[7] use explicit control messages to initiate the clusters, our scheme has the ability to passively exploit the underlying routing protocol to gain topology knowledge. Another key difference is that the above methods require that all traffic must pass through the cluster-head, while our approach allows some traffic classes to take an optional (shortest-path) route towards the sink.

III. CLUSTERING

A. K-means

K-means is a classical and simple method for clustering that has been applied to several problem domains, including sensor networks, as demonstrated by Gong et al. [7].

When applied to sensor node clustering, the procedure is as follows: (1) the number of clusters k must be predetermined. (2) k points are placed in the geographical space represented by the nodes being clustered. These points represent the cluster centroids. (3) Each node is assigned to the cluster with the closest centroid (in terms of Euclidian distance). (4) The positions of the k centroids are recalculated as the mass center of each cluster. Then, (3-4) are repeated until the centroids no longer move. In [7], the nodes with the minimum distance to the cluster centroid and highest residual energy are elected as cluster-heads.

While this algorithm outperforms LEACH, its disadvantage is that the number of clusters must be predetermined (or estimated), and that the exact geographical position of the nodes must be known.

B. Edge–betweenness community detection

Edge-betweenness community detection is a method proposed by Newman and Girvan [8]. Community detection algorithms are known from physics literature, (i.e., a community is a region of the network with dense connections) and have been successfully used to capture interactions in ad hoc networks [9]. The algorithm tries to find the communities of the network with the maximum modularity value. The modularity measure is based on the formula $\mathcal{Q} = \sum_{i=0}^{m} \left(e_{ii} - a_i^2\right)$ where m is the number of detected communities, e_{ii} represents the fraction of links in the network that connect the nodes in community i, and a_i represents the fraction of links that connect two nodes in community i. The algorithms proposed by Newman and Girvan [8] all find good approximations for the maximum modularity. The algorithm (*EB*) searches for the division of the network with the greatest modularity value by removing links with high importance in the network (see Algorithm 1).

Algorithm 1 Edge-betweenness

```
EB $(\mathcal{G})$
  1) $\mathcal{G}' = \mathcal{G}$
  2) $for\ i = 1 \ldots |\mathcal{L}|$
       a) $\mathcal{G}' = \{\mathcal{G}' \setminus \{e_{b_i}\} \mid b_i = max\,(betweenness\,(\mathcal{G}'))\}$
          $\mathcal{P}_i = \{connected\,(\mathcal{G}')\}$
       b) $\mathcal{Q}_i = \mathcal{Q}\,(\mathcal{P}_i)$
  3) return $\{\mathcal{P}_l \mid l = max\,(\mathcal{Q}_i)\}$
```

The algorithm recursively computes the betweenness score of each link in $\mathcal{L}$ defined by the number of shortest paths going through a link. The link with the highest betweenness score is removed from the graph, and the modularity value is recomputed. The algorithm is applied until there are no more links left. The communities are determined by the partitioned network obtained in the step with the maximum modularity value.

As opposed to most existing clustering methods, EB-clustering does not put any a priori constraints on the cluster structures (e.g., cluster diameter, number of nodes in a cluster or number of clusters). While K-means requires that a localization scheme is present in the network, EB-clustering only relies on the network topology. Notice that K-means assumes that geographically adjacent nodes also are 1-hop neighbors. This is not always the case for sensor networks. As shown in Fig. 1, this assumption can lead to suboptimal clusters and excessive paths between cluster members and the cluster-head.

C. Fetching topology information

The prerequisite for EB-clustering is to have an updated view of the network topology. Such information can either be obtained *actively* by exchanging explicit control messages between the nodes, or *passively* by taking advantage of information available by consulting the underlying routing protocol. Our approach belongs to the latter category, and performs the topology fetching without the need for extra messages. Consequently, the overhead of enabling clustering in the network can be drastically reduced.

In our approach, the cluster construction is separated from the routing layer, and standard routing is therefore maintained. The approach taken by [5]–[7] on the other hand, forces all traffic to be routed via the cluster-heads, which is not always in the shortest path between an arbitrary node and the sink. This is a suboptimal solution for heterogeneous networks containing several sensor types. In surveillance systems for example, all sensor nodes can contain passive IR, sound and vibration sensors to detect and track a target, while a few nodes are equipped with a digital camera or active IR for target verification. Our approach supports such applications using policy-based routing. Alarms and measurements are considered easy to aggregate (homogeneous data) and can be routed directly to the designated cluster-head, which is responsible for data aggregation (to reduce data transmissions) or filtering (to reduce the false alarm rate). Meanwhile, data from special purpose sensors, such as imaging sensors, can not be aggregated and should therefore follow the shortest path to the sink.

In this paper, we study both centralized and distributed clustering methods and examine the use of two different routing protocols to obtain the topology information. The proposed schemes are:

1) OLSR distributed scheme.
2) OLSR centralized scheme.
3) DYMO-low centralized scheme.

In the next two sections we describe how to combine EB-clustering with these routing alternatives.

IV. OLSR Schemes

A. Introduction

Optimized Link State Routing (OLSR) [10] is proposed by the IETF aiming at Mobile Ad-hoc Networks (MANET). Although OLSR is seldom considered viable for sensor networks due to its proactive behavior and the possibly large routing table, we argue that some classes of WSNs may benefit from the use of OLSR. OLSR has gained considerable popularity because of its versatility and extensibility, and simple extensions can provide several attractive features, such as e.g., multicast, multiple interfaces and service discovery. If such features are needed in the WSN, using OLSR may simplify the design compared to adding these features on top of a less advanced protocol.

For our purpose, OLSR provides the attractive feature that each node keeps an updated view of the network topology. This feature can be used to determine clusters in the network in a distributed fashion, as described in the next section.

B. OLSR distributed scheme

The *OLSR distributed scheme* employs the OLSR routing protocol repositories on each of the nodes to gain information about the network topology. This information is then used to determine the network clusters locally using EB-clustering. The challenge with this approach is that it relies on consistent cluster calculation in the network. To ensure that all the nodes determine exactly the same clusters, each node needs to obtain accurate topology information. However, if default OLSR settings are used, only partial link-state can be obtained.

The partial link-state in OLSR is caused by the intention to limit the communication overhead by reducing the number of links advertised and the number of nodes that advertises them. Using default OLSR settings, only nodes chosen as MultiPoint Relays (MPRs) create topology control (TC) messages. A TC message only contains the advertised link set of a node limited to its MPR selector set. Hence, all neighbors will not be reported in the TC message, and for our purpose, this means that the entire topology (including all links) cannot be detected. Consequently, exact and consistent cluster determination cannot be ensured.

Mechanisms to extend the network topology knowledge in OLSR are previously studied in [11] and [12]. In [11], the authors investigate different options by tuning the MPR-Coverage settings and by increasing the amount of information in each TC message. One way to let an MPR report all links is to alter the `TC_REDUNDANCY` parameter from TC_0 to TC_2. By doing this, the advertised link set of the node include the full neighbor link set. However, as pointed out in [12], the nodes generating TC messages are not constrained to MPRs only when using this setting. The authors therefore suggest applying TC generation with full link set only to those nodes that are selected as an MPR by another node. This new proposed setting is named TC_4 (this term is also applied in our research). Notice that if a link exists between two non-MPR nodes, its existence is not reported in any TC messages. This can be resolved by changing the `MPR-coverage` setting, as proposed in [11]. By altering this parameter a node can increase the preferred number of MPRs in its MPR set increasing the probability that all links are reported.

To verify the performance of the distributed clustering scheme, we examine the consistency of the identified clusters while altering the `TC_REDUNDANCY` parameter.

C. OLSR centralized clustering scheme

The *centralized* clustering scheme solves the beforementioned cluster consistency issue. In this case, the clusters are determined by employing EB-clustering at the sink node only. The partial link-state of OLSR is not critical since each node will unambiguously belong to one single cluster. The drawback of this approach is that a separate protocol is needed to elect and inform cluster-heads and to tell each node which cluster it belongs to. However, this can be solved either by creating an OLSR extension, or using a simple application layer protocol.

It is worth noting that distributed protocol designs are traditionally preferred before centralized designs in networking systems due to the fault tolerance and lack of scalability of the latter approach. We argue that in most WSNs, the sink node is already a single point of failure. The fault tolerance is therefore not increased by centralizing the clustering algorithm [13]; in fact, this approach simplifies the protocol design and the implementation. Further, the scalability of the centralized algorithm is not a big concern compared to a distributed design, as the sink node can be equipped with several orders of magnitude more memory and CPU than the sensor nodes.

V. DYMO-LOW SCHEME

In the *DYMO-low centralized scheme*, we propose a few modifications to the DYMO-low routing protocol [14] to fetch topology information and to facilitate centralized clustering. DYMO-low is intended for use on IEEE 802.15.4 devices and is based on the principle of flooding route requests (RREQ) and unicasting route replies (RREP) as known from AODV and DYMO. However, DYMO-low is considerably simplified to better match the limitations of 802.15.4.

Our extension introduces two new messages to the protocol, Topology Route Request (TRREQ) and Topology Route Reply (TRREP). As with the centralized OLSR scheme, the sink node performs the EB-clustering calculation, and a separate protocol is employed to elect and inform the cluster-heads. The difference between this approach and the OLSR centralized approach is that DYMO-low is a reactive protocol and does not disseminate routing information regularly as is the case with OLSR. This can save considerable bandwidth if the proactive behavior of OLSR is not needed. Another benefit with this approach compared to OLSR is that our DYMO-low implementation can provide *full* topology information, whereas OLSR does not provide this without implementing the beforementioned extensions, with the penalty of increased OLSR overhead. The protocol operation consists of a request phase and a reply phase.

The sink first initiates the network by announcing its address via a TRREQ. This message can be seen as a proactive route request destined to *all* nodes in the network. The TRREQ is flooded similarly as a regular DYMO-low routing request (RREQ), and the nodes which receive the TRREQ, retransmits the packet only once. This means that all nodes will receive a copy of the TRREQ from each of its neighbors. When a node receives a TRREQ packet, it stores the address of its neighbor. As the TRREQ disseminates from the sink to the entire network, all nodes will eventually obtain a list including each of its one-hop neighbors with no more cost than a regular Route Request process.

Upon receiving a TRREQ packet, a node responds back to the sink using a TRREP. This transmission takes place after a random time delay to avoid network congestion and collisions. The response message extends the regular Route Reply defined in DYMO-low, with a list of the one-hop neighbors. The sink will eventually receive TRREPs from all the sensor nodes in the network. EB-clustering will then use this information to determine the clusters. Note that each link in the network is reported twice (once from each link end). The duplicated information can enable reconstruction of missing TRREP information.

It can be argued that this approach cannot be classified as passive clustering since we alter the routing protocol to fetch the topology. However, our extension in fact reduces the number of control messages compared to standard DYMO-low. After a complete request/reply phase, all nodes in the network have a valid route to the sink, making them ready to perform their sensing task immediately. If standard DYMO-low is used, route requests must be initiated from each node in the network to accomplish this. This leads to a tremendous overhead due to the flooded route requests. Our approach, on the contrary, limits this to just *one* sink-initiated route request and considerably reduces the number of messages flooded in the network. The request/reply and EB-clustering process can be initiated either automatically or by a network operator.

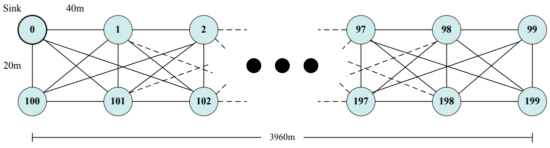


Fig. 2. The main scenario used in the simulation and analysis.

VI. RESULTS AND ANALYSIS

The motivation behind the simulations in this section is threefold. First, we analyze the accuracy of the topology knowledge in OLSR and how inaccuracies affect the cluster consistency for distributed clustering. Second, we compare the overhead posed by the different schemes. Finally, we study the energy savings by employing the clustering scheme considering different cluster-head election strategies, and different distances between a target (sensed by the WSN) and the sink. We compare EB-clustering with K-means clustering.

We implemented the DYMO-low Internet Draft in the NS-2.34 network simulator and added the proposed extensions to the protocol to enable neighbor detection and reporting. For the OLSR experiments, we used UM-OLSR [15], which we modified to provide extended topology knowledge. The clustering methods were implemented using iGraph. Unless otherwise mentioned, default OLSR settings were used. IEEE 802.11 DCF was used as the MAC protocol.

Two scenarios were created for the testing and analysis. The initial setup (scenario 1) consists of 200 nodes aligned along a virtual road, see Fig. 2. The inter-node distance is 40m horizontally and 20m vertically, covering a total area of 20x3960m. The transmission range is set to 100m. Scenario 2 is a small modification of Scenario 1 made by removing the nodes $4, 8, 12 \dots 96$. Scenario 2 in this way provides a layout with defined groups of nodes.

A. Topology knowledge and cluster consistency

First we consider only Scenario 1 and employ OLSR routing. Fig. 3 shows the average accuracy of the topology knowledge at each of the nodes. When using default OLSR, topology knowledge accuracy was only 35%. Using TC_2, the accuracy increased to 90%, while with TC_4, the accuracy was 75%. These results correspond to those presented in [12]. The reduced topology knowledge observed at the network ends

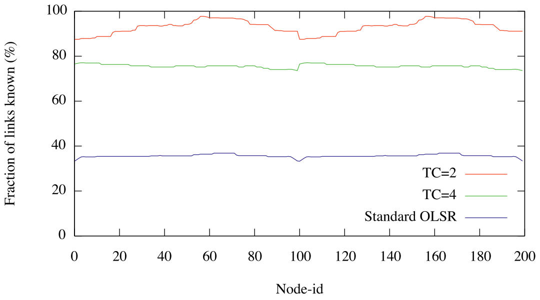


Fig. 3. Accuracy of topology knowledge in the network

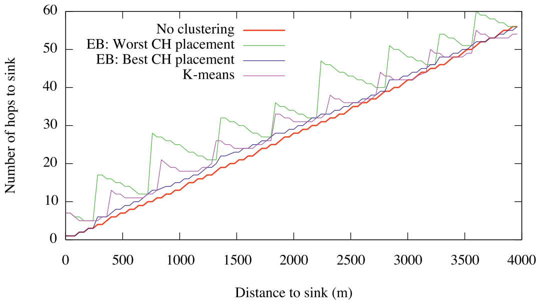


Fig. 4. Number of hops as a function of distance between a detected event and the sink.

TABLE I
PERFORMANCE OF THE SCHEMES. DIFFERENT OLSR DISTRIBUTED CLUSTERING COMPARED WITH DYMO-LOW CENTRALIZED CLUSTERING.

Protocol	Top-know	Overhead	Clust-cons1	Clust-cons2
OLSR TC_0	35%	39.5 KB/s	88.9%	91.1%
OLSR TC_2	90%	76.7 KB/s	96.3%	96.0%
OLSR TC_4	75%	58.1 KB/s	90.6%	93.4%
DYMO_low	100%	483KB/round	(100%)	(100%)

(i.e., the nodes 0,99,100,199) is caused by collisions (and loss of topology information) in the center of the network. As a comparison, when running the centralized DYMO-low scheme 100% accuracy is achieved in the same scenario.

We now examine how the topology inaccuracies affect the consistency of the clusters when EB-clustering is employed on each node. To compare the communities detected at the different nodes, we represent the node-to-cluster memberships in matrixes, and compare the matrixes created at each of the nodes. Due to different topology knowledge, a small percentage of the detected cluster memberships differ among the nodes. The values in table I show the percentage of the detected cluster membership information that is equal among all nodes. There is a tendency that local information is correct, and the membership inconsistencies are on distant clusters only. EB-clustering here works remarkably well even with limited OLSR topology knowledge, but increasing the topology knowledge further improves the clustering consistency. An input scenario with more clearly defined groups (Scenario 2) also leads to more consistent clusters. This is caused by the fact that EB-clustering creates communities based on counting the number of shortest paths going through each link (betweenness-score), and this scenario has more links with salient betweenness-score.

When clustering is employed *centrally*, the reduced topology accuracy is not crucial. Even with standard OLSR, our experiments show that the clusters fit the physical layout of the nodes well, although not as accurate as with increased OLSR topology knowledge or by using DYMO-low.

B. Overhead

Table I also show the overhead for the different routing alternatives. As the OLSR protocol exchanges control messages periodically, overhead is almost constant regardless of the lifetime of the clusters and the rotation of the cluster-heads. For DYMO-low, messages are only sent when the clusters are regenerated (referred to as one *round*[1]). For DYMO-low, all TRREQ and TRREP transmissions for one round are included in the measures. We also let CHs flood their existence to the network using TRREQs. The results show that the centralized DYMO-low protocol leads to the best clusters (full topology is obtained) and also the smallest overhead in scenarios with slow CH rotation ($> 12s$). We assume that the distributed clustering scheme may be efficient in mobile networks, which require frequent regeneration of the clusters and where network partitions prohibit centralized control.

C. Data aggregation

Next, we evaluate the clustering scheme considering different cluster-head election strategies, and different distances between a target (sensed by the WSN) and the sink. We extend Scenario 1 to include a vehicle that moves along the WSN, and apply DYMO-low centrally. Sensor nodes detecting the vehicle transmit this information to the cluster-head in its cluster. Most clustering schemes in the literature employ a round-robin scheme to alternate the role of the CH to balance the energy consumption. We find it interesting to examine the performance of EB-clustering with extreme CH placements. The optimal CH placement is found when the elected CH-node is the node in the cluster that minimizes the average number of hops between a node in the cluster and the sink, while the worst is found when this number is maximized.

Fig. 4 shows the number of hops necessary to transmit *one* sensor reading using standard routing, compared with EB-clustering with worst and best CH. This is compared with K-means using centroid CHs (k is predefined to match the cluster number proposed by EB-clustering). We observe that electing the optimal CH nodes hardly increases the path-length compared to standard routing, while the worst placements increases the path length considerably. Uniform rotation of

---

[1] If the topology has changed between two rounds, a new set of clusters is generated. The optimal round frequency depends on the expected data traffic and link stability and is not studied in this paper.

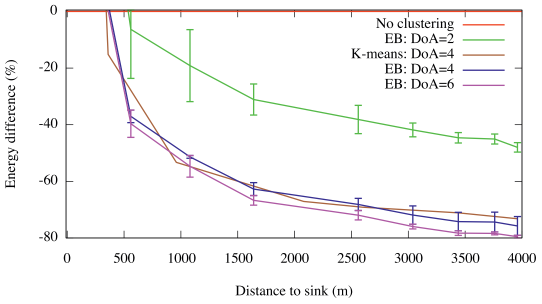


Fig. 5. Energy consumption as a function of distance between a detected event and the sink. Bars indicate worst and best cluster-head placements.

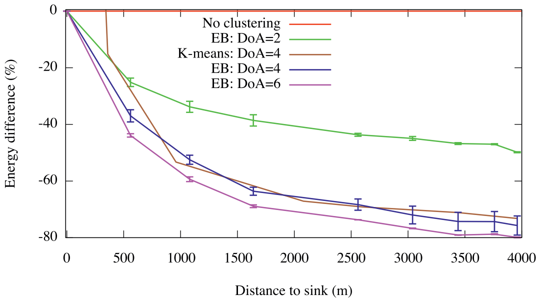


Fig. 6. Energy consumption as a function of distance between a detected event and the sink. Bars indicate min/max.

cluster-heads results in an average of 32% increase in the path length. K-means gives similar average path lengths as EB-clustering. In a real implementation we anticipate that K-means gives longer paths than EB-clustering, since the network topology not always reflect the geographical positions of the sensor nodes (cf. Fig. 1).

Now we focus on the same setup, but apply in-network data-aggregation at the CHs. We apply the term Degree of Aggregation (DoA) from [1], representing the number of messages that the CH receives and aggregates before transmitting data to the sink. DoA depends on the sensing range, the network density, and the signature of the tracked object. The effect of manipulating DoA is shown in Fig. 5. As seen in the figure, the benefit of employing clustering is limited when the detected object is close to the sink and the DoA is low. However, assuming that an event can occur (i.e., a vehicle or intruder is detected) at any position along the network, a DoA of 4 and uniform CH placement, 49% energy reduction can be expected. K-means produce comparable results.

Since an EB-clustering node (be it central or distributed) has full knowledge of *all* clusters in the network, the above result can be optimized. Instead of letting the aggregation role rotate among the cluster members only, we instead exploit nodes from the upstream neighbor cluster. This eliminates the problem of routing packets in the wrong direction. In Fig. 6 we apply rotation only among border-nodes. Here, a DoA of 4 gives an average reduction of 58%. Even a modest DoA of 2, gives an average energy reduction of 36% compared to standard routing.

VII. CONCLUSIONS

We have proposed a method to use the edge-betweenness community detection algorithm to determine the clusters and to facilitate in-network data aggregation in roadside sensor networks. The method omits the need for exact geographical positions as in K-means. We have presented both centralized and distributed designs, and results show that clusters can be generated in a consistent way, even with reduced OLSR topology knowledge. The best results are obtained using centralized clustering and our DYMO-low routing protocol scheme. The average energy reduction is 20–62% compared to standard routing, and outperforms K-means. Future works include synchronizing idle-listening within the clusters and implementing the protocols in a test bed.

REFERENCES

[1] T. He, S. Krishnamurthy, J. A. Stankovic, T. Abdelzaher, L. Luo, R. Stoleru, T. Yan, and L. Gu, “Energy-efficient surveillance system using wireless sensor networks,” in *In Mobisys*. ACM Press, 2004, pp. 270–283.

[2] S. Lindsey, C. Raghavendra, and K. Sivalingam, “Data gathering algorithms in sensor networks using energy metrics,” *Parallel and Distributed Systems, IEEE Transactions on*, vol. 13, no. 9, pp. 924 – 935, sep. 2002.

[3] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva, “Directed diffusion for wireless sensor networking,” *IEEE/ACM Trans. Netw.*, vol. 11, no. 1, pp. 2–16, 2003.

[4] I. Solis and K. Obraczka, “The impact of timing in data aggregation for sensor networks,” in *2004 IEEE ICC*. IEEE, 2004, pp. 3640–3645.

[5] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “Energy-Efficient Communication Protocol for Wireless Microsensor Networks,” in *HICSS ’00*, 2000, p. 8020.

[6] W. K. Lai, C.-S. Shieh, and Y.-T. Lee, “A cluster-based routing protocol for wireless sensor networks with adjustable cluster size,” in *ChinaCOM 2009*, Aug 2009, pp. 1 –5.

[7] Y. Gong, G. Chen, and L. Tan, “A Balanced Serial K-Means Based Clustering Protocol for Wireless Sensor Networks,” *WiCOM ’08. Proceeding*, pp. 1 –6, oct. 2008.

[8] M. E. J. Newman and M. Girvan, “Finding and evaluating community structure in networks,” *Physical Review E*, vol. 69, no. 2, pp. 026 113+, Feb 2004.

[9] O. V. Drugan, T. Plagemann, and E. Munthe-Kaas, “Detecting Communities in Sparse MANETs,” *IEEE/ACM Transactions on Networking*, Accepted for publication in 2011.

[10] T. Clausen and P. Jacquet, “Optimized Link State Routing Protocol (OLSR),” RFC 3626 (Experimental), Oct. 2003.

[11] T. Clausen, P. Jacquet, and L. Viennot, “Investigating the Impact of Partial Topology in Proactive MANET Routing Protocols,” in *WPMC*. IEEE, 2002.

[12] P. E. Villanueva-Pena, T. Kunz, and P. Dhakal, “Extending network knowledge: making OLSR a quality of service conducive protocol,” in *IWCMC ’06*. New York, NY, USA: ACM, 2006, pp. 103–108.

[13] B. Raman and K. Chebrolu, “Censor networks: a critique of ”sensor networks” from a systems perspective,” *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 3, pp. 75–78, 2008.

[14] K. Kim, G. Montenegro, S. Park, I. Chakeres, and C. Perkins, “Dynamic MANET On-demand for 6LoWPAN (DYMO-low) Routing,” Internet Engineering Task Force, Internet-Draft, Jun. 2007, Expired.

[15] University of Murcia, “UM-OLSR,” no. http://masimum.dif.um.es/, Accessed 2010.

Paper C :

Constrained-based Multiple Sink Placement for Wireless Sensor Networks

J. Flathagen, Ø. Kure, and P. E. Engelstad

Presented at the International Workshop on Wireless Sensor, Actuator and Robot Networks (WiSARN-FALL), 2011
Paper published in proceedings of IEEE International Conference on Mobile Adhoc and Sensor Systems, MASS 2011, pp. 783-788, ISBN: 978-1-4577-1345-3.

Constrained-based Multiple Sink Placement for Wireless Sensor Networks

Joakim Flathagen*‡¶, Øivind Kure‡¶ and Paal E. Engelstad§¶
*Norwegian Defence Research Establishment, ‡Q2S NTNU, ¶UNIK, §University of Oslo

***Abstract*—A wireless Sensor Network (WSN) consists of many low-cost and energy-constrained sensing nodes. One method that offers a great potential for improving both the lifetime and the durability of WSNs is to deploy multiple data sinks instead of the standard approach relying on just one sink. In this paper we focus on multiple sink deployment problems and discuss different methods to estimate the optimal placement of a given number of sinks. Most previous works study unconstrained sink node placement, assuming that the sinks can be placed anywhere. In practice, there may be areas which are occupied by obstacles, or are beyond wireless range, and therefore not viable for sink placement. Our method inherently considers deployment constraints by inspecting the routing topology and therefore avoids connection black holes when proposing optimal sink locations. We have used an anycasting tree-routing scheme, and have performed extensive simulations in a wide range of realistic scenarios. The results show that a constraint-based deployment algorithm is paramount to get the full potential of multiple sink WSNs.**

I. INTRODUCTION

A wireless Sensor Network (WSN) consists of many small and low-cost sensing nodes. The two basic challenges in WSNs are *energy efficiency*, due to the battery-powered sensors, and *scalability*, due to a potential high number of devices needing to interoperate. In this paper we aim to prolong the network lifetime and improve the scalability by deploying *multiple sinks*. In addition to reducing the average path length between a sensing node and the corresponding data sink, the use of multiple sinks also provides energy fairness by load balancing. The method also gives redundancy if one of the sink-nodes should fail due to energy shortage, or if it is vandalized or stolen.

While finding the optimal number of sinks is by nature an off-line problem mainly constrained by deployment cost, determining the optimal *placement* of the sink nodes is a more difficult challenge. The initial deployment of the WSN can be done either in a structured or planned manner by a network designer, or in a semi-random way (e.g., an air-drop). In any case, the optimal placement of the sinks cannot be known a priori, and there is a need for heuristics to facilitate relocation of existing sinks or to position new sinks in the network. Our algorithms aim to find the optimal sink locations for a given network topology and coverage. The algorithms are employed at a separate computer and sink relocation is then performed either manually or by mobile sinks or robots.

Most works study *unconstrained* sink node placement, assuming that the sinks can be placed anywhere. In practice, there may be areas which are occupied by obstacles, or are out of wireless range, and therefore not viable for sink placement. Hence, in this paper, we study *constrained* sink node placement, meaning that the sinks can only be placed in a subset of the WSN scene. Via extensive simulations we show that the constrained approach leads to improved goodput and lifetime compared to the unconstrained approach.

Before presenting our own schemes, it is worth reviewing some of the preceding work regarding multiple sink deployment in WSNs.

II. RELATED WORK

Oyman et al. [1] propose to find the optimal placement of multiple sinks using the well-known K-means clustering. The cluster centroids for the k clusters are chosen as the optimal placement for the sinks. The approach is used to minimize the number of sinks for a predefined minimum operation period, and to find the minimum number of sinks while maximizing the network lifetime. The K-means method is further described and used as a baseline later in the paper. The approach presented in [1] requires global location information to find the optimal sink placements. Vincze et al. [2] aim to relax this requirement by approximating the location of nodes with unknown positions. The system is, however, based on a geographical routing protocol, which requires a functional location system in the WSN.

The approaches taken in [1], [2] study *unconstrained* sink placement. This limits their practical use. As discussed in the introduction, such schemes are based on the assumption that there are no physical boundaries limiting the proposed placement of the sinks. The presumed optimal sink locations found by the algorithms are therefore not necessarily viable in practice due to physical constraints in the scene. A proposed location may actually end up being outside radio-range of the surrounding sensor nodes. The work by Dai et al. [3] aims to solve this problem by only proposing sink positions at locations that are known to be in communication range with at least a subset of the network.

To accomplish this, they restrict sink placement only to locations already occupied by sensing nodes. However, since their network model is restricted to Manhattan grid layouts and assumes uniform link lengths and link weights, the approach is not useful for semi-structured deployments. In this sense, the works [4] and [5] are therefore considered more flexible. Although both works study relay node placement, they can be adapted to the sink node placement problem. Deployment constraints are used to limit relay node placements at some pre-specified candidate locations only, meaning that the proposed locations are not restricted to known sensor node locations as in [3]. Their methods are more flexible and practical in a real setting, but require that the deployment algorithm a priori knows the deployment constraints. This requirement cannot always be fulfilled.

The deployment strategies we present in this paper (SPP and RMP) distinguishes from the beforementioned proposals since we allow *any* network topology. Also, sink deployment constraints are not an input parameter to the algorithms but are instead learned by inspecting the link information.

III. SINK PLACEMENT ALGORITHMS

To effectively determine the optimal placement for multiple sinks, network information must be gathered globally or estimated. We distinguish the different schemes in two categories: (i) those that require knowledge about the geographical positions of all sensor nodes (geo-aware); and (ii), those that rely on the network topology (topology-aware). In the following, we present four different sink deployment strategies, two in each category. The first method is similar to the one previously proposed by Oyman et al. [1]. It also shares resemblance with the method proposed by Vincze et al. [2]. The tree final methods are considered novel to our paper.

A. K-means placement (KSP)

K-means is a classic and simple method for clustering that has been applied to several problem domains. When applied to sensor sink node placement, the cluster memberships proposed by the algorithm is ignored. K-means is simply used to find the cluster centroids given a set N of n sensor nodes and their geographical positions $P = \{p_1, p_2, \ldots, p_n\}$. In this way, K-means can find the optimal set of sink locations $S^* = \{s_1, s_2, \ldots, s_k\}$ given a predefined number of sinks k. The method works as follows:

1) The preferred number of sinks k is predetermined.
2) k points $s_1, \ldots, s_k$ are placed in the geographical space bounded by the nodes being clustered, P. These points represent the cluster centroids, which will eventually constitute the sink locations.
3) Each sensor node is assigned to the cluster with the closest (Euclidean) centroid s.
4) The k centroids are repositioned to the mass center of each cluster.
5) Repeat steps 3-4 until the centroids no longer move.

By iteratively minimizing the within-cluster sum of squares, the final cluster centroids are found and chosen as the optimal placement for the sinks:

$S^* = arg\ min_S \sum_{i=1}^{k} \sum_{N_j \in S_i} ||p_j - s_i||^2$

The prerequisite to run K-means sink placement algorithm (KSP) is exact knowledge of each sensor node location. The location information can be obtained either by GPS positioning or by special localization schemes [6], [7]. In any case, the location information must be gathered from the sensor nodes to a central entity running KSP. This can be done using a mobile robot node or by temporarily installing one or more static sinks at random locations in the network.

B. K-medoid placement (KDP)

K-medoid clustering is closely related to K-means and is an excellent candidate algorithm for sink node localization. Instead of using cluster centroids, K-medoid builds on the concept of *medoids*. A medoid is defined as the most central object in a cluster. For our purpose, this is an attractive feature, since the algorithm can find the position of any k nodes in N that are most central instead of proposing *new* sink locations. The method therefore provide constrained placement, and our hypothesis is therefore that K-medoid is a better candidate for sink placement than K-means. Our K-medoid sink placement is based on Partitioning Around Medoids clustering (PAM), originally proposed by Kaufman and Rousseeuw [8]. The method works as follows:

1) Randomly select k of the n nodes to represent the initial medoids. The medoid positions will later represent the sink locations.
2) Each node is associated with the closest (Euclidean) medoid.
3) For each medoid m and non-medoid n, the pair (m,n) is swapped and the configuration cost is computed.
4) The configuration with the lowest cost is selected and stored in M.
5) Repeat steps 2-4 until there is no change in the medoid set.

The optimal sink locations are given by the positions of the medoid nodes in M^*, found by:

$M^* = arg\ min_M \sum_{i=1}^{n} min_{j=1}^{k} ||p_i - m_j||$

The above algorithm shares the same prerequisites as mentioned above for KSP, since all individual node locations must be known a priori.

C. Shortest path placement (SPP)

All multiple sink deployment strategies that require location information suffer from the following shortcomings:

1) The geographical positions of the sensor nodes must be known. To obtain the individual node positions, a localization and collection scheme must be present in the network.
2) Since the methods are based on Euclidean distance, the algorithms inherently assume that all sensor nodes share the same transmission range and that geographically adjacent nodes also are 1-hop neighbors. This is not always true in obstructed environments.

To overcome both these limitations, our Shortest Path Placement algorithm (SPP) can instead of requiring the geographical positions, take advantage of the network topology information to determine the optimal sink locations. By letting the sink placement algorithm take advantage of the topology information directly, instead of using the estimated node positions (which are imprecise and often derived from the topology anyway [6], [7]), the overall system design is radically simplified.

Our SPP algorithm builds on KDP and differs mainly in the distance measure employed. We model the network as an undirected graph G represented as a tuple $G(V, E)$ where V is the set of vertices representing the sensor nodes and E is the set of edges. Each edge represent a bidirectional communication channel between a pair of nodes i and j. We then construct an adjacency matrix A, where $a_{ij} = 1$ if there is an edge from vertex i to vertex j. If $i = j$, $a_{ij} = 0$. If there is no edge between i and j, $a_{ij} = \infty$. The all pairs shortest path matrix D is then computed from A using Dijkstras algorithm [9]. The shortest path distance between i and j is defined as d_{ij}. This measure now constitute the distance measure which replaces the Euclidian distance measure used in the KDP algorithm introduced above such that:

$$M^* = arg\ min_M \sum_{i=1}^{n} min_{j=1}^{k} d_{ij}$$

The algorithm finds k nodes (sinks) in the network that minimizes the average number of hops in respect to the remaining nodes in the network. The prerequisite to run SPP is that all links in the network are known a priori. As for the before-mentioned algorithms, such information can be gathered using a mobile node or by temporarily installing one or more sinks in the network. Notice that the collection of link information is inherently performed in many routing protocols, and this requirement is therefore easier to fulfill than obtaining the exact node positions.

D. Routing Metric placement (RMP)

Wireless sensor networks are error prone in nature and it is evident that poor link quality causes problems for packet delivery and routing. Hence, there are numerous works focusing on increasing the reliability by using better routing metrics, e.g., ETX, ETT or LQI. We provide an extension of the SPP algorithm that uses a *metric* for each edge before performing the shortest path calculation. The employed metric should preferably be the same metric as the one used by the routing protocol. The sink placement will then be optimized according to the chosen routing metric instead of being optimized to a separate (and often irrelevant) measure such as the Euclidean distance between the nodes.

As a proof-of-concept we use the link quality estimate (LQI) from 802.15.4 MAC layer to provide simple constraint based routing. The idea is implemented such that if the initial link quality estimate is below a certain threshold value (i.e., due to environmental constraints or path loss), we consider the link as *weak*. If the estimate is above this value, the link is considered *good*. By using this kind of routing constraint, the sink placement algorithm can be used to select the k sink node locations that maximize the overall link quality.

We extend the adjacency matrix A explained for SPP such that link constraints can be included in the calculations. This is implemented in the following manner:

$$a_{ij} = \begin{cases} 1 & \text{if link } i,j \text{ exists;} \\ 1+c & \text{if link } i,j \text{ is weak;} \\ 0 & \text{if } i=j\text{;} \\ \infty & \text{otherwise} \end{cases} \tag{1}$$

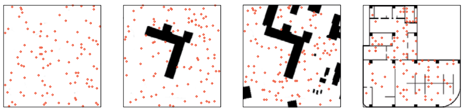
The constant c is used to take account for links which are considered weak. In our experiments, $c = 0.5$. The all pairs shortest path matrix D is computed from A, and inherently includes the link quality constraints. The shortest path distance between i and j is defined as d_{ij} and is used to find the sink locations as shown for SMP. RMP in this way finds the k nodes in the network that maximizes the average link quality. Placing the k sink nodes at these locations will presumably lead to fewer MAC retransmissions, fewer collisions and extended network lifetime.

IV. ANYCAST ROUTING IN MULTIPLE SINK NETWORKS

In multiple sink WSNs, the sensor nodes usually transmits data to *one* arbitrary sink and do not particularly care which sink is used. In such an anycasting paradigm, the routing protocol is responsible for transmitting datagrams to at least one of the sinks that accept datagrams with a certain anycast address.

For the purpose of the studies in this paper, we have developed a tree based routing protocol. The protocol establishes an anycast collection tree routed at the sinks. All nodes transmit beacons indicating their distance to the sink, whereas sink nodes report a distance of 0. The protocol uses the link quality indicator (LQI) from the physical layer in addition to the hop distance in the routing decision. The LQI value of a link is measured upon beacon reception. If the LQI value is below a certain threshold value, the link is considered *weak*. The route cost then becomes a combination of the number of hops N_H and the number of weak links N_W. A route a is said to be better than route b if $N_W(a) < N_W(b)$ *or* $N_W(a) = N_W(b)$ *and* $N_H(a) \leq N_H(b)$. Thus, a data packet will follow the path that minimizes both the number of hops and the number of weak links between a node and a sink.

V. ONE-SINK PLACEMENT



(a) S1 (b) S2 (c) S3 (d) S4

Figure 1. The four scenarios used in the simulations

To obtain valuable understanding of the differences between our proposed deployment algorithms, we first study networks containing just one sink ($k = 1$). As a point of comparison for sink placement we use a simple center placement strategy. The strategy merely places the sink at the center of the area. Should the center position be blocked by an obstruction (i.e., wall or building), the sink is located at the nearest non-obstructed position. In this way, the model is supposed to mimic deployment as if performed by a physical network operator or a robot.

To ensure that our results are not biased by our selection of a particular network layout, we consider four different network scenarios as shown in Fig. 1. The first scenario represents an open area with no obstructions. The second scenario represents the same area but with a large obstruction (building). More buildings are added in the third scenario. The fourth scenario is an indoor office area. In all scenarios, we define that signals communicated through walls and buildings observe a different radio propagation condition than signal communication line-of-sight through open air. We use the ShadowingVis propagation model in ns-2.34 to model this behavior in the simulated areas.

For all scenarios, each sensor node transmits a 50-byte sensor reading packet each 100s addressed to the sink anycast address. The readings are transmitted during the entire lifetime of the network. We define the network lifetime as the point in time when the first sensor node runs out of energy. The simulation parameters, including the transmission and reception energy usage, are given in Table I. For simplicity we assume that the energy consumption during idle periods is negligible. All parameters are kept equal for the different deployment strategies, meaning that the only variable affecting the simulation results is the actual choice of sink deployment strategy. Initially, we place two sinks at two random locations. These sinks are used to collect neighbor information and link quality estimates, which are subsequently used in the calculations. For KSP and KDP, we assume that the geographical positions of the nodes are exact and known a priori.

Table I
SIMULATION PARAMETERS

Simulator	NS-2.34
Propagation model pathlossExp_ std_db_ dist0_	ShadowingVis 1.5/4.0 (Open/Obstructed) 2.0/1.9 1.0/1.0
Number of nodes	100
Number of random topologies	10
Area	125m x 125m (S1-S3) 32m x 32m (S4)
MAC protocol Frequency CSTresh_ RXThresh_ RXpower TXpower Initial Energy	IEEE 802.15.4 2.4 GHz 1.20174e-07 1.20174e-07 35.28mW 31.32mW 1.0 Joule
Traffic parameters	CBR 50 bytes
Data rate	1pkt/100s/node

A. Results and analysis

Figure 2 show the lifetime for all scenarios and for all sink deployment algorithms. We observe that for scenario 1, the difference in lifetime is minimal between the five methods. This is expected considered that $S1$ represent a non-obstructed area, and with a reasonably high network density. For the scenarios $2-4$, we observe that the topology aware algorithms give remarkable lifetime improvements compared to both the geo-aware algorithms and the naïve center placement strategy. By concurrently studying Figure 2 and Figure 3, we observe that system lifetime relates to the average number of transmissions required to successfully transmit a packet from a source to the sink. This gives an insight of the quality of the links selected. Retransmissions due to packet loss cause more energy to be used on transmitting and receiving messages, which in turn reduces the system lifetime.

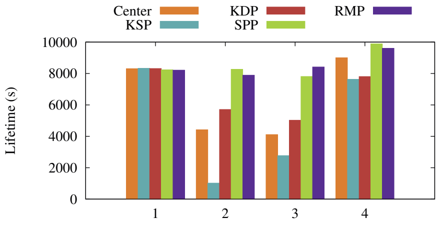


Figure 2. Network lifetime

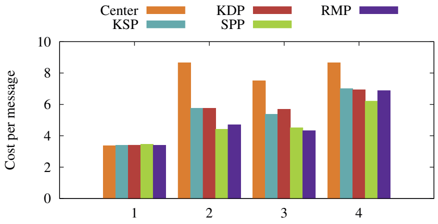


Figure 3. Average cost per sensor message

Figure 4 show the total number of sensor messages received at the sink (goodput) during the lifetime of the network. The figure in this way show the effective work performed by the sensor network during its system lifetime. KSP shows reduced performance for some topologies. This phenomenon is caused by the fact that the KSP strategy can propose sink locations in connection holes (no neighbors), or on top of obstructions. We did not reposition the sink to a better location in these cases. Center placement is therefore somewhat better, since obstructions are avoided in this model. However, there is still no guarantee that connection holes are avoided, and Center therefore has a lower average performance than the best deployment strategies.

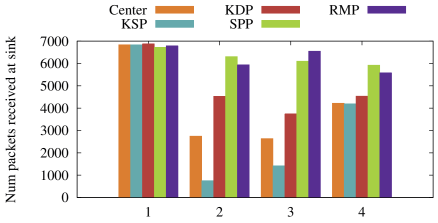


Figure 4. Total number of sensor messages received at the sink

Figure 5 show the percentage of nodes communicating with the sink during the system lifetime. This result gives a picture of how well the sink placement matches the network topology. Since all sensor nodes are randomly deployed within the open area, a small percentage of isolated nodes are expected regardless of the sink deployment procedure. However, the figure shows that an intelligent sink deployment procedure can minimize the number of isolated nodes. Again, we observe that the topology-aware strategies performs better that the other strategies.

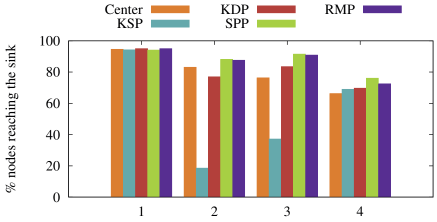


Figure 5. Percentage of nodes able to communicate with the sink

B. Summary

The following conclusions can be drawn from the above results:

- The network environment plays a huge part of the picture when comparing the performance of the schemes. When using a simple scenario ($S1$), all schemes give comparable results. However, in more complex environments which includes obstructions, SPP and RMP gives the longest lifetime, the highest number of packets received, and the lowest number of isolated nodes.
- RMP is the best choice when the network is sparse and there is a high number of low quality links in the network (i.e., many obstructions, as in $S3$). In a dense network ($S4$) and in a network with fewer obstructions ($S2$), SPP is the best choice. We anticipate that RMP may perform better under all network conditions if a more advanced network metric is used.
- We observe that even the simplest mechanism performs well under unconstrained and ideal conditions such as $S1$, while it performs poorly in obstructed environments. This result leads to the conclusion that previous sink deployment mechanisms only validated in simple simulation scenarios may be of little use in real world implementations.

VI. MULTIPLE SINK PLACEMENT

We now study the multi-sink problem and analyze the influence of increasing the number of sinks on the lifetime and total number of packets received. For the multi-sink case, we assume that the system does not particularly care which sink each sensor node uses as long as the lifetime is elongated and that the network load is balanced. We also assume that the sinks are either connected through a fixed network, or are manually collected by a network operator or robot after a certain period of time.

As the Center algorithm performed poorly for $k = 1$ and is difficult to apply for $k > 1$, we only consider the strategies KSP, KDP, SPP, and RMP. Also, we focus on scenario 3 only, since this scenario gave the results with the widest diversity for the different strategies in the one-sink case. We now investigate whether the difference between the strategies is consistent also when k increases. We apply the same simulation methods as described in Section V.

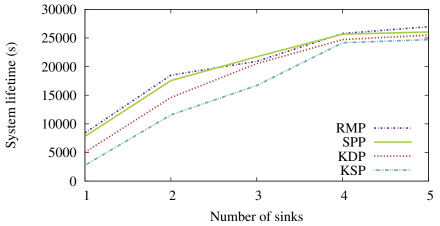


Figure 6. Lifetime of the sensor network

Figure 6 show the network lifetime related to the number of sinks for the different deployment strategies. We observe that the network lifetime first increases almost proportionally to the number of sinks, which is expected since the average path length decreases. It is also interestingly to see that the lifetime difference between the strategies observed for the one-sink case is sustained also when the number of sinks increases. This proves that it is extremely important to find the optimal sink placement even in the multi-sink case. It is, however, obvious that when a very high number of sinks is available (in this case $k \gg 5$), the choice of deployment strategy eventually becomes irrelevant. As in the one-sink case, we observe that the topology aware algorithms give remarkable lifetime improvements compared with the geo-aware algorithms. RMP increases the lifetime with 60% for $k = 2$, and 25% for $k = 3$ compared to KSP. In fact, *two sinks* deployed with SPP or RMP gives significantly longer lifetime than *tree sinks* deployed with KSP.

To get the full picture of how important it is to place the sinks wisely, Figure 6 must be seen in relation with Figure 7. Figure 7 shows the number of successfully received sensor readings at the sinks (goodput) during the system lifetime. We observe that with the topology-aware methods, SPP and RMP, the number of messages received during the system lifetime is significantly increased compared to the geo-aware methods, KSP and KDP.

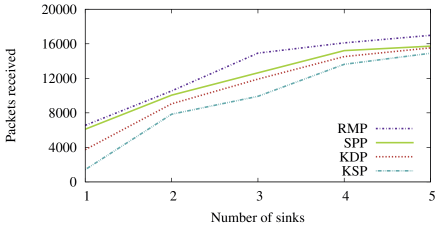


Figure 7. Total number of received packets

VII. CONCLUSIONS

In this paper, we have shown that deploying multiple sinks in WSNs offers a tremendous potential for improving both the lifetime and goodput. Most related work in the literature only considers *unconstrained* sink deployment mechanisms. Extensive simulation results show that such methods are insufficient since even the simplest deployment mechanisms performs well under unconstrained and ideal conditions, while they perform poorly in constrained environments. The results show that a constraint-based deployment algorithm is paramount to get the full potential of multiple sink WSNs.

REFERENCES

[1] E. I. Oyman and C. Ersoy, "Multiple sink network design problem in large scale wireless sensor networks," in *In: Proc. of 1 st Int. Conf. on Communications (ICC 2004)*, 2004.

[2] Z. Vincze, R. Vida, and A. Vidacs, "Deploying multiple sinks in multi-hop wireless sensor networks," in *In: Pervasive Services, IEEE International Conference on*, July 2007, pp. 55–63.

[3] S. Dai, C. Tang, S. Qiao, K. Xu, H. Li, and J. Zhu, "Optimal Multiple Sink Nodes Deployment in Wireless Sensor Networks Based on Gene Expression Programming," in *ICCSN 2010*, February 2010, pp. 355–359.

[4] S. Misra, S. D. Hong, G. Xue, and J. Tang, "Constrained relay node placement in wireless sensor networks to meet connectivity and survivability requirements," in *In: IEEE INFOCOM 2008 Proceedings*, April 2008, pp. 281–285.

[5] D. Yang, S. Misra, X. Fang, G. Xue, and J. Zhang, "Two-Tiered Constrained Relay Node Placement in Wireless Sensor Networks: Efficient Approximations," in *In: IEEE Secon 2010 Proceedings*, June 2010, pp. 1–9.

[6] Y. Shang, W. Ruml, Y. Zhang, and M. P. J. Fromherz, "Localization from mere connectivity," in *MOBIHOC*, 2003, pp. 201–212.

[7] J. Flathagen and R. Korsnes, "Localization In Wireless Sensor Networks Based on Ad hoc Routing and Evolutionary Computation," *MILCOM*, pp. 1173–1178, 2010.

[8] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley-Interscience, March 1990.

[9] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Num Math*, vol. 1, no. 1, pp. 269–271–271, Dec. 1959.

Paper D :

Integrating Wireless Sensor Networks in the NATO Network Enabled Capability using Web Services

J. Flathagen and F. T. Johnsen

In proceedings of the IEEE Military Communications Conference, MILCOM, Baltimore, MD, USA, November 7–10, 2011, pp. 828–833, ISBN: 978-1-4673-0080-3.

Integrating Wireless Sensor Networks in the NATO Network Enabled Capability using Web Services

Joakim Flathagen*‡¶ and Frank T. Johnsen*
*Norwegian Defence Research Establishment (FFI), ‡Q2S NTNU, ¶UNIK
Email: joakim.flathagen@ffi.no, frank-trethan.johnsen@ffi.no

***Abstract*—Wireless Sensor Networks (WSNs) are expected to provide greatly enhanced situational awareness for warfighters in the battlefield. Sensors widespread in the battlefield are however, of very limited value unless the sensors are reliable during the entire operation and the information produced is accessed in a timely manner. In this paper we focus on these issues by enabling WSNs as a capability in the NATO Network Enabled Capability (NNEC) using Web services. We demonstrate that Web services is an enabling technology for information-sharing, facilitating presentation of sensed data and alarms to a battlefield management system. In addition, we show the feasibility of using a Web services approach as a query processing tool enabling multi-sensor fusion and data aggregation in the WSN domain. The networking protocols can in this way inherently adjust data-aggregation and -processing criteria according to the requirements posed by external subscriber systems. In this way, energy efficiency, which is paramount in WSNs, is optimized without sacrificing the flexibility of Web services. Our proposed methods are tested using practical experiments with TelosB sensing nodes.**

***Index Terms*—Wireless Sensor Networks, Web services, Collection Tree Protocol**

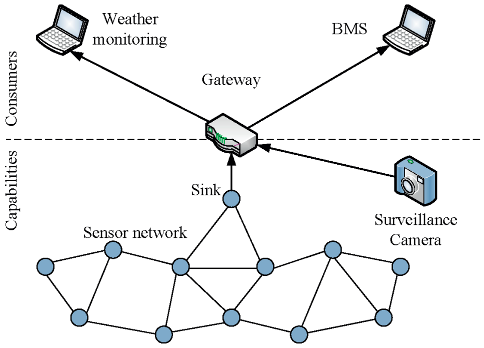


Fig. 1. Sensor network enabled as a service providing capabilities to different consumers. The gateway may invoke additional services to provide a composite service.

I. INTRODUCTION

Recent advances in integrated circuit design, micro electromechanical sensors and wireless network technology have enabled the development of low cost wireless sensors that can be deployed in large quantities. Wireless Sensor Networks (WSNs) can sense and gather information about the environment automatically and unattended. In the tactical domain, great benefit can be achieved by using covert miniaturized sensors, as they are difficult to avoid by a possible intruder and less subject to vandalism or theft compared to traditional sensor systems. Further, the network protocol redundancy and the vast number of sensing nodes improve reliability and minimize the false alarm probability compared to previous sensor systems.

Sensors widespread in the battlefield are, however, of very limited value unless the information is accessed and shared in a timely manner [1]. One of the main goals of the NATO Network Enabled Capability (NNEC) is to address this issue by facilitating seamless linking of sensors, decision makers and weapon systems. The NNEC feasibility study has identified *Web services* as the key enabling technology for NNEC [2]. Web services technology is based on a number of standards, which help ensure that different implementations from different vendors are interoperable. In this paper we explore enabling wireless sensor networks as a capability in NNEC using Web services. Since WSNs have scarce resources in terms of available bandwidth, battery, and computational power, it does not make sense to attempt to service-enable each and every sensing node. Instead, we use a wrapping approach, thus allowing existing mechanisms to be used within the WSN, while nodes external to the WSN may configure and receive information from the network using Web services. External consumer systems are for example Battlefield Management Systems (BMS) or Weather Monitoring Stations, see Fig. 1.

We do not, however, consider Web services only as an information-sharing and interoperability entity. In our architecture, we also suggest the use of a Web services gateway as a query processing system publishing relevant sensing and alarm-criteria to the WSN domain. The networking protocols can in this way inherently adjust data aggregation and processing criteria according to the requirements posed by the external subscriber systems. In this way, energy efficiency, which is paramount in WSNs, is optimized without sacrificing the flexibility of Web services.

The paper presents our Web services based WSN architecture and a real case-study to demonstrate our ideas applied to a tactical scenario. Before presenting our own setup and results in detail, it is worth reviewing some of the previous and related research.

II. RELATED WORK AND BACKGROUND

Directed diffusion [3] was one of the first initiatives to create a combined routing and query system for WSNs. In DD, the queries are formatted as *interest* messages which are disseminated to all sensing nodes. Gradients from each sensing node back to the base station are set up during the interest dissemination. Since the interest messages are not reliably transmitted throughout the network, the base station must periodically retransmit the interest message. Directed diffusion supports in-network data processing and aggregation, and the interest message formation allows publish-and-subscribe to occur at a very fine-grained level. However, the protocol is based on a query-driven on demand data model, and is not efficient for event-initiated alarm scenarios, such as e.g., tactical surveillance. The interest message formation in Directed Diffusion is using a proprietary format and is therefore not appropriate when used in a multi-consumer WSN such as the one in Fig. 1. Query processing systems such as TinyDB [4] aim to provide a flexible and simple query API by enabling queries written in a SQL-like language inspired from Data base systems. Hence, queries can be formulated remotely by multiple consumers using different physical entities. As opposed to DB systems, the queries here operate on real-time streams of data passing through memory rather than performing queries to a disk. TinyDB queries are input to the base station node, which sends an optimized version of the query to the sensor network. In the network, the sensing nodes that have data satisfying the query predicates, formulate an answer. These answers are returned to the base station (or sink). Data can be transformed, combined, and summarized according to the query.

If WSN-interaction is necessary in a multi-consumer setting, *Web services* provide higher flexibility and increased interoperability compared to extending querying protocols to each consumer. Notice that there are many definitions of ”Web services”. The core idea is the same (i.e., using XML-formatted data for information exchange), but some of the finer details may vary. For example, the REST approach ignores most of the Web services standards and specifications, meaning that REST is too restrictive if one wants to implement a pervasive SOA for military networks. We need the flexibility of a broader spectrum of the Web services specifications for NNEC. Thus, when we discuss Web services in this paper we use the definition by the W3C [5]: ”A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP-messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards.”

A Web service based WSN can be realized either by service-enabling each and every sensing node or by providing a Web service gateway that hides the inner WSN protocols. The work by Delicato et al. [6] was an early architecture work belonging to the first category proposing to integrate full SOAP support in the WSN sensing nodes. The full SOAP will however, often lead to tremendous overhead due to the verbose XML format. Although compression can reduce the overhead of XML significantly, and binary coding such as Efficient XML can enable XML to be used at the tactical edge [7], our previous research [8] has shown that the overhead associated with compression libraries make them unsuitable for use on severely limited devices. Thus, in contrast to other WSN implementations, such as [9], we do not attempt to employ XML compression in our WSN in this paper.

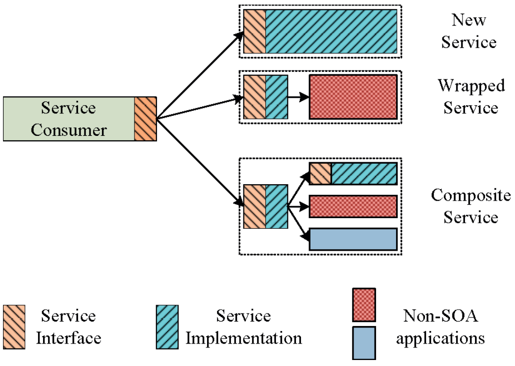


Fig. 2. Creating services (adapted from [12])

An alternative method to reduce the overhead is to convert the XML messages to a more optimized format at a gateway before relaying them to the WSN devices. The authors of [10] for example, propose WSN-SOA to reduce XML formats to a size applicable for 802.15.4 devices, while Bressan et al. [11] rely on the Constrained RESTful Environments (CoRE) based on REST. We argue that there is no point in extending Web services to every sensing node. In contrast, the WSN should (from the Web services perspective) be seen as one single sensing unit, providing filtered and aggregated sensed data to one or more consumers. Therefore, a gateway should be responsible for interacting with the WSN nodes on the back-end side, and the consumers on the front-end side. This approach lets the WSN designers focus on energy-efficient protocols inside the WSN, thus limiting the need for implementing computationally intensive standards to the gateway which provides an interface to the outside world.

III. SOA FOR WIRELESS SENSOR NETWORKS

Web services technology is based on a number of standards, which help ensure that different implementations from different vendors are interoperable. In this paper we explore enabling WSNs as a capability in NNEC using Web services. There are several ways of realizing a capability as a service. For example, a service may be created from scratch, it may function as a front-end to a legacy system, or it may be a combination of existing services, as illustrated in Figure 2.

Since WSNs have scarce resources in terms of available bandwidth, battery and computational power, it does not make sense to attempt to service-enable each and every sensing node. Instead, we use the wrapping approach, thus allowing existing mechanisms to be used within the WSN, while nodes external to the WSN may configure and receive information from the network using Web services. Even if the SOAP messages themselves do not have to be transmitted to every sensing node, it is crucial that available query information inside the XML payload is utilized to optimize the overall system performance.

The first contribution of our proposed architecture is therefore to provide a Web services wrapper that enables external consumers to interoperate with the sensor network using XML and Web services. The interaction operates in both directions. The second contribution of the architecture is query dissemination and collection formation that is adaptive and based on the requests posed by the Web service consumers. The architecture is shown in Fig. 3 and is described subsequently.

A. *Our gateway: A Web service wrapper*

The gateway contains the Web services wrapper and provides an interface (a front-end) to the WSN using established Web services standards. A WSDL file defines the interface, data types and message flow, whereas SOAP is employed for message transmission. This part of the wrapper is accessible to other systems using COTS Web services technology. The Web service interface allows external clients to configure queries for the WSN, and register a service endpoint (EP) for pushed information. In other words, our wrapper supports the publish/subscribe pattern, in that clients register a query (step 1, subscription providing recipient EP) and results of this query (be it periodic reports or spontaneous alarms) are sent (i.e., published directly to the consumers in steps 6 and 7) to the registered service endpoint. A client connecting to the gateway is typically a BMS, requesting alarm reports when a subset of the sensing nodes detects an intruder which is trespassing the area monitored. Such an example query is shown in Listing 1.

Listing 1. XML Query requesting alarm reports when at least four IR detectors are trigged

```
<GetIntruder>
    <MinPIRDetections>4</MinPIRDetections>
    <LightMaxThreshold>1lux</LightMaxThreshold>
    <Duration>30d</Duration>
    <IncidentReport>http://10.0.0.2/</IncidentReport>
</GetIntruder>
```

When the WSN reports to the gateway (step 3) about a detected target, the gateway sends a request to a separate Web service enabled camera (step 4) to take a picture covering the area monitored. The target information (from step 3) and the picture provided (by step 5) are combined to a report sent to the BMS endpoint (e.g., step 6 and/or 7). COTS Web services technology is used to implement step 1 as well as steps 4 through 7, limiting proprietary solutions only to the functionality implemented in the back-end system, i.e., steps 2 and 3. Thus, our prototype follows the guidelines of the NNEC FS, using Web services technology to loosely couple services and clients.

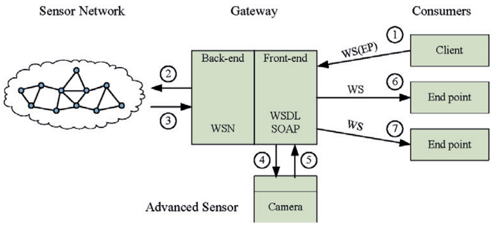


Fig. 3. Architecture

Another Web services client could be a weather monitoring station, requesting periodic temperature or humidity reports. A typical temperature report query, requesting individual temperature readings from each sensing node each 30 minute, is formatted in XML as shown in Listing 2.

Listing 2. XML Query requesting temperature reports from all sensing nodes

```
<GetTemperature>
    <CollectionStyle>Individual</CollectionStyle>
    <Duration>30d</Duration>
    <Interval>30m</Interval>
    <IncidentReport>http://10.0.0.1/</IncidentReport>
</GetTemperature>
```

In addition to supporting third party consumer applications, the architecture can also provide special case Web services for example to provide network developers with real-time information about the network at any given time, either during the initial deployment, create mid-life status reports, or to assist redeployment of energy exhausted nodes. These reports can be forwarded to a dedicated monitoring endpoint.

At the back-end, the gateway communicates with the WSN using two different traffic patterns: Dissemination (step 2) and Collection (step 3).

B. *Dissemination of queries*

The Web services wrapper shown in Fig. 3 interfaces with a back end, where the incoming XML configuration requests are transformed to a much more resource efficient, proprietary format used in our WSN. The format uses a compact and simple representation of sensor queries. Keys and attributes are represented as small integer values instead of text strings. Typical keys here are sensor type identificators and attributes are threshold values and timer values. A typical XML formatted Web services query of 200-300 bytes is translated to a small 10-15 byte message.

To disseminate the compact query through the WSN a dissemination protocol is required. Since messages can be lost due to e.g., collisions, channel noise or even buffer overflow, the dissemination protocol needs to be reliable. In addition, message synchronization could be necessary after a node reboot, e.g., if an application failure causes the watchdog timer to elapse. This means that simple flooding of the queries is not sufficient. In our implementation we have used Drip dissemination [13] to account for the above circumstances. The constrained power budgets in WSNs often lead to slow converging dissemination protocols. Drip copes with this issue by building a reliable transport layer on top of the Trickle algorithm.

C. Collecting sensed data

In most sensor networks, the majority of the network traffic is destined to the sink. For such networks, a collection tree traffic pattern is preferred in rather than other ad hoc network protocols. Instead of implementing our own collection tree, we use the Collection Tree Protocol (CTP) [14] which is the defacto collection protocol in TinyOS and is used successfully in many real WSN deployments. CTP consists of two parts; (i) data path validation, to quickly discover and fix routing inconsistencies by taking advantage of the data traffic; and (ii), adaptive beaconing using the Trickle algorithm, which optimizes the standard trade-off between low routing beaconing traffic overhead and low route repair latency. The anycast pattern employed by CTP also enables the possibility to extend our setup to a multiple sink architecture for increased reliability and reduced overall power consumption.

D. Aggregation

In our architecture, we focus on balancing the trade-off between the limited resources of the WSN and the required system performance necessary to fulfill the Web services query predicates. A query may for example ask for detailed reports requiring that every sensed value should be collected from the WSN and presented in a combined form in a Web services report. Alternatively, the query could indicate that a small report of filtered or aggregated measurements is preferred. Data aggregation in the Web services architecture can be employed either at the Web services gateway, or inside the WSN. From an energy-efficiency point of view, the latter alternative is preferred. To accomplish this, we have implemented a flexible data aggregation scheme running on the WSN nodes. Although the standard CTP does not include aggregation, the forwarding engine in CTP allows a routing extension to intercept the packets relayed by an intermediate node. Different aggregate functions can therefore alter the data upon interception as the sensed data traverses the collection tree.

Most data queries requests for periodically transmitted reports (e.g., each minute, each hour or each day). However, as the period timers are not fully synchronized among the nodes, there is an unknown time gap t between the first and the last node producing data in each period. Each node in the aggregation tree will therefore observe a gap $g \leq t$ between the arrival times of the sensing messages it receives from its child nodes. This time gap represents a challenge in WSN designs. If data freshness is paramount, each node should send its own measurements immediately when its period timer elapses, and retransmit all upstream messages immediately upon reception (i.e., no data aggregation). On the other hand, if the optimization objective is energy efficiency, each node should wait for a time $\geq t$ to account for all messages delivered from its child nodes before aggregation and transmission. The optimum balance between data freshness and energy efficiency can be found by optimizing the aggregation timeout of each node. One solution is to take advantage of the node position in the routing tree, as shown by Solis and Obraczka [15]. Our data aggregation algorithm on the contrary, minimizes the aggregation delay on each node without any routing protocol information. Rather, the node can learn g (the expected time difference between the messages received from its child nodes) by observing the inter arrival time of the packets received. The child node that triggers the end time of the period g is used as a synchronizer node trigging sensing, aggregation and transmission of the final data packet. Each node chooses the child node that constitutes the start of the maximum interarrival time in one periodic cycle as its synchronizer node (see Algorithm 1).

Algorithm 1 Data aggregation

```
Intercept $(Message\ m, Node\ n, f(m) = NO|MIN|MAX|AVG)$
    $if\,(f(m) == NO)\,send\,(m)\,, exit$
    $\mathcal{M} = aggregate\,(\mathcal{M}, m, f(m))$
    $\mathcal{T}\,(l) = t_{last} - t_{now}$
    $s = \arg\max_{i} \mathcal{T}\,(i)$
    $if\,(n == s)$
        $\mathcal{M} = aggregate\,(\mathcal{M}, M_{this}, f(m))$
        $send\,(\mathcal{M})$
    $l = n$
    $t_{last} = t_{now}$

On_Synchronizer_timeout $(s)$
    $\mathcal{M} = aggregate\,(\mathcal{M}, M_{this}, f(m))$
    $send\,(\mathcal{M})$
    $\mathcal{T} = 0$
    $s = 0$
```

If the synchronizer node times out (e.g., the CTP routing tree has changed), the node immediately transmit the aggregate of its temporarily stored data and sensed data and chooses a new synchronizer node on the next period. If no synchronizer is found, the node is a leaf node, and transmits sensed data immediately after its period timer has elapsed. Our aggregation scheme supports the following aggregation functions: *Average*, *Minimum*, *Maximum* and *No aggregation*, and adapts according to the queries transmitted from the gateway back end. *No aggregation* means that all measurements are delivered to the Web services gateway. Here, the measurements are combined to a joined report before reporting to the EP. The join-process could also include aggregation, but in-network aggregation is preferred.

IV. EVALUATION

A. Experimental setup

The experiment was set up as shown in Fig. 3. The WSN consisted of 20 wireless tmote sensing nodes [16] running TinyOS 2.1.1. The nodes were equipped with the following sensors: sound, light, temperature, humidity, ultrasound, and passive IR (PIR) (see Fig. 4). The gateway with the Web

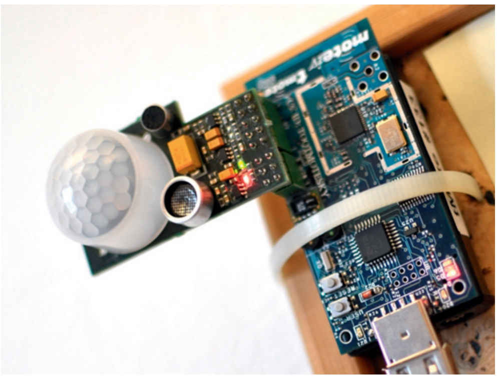


Fig. 4. The test network contains 20 tmote sensing nodes with IR-detectors

services wrapper ran on a iEi industrial computer with Linux, while the camera Web service was installed on a separate standard computer with a camera attached. The Web services consumers consisted of our trial client software on two standard computers.

We focused on two scenarios. First, a target detection scenario. In this scenario, the PIR sensors were used to detect possible targets trespassing the monitored area. The separate imaging sensor was used to take a picture of the target to provide target verification. The minimum number of PIR detectors detecting the target before classifying the event as an alarm, was configurable by Web services query created by the consumer system. Such an example query is shown in Listing 1. In the second scenario, an external system requested weather reports that should be presented periodically. An example of this query is shown in Listing 2. Besides performing functional testing of the architecture, we tested the effectiveness of the data format and data aggregation to obtain deeper insight of the system.

B. SOAP-based query vs. reduced query

We quantify the effectiveness of our reduced data format (RF) by comparing it with equivalent SOAP-based Web services. To reduce the unnecessary overhead, we removed the standard SOAP headers before dissemination with Drip. The query used for the experiment is shown in Listing 2. Our reduced information format message (12 bytes) was disseminated using the same method. Because of the very limited available memory on the tmote sensing node, we did not implement an XML parser but focused merely on the dissemination procedure in our experiment.

We performed 20 disseminations for each message format for networks with sizes 5,10 and 20 nodes respectively. The average node degrees in the networks were between 3 and 5. The 95% confidence intervals are given in the figures. Fig. 5 shows the time elapsed until all nodes had successfully received the query. Although Drip guarantees data delivery in a connected network, the delivery time can be severe, and increases with the size of the message disseminated. Overall, the RF format reduces the dissemination time to about a fifth of the time observed when disseminating XML.

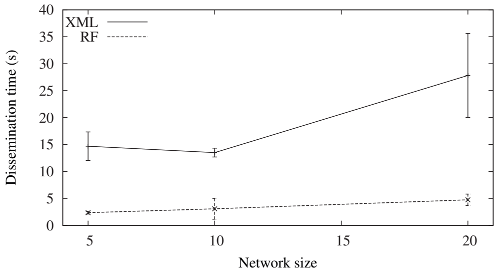


Fig. 5. The time required to fully synchronize the network

Fig. 6 shows the total energy spent on the dissemination process. The energy spending is calculated by observing the CC2420 radio load on each node, and accounting for the current draw of the tmote in RX/TX/Idle states from [16]. The XML encoded message results in more than eight times the power consumption compared to using the reduced format messages. These results illustrate that XML queries can indeed be transmitted to every node. However, in order to ensure reliable dissemination, the huge message size, which is difficult to avoid with XML, increases the energy consumption and prolongs the dissemination delay compared to using a more optimized format. It is also worth noting that XML gives no particular advantage compared to the reduced format in our homogeneous sensor network. A highly heterogeneous network may, on the other hand, benefit from of XML.

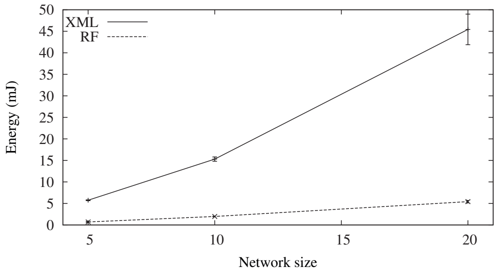


Fig. 6. The total energy spent in the dissemination process

C. In-network aggregation vs. gateway-aggregation

The Web services query predicates determine the proper aggregate function of the network system. In-network data aggregation is more complex to implement than relying on data aggregation only at the gateway. With this in mind, it is interesting to examine the performance of these two radically different strategies. With the first strategy, individual sensor readings were requested each 20s from all nodes. In this case,

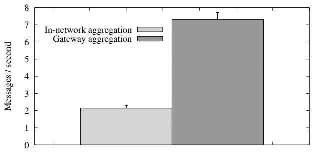


Fig. 7. The effect of data aggregation with 20 sensing nodes.

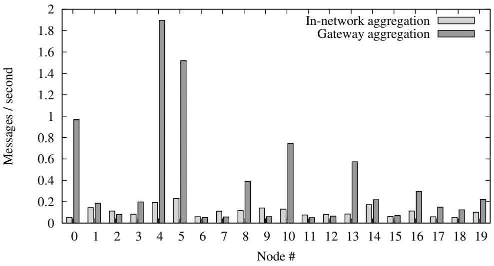


Fig. 8. The message flow distribution in the network with and without data aggregation

the aggregation took place at the gateway and there is no aggregation in the WSN. With the next strategy, the sensor nodes employed the aggregation strategy presented in section III-D. We used a 20-node tmote test-bed and performed 10 one-hour runs for both strategies.

Fig. 7 illustrates the total network load (in messages processed per second) both for the case with in-network aggregation and for gateway aggregation. The 95% confidence intervals are given in the figure. We observe that in-network aggregation significantly reduces the message load in the network. In Fig. 8, we examine the load on each sensing node separately. The figure shows that when in-network aggregation is enabled, message load is also better distributed.

From the literature, we know that the effect of in-network data aggregation increases with the size of the network. However, our results show that even a small network such as our 20-node network, can benefit greatly by employing in-network data aggregation. CTP focuses on establishing stable (low-ETX) routes rather than short routes. Hence, the number of hops involved in an arbitrary message transmission may be high, and the effect of in-network data-aggregation incrreases accordingly.

V. CONCLUSION

The results from our test-bed implementation shows that our Web services based architecture is feasible in a real setting. We were able to show that the WSN can take advantage of the attribute information in Web services queries provided by NNEC consumers, and that we could optimize the message flow by employing appropriate in-network data aggregation. It should be noted, however, that even if the Web services middleware we used has been identified as a key enabler for NNEC, there is a need for further standardization within NATO. Here, we have shown that it is feasible to use the technology in an NNEC setting, but for actual use in a coalition the interface to the WSN gateway (i.e., the WSDL) must be standardized as well. Finally, we were able to show that the Web services gateway can effectively combine the WSN service with an advanced Web service (camera) to provide a composite service to e.g., a BMS.

REFERENCES

[1] J. L. Paul, “Smart sensor Web: tactical battlefield visualization using sensor fusion,” *Aerospace and Electronic Systems Magazine, IEEE*, vol. 21, no. 1, pp. 13–20, January 2006. [Online]. Available: http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1581121

[2] P. Bartolomasi, T. Buckman, A. Campbell, J. Grainger, J. Mahaffey, R. Marchand, O. Kruidhof, C. Shawcross, and K. Veum, “NATO network enabled capability feasibility study,” Version 2.0, October 2005.

[3] C. Intanagonwiwat, R. Govindan, and D. Estrin, “Directed diffusion: a scalable and robust communication paradigm for sensor networks,” in *Proceedings of the 6th annual international conference on Mobile computing and networking*, ser. MobiCom ’00. New York, NY, USA: ACM, 2000, pp. 56–67.

[4] S. R. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, “TinyDB: an acquisitional query processing system for sensor networks,” *ACM Trans. Database Syst.*, vol. 30, pp. 122–173, March 2005.

[5] H. Haas and A. Brown, “Web services glossary W3C working group note 11 february 2004,” in *W3C*, 2004. [Online]. Available: http://www.w3.org/TR/ws-gloss/

[6] F. C. Delicato, P. F. Pires, L. Pirmez, and L. F. R. d. C. Carmo, “A flexible web service based architecture for wireless sensor networks,” in *Proceedings of the 23rd International Conference on Distributed Computing Systems*, ser. ICDCSW ’03. Washington, DC, USA: IEEE Computer Society, 2003, pp. 730–.

[7] J. Schneider, “Efficient XML: Taking Net-Centric Operations to the Edge,” in *13th International Command and Control Research and Technology Symposium (ICCRTS)*, June 2008.

[8] D. Hadzic *et al.*, “Web services i nettverk med begrenset datarate (in Norwegian),” in *FFI report 2006/03886*, 2006.

[9] N. Glombitza, D. Pfisterer, and S. Fischer, “Integrating wireless sensor networks into web service-based business processes,” in *Proceedings of the 4th International Workshop on Middleware Tools, Services and Run-Time Support for Sensor Networks*, ser. MidSens ’09. New York, NY, USA: ACM, 2009, pp. 25–30.

[10] J. Leguay, M. Lopez-Ramos, K. Jean-Marie, and V. Conan, “An efficient service oriented architecture for heterogeneous and dynamic wireless sensor networks,” in *Local Computer Networks, 2008. LCN 2008. 33rd IEEE Conference on*, October 2008, pp. 740–747.

[11] N. Bressan, L. Bazzaco, N. Bui, P. Casari, L. Vangelista, and M. Zorzi, “The Deployment of a Smart Monitoring System Using Wireless Sensor and Actuator Networks,” October 2010, pp. 49–54.

[12] Y. Natis, “Gartner research ”service-oriented architecture under the magnifying glass”,” Application Integration & Web Service, Summit 2005, April 18-20, 2005.

[13] G. Tolle and D. Culler, “Design of an application-cooperative management system for wireless sensor networks,” in *Wireless Sensor Networks, 2005. Proceeedings of the Second European Workshop on*, Feb. 2005, pp. 121–132. [Online]. Available: http://dx.doi.org/10.1109/EWSN.2005.1462004

[14] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis, “Collection tree protocol,” in *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, ser. SenSys ’09. New York, NY, USA: ACM, 2009, pp. 1–14.

[15] I. Solis and K. Obraczka, “The impact of timing in data aggregation for sensor networks,” in *2004 IEEE International Conference on Communications (IEEE Cat. No.04CH37577)*. IEEE, 2004, pp. 3640–3645 Vol.6. [Online]. Available: http://dx.doi.org/10.1109/ICC.2004.1313222

[16] J. Polastre, R. Szewczyk, and D. Culler, “Telos: enabling ultra-low power wireless research,” in *IPSN ’05: Proceedings of the 4th international symposium on Information processing in sensor networks*. Piscataway, NJ, USA: IEEE Press, 2005.

Paper E :

O-CTP: Hybrid Opportunistic Collection Tree Protocol for Wireless Sensor Networks

J. Flathagen, E. Larsen, P. E. Engelstad and Ø. Kure
Presented at the Seventh IEEE International Workshop on Practical Issues in Building Sensor Network Applications (SenseApp), 2012
Paper published in workshop proceedings of the IEEE Conference on Local Computer Networks, LCN, Clearwater, USA, October 22–25, 2012, pp. 943 - 951, ISBN:978-1-4673-2129-7.

O-CTP: Hybrid Opportunistic Collection Tree Protocol for Wireless Sensor Networks

Joakim Flathagen*‡¶, Erlend Larsen*, Paal E. Engelstad§¶ and Øivind Kure‡¶
*Norwegian Defence Research Establishment (FFI), ‡Q2S NTNU, ¶UNIK, §University of Oslo
Email: {jfi,erl}@ffi.no,{paale,okure}@unik.no

***Abstract*—Radio interference or deliberate jamming attacks can cause highly unpredictable communication in Wireless Sensor Networks (WSNs). Most prevalent WSN platforms consist of low-cost hardware with no effective measures against these threats. Most proposed countermeasures require a more advanced hardware design or radical changes to the 802.15.4 MAC protocol. These alternatives can be very difficult or even impossible to apply to existing WSN designs. In this paper we do not attempt to change the hardware or the MAC protocol. Instead we investigate how WSN routing protocols behave when the network is affected by interference. The paper proposes enhancements of CTP, the de-facto tree-based routing protocol for WSN, using opportunistic routing. We compare our approach with a wide range of protocols: CTP, TYMO, MultihopLQI, broadcast and geographic opportunistic routing in a real-life TelosB testbed subjected to different interference levels. The results show that our hybrid protocol, O-CTP, both improves the data delivery rate and reduces the cost when compared to standard routing protocols.**

***Index Terms*—Interference, Jamming, Opportunistic routing, Wireless Sensor Networks**

I. INTRODUCTION

Wireless Sensor Networks (WSNs) often suffer from highly unpredictable wireless communication conditions. The quality of the communication depends on several factors such as the deployed environment, the frequency spectrum and modulation schemes utilized, and the communication devices themselves [1]. The multi-hop nature of WSNs further increases the problem. Results on deployed networks and testbeds show that typical delivery ratios are between 70 and 99% [2]–[4], but could even go as low as 20-40% [5]–[7]. One reason for the unpredictable packet delivery rate is that the wireless channel fluctuates significantly with time. People or vehicles entering the sensed area, or even rain and wind, give unreliable RF propagation. Interference in the chosen frequency band adds further weight to the problem. For an IEEE 802.15.4 equipped sensing node operating at 2.4GHz, possible sources of interference include other radio transmitters operating in the same frequency band (e.g., 802.11, Bluetooth or video transmitters), harmonic interference from other bands, microwave ovens and military radars. An opponent may also use interference intentionally to disrupt communications (i.e., radio jamming) [8].

Much work has been dedicated to create effective measures against interference and jamming in WSNs. The most effective methods involve changes to the physical layer, e.g., moving from the standard Direct-sequence spread spectrum (DSSS) in 802.15.4 to Frequency-hopping spread spectrum (FHSS) or using directional antennas. Some methods focus on changing the MAC protocol [6]. Few of these countermeasures can, however, be effectively applied to the prevalent WSN platforms today (i.e., TelosB, Mica and IRIS), without redesigning the platform. The focus in this article is therefore to study how the delivery rate can be maximized even in interfered environments, simply by choosing the routing protocol cleverly.

Traditional routing protocols for WSNs deal with dynamics in the underlying network structure by using various metrics, e.g., the number of hops [9], radio link quality [10] or Expected Transmission Count (ETX) [4]. Despite these attempts, the metric calculations have difficulties in coping with the rapid changes in the unreliable wireless medium, making it difficult to choose the optimal next hop node. This observation has led to the development of opportunistic routing [11]–[13]. Opportunistic routing is proven to be very effective in error-prone wireless networks, since it allows *any* node that is closer to the destination to participate in packet forwarding. The overhead that comes with opportunistic routing is, however, a difficult problem to tackle. Our experiments show that opportunistic routing is most relevant when the network is subjected to high and unpredictable interference and traditional routing thus performs badly.

The main contributions in this paper are:

- A presentation of a new hybrid opportunistic protocol (O-CTP), which uses traditional routing when the network is stable and has reasonably little packet loss, but switches to opportunistic forwarding when the network is subjected to interference or jamming.
- An empirical comparison of six routing protocols in an interfered environment using a testbed of 20 TelosB sensing nodes. We employ four different interference patterns and show that O-CTP gives the overall best balance between packet delivery ratio and overhead.

The rest of this article is organized as follows. Section II reviews related work. Section III describes O-CTP in detail. The test and experiment setup is described in section IV. Section V and VI offer experimental results. Finally, in section VII we conclude the article.

II. RELATED WORK AND BACKGROUND

In this section, we review the prior research addressing the issues of routing in WSNs. We focus primarily on protocols that are implemented and tested in real-world environments.

First, we discuss traditional routing protocols and then we explain different opportunistic alternatives. Finally, we explain why there is room for improvement in WSN routing.

A. *Traditional routing*

TYMO [9] and NST-AODV [14] both originate from the ideas behind DYMO and AODV, which are protocols tailored to mobile ad-hoc networks. There are three basic problems that arise with these protocols in WSNs. 1) The hop count metric does not provide good performance since it treats all hops as equal. 2) Routes are based on the end-to-end principle, meaning that they are costly both to establish and to maintain in a lossy environment. 3) The protocols do not exploit the fact that most traffic is destined to one node (i.e., the sink).

Convergecast routing protocols are proposed to address the above issues. In convergecast protocols, such as MultihopLQI [10] and Collection Tree Protocol (CTP) [4], all traffic is assumed destined to a single sink node. The sink node constitutes the root in the routing tree. Each node uses a gradient minimization approach to determine the next hop (i.e., its parent). MultihopLQI uses the Link Quality Indicator (LQI) from the physical layer to additively obtain the gradient towards the sink. LQI is proven to be more stable in selecting the best paths than using hop-count [15]. Beaconing (with fixed interval) is used by all nodes to measure LQI and to support changes in the topology. CTP builds on MultihopLQI but distinguishes from it on two central features: 1) It uses the Expected number of transmissions (ETX) as its routing metric as opposed to LQI: Starting with an ETX of 0 at the sink, each node calculates its own ETX as the ETX reported by the parent plus the ETX of its own link to the parent. 2) CTP uses adaptive beaconing by extending the Trickle algorithm [16] to reduce the route repair latency and send fewer beacons when the network is stable. To adapt quickly to topology changes, the trickle timer interval is reset whenever a routing loop is detected or the routing cost decreases significantly.

It is worth noting that NST-AODV, TYMO, MultihopLQI and CTP are implemented in TinyOS and tested in several real WSNs [4], [9], [10], [14].

B. *Opportunistic routing*

Traditional routing protocols aim to find the optimal paths through a network by daisy-chaining the links with the presumed best qualities. This approach stems from protocols found in fixed infrastructure and is ideal when there are minimal network dynamics. The metric calculations, however, have difficulties coping with the rapid fluctuations in the wireless domain. Consequently, the routing decisions may be based on historic and outdated metrics. Opportunistic routing differs from traditional routing since it exploits, rather than attempting to hide, the broadcast nature of the wireless medium [11]–[13], [17], [18]. In opportunistic routing, a node does not preselect a preferred forwarder according to a set of (possibly outdated) metrics. Instead, opportunistic routing exploits the fact that there might be many potential forwarders in a node‘s vicinity able to receive the broadcast packet. The designated forwarding nodes may differ from one packet to the next. Hence, channel fluctuations are implicitly taken into account since the forwarding decision is carried out while the packet moves through the network.

Various opportunistic routing protocols differ mainly in the way the relay nodes decide on which node should retransmit the packet. In the seminal opportunistic routing protocol ExOR [11], the sender chooses a candidate subset of all its neighboring nodes that could bring the packet closer to its destination. This list is prioritized according to distance and put in the packet header. Each recipient delays a certain time depending on its position in the list before forwarding the packet. LAOR [17] and GeRaF [19] take a similar approach. Other protocols, such as TORP [13] use ETX to choose the candidate subset. MORE [12] relaxes the need to coordinate the forwarding, since the approach combines opportunistic routing with network coding. ORW [20] is a promising opportunistic routing scheme tailored directly to duty-cycled networks and can supplement our work in a future version.

C. *Towards a hybrid protocol*

Although there are numerous papers that study opportunistic routing analytically or via simulations [11], [12], [17], there are few papers that investigate real-world implementations. The works by Carnley et al. [13], Joe et al. [18] and Landsiedel et al. [20] are rare exceptions. There are also few papers that specifically analyze the trade-off between traditional routing and opportunistic routing. Shah et al. [21] use simulations to conclude that opportunistic routing is superior to geographical routing when the channel quality is low. Carnley et al. [13] show that TORP improves throughput and lowers the overhead compared to CTP in some scenarios.

To the best of our knowledge, this is the first paper that analyzes the trade-off between traditional routing and opportunistic routing in interfered environments. Further, we are the first to provide a routing solution that is based on a hybrid approach.

III. O-CTP: A HYBRID OPPORTUNISTIC COLLECTION TREE PROTOCOL

The hybrid protocol presented in this paper is called Opportunistic Collection Tree Protocol (O-CTP). O-CTP consists of three fundamental parts:

1) The traditional routing part, which is largely based on CTP.
2) An opportunistic routing part, which is employed when traditional routing is no longer effective.
3) A set of *triggers*, which enables switching between traditional routing and opportunistic routing.

Before digging into the protocol specification, it is worth discussing the intuition underlying our protocol design.

A. *Why opportunistic routing is a trade-off*

It is helpful to consider the simple network presented in Fig. 1. In the network example there are three possible routes from source s to the destination d. The three alternative routes

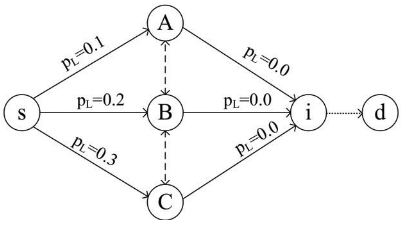


Fig. 1. Opportunistic routing exploits the broadcast nature of wireless networks. Node s does not preselect a preferred forwarder but exploits the fact that there might be many potential forwarders in a node‘s vicinity able to receive the broadcast packet

go via either of the nodes A,B or C to i. The three possible links from s are all subjected to some degree of packet loss varying from 10% to 30%. For the remaining path we assume no packet loss. In the following discussion, we use CTP as an example of a traditional routing protocol. CTP will choose A as the preferred forwarder for s, since choosing A minimizes the overall ETX from s to d. Hence, a packet loss of 10% can be expected for the first hop. *Opportunistic routing* on the other hand, takes a different approach, since it exploits the fact that all transmissions are broadcast. Hence, it does not preselect a single forwarder, but assumes that at least one of the neighbors receives and forwards the packet. In the case in Fig. 1, all the nodes A, B, C are able to receive a broadcast packet from s. The combined packet loss probability for the first hop is now reduced to $0.1 \times 0.2 \times 0.3 = 0.006$, which is a tremendous improvement over the CTP protocol. The performance of CTP is, however, not as depressive as it might first seem, since CTP employs retransmissions (up to 31 times as default). Consequently, the overall delivery rate can therefore be expected to be very close to 100%. Taking in account the retransmissions, the expected cost (transmissions per packet) to reach i using CTP is about 2.11 ($\frac{1}{1-0.1}$ for the first hop and 1.0 for the second).

The basic problem that arises with opportunistic routing is that the forwarding nodes are not necessarily able to hear each other. In our example, B will overhear all retransmissions performed by A or C, and since it is wasteful for B to forward those packets it effectively suppresses duplicate forwarding. But since A can not hear C and vice versa, they will both forward the same packet. Such duplicates are not only wasteful in terms of energy. They also increase the collision probability. Despite much research in reducing duplicates, there is no effective mechanism to eliminate such duplicates entirely [22]. Assume now that each of the nodes A,B,C has a probability of $P_{FA} = P_{FB} = P_{FC} = \frac{1}{3}$ to be the first forwarder and that the opportunistic routing protocol performs retransmissions. The expected cost can be calculated as the sum of the expected number of transmissions for each hop. For the fist hop, the expected number of transmissions is $\frac{1}{1-0.006}$, while the second hop gives $2P_{FA}+P_{FB}+2P_{FC}$. This gives a total cost of 2.67, which exceeds the CTP cost. Since duplicates will occur on the second hop when OR is used, CTP is the most effective protocol in this example.

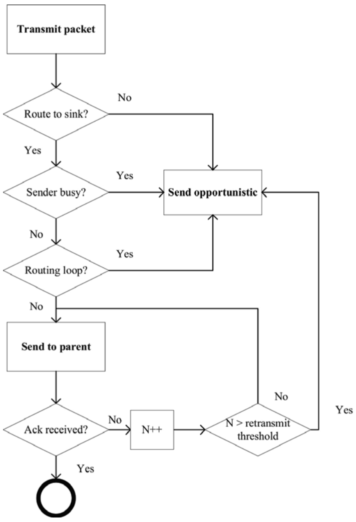


Fig. 2. The basic operation of O-CTP

As previously discussed, the link loss is never stable as in the above example, but fluctuates with time. Imagine now that the packet loss probability on the link $s \rightarrow A$ suddenly increases to 90% due to some interference. CTP will still choose A as its preferred forwarder for some time. The overall cost on the route s to i via A now increases to 11 ($\frac{1}{1-0.9} + 1$). The high number of retransmissions required to achieve 100% delivery rate quickly translates to a huge waste of energy. For opportunistic routing, the situation is practically unchanged, since the packet loss on the first hop is now $0.9 \times 0.2 \times 0.3 = 0.054$ resulting in a total cost of 2.72. Hence, the cost does not increase significantly from the previous situation. In this example, the opportunistic protocol outperforms CTP.

We have now illustrated why traditional routing performs best when the network conditions are fairly good and predictable, while opportunistic routing performs best when the network conditions are poor and unpredictable. Our hypothesis is that a hybrid protocol, which is able to change its operation based on the current network dynamics, could benefit from both of these worlds and give an overall improved performance.

B. When to switch from traditional routing to opportunism?

We decided to build our hybrid protocol based on CTP, since this is the de-facto collection protocol for real-world deployed WSNs and has shown high delivery ratio in previous studies. The basic idea of O-CTP is to switch from CTP operation to opportunism whenever the network is subjected to interference. A best-of-both-worlds protocol is very difficult to construct, since there is no fail-free trigger that allows the protocol to switch to opportunistic routing at the optimal moment. The central component of O-CTP is therefore the triggering part.

The trigger could be built as dependent on cross-layer communication. However, since CTP is built to be independent of layer 1 and layer 2, we decided not to break this hardware-independency by introducing cross-layering. There are, however, some possibilities to monitor the underlying network status directly from the forwarding engine in CTP. We have used these to trigger opportunistic forwarding. This is a distributed decision, and all nodes can decide the forwarding method for its current packet transmission. A switch to opportunistic sending is performed if one of the following situations occur within the CTP routing protocol:

1) *There is no route to the sink (i.e., no parent).* Even if CTP is in a no-route state, there might be many possible routes available that could be used immediately by the opportunistic protocol.
2) *Sender is busy.* Normally, in CTP, the forwarding engine denies packet forwarding if the forwarding layer is busy. However, in this state, packets can still be forwarded opportunistically.
3) *Routing loop detected.* Even if standard CTP has mechanisms to deal with loops, we observed that loops occur very frequently in interfered networks. Since the detection of a loop means that there is a problem somewhere in the routing tree, O-CTP is implemented such that when a loop is detected, the packet is forwarded opportunistically.
4) *The retransmit threshold has expired.* In standard CTP, the forwarding engine gives up packet forwarding when the retransmit threshold expires. In O-CTP, the packet is forwarded opportunistically instead.

Either of the above circumstances indicate that there is a problem with the packet forwarding, which means that opportunism is beneficial. These trigger mechanisms are evaluated empirically in section V. The decision on whether to forward a packet opportunistically or not is memory-less (cf. Fig. 2) and it is not necessary to use a trigger to switch back from opportunistic forwarding to traditional forwarding. In other words, a packet following a previous packet that was forwarded with opportunistic routing, may be forwarded with opportunistic routing or traditional routing depending on the current state of the forwarding engine.

C. The opportunistic part of O-CTP

There are several previous routing protocols that shares salient opportunistic routing features, e.g., ExOR [11], LAOR [17], BRL [23], GeRaF [19] and IGF [24]. Many of the protocols in this category can serve the purpose as the opportunistic routing part of O-CTP. Since none of these opportunistic protocols are publicly available for TinyOS, we implemented our own protocol to validate the hybrid routing approach in O-CTP. Our protocol is a geographic-opportunistic routing protocol (GEOPP) that covers the basic opportunistic principles presented in previous research.

The key difference between various opportunistic protocols is how the forwarding decision is performed. For example, IGF, BRL, and GeRaF, employ RTS/CTS handshaking between the source and the possible forwarders before transmitting the data packet. The motivation behind the RTS/CTS approach is to pre-elect one single forwarder and in this way limit the number of possible duplicates. However, the drawback is that even after a successful RTS/CTS exchange, the probability of successfully receiving a larger data message might be very low [25]. Another method, used by LAOR [17] and ExOR [11], is to specify a list of forwarding nodes in the packet header. The list is sorted in decreasing order of progress towards the sink, and hence, represents the priority of the forwarders. The shortcoming of this approach is that all potential forwarders can not possibly be added to the list since the header size is limited. This limitation can leave some long-progress paths underutilized. Considering the example in Fig. 1, there could be a small possibility that a transmission from s might reach i directly. This opportunity will be left unused if only A, B, C is stated in the forwarding list. Further, if any of the links $s \rightarrow A, B, C$ are downstream unidirectional, they will be left unused since s has no knowledge of them.

Due to our interest in making a working system, we had to trade off some advanced protocol ideas presented in previous research for simpler ones. In GEOPP, there is no RTS/CTS scheme. Neither is there any forwarding list in the packet header. Hence, there can be many possible forwarders receiving the same packet. To make sure that a minimum number of these neighbors forward the packet, each neighbor computes a dynamic forwarding delay (DFD) as in ExOR, depending on its position relative to the sink. The node with a small progress towards the sink computes a higher delay than a node with a large progress. Assuming that all nodes know their own location and the sink location, the DFD is simple to calculate. The node that computed the smallest DFD (i.e., the node which is closest to the sink) forwards first. The other forwarders overhearing this retransmission, stops their DFD-timer and deletes the packet from their forwarding queue. In addition, the node transmitting the packet uses the overheard retransmission as an implicit acknowledge indicating that the packet is undergoing a positive progress towards the sink. If no such implicit acknowledge is heard, the node may choose to retransmit the packet (still opportunistically) up to a predefined number of times. Notice that the problem with most geographical routing protocols is that packets can be routed to a dead-end, where there is no neighbor closer to the destination. The aim of this paper has not been to attempt to solve this problem, and GEOPP therefore lacks a solution for the dead-end problem. Although this issue should be investigated, we do not consider it as a big problem here since GEOPP is a fallback solution used only when CTP fails.

Since the forwarding area in GEOPP covers all nodes with a positive progress towards the sink, GEOPP can expect a high delivery ratio but also a relatively high cost compared to some of the other opportunistic routing protocols due to more duplicated packets. Finally, even if GEOPP is presented here as an integral part of O-CTP, it is, as shown in the empirical analysis later in the paper, possible to run the protocol stand-

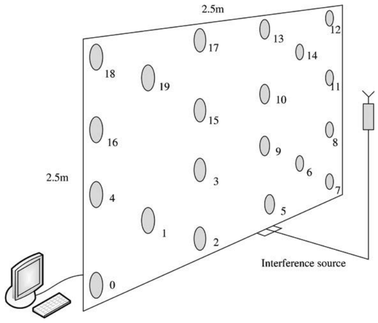


Fig. 3. The testbed consists of 20 TelosB sensing nodes and a 2.4GHz software controlled interference source. Node 18 is the sink collecting all information.

alone as a pure opportunistic protocol.

IV. MATERIALS AND METHODS

A. testbed

To evaluate the performance of different routing protocols in a realistic setting, we implemented a real testbed (cf. Fig. 3). The testbed consisted of 20 TelosB sensing nodes [26] covering an area of $2.5 \times 2.5m^2$. The TelosB has a 4 MHz MSP430 processor, 10 KB of RAM and 48KB program memory. TelosB uses the Chipcon CC2420 radio in the 2.4GHz band, an IEEE 802.15.4 compatible radio with O-QPSK modulation with DSSS at 250kbps. The output power was set to -25dBm, which gave a multihop network with an average node degree of 6. The nodes were connected to a standard laptop using a combination of USB cables and hubs. This USB backbone was used for reprogramming and debugging. Node 18 was the designated sink, forwarding packets to the computer over USB.

TABLE I
THE FOUR DIFFERENT INTERFERENCE PATTERNS EMPLOYED IN THE EXPERIMENTS AND THE RESULTING AVERAGE PACKET LOSS

	Interference pattern			
	No	**Low**	**Medium**	**High**
Dutycycle (T_{on},T_{off})	0,∞	10s,60s	10s,30s	20s,30s
Avg packetloss	2%	13%	23%	33%

Network interference can come from various sources. To allow interference in a controlled fashion, we used an ATT Q30 2.4GHz signal jammer, which was placed 1m from the testbed surface (cf. Fig. 3). Our goal was to introduce realistic interference and not complete jamming, and the jammer-antennas were therefore equipped with 20dB damping. Since most interference sources (be it radar, video links or 802.11) are transient, we used duty cycling of the signal jammer controlled from software for the experiments. This approach enabled both realistic and reproducible results. By employing different interference patterns, from continuously off to increasingly more aggressive interference, we could manipulate the packet loss in the network in a predictable manner. Typical packet losses for communication from the sensing nodes towards the sink for the different interference patterns are presented in Table I.

B. Protocols

For the purpose of the experiments in this paper, O-CTP was implemented for TinyOS 2.x. In our empirical study, we compare O-CTP with the most prevalent routing protocols for WSNs: CTP [4], MultihopLQI [10] and TYMO [9]. We use the default parameter setting for all three protocols. We also compare with the pure opportunistic protocol GEOPP, and with naïve broadcast (BCAST). Our BCAST implementation works as follows: Message originators send broadcast packets. A node hearing a BCAST transmission, records the sequence number and the originator (to avoid duplicate retransmissions) and retransmits the packet. Eventually, the packet reaches its destination (i.e., the sink). BCAST can be seen as the simplest routing protocol available. Since it also can be categorized as opportunistic (it uses multiple forwarding nodes), it serves well as a baseline for comparison in our study.

V. ANALYZING O-CTP TRIGGERS

To obtain valuable understanding of O-CTP, we first investigate the triggers initiating opportunistic forwarding. Table II shows the relationship between the traffic sent with opportunistic routing and the traffic sent with traditional routing when the network is exposed to different interference patterns. Further, the table shows the fraction of the opportunistic routing traffic directly traced to each trigger. In this experiment, the retransmit threshold was set to 3. For each of the interference settings, we ran 10 experiments lasting one hour each. As shown, the share of the opportunistic data traffic increases with increasing interference. Another observation is that the expiration of the retransmit threshold contributes to most of the opportunistic data traffic. The other incidents (i.e., no parent, sender is busy, routing loop) do not occur very often. In practice, the retransmit threshold is the critical parameter in optimizing the performance of O-CTP and manipulating this threshold is the logical next step in the investigation.

TABLE II
THE AMOUNT OT TRAFFIC TRANSMITTED OPPORTUNISTICALLY (FOR EACH TRIGGER) AND USING TRADITIONAL ROUTING

Opportunistic	**Interference pattern**			
trigger	**No**	**Low**	**Medium**	**High**
No parent	5.5%	6.9%	6.8%	8.2%
Sender busy	0%	1.1%	0.4%	0%
Loop	0%	1.6%	1.7%	3.4%
RTX expired	2.9%	11.8%	18.7%	27.5%
None (traditional routing)	91.6%	78.6%	72.4%	60.9%

Fig. 4 shows the effect of manipulating the retransmit threshold on the delivery ratio. We ran one one-hour experiment for each retransmit threshold between 1-40 for each interference setting - a total of 160 experiments. The astute reader can notice some small irregularities in the results in Fig. 4. They are natural, since we ran only one experiment per data point. Despite this fact, the trends are clear. When

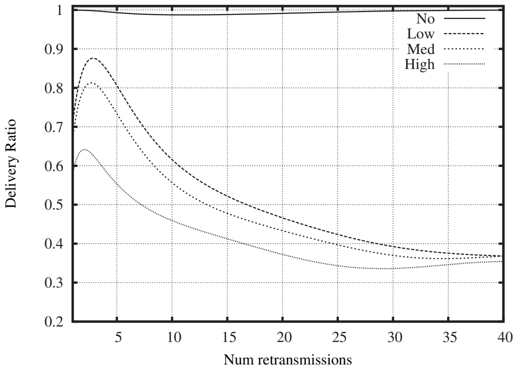


Fig. 4. Delivery ratio for O-CTP with different retransmit thresholds

there is minimal interference, the retransmit threshold setting is not crucial. The default setting in CTP is rather high (31). This is reasonable, since in a sink-routed tree, the next packet in the queue has the same destination as the current packet (i.e., the sink). Consequently, the outcome of transmitting the next packet in the queue will be the same as the current one [4]. For O-CTP, however, a high retransmit threshold for the traditional routing part is not beneficial for two reasons. *First*, a high number of retransmissions indicate that there is a problem with interference, meaning that the packet delivery could have been improved by switching to opportunism at an earlier stage. *Second*, retransmitting a packet several times puts a high load on the network. This can influence other on-going transmissions, which again increases contention and collisions. We also experienced that the probability for creating routing loops increased with increased retransmit threshold. A late switch to opportunism in a saturated and interfered network (with possible loops) gives no improvement for packet delivery. Based on the results shown in Fig. 4 the retransmit limit for traditional routing was set to 3 (triggering opportunistic forwarding) in the subsequent experiments. For GEOPP, we remember that the retransmission function is based on listening to implicit acknowledgements. Since these acknowledgements are unreliable (requiring symmetric links), incrementing the retransmission threshold therefore increases the cost as well. Retransmissions also contribute to more duplicate packets in the network. We observed that a high retransmit threshold setting for the opportunistic routing protocol indeed improves packet delivery, but the cost of bringing the delivery rate close to 100% could be extremely high in an interfered network. For the subsequent experiments, the retransmit threshold for opportunistic routing was set to 2 to balance reliability and cost.

VI. ROUTING PROTOCOL COMPARISON

In this section we evaluate O-CTP using two empirical experiments. The first experiment investigates how the three protocols O-CTP, CTP and pure opportunism (GEOPP) react to interference. In the second experiment we study O-CTP against five routing protocols in various interference scenarios.

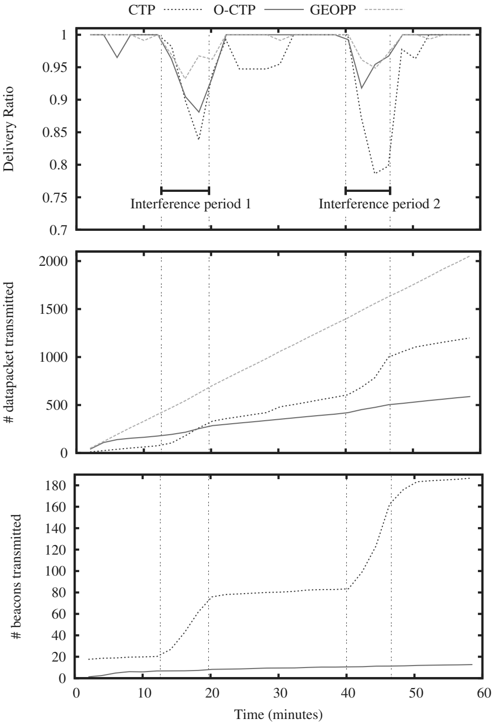


Fig. 5. The delivery ratio and overhead of CTP, O-CTP and GEOPP when the network is under medium interference.

In comparing the protocols, three key performance metrics are evaluated. 1) *Packet delivery ratio* – which is defined as the number of packets received (duplicates not included) divided by the number of application packets transmitted, 2) *the number of data packets transmitted* – which gives a picture on the number of retransmissions and duplicates created by the protocol, 3) *the number of beacon messages transmitted* – which is the overhead of maintaining the routing protocol tables.

A. O-CTP related to CTP and pure opportunism

First, we perform an experiment with mixed interference. For the experiment, we have used the testbed setup explained previously. We ran CTP, O-CTP and GEOPP (isolated) on the testbed for one hour. The packet rate was fixed at one packet per node per 20s, which represents a typical medium duty cycle sensor network. Between 12-19 and 40-46 minutes, we ran the signal jammer with the medium interference pattern. The rest of the test period elapsed without any interference.

Fig. 5 shows the delivery ratio averaged each 2 minutes. In the periods without interference, the delivery ratio is close to 100% for all three protocols. During interference, all protocols

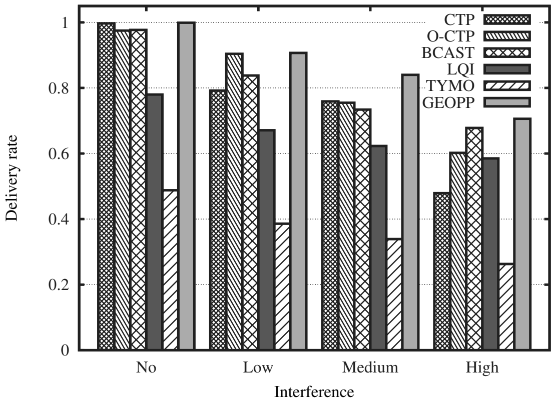


Fig. 6. The delivery ratio on different interference patterns

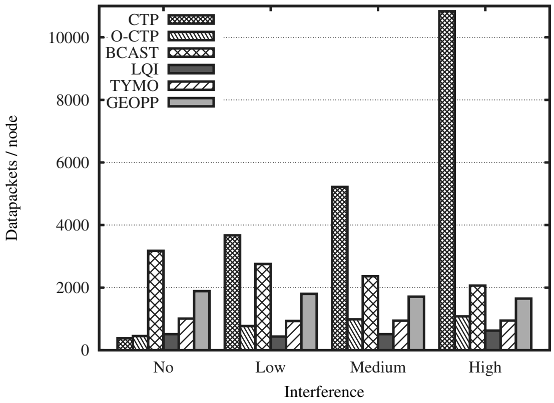


Fig. 7. Datapackets transmitted by each node on different interference patterns

are affected. CTP lose most packets, O-CTP is affected to a lesser extent and GEOPP loses the fewest packets. This is in compliance with our previous analysis. The same figure also shows the cumulative number of data packets and beacon packets transmitted per node. CTP increases both the number of beacon packets and the number of data packets during the interference period. One part of the data packet increase is traced to a rise in the number of retransmissions, and one part is caused by routing loops, which are inevitable when the parent change rate increases. The data packet rate of O-CTP changes slightly during interference since the number of opportunistic transmissions increases. For GEOPP, the data packet rate is stable during the test period. Notice that CTP transmits more data packets than O-CTP even during the non-interference time. Even though our jammer is turned off during this period, weak links in the network can occur, leading to packet retransmissions or loops. In such cases O-CTP performs better. It is important to note that it is possible to reduce the overhead of CTP significantly by altering the routing parameters. By increasing the minimum trickle interval from 64ms to 30000ms and reducing the number of retransmissions from 31 to 3, we were able to reduce the overhead to almost $\frac{1}{10}$ of the numbers presented in Fig. 5. However, the major disadvantage was that the delivery ratio was reduced with 15-20%, so this setting can not be recommended.

In the comparison, O-CTP presents excellent packet delivery ratios (albeit lower than GEOPP) and it clearly has the lowest overhead. In the next section we measure the performance of O-CTP under a wider range of conditions, and compare with an extended set of routing protocols.

B. Comparing six routing protocols

The routing protocols we consider here are CTP, O-CTP, BCAST, MultihopLQI (LQI), TYMO, and GEOPP. Each routing protocol is tested for one hour for each interference setting (i.e., ”no”, ”low”, ”medium”, and ”high”), repeated ten times and the results are averaged.

Fig. 6 shows the packet delivery ratio for each routing protocol and for each interference setting. Let us first focus on the situation without any external interference. We observe that the packet deliveries for CTP, O-CTP, BCAST and GEOPP are very similar. Compared with these, LQI loses about 20% more packets and TYMO about 50% more packets. When increasing the interference from ”no” to ”low”, CTP and BCAST loses 10-15% more packets than O-CTP. By increasing the interference further, BCAST and GEOPP (pure opportunistic routing) show the best performance, while CTP seems to be very sensitive to high interference. This observation is in compliance with our previous analysis. In all cases LQI and TYMO are outperformed by O-CTP, BCAST and GEOPP.

Fig. 7 shows the average number of data packets transmitted per node during the test. The first observation is that CTP is very effective when there is no interference. This shows that the ETX routing works excellent as long as the links are stable. However, even with low interference, CTP has a vast overhead, which increases tremendously when the interference increases. The rise is caused by CTP’s quick reaction to topology changes, which increases the parent change rate and again increases the probability for routing loops. Interestingly, BCAST is more efficient than CTP in interfered environments. Our hybrid protocol, O-CTP, shows higher overhead than CTP in the ”no interference”-setting. This is due to the fact that a fraction of the traffic is sent opportunistically (see table II), with unavoidable duplicates. When there is much interference, the hybrid protocol sends an even larger part of the traffic opportunistically, and this is also reflected by the overhead. Nevertheless, the overhead with standard CTP is higher with one order of magnitude. The hybrid approach also reduces the overhead with 50-80% compared to pure opportunism. In all cases, LQI demonstrates much lower data packet load than the other protocols; however, it comes at a price, since the delivery ratio is significantly reduced (cf. Fig. 6).

For CTP, the number of beacon messages increases tremendously even with little interference (cf. Fig. 8). The problem

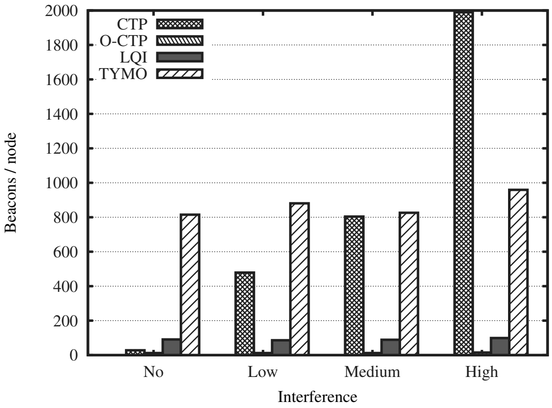


Fig. 8. Beacons transmitted by each node on different interference patterns

worsens when adding more interference. This phenomenon is mainly caused by the trickle timer controlling the beacon interval, which is reset to a (default) 64ms interval whenever a parent is lost or a neighbor node detects a topology problem. TYMO shares the high overhead problem, albeit its cause is different. One reason is that TYMO floods the entire network in order to find the route to the sink; a process that is performed very often. Another reason is that TYMO is not capable of constructing routes over asymmetric links. Compared with Fig. 7, we see that the number of beacon packets and data packets combined for TYMO, surpasses the number of data packets for BCAST. Although we have only tested one testbed size, there is no reason to believe that TYMO is better than BCAST for larger networks. O-CTP shows stable beacon results regardless of the network environment. Obviously, for BCAST and GEOPP there is no routing traffic, since both protocols are beaconless.

C. Discussion of the results

It is worth discussing our results compared to other studies on real WSNs. TYMO performed badly in all our experiments, which complies well with results from other recent studies [7], [27]. Nevertheless, we believe that there might be room for improvement by taking advantage of some more advanced AODV-features. CTP and MultihopLQI have been studied numerous of times recently [2], [4], [28]. Most studies conform to our conclusion that CTP has overall better packet delivery than MultihopLQI. The work by Gnawali et al. [28] is the only one studying CTP under interference. However, in our setup, CTP showed much higher overhead than the results presented in their paper. Carnley et al. [13] and Landsiedel et al. [20] support our finding that opportunism can indeed outperform CTP.

VII. CONCLUSIONS AND FUTURE WORK

Radio interference or deliberate jamming attacks can cause highly unpredictable communication in WSNs. While advancements in hardware design and MAC protocols can improve packet delivery, we have investigated a simpler approach using hybrid opportunistic techniques on the routing layer. Our hybrid protocol (O-CTP) is designed by combining the high packet delivery ratio of opportunistic routing in error-prone wireless networks, and the energy efficiency of traditional routing in stable networks. In the paper we used a real testbed and showed that O-CTP improves both packet delivery and system lifetime in an interfered network compared to five other protocols.

There is still a huge potential for improvement of O-CTP. Future works include improvements in the trigger (e.g., using cross-layering) making the protocol react faster to interference, and techniques to reduce the number of duplicate packets. Further, the protocol should incorporate the challenges posed with duty-cycled sensing nodes.

REFERENCES

[1] J. Zhao and R. Govindan, “Understanding packet delivery performance in dense wireless sensor networks,” in *Sensys ’03*. ACM, 2003, pp. 1–13.

[2] S. Lin, G. Zhou, K. Whitehouse, Y. Wu, J. Stankovic, and T. He, “Towards stable network performance in wireless sensor networks,” in *RTSS’09*. IEEE, 2009, pp. 227–237.

[3] T. Liu, A. Kamthe, L. Jiang, and A. Cerpa, “Performance Evaluation of Link Quality Estimation Metrics for Static Multihop Wireless Sensor Networks,” in *IEEE SECON ’09*, june 2009, pp. 1 –9.

[4] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis, “Collection tree protocol,” in *SenSys ’09*. ACM, 2009, pp. 1–14.

[5] V. Shnayder, B. Chen, K. Lorincz, T. Fulford-Jones, and M. Welsh, “Sensor networks for medical care,” in *SenSys 05*, 2005, pp. 314–314.

[6] A. Wood, J. Stankovic, and G. Zhou, “DEEJAM: Defeating energy-efficient jamming in IEEE 802.15. 4-based wireless networks,” in *SECON’07*. IEEE, 2007, pp. 60–69.

[7] J. Vanhie-Van Gerwen, E. De Poorter, B. Latré, I. Moerman, and P. Demeester, “Real-life performance of protocol combinations for wireless sensor networks,” in *2010 IEEE SUTC*, 2010, pp. 189–196.

[8] A. Mpitziopoulos, D. Gavalas, C. Konstantopoulos, and G. Pantziou, “A survey on jamming attacks and countermeasures in WSNs,” *Communications Surveys & Tutorials, IEEE*, vol. 11, no. 4, pp. 42–56, 2009.

[9] R. Thouvenin. TYMO source code. [Online]. Available: http://www.tinyos.net/tinyos-2.x/tos/lib/net/tymo/

[10] G. Tolle. MultihopLQI source code. [Online]. Available: http://www.tinyos.net/tinyos-2.x/tos/lib/net/lqi/

[11] S. Biswas and R. Morris, “Opportunistic routing in multi-hop wireless networks,” *ACM SIGCOMM’04*, vol. 34, no. 1, pp. 69–74, 2004.

[12] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, “Trading structure for randomness in wireless opportunistic routing,” *ACM SIGCOMM Computer Communication Review*, vol. 37, no. 4, pp. 169–180, 2007.

[13] J. Carnley, B. Sun, and S. Makki, “TORP: TinyOS Opportunistic Routing Protocol for Wireless Sensor Networks,” in *Consumer Communications and Networking Conference (CCNC), 2011 IEEE*. IEEE, pp. 111–115.

[14] C. Gomez, P. Salvatella, O. Alonso, and J. Paradells, “Adapting AODV for IEEE 802.15.4 Mesh Sensor Networks: Theoretical Discussion and Performance Evaluation in a Real Environment,” in *WOWMOM ’06*, 2006, pp. 159–170.

[15] P. Swieskowski and G. Werner-Allen, “Improving the Performance of a Data Collection Protocol,” *Division of Engineering and Applied Sciences, Harvard University*, 2005.

[16] P. Levis, N. Patel, D. Culler, and S. Shenker, “Trickle: A self-regulating algorithm for code propagation and maintenance in wireless sensor networks,” in *Proc. of the USENIX NSDI Conf.*, 2004, pp. 2–2.

[17] X. Yang, J. Yin, and S. Yuan, “Location-aided opportunistic routing for mobile ad hoc networks,” in *WiCom’09*. IEEE, 2009, pp. 1–5.

[18] I. Joe and D. Kim, “An opportunistic routing protocol for underground wireless sensor networks,” in *SNPD’09*. IEEE, 2009, pp. 602–605.

[19] M. Zorzi and R. Rao, “Geographic random forwarding (GeRaF) for ad hoc and sensor networks: multihop performance,” *IEEE transactions on Mobile Computing*, pp. 337–348, 2003.

[20] O. Landsiedel, E. Ghadimi, S. Duquennoy, and M. Johansson, “Low power, low delay: opportunistic routing meets duty cycling,” in *Proceedings of the 11th international conference on Information Processing in Sensor Networks*. ACM, 2012, pp. 185–196.

[21] R. Shah, S. Wietholter, A. Wolisz, and J. Rabaey, “When does opportunistic routing make sense?” in *PerCom 2005*. IEEE, 2005, pp. 350–356.

[22] A. Triviño-Cabrera, “Survey on Opportunistic Routing in Multihop Wireless Networks,” *IJCNIS*, vol. 3, no. 2, 2011.

[23] M. Heissenbüttel, T. Braun, T. Bernoulli, and M. Wälchli, “BLR: beacon-less routing algorithm for mobile ad hoc networks,” *Computer communications*, vol. 27, no. 11, pp. 1076–1086, 2004.

[24] B. Blum, T. He, S. Son, and J. Stankovic, “IGF: A state-free robust communication protocol for wireless sensor networks,” *Tech Rep. Dept Comp. Sci, Univ of VA*, 2003.

[25] J. Sanchez, P. Ruiz, and R. Marin-Perez, “Beacon-less geographic routing made practical: challenges, design guidelines, and protocols,” *Communications Magazine, IEEE*, vol. 47, no. 8, pp. 85–91, 2009.

[26] J. Polastre, R. Szewczyk, and D. Culler, “Telos: enabling ultra-low power wireless research,” in *IPSN’05*. IEEE Press, 2005, pp. 48–es.

[27] J. Lee, B. Kusy, T. Azim, B. Shihada, and P. Levis, “Whirlpool routing for mobility,” in *MobiHoc’10*. ACM, 2010, pp. 131–140.

[28] O. Gnawali, L. Guibas, and P. Levis, “A case for evaluating sensor network protocols concurrently,” in *WINTECH’10*. ACM, 2010, pp. 47–54.

Paper F :

Experiences from deploying a Wireless Sensor Network for Military Base Protection

J. Flathagen, R. Korsnes, E. Larsen, V. Pham, T. M. Mjelde, J. Sander

Submitted to IEEE Sensors Journal, 2013

Is not included due to copyright