

TECHNISCHE UNIVERSITÄT DRESDEN

FAKULTÄT INFORMATIK

**Impact of Random Deployment on Operation and  
Data Quality of Sensor Networks**

Habilitationsschrift

zur Erlangung des akademischen Grades

Dr.-Ing. habil.

auf dem Fachgebiet Informatik

Dr.-Ing. Waltenegus Dargie

Geboren am 24. Mai 1969 in Neghelle Borena, Äthiopien

Dresden 31. March 2010



## Acknowledgement

I am most grateful to Prof. Alexander Schill for giving me the opportunity to work with him; and for opening a great and wide door to research, publications, and participation in scientific activities. His kindness, readiness to help, and his remarkably positive attitude were like reliable anchors in a troubled sea.

I consider myself very fortunate for working with so many wonderful and outstanding graduate students at the Technical University of Dresden – they are many in number, but some of them are Daniel Hofmann, Chao Xiaojuan, Rami Mochaourab, Robert Krüger, and Qian Dong.

I acknowledge the contribution of Prof. Mieso K. Denko in my research career. Cooperating with him in various research activities and participating in lively and critical discussions were always rewarding. Moreover, he has offered me a generous opportunity to work with him in the context of several scientific conferences and workshops as well as reputable journals.

I am thankful of the encouragement and timely feedback I received from Prof. Kay Römer during the preparation of the summary of this work.

While I aim to earn credits for the research results presented in this compilation, I am also aware of the intrinsic and extrinsic contribution of several people that surround me at home, at the Technical University of Dresden, and around the world.

# Examination Committee

## **Prof. Dr. Andreas Pfitzmann (Chair)**

*Chair of Privacy and Data Security, Faculty of Computer Science, Technical University of Dresden*

## **Prof. Dr. rer. nat. habil. Dr. h. c. Alexander Schill**

*Chair of Computer Networks, Faculty of Computer Science, Technical University of Dresden*

## **Prof. Dr.-Ing. Eduard Jorswieck**

*Chair of Communications Theory, Faculty of Electrical Engineering, Technical University of Dresden*

## **Prof. Dr. Kay Römer**

*Institute of Computer Engineering, University of Lübeck*

## **Prof. Dr. Uwe Aßmann**

*Institute of software and multimedia engineering, Technical University of Dresden*

## **Prof. Dr. Christel Baier**

*Chair of Algebraic and Logical Foundations of Computer Science, Faculty of Computer Science, Technical University of Dresden*

## **Prof. Dr.-Ing. habil. Rainer G. Spallek**

*Chair of VLSI-Design, Diagnosis and Architecture, Faculty of Computer Science, Technical University of Dresden*

## **Prof. Dr.-Ing. habil. Martin Wollschlaeger**

*Chair of Industrial Communications, Faculty of Computer Science, Technical University of Dresden*

I dedicate this work to my wife Kathy, and to my children, Pheben and Joshua,  
with love



# Table of Contents

1. Summary (12 pages)
2. Modelling the Energy Cost of a Fully Operational Wireless Sensor Network. (13 pages)
3. A Topology Control Protocol based on Eligibility and Efficiency Metrics (28 pages)
4. Analysis of Error-agnostic Time and Frequency Domain Features Extracted from Measurements of 3D Accelerometer Sensors (8 pages)
5. Adaptive Audio-Based Context Recognition (11 pages)
6. Recognition of Complex Settings by Aggregating Atomic Scenes (8 pages)

# Impact of Random Deployment on Operation and Data Quality of Sensor Networks

Waltenegus Dargie

*Chair of Computer Networks, Department of Computer Science, Technical University of  
Dresden, 01062, Dresden, Germany*

---

---

## 1. Summary

Several applications have been proposed for wireless sensor networks. The application of Mainwaring et al. [15] gathers data from humidity, temperature, barometric pressure, and light sensors to monitor the activities of seabirds. Kim et al. [11] use wireless sensor networks for structural health monitoring, in which the structural integrity of bridges and buildings is inspected using accelerometer sensors. The application of Werner-Allen et al. [23] monitors active volcanoes using seismic and infrasonic sensors. The underlying network was able to capture 230 volcano events in just over three weeks. The application of Stoianov et al. [20] uses hydraulic and acoustic/vibration sensors to monitor large diameter, bulk-water transmission pipelines. Likewise, wireless sensor networks are proposed for precision agriculture [7, 6], healthcare [21], and underground mining [17].

Among the desirable features that inspired so many, one is the ease of deployment. Since the nodes are capable of self-organization, they can be placed easily in areas that are otherwise inaccessible to or impractical for other types of sensing systems. In fact, some have proposed the deployment of wireless sensor networks by dropping nodes from a plane, delivering them in an artillery shell, or launching them via a catapult from onboard a ship [2]. Arora et al. [3] report that an actual aerial deployment has been carried out using an unmanned aerial vehicle (UAV) at a Marine Corps combat centre in California – the nodes were able to establish a time-synchronized, multi-hop communication network for tracking vehicles that passed along a dirt road. While this has a practical relevance for some civil applications (such as rescue operations), a more realistic deployment involves the careful



planning and placement of sensors.

Even then, nodes may not be placed optimally to ensure that the network is fully connected and high-quality data pertaining to the phenomena being monitored can be extracted from the network. A good example is a wireless sensor network that monitors gas ( $\text{SO}_2$ ,  $\text{H}_2\text{S}$ ,  $\text{NH}_3$ , etc.) pipelines in oil refineries. One can consider two types of deployment: (1) area deployment, and (2) spot deployment. In area deployment, the entire field is covered by sensors, so that no “blind” spots will exist. This type of deployment is suitable if one expects a leakage to occur anywhere in the field; but it is expensive. In spot monitoring, specific spots in pipelines (such as bends and joints) are considered more likely places for a leak to occur. Subsequently, the nodes are placed at or near these spots. The second type of deployment is more economical and feasible for many real-world applications.

Spot deployment, however, entails a physical as well as a logical random distribution of nodes. Physical randomness is unavoidable because of the irregularities of the pipelines. Since the bends and joints are not uniformly distributed, the nodes will not be either. This may result in disconnection due to the absence of intermediate nodes between two or more clusters of nodes. Moreover, some nodes may exhaust their energy more quickly than others because they are intensively used as vital relaying nodes. Logical randomness occurs because of the mobility of the phenomena – once there is a gas leak, it diffuses at a velocity the magnitude and direction of which depends on several factors, including the direction of the wind and the density of the gas. Since the nodes are placed based on the likelihood of leakage occurrence, some may not be able to capture the mobility of the gas and may potentially deliver imperfect or even erroneous observation.

In the literature, these problems are partially addressed through dense deployments [26, 25, 22]. While this can be a plausible solution, it cannot always be supported due to mobility or space constraints. For example, in supply-chain management, containers house the items being monitored as well as the sensor nodes [16]; in healthcare applications, patients or nurses should not be burdened with or hindered by too many sensor nodes [19, 12].

This work aims to address the problem of random deployment through two complementary approaches:

The first approach aims to address the problem of random deployment from a communication perspective. It begins by establishing a comprehensive mathematical model to quantify the energy cost of various concerns of a fully operational wireless sensor network. Based on the analytic model, an energy-

efficient topology control protocol is developed. The protocol sets eligibility metric to establish and maintain a multi-hop communication path and to ensure that all nodes exhaust their energy in a uniform manner.

The second approach focuses on addressing the problem of imperfect sensing from a signal processing perspective. It investigates the impact of deployment errors (calibration, placement, and orientation errors) on the quality of the sensed data and attempts to identify robust and error-agnostic features. If random placement is unavoidable and dense deployment cannot be supported, robust and error-agnostic features enable one to recognize interesting events from erroneous or imperfect data.

### *1.1. Energy Model*

Any strategy that aims to enhance the quality of sensed data should also consider the energy cost it introduces into the network. Energy is a crucial and scarce resource in wireless sensor networks. In most cases, it is costly to recharge or replace batteries, and how energy is consumed can directly affect the scope and usefulness of the network. For example, Chintalapudi et al. [9] deploy a wireless sensor network inside a four-storey office building to monitor the response of the building to a forced excitation. Likewise, Kim et al. [11] deploy a wireless sensor network on the Golden Gate Bridge to monitor the response of the structure to ambient excitations (movement of vehicles and wind). In both cases, the researchers' field observations suggest that such networks are most useful for intermittent monitoring, because of the significant power consumption during aggressive oversampling, which was needed in order to compensate for high packet loss rates (an average packet loss rate of 30% was observed in one of the deployments setting). Subsequently, it is vital to bear in mind the scarcity of this resource when a communication protocol is developed.

As one of the contributions of this work, a comprehensive and realistic mathematical model of a fully operational wireless sensor network is developed. While work already exists on the modelling of the energy consumption of wireless sensor networks, much of the focus has been on the link and network layers [8, 14]. To the best knowledge of this author, this is the first comprehensive model that takes aspects of the physical, link, network, and application layers into account to fully quantify the energy cost of an operational network. The model takes toxic gas detection in an oil refinery as a scenario, and defines the sensing task as a combination of periodic and event-based reporting. The energy model, however, is by no means limited to toxic

gas detection. It can be used to characterize and examine any event-based, query-based, or periodic-sensing application.

As a periodic-sensing application, the network periodically reports the concentration of  $\text{H}_2\text{S}$  and  $\text{NH}_3$  to a remote control station. As an event-based application, aggressive and repetitive reporting is supported when the concentrations of these gases exceed certain thresholds. A flat topology in which all of the nodes play the same role, i.e. both as sensing and as relaying nodes, is assumed. The model enables to estimate the lifetime of such networks; and the energy budget can be broken down into different concerns. The model takes into account various aspects, including the network density, the duty cycle, the size of the contention window of the media access control (MAC) protocol, the average number of intermediate hops, the message diffusion and gradient creation strategy, and the number of leakage sources.

Moreover, the model enables to express and estimate the cost of network management – self-organization, neighbour discovery, and periodic schedule synchronization – as a separate concern from the sensing task (i.e., the sensing, processing, and transfer of interesting events). As an example, the model shows that synchronization of sleeping schedules and periodic neighbour discovery in hybrid MAC protocols costs a disproportionate amount of energy. This assertion is in agreement with the current implementation of TinyOS<sup>1</sup>, the most widely used runtime environment in wireless sensor networks. TinyOS implements B-MAC [18], which is based on an adaptive preamble, instead of S-MAC [24], which, even though highly referenced in the literature, is based on asynchronous sleeping schedule.

### *1.2. Topology Control*

The random deployment assumption requires a robust and energy-efficient topology control protocol to ensure that the network is connected regardless of the way the nodes are distributed in the network. At the same time, it is also important to ensure that nodes in the network have a fair share of the packet forwarding and data aggregation task so that they exhaust their energy in a uniform manner – this prevents the network from fragmenting prematurely. In contrast, most existing flat topology networks function based on either flooding or gossiping strategies. While highly flexible, routing (topology) protocols based on these strategies are known for their energy

---

<sup>1</sup>As of November 24, 2009, the current version is TinyOS 2.1.0. Source: <http://tinyos.net/scoop/section/Releases>.

consumption. The topology control protocol (1) sets a limit on the maximum number of neighbouring nodes with which a node should collaborate, and (2) computes the optimal transmission distance to ensure that the network is connected.

The topology protocol proposed in this work is based on the energy model described in Section 1.1 and the work of Bhardwaj et al. [4]. Bhardwaj et al. develop an optimal multi-hop path that maximizes network lifetime. Each node in the network establishes a link with a node that is a *characteristic distance*,  $d_{char}$ , away from it. This distance is the most energy-efficient transmission distance and it is calculated by taking several parameters (the channel's path-loss index and the power consumption of the hardware components of the transceiver) into account. However, the model makes two essential assumptions: (1) there are intermediate nodes that are placed on the straight line that connects a source node with the sink node; and (2) there is always a relaying node that is a distance  $d_{char}$  away from a data forwarding node. These two assumptions are difficult to satisfy. Therefore, our proposed topology control defines eligibility metric for nodes that are placed away from the straight line that connects a source node with the sink node, and which may not satisfy the  $d_{char}$  assumption to participate in establishing a multi-hop link. The features of the topology control protocol are summarized as follows:

1. it takes the limitation of a node's hardware (the transceiver) and some of the characteristics of the channel into account to define the optimal communication distance;
2. it aims to optimize the energy consumption of any arbitrary multi-hop link based on local knowledge, i.e., knowledge about neighbours; and
3. it defines eligibility metric to ensure that intermediate nodes are fairly selected, so as to guarantee a uniform energy consumption throughout the network.

The performance of the topology control protocol was evaluated with respect to node degree, robustness, graph stretch factor, normalized energy dissipation as a function of node density, and variance in energy reserve. Compared to graph-based topology control protocols (based on the K-neighbourhood, relative neighbourhood graph, or Gabriel graph), the topology control protocol performs particularly well in that it allows nodes to exhaust their energy in a more uniform manner. This saves the network from fragmenting prematurely.

### 1.3. Signal Processing

The problem of random deployment cannot be solved by simply designing robust and energy-efficient communication protocols. Considering the resource scarcity in wireless sensor nodes and the harsh environments in which the networks operate, data may still be corrupted or lost, and nodes may be displaced from their initial position or suffer from thermal agitations. All of these factors contribute to the delivery of imperfect data. In the end, the applications that consume the raw sensor data should decide how to deal with imperfect sensing.

This work aims to complement communication protocols with effective and efficient signal processing strategies. Its main contributions are a thorough investigation into error-agnostic time and frequency domain features for processing data from acoustic and accelerometer sensors. A close scrutiny into the proposed applications reveals that acoustic and accelerometer sensors are frequently employed in wireless sensor networks. For example, acoustic sensors are used in human activity recognition [10], active volcano monitoring [23], traffic automation [3], and pipeline monitoring [20]. Likewise, accelerometer sensors are used in healthcare applications (Parkinson’s disease and epilepsy) [13, 12], structural health monitoring [11], and supply-chain management [16].

The investigation is categorized into feature extraction and scene recognition. In the first case, the aim is to investigate the expressive power of a large number of time and frequency domain features. Data from accelerometer sensors representing the movements of people (slow acceleration) and cars (fast acceleration) are investigated. Both quantitative and qualitative analyses are performed to examine the existence of correlation between measurements representing the same type of events (movements) under different deployment settings. The investigation reveals that while most frequency domain features are error-agnostic, time domain features such as correlation coefficients are expressive if absolute acceleration values are considered. Moreover, these types of features can be extracted locally. Nevertheless, the analysis shows that features representing slow movements are in general are very sensitive to deployment errors.

In the second case, the main aim is to investigate different types of pre-processing configurations during feature extraction – sampling rate, sampling duration (sample size), and frame overlap during short time Fourier transformation. Activity recognition (context recognition) is taken as an application

in which Bayesian networks and Hidden Markov models are used as recognition schemes. These schemes take as inputs the features extracted from acoustic sensors to estimate higher-level human activities.

## 2. Organization

This compilation consists of five journal articles. The paper by Dargie et al. (DAR09A) presents the results of the analytic model for the energy cost discussed in section 1.1; it also establishes the foundation for the topology control protocol presented in section 1.2.

The paper by Dargie et al. (DAR09B) presents the design and implementation of the topology control protocol. This paper is an extended version of the paper that was presented at AINA 2009 conference (DAR09E).

The last three papers focus on the signal processing aspects discussed in section 1.3. The paper by Dargie and Denko (DAR09C) focuses on the analysis of error-agnostic feature extraction. The paper by Dargie (DAR09D) presents a detailed account of the pre-processing configurations during feature extraction, and the trade-off between scene (context) recognition and processing time. Finally, the paper by Dargie and Tersch (DAR08A) presents the recognition accuracy associated with a large number of everyday human activities based on frequency domain features extracted from acoustic sensors.

### 2.1. Contribution

It is very hard to draw a line that distinguishes my own contribution from the contributions of the other co-authors in this compiled work. The papers are results of intensive team work that extends over a period of three years. Most of the co-authors were graduate students who wrote their theses under my supervision. Therefore, only a high-level description of my contribution can be provided.

In (DAR09A), I was responsible for defining the sensing task for toxic gas detection, the topology of the network, and the communication protocols. Moreover, I was responsible for the link-layer mathematical model. I produced more than 50% of the paper.

In (DAR09B), my main contribution was in defining the scope and usefulness of the topology control protocol and in providing a comprehensive analysis of the state-of-the-art. I also collaborated with Mr Mochaourab in

developing the eligibility metric for the newly developed topology control protocol. I contributed more than 30% of the paper's content.

In (DAR09C), my main contribution was the selection of the time and frequency domain features and the definition of the experimental setting. I also participated during the analysis of the measurements. The accelerometer data were collected by Mr Robert Krüger. I contributed more than 70% of the paper's content.

In (DAR09D), my contribution was the development of the adaptive context-recognition architecture and the implementation of some of its components. Mr Daniel Hofmann contributed by implementing the recognition engine and by gathering and analysing the acoustic data.

In (DAR08A), my contribution was the design and implementation of the conceptual framework for context recognition. Mr Tobias Tersch contributed by setting up the Bayesian network and by computing the conditional probabilities. I contributed about 60% of the paper's content.

## References

- [1] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury. A survey on wireless multimedia sensor networks. *Comput. Netw.*, 51(4):921–960, 2007.
- [2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Comput. Netw.*, 38(4):393–422, 2002.
- [3] A. Arora, P. Dutta, S. Bapat, V. Kulathumani, H. Zhang, V. Naik, V. Mittal, H. Cao, M. Demirbas, M. Gouda, Y. Choi, T. Herman, S. Kulkarni, U. Arumugam, M. Nesterenko, A. Vora, and M. Miyashita. A line in the sand: a wireless sensor network for target detection, classification, and tracking. *Comput. Netw.*, 46(5):605–634, 2004.
- [4] M. Bhardwaj, T. Garnett, and A. Chandrakasan. Upper bounds on the lifetime of sensor networks. In *IEEE International Conference on Communications (ICC 2001)*, volume 3, pages 785–790, 2001.
- [5] G. Bianchi. Performance analysis of the ieee 802.11 distributed coordination function. *IEEE Journal on Selected Areas in Communications*, 18(3):535–547, 2000.

- [6] J. Burrell, T. Brooke, and R. Beckwith. Vineyard computing: sensor networks in agricultural production. *IEEE Pervasive Computing*, 3(1):38–45, 2004.
- [7] A. Camilli, C. E. Cugnasca, A. M. Saraiva, A. R. Hirakawa, and P. L. P. Corrêa. From wireless sensors to field mapping: Anatomy of an application for precision agriculture. *Comput. Electron. Agric.*, 58(1):25–36, 2007.
- [8] C.-F. Chiasserini and M. Garetto. An analytical model for wireless sensor networks with sleeping nodes. *IEEE Transactions on Mobile Computing*, 5(12):1706–1718, 2006.
- [9] K. Chintalapudi, T. Fu, J. Paek, N. Kothari, S. Rangwala, J. Caffrey, R. Govindan, E. Johnson, and S. Masri. Monitoring civil structures with a wireless sensor network. *IEEE Internet Computing*, 10(2):26–34, 2006.
- [10] C. Hermann and W. Dargie. Senceive: A middleware for a wireless sensor network. In *AINA '08: Proceedings of the 22nd International Conference on Advanced Information Networking and Applications*. IEEE Computer Society, Washington, DC, USA, 2008, pages 612–619.
- [11] S. Kim, S. Pakzad, D. Culler, J. Demmel, G. Fenves, S. Glaser, and M. Turon. Health monitoring of civil infrastructures using wireless sensor networks. In *IPSN '07: Proceedings of the 6th International Conference on Information Processing in Sensor Networks*. ACM, New York, NY, USA, 2007, pages 254–263.
- [12] S. Kumar, K. Kambhatla, F. Hu, M. Lifson, and Y. Xiao. Ubiquitous computing for remote cardiac patient monitoring: a survey. *Int. J. Telemedicine Appl.*, 2008:1–19, 2008.
- [13] K. Lorincz, B.-r. Chen, G. W. Challen, A. R. Chowdhury, S. Patel, P. Bonato, and M. Welsh. Mercury: a wearable sensor network platform for high-fidelity motion analysis. In *SenSys '09: Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*. ACM, New York, NY, USA, 2009, pages 183–196.
- [14] R. Madan, S. Cui, S. Lall, and A. J. Goldsmith. Modeling and optimization of transmission schemes in energy-constrained wireless sensor networks. *IEEE/ACM Trans. Netw.*, 15(6):1359–1372, 2007.



- [15] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson. Wireless sensor networks for habitat monitoring. In *ACM International Workshop on Wireless Sensor Networks and Applications (WSNA 2002)*, 2002, pages 88–97.
- [16] M. Malinowski, M. Moskwa, M. Feldmeier, M. Laibowitz, and J. A. Paradiso. Cargonet: a low-cost micropower sensor node exploiting quasi-passive wakeup for adaptive asynchronous monitoring of exceptional events. In *SenSys '07: Proceedings of the 5th International Conference on Embedded Networked Sensor Systems*. ACM, New York, NY, USA, 2007, pages 145–159.
- [17] M. Ndoih and G. Delisle. Geolocation in underground mines using wireless sensor networks. In *Antennas and Propagation Society International Symposium*, 2005, pages 229–232.
- [18] J. Polastre, J. Hill, and D. Culler. Versatile low power media access for wireless sensor networks. In *SenSys '04: Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*. ACM, New York, NY, USA, 2004, pages 95–107.
- [19] E. I. Shih, A. H. Shoeb, and J. V. Guttag. Sensor selection for energy-efficient ambulatory medical monitoring. In *MobiSys '09: Proceedings of the 7th International Conference on Mobile Systems, Applications, and Services*. ACM, New York, NY, USA, 2009, pages 347–358.
- [20] I. Stoianov, L. Nachman, S. Madden, and T. Tokmouline. Pipenet: a wireless sensor network for pipeline monitoring. In *IPSN '07: Proceedings of the 6th International Conference on Information Processing in Sensor Networks*. ACM, New York, NY, USA, 2007, pages 264–273.
- [21] U. Varshney. Pervasive healthcare and wireless health monitoring. *Mob. Netw. Appl.*, 12(2–3):113–127, 2007.
- [22] M. C. Vuran and I. F. Akyildiz. Spatial correlation-based collaborative medium access control in wireless sensor networks. *IEEE/ACM Trans. Netw.*, 14(2):316–329, 2006.
- [23] G. Werner-Allen, K. Lorincz, M. Welsh, O. Marcillo, J. Johnson, M. Ruiz, and J. Lees. Deploying a wireless sensor network on an active volcano. *IEEE Internet Computing*, 10(2):18–25, 2006.

- [24] W. Ye, J. Heidemann, and D. Estrin. Medium access control with coordinated adaptive sleeping for wireless sensor networks. *IEEE/ACM Trans. Netw.*, 12(3):493–506, 2004.
- [25] J. Zhao and R. Govindan. Understanding packet delivery performance in dense wireless sensor networks. In *SenSys '03: Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*. ACM, New York, NY, USA, 2003, pages 1–13.
- [26] X. Chao, W. Dargie, and L. Guan. Energy Model for H2S Monitoring Wireless Sensor Network. *CSE '08: Proceedings of the 2008 11th IEEE International Conference on Computational Science and Engineering*. IEEE Computer Society, Washington, DC, USA, 2008, pages 402–409.

## Own Contribution

- DAR08A W. Dargie and T. Tersch. Recognition of complex settings by aggregating atomic scenes. *IEEE Intelligent Systems*, 23(5):58–65, 2008. DOI=<http://dx.doi.org/10.1109/MIS.2008.90>.
- DAR09A W. Dargie, C. Xiaojuan, and M. K. Denko. Modelling the energy cost of a fully operational wireless sensor network. *Springer Journal of Telecommunication Systems*, Preprint, 2009. DOI=<http://dx.doi.org/10.1007/s11235-009-9228-z>.
- DAR09B W. Dargie, A. Schill, R. Mochaourab, and L. Guan. A topology control protocol based on eligibility and efficiency metrics. *The Journal of Systems and Software (Elsevier)*, submitted September 2009.
- DAR09C W. Dargie and M. K. Denko. Analysis of error-agnostic time and frequency domain features extracted from measurements of 3D accelerometer sensors. *IEEE Systems Journal*, accepted 12 November 2009.
- DAR09D W. Dargie. Adaptive audio-based context recognition. *IEEE Transactions on System, Man, and Cybernetics, Part A*, 39(4):715–725, 2009. DOI=<http://dx.doi.org/10.1109/TSMCA.2009.2015676>.
- DAR09E W. Dargie, R. Mochaourab, A. Schill, and L. Guan. A topology control protocol for 2D Poisson distributed wireless sensor networks. In *Proceedings of the 2009 International Conference on Advanced Information Networking and Applications Workshops*, May 26–29, 2009. WAINA. IEEE Computer Society, Washington, DC, 2009, pages 582–587. DOI=<http://dx.doi.org/10.1109/WAINA.2009.63>.

# Modelling the energy cost of a fully operational wireless sensor network

Waltenegus Dargie · Xiaojuan Chao · Mieso K. Denko

© Springer Science+Business Media, LLC 2009

**Abstract** Several applications have been proposed for wireless sensor networks, including habitat monitoring, structural health monitoring, pipeline monitoring, precision agriculture, active volcano monitoring, and many more. The energy consumption of these applications is a critical feasibility metric that defines the scope and usefulness of wireless sensor networks. This paper provides a comprehensive energy model for a fully functional wireless sensor network. While the model uses toxic gas detection in oil refineries as an example application, it can easily be generalized. The model provides a sufficient insight about the energy demand of the existing or proposed communication protocols.

**Keywords** Wireless sensor networks · Energy-model · Energy-efficient protocols · Lifetime of a wireless sensor network

## 1 Introduction

Several applications have been proposed for wireless sensor networks. The application of Mainwaring et al. [12] gathers

data from humidity, temperature, barometric pressure, and light sensors for monitoring the activities of seabirds. Kim et al. [10] use wireless sensor networks for structural health monitoring, in which the structural integrity of bridges and buildings is inspected using accelerometer sensors. The networks are tasked with measuring the response of a structure to an ambient excitation (heavy wind or passing vehicles) or a forced shake (using shakers or impact hammers). The application of Werner-Allan et al. [17] monitors active volcano using seismic and infrasonic sensors. The underlying network was able to capture 230 volcano events just over three weeks. The application of Stoianov et al. [15] uses hydraulic and acoustic/vibration sensors for monitoring large diameter, bulk-water transmission pipelines.

The most prevalent concern in wireless sensor networks is the limited lifetime. The nodes operate with exhaustible batteries; and recharging or replacing these batteries, given the sheer size of the network and the deployment settings, is a significant hurdle. For example, because of the energy constraint, Kim et al. [10] suggest that wireless sensor networks can only be used during occasional inspection of bridges and buildings, thereby limiting their scope as well as usefulness. Subsequently, almost all types of communication protocols and data processing algorithms target efficient use of energy and optimization of network lifetime as their design goal.

In this paper, we carefully analyse the energy cost of a fully operational wireless sensor network. The application we use for our analysis will be toxic gas detection in oil refineries. We will consider highly referenced, energy-aware protocols for establishing and running the network. We shall give particular consideration to the link and network layer as well as to the self-organization (neighbor discovery and interest dissemination) aspects, as these claim a significant portion of the energy budget. Finally, we shall provide comprehensive analytic and simulation models based on which

---

W. Dargie (✉)  
Chair of Computer Networks, Technical University of Dresden,  
01187 Dresden, Germany  
e-mail: [waltenegus.dargie@tu-dresden.de](mailto:waltenegus.dargie@tu-dresden.de)

X. Chao  
9th Floor Tower B, CEC Plaza, NO. 3 Dan Ling Street,  
Hai Dian District, Beijing 100080, China  
e-mail: [chaoxj@gmail.com](mailto:chaoxj@gmail.com)

M.K. Denko  
Department of Computing and Information Science,  
University of Guelph, Guelph, Ontario, Canada, N1G 2W1  
e-mail: [denko@cis.uoguelph.ca](mailto:denko@cis.uoguelph.ca)

the lifetime of the network can be estimated. The models take into account node density, distributed sleeping schedules, multi-hop communication and time synchronization.

The rest of this paper is organized as follows: In Sect. 2, we discuss related work; in Sect. 3 we will briefly discuss toxic gas detection in refineries; in Sect. 4, we will establish basic assumptions for the network model; in Sects. 5 and 6, we will provide a comprehensive analysis and simulation of the energy cost. Finally in Sect. 7, we will discuss our experiences and observations and provide concluding remarks and outline for future work.

## 2 Related work

Tseng [16] provide an analytic energy model for estimating the energy consumption of a wireless sensor network that employs the S-MAC medium access control protocol [18]. The model takes the cost of control messages (RTS/CTS/ACK/DIFS) and the duty cycle of the sleeping schedule of individual nodes into account. The model attempts to define and estimate the energy consumption of various operation modes. In [19], an analytic, integrated data-link layer model is presented. The model enables to estimate the energy cost of link layer protocols. The strength of the model is in its capability to give insight about the effect of a link layer decision on other layer concerns, including channel assignment, rate of transmission, power and management. However, the framework does not offer a comprehensive understanding of the energy cost of the entire network.

Feeney [6] propose an analytical model for examining the energy cost of routing in a mobile ad hoc networks. The work attempts to demonstrate the trade-off between energy consumption and reliability. Two popular routing protocols are chosen for the analysis: Dynamic Source Routing (DSR) [13] and Ad hoc On-demand Distance Vector (AODV) [11]. These two protocols support routing in flat topology networks, with all nodes participating equally in the routing process. Moreover, both protocols are on-demand protocols, in which nodes discover and maintain routes as needed. DSR heavily depends on the cache of network wide topology information extracted from source routing headers, while AODV is a destination-oriented protocol based on the distributed Bellman-Ford algorithm. Both protocols are adaptive for dynamic topology. Similar to other energy models, the network interface has four possible energy consumption states: transmitting, receiving, idle, and sleeping. The idle mode is the default mode for ad hoc environment. The energy cost is calculated as a function of packet size. The unit energy of a packet is decided by the sender, the intended receiver(s), and the nodes overhearing the message. Chao et al. [4] report an initial result of this work, but its mathematical model was not fully developed.

## 3 Toxic gas detection

The application we use to analyse the energy cost of the communication protocols in wireless sensor networks is a toxic gas detection application in oil refineries. There are two reasons for choosing toxic gas detection: (1) Oil refineries cover extensive areas, requiring large scale sensing to detect oil and gas leakages in pipelines. This fits into the basic assumption that a wireless sensor network is made up of hundreds and thousands of wireless sensing nodes. (2) Presently, a good portion of the oil industry is replacing cable based sensing systems by portable and wireless devices which can easily be deployed and maintained. The next evolution in toxic gas detection will be towards wireless sensor networks. For the detail description of the various toxic gases that should be sensed, we refer our readers to [4] and [5].

## 4 Network model

Our analysis and simulation of the network's energy consumption and lifetime is based on a network model. The network model establishes the basic assumptions concerning the network's topology, the distribution and density of nodes, and the way the network is connected. Moreover, it defines the network's sensing task.

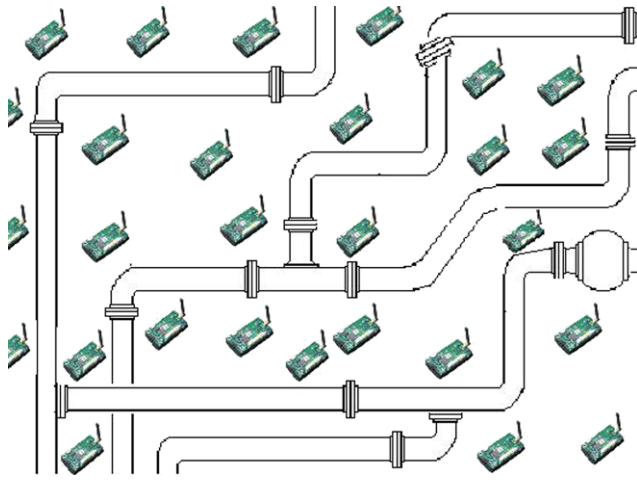
Deployment refers to the way wireless sensor nodes are placed in areas where the sensing task should be carried out. This decision directly affects the quality of sensing as well as the overall energy consumption of the entire network. While there can be three basic monitoring strategies—spot, area and fence—for toxic gas monitoring, spot monitoring is the most suitable strategy [4].

Coverage is another significant performance metric. In [2], it indicates how well a given area can be monitored by the network. Even though there are some existing models for estimating the number of sensors required to cover the entire sensing field with a probability,  $p$ , of detection an event, coverage is deployment dependent. For a spot monitoring scenario, even though the whole area is not necessarily covered, all potential leakage sources are monitored.

As shown in Fig. 1,  $N$  nodes are distributed randomly on a rectangular area  $A$  of size  $A = a \times b$ . Without loss of generality, we assume that  $a \leq b$ . The node distribution can be modelled as a two-dimensional Poisson distribution with average density,  $\lambda$ . The probability of finding  $k$  nodes in  $A$  is given by:

$$P(k \text{ nodes} \in A) = e^{-\lambda A} \frac{(\lambda A)^k}{k!} \quad (1)$$

The connectivity figure speaks about the existence of a communication link between a source anywhere in the network and a single sink. In multi-hop communication, there



**Fig. 1** 2D Poisson distributed node deployment

is at least one multi-hop path between a node and the sink (base station). The probability that a network is connected, i.e., all nodes can communicate with the sink either directly or with the support of intermediate nodes, mainly depends on the node density and the transmission range of individual nodes. If the nodes are assumed to be homogeneous, the relationship between connectivity probability, transmission range and node density is estimated by<sup>1</sup> [3]:

$$p(\text{conn}) \cong (1 - e^{-\lambda \pi r_0^2})^n \quad (2)$$

where  $p(\text{conn})$  is the probability that the network is connected;  $\lambda$  is the density of the network,  $n/A$ ;  $r_0$  is the threshold transmission range; and  $n \gg 1$  is the number of deployed nodes. The deployment scenario for our case is depicted in Fig. 1. The spot monitoring strategy is complemented by additional randomly deployed nodes for improved connectivity. Each node has the same radio transmission range  $R$ , and two nodes can communicate via a wireless link if their Euclidean distance is less than the transmission range, i.e.,  $d \leq R$ . For simplification, fading and path efficiency are not taken into account; we do not consider also the presence of obstacles in the path of propagation.

Finally, the sensing task for which the network is deployed determines the data traffic size in the network. For toxic gas detection, there are two essential concerns: the long and short term impact of toxic gases release. Hence, every sensor node should periodically (a tunable parameter) report the concentration of  $\text{H}_2\text{S}$  and  $\text{NH}_3$  to a sink. This is defined as a normal case with a normal priority. In case of a leakage that surpasses a threshold defined by the safety board of the refinery (this is usually a concentration between 10 and 15 ppm, an alarm should be fired off within 30 seconds. This is characterized as an abnormal condition with high priority.

<sup>1</sup>This is without taking the border effect into account.

## 5 Energy model

In Sect. 4, we presented a number of factors that affect the quality of sensing and the lifetime of a wireless sensor network. In this section, we shall translate those factors into quantifiable terms so that we can estimate the energy cost. The model together with the sensing task description, and the specification of the hardware devices and the communication protocols will be sufficient to estimate the lifetime.

The communication protocols we employ to establish the wireless sensor network are the S-MAC [18], for medium access control, and the Directed Diffusion [8], for supporting self-organization and routing. The justification for these protocols is given in more detail elsewhere [4]. A more technical assessment of these protocols can be found in [20] and [21].

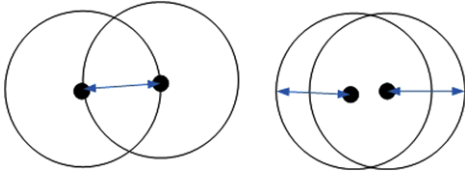
Hop count is an essential performance parameter and indicates how many hops a packet is relayed in average for a given distance in a network. For a deterministic topology, this hop-count estimation is a simple geometry problem. For a random network model, however, a combination of statistics and probability theory is required. Fortunately, there are many existing models already. To calculate the minimum hop count, we determine the distance  $S$  between two random nodes and divide it by the transmission range  $R$ . In the literature, the random distance formula [1] is widely adopted. It is based on the calculation of the random distance distribution within a rectangular area:

$$\begin{aligned} E\{S\} = & \frac{1}{15} \left[ \frac{a^3}{b^2} + \frac{b^3}{a^2} + \sqrt{a^2 + b^2} \left( 3 - \frac{a^2}{b^2} - \frac{b^2}{a^2} \right) \right] \\ & + \frac{1}{6} \left[ \frac{b^2}{a} \ln \left( \frac{a + \sqrt{a^2 + b^2}}{b} \right) \right. \\ & \left. + \frac{a^2}{b} \ln \left( \frac{b + \sqrt{a^2 + b^2}}{a} \right) \right] \end{aligned} \quad (3)$$

Taking  $E(S)$  as the expected distance between the source and the destination in our random network, the lower bound of the expected value of hop count can be expressed as:

$$H_{\min} = E\{S\}/R \quad (4)$$

To achieve a more realistic analysis, transmission error due to packet loss and collision should be included in the energy model. Because the listening time in S-MAC is fixed, a fixed contention window is better for coordination and synchronization than an exponential back-off. However, a fixed contention window can cause significant packet loss. We used Bianchi's model [7] to estimate packet loss due to collision at the link layer. According to the model, the prob-



**Fig. 2** Possible intersections of two neighbor nodes

ability of successful transmission,  $p_{succ}$ , can be calculated as:

$$p_{succ} = \frac{(\lambda - 1)\tau(1 - \tau)^{\lambda-2}}{1 - (1 - \tau)^{\lambda-1}} \quad (5)$$

where  $\lambda$  refers to the network's node density and  $\tau = \frac{2}{(CW+1)}$  and  $CW$  is the carrier sensing contention window.

All data packets in S-MAC, except interest dissemination, are unicast and will cause *RTS/CTS/ACK* control overhead. To estimate the energy cost of adaptive listening, it is useful to estimate the number of neighbors which potentially overhear the *RTS/CTS* message, i.e., the average neighbors in enclosure of the sender and the receiver. This can be calculated by first getting the overlaps of communication coverage between two random neighbors. When two nodes becomes neighbors, their transmission circles intersect, in which case Fig. 2 shows the two extreme scenarios. The intersection area can be described as [14]:

$$2R^2 \cos^{-1} \left( \frac{d}{2R} \right) - \frac{1}{2}d\sqrt{4R^2 - d^2} \quad (6)$$

where  $0 \leq d \leq R$  and  $d$  is the Euclidean distance between two nodes. Taking the assumption of the random Poisson distribution of the nodes into account, then  $d$  is bound in  $(0, R)$  with a uniform probability distribution. Accordingly,  $d$  will be:

$$d = \frac{R}{2} \quad (7)$$

Then the average overlap of two circles can be described by:

$$A_{intersec} \approx 2.152R^2 \quad (8)$$

The enclosure area for neighbors of a sender or a receiver is

$$A_1 = 2\pi R^2 - 2.152R^2 \approx 4.131R^2 \quad (9)$$

with

$$\lambda = \frac{N \times \pi R^2}{a \times b} \quad (10)$$

And,

$$N_{neighb} = \frac{A_1}{a \times b} \times N \approx \frac{A_1}{\pi R^2} \times \lambda = 1.314\lambda \quad (11)$$

## 5.1 Energy consumption analytic model

For a thorough analysis of the energy model (from (12) to (65)), the variables (parameters) listed in Table 1 and their corresponding descriptions should be referred to. Additional variables will be explained according to their context of use.

We propose two analytic models to estimate the energy consumption of a toxic gas detection network. We call the first model Pure Synchronization Energy Model (PSE) and the other Full Application Energy Model (FAE). In PSE, there will not be data transmission in the network; nodes communicate with each other to perform synchronization (i.e., exchanging sleeping schedules). Most existing S-MAC based energy models assume that the whole network is synchronized without actually considering the energy consumed by the synchronization process. We present the PSE model to provide a realistic picture of the contribution of time synchronization on the overall energy consumption. In the simulation section, we shall demonstrate that synchronization and periodical neighbor discovery cost more energy than data transmission. FAE models a fully functional network in which both periodical and incidental data transmission and time synchronization are taking place.

### 5.1.1 Pure synchronization energy model

S-MAC carries out time synchronization in 4 steps: In the first step, every node is initially active for  $sync_p$  cycles, waiting for the arrival of SYNC packet from other nodes. The energy consumption of this phase is expressed as:

$$E_{listen\_sync} = N \times sync_p \times T_{frame} \times P_{idle} \quad (12)$$

In the second step, nodes periodically resynchronize to avoid clock drift. During this time,  $t$ , the number of attempts every node sends SYNC packet is expressed as,

$$N_{sync\_sent\_try} = (t / (sync_p \times T_{frame}) - sync_p) \quad (13)$$

Due to packet collision and loss, only a portion of these packets will be successfully received:

$$N_{sync\_sent} = N_{sync\_sent\_try} \times p_{succ} \quad (14)$$

The energy consumed during sending SYNC packets at this stage is given by:

$$E_{sync\_per\_node\_sent} = E_{trans} + E_{idle} + E_{sleep} \quad (15)$$

where

$$E_{trans} = M_{sync} / R_{data\_rate} \times P_{trans} \quad (16)$$

**Table 1** Variables definition

Variables	Definition
$P_{trans}, P_{sleep}, P_{recv}, P_{idle}$	Energy consumption per time unit of four modes
$M_{sync}, M_{RTS}, M_{CTS},$ $M_{ACK}, M_{interest}, M_{data}$	Size of SYNC, RTS, CTS, ACK, Interest, and data Message
$t_{sync\_cw}, t_{data\_cw}$	Size of contention window of SYNC and Data Message
$t_{backoff}, t_{DIFS}, t_{SIFS}$	Size of backoff, DCF, Inter Frame Space and Short Inter Frame Space
$t_{idle}$	Idle period of every transmission/reception pair of one data packet
$t_{adapt}$	Adaptation time, frame length dependent
$I_{normal}, I_{abnormal}$	Interest propagation frequency for Normal and Abnormal case
$i_{normal}, i_{abnormal}$	Normal and Abnormal event report interval
$d_{report\_abnormal}$	Report duration after a leak is detected
$R_{data\_rate}$	Data rate
$duty\_cycle$	Duty cycle
$T_{frame}$	Frame length
$p_{succ}$	Probability of successful packet transmission/reception
$R_{retry}$	Max retry times
$f_{srch\_cycle}$	Frequency of neighbor Discovery
$sync_p$	The initial Synchronization period
$N_{neigh}$	Average neighbors in enclosure of sender and receiver
$\lambda$	Network density
$N$	Total number of nodes in an area, $A$
$N_{leak}$	The number of nodes that detected leakage
$H_{min}$	Minimum hop count (Topology dependent)

And,

$$E_{idle} = P_{idle} \times (T_{frame} \times duty\_cycle - M_{sync}/R_{data\_rate}) \quad (17)$$

$$E_{sleep} = P_{sleep} \times T_{frame} \times (1 - duty\_cycle) \quad (18)$$

$$E_{sync\_per\_node\_recv} = (1 - \lambda) \times (E_{recv} + E_{idle\_recv} + E_{sleep\_recv}) \quad (19)$$

The energy consumed during receiving the SYNC packets is expressed as,

$$E_{recv} = (M_{sync}/R_{data\_rate} \times P_{recv}) \quad (20)$$

When a SYNC packet arrives at a receiving node, it either succeeds or fails due to collision or channel error. Since a failure reception consumes the same amount of energy as a successfully received packet, we merge both scenarios together. In other words, all  $(\lambda - 1)$  neighbor nodes will receive the SYNC packet regardless of its usefulness. Then

total amount of energy consumed during periodical SYNC packet sending and receiving is therefore calculated as:

$$\begin{aligned} E_{period\_sync\_pure} &= N_{sync\_sent\_try} \times N \times E_{sync\_per\_node\_sent} \\ &+ N_{sync\_sent} \times N \times E_{sync\_per\_node\_recv} \end{aligned} \quad (21)$$

In the third step, every node periodically performs neighbor discovery by listening for the whole  $sync_p$  cycles as described in the first stage, i.e., for every  $sync_p \times f_{srch\_cycle}$  cycles. Note, however, that not all node enter into neighbor discover phase at the same time since those nodes that lose during contention for channel access will compete only in the next contention cycle, after sending a SYNC packet. Thus the periodical neighbor discovery will be delayed due to collision.



$$E_{nb\_srch} = N \times sync_p \times p_{succ} \times T_{frame} \times P_{idle} \times \frac{t}{T_{frame} \times sync_p \times f_{srch\_cycle}} \quad (22)$$

Finally, to calculate the energy consumption of transmitting empty frames,<sup>2</sup> first, we find out the number of empty frames.

$$N_{empty} = (t/T_{frame} - sync_p - (t \times p_{succ}) / (T_{frame} \times f_{srch\_cycle}) - N_{sync\_sent\_try} + N_{sync\_sent} \times (\lambda - 1)) \times N \quad (23)$$

Here  $t/T_{frame}$  gives the total number of frames per node for the period,  $t$ .  $sync_p$  is the probable duration of the initial synchronization time in which a node waits for SYNC packet from other nodes.  $\frac{t}{(T_{frame} \times f_{srch\_cycle})}$  is the neighbor discovery frames; and  $N_{sync\_sent} \times \lambda$  expresses the number of frames for sending and transmitting periodical SYNC packets. Subsequently, the expected energy consumption for synchronization is expressed as:

$$E_{empty} = N_{empty} \times T_{frame} \times (P_{idle} \times duty\_cycle + P_{sleep} \times (1 - duty\_cycle)) \quad (24)$$

The energy consumption for SYNC overhead without data transmission is given by:

$$E_{sync\_pure} = E_{listen\_sync} + E_{period\_sync\_pure} + E_{nb\_srch} + E_{empty} \quad (25)$$

## 5.2 Full application energy model

In this model, the energy model contains two parts: the energy consumption due to synchronization and the energy consumption due to data transmission. The four stages of synchronization discussed in Sect. 5.1.1 apply for the Full Application Energy Model as well. The amount of energy consumed during listening for Sync packets and neighbor discovery is the same as in the previous case. However, even though both SYNC and data packets can be processed in the same frame, in stage 2 of the synchronization stage, we calculated only the energy for sending/receiving SYNC packets. The energy consumed during the remaining time can be accounted for data transmission or receiving; or for idly listening. Suppose  $N_{sync\_sent}$  is the number of times every node sends SYNC packets successfully and  $N_{sync\_sent\_try}$  is the number of times a node broadcasts SYNC packets.

<sup>2</sup>Here we define empty frames as frames that contain scheduled idle time only. In these frames, we need only calculate the energy consumed during idle time.

The energy consumption during sending and receiving every SYNC packet is given as follows:

$$E_{sync\_per\_node\_sent} = \frac{M_{sync}}{R_{data\_rate}} \times P_{trans} + P_{idle} \times (t_{sync\_cw} + t_{backoff} + t_{DIFS}) \quad (26)$$

$$E_{sync\_per\_node\_recv} = (\lambda - 1) \times M_{sync} / R_{data\_rate} \times P_{recv} \quad (27)$$

Because of the reason stated in step 2 of the PSE model, we merge both scenarios together.

$$E_{period\_sync} = N_{sync\_sent\_try} \times N \times E_{sync\_per\_node\_sent} + N_{sync\_sent} \times N \times E_{sync\_per\_node\_recv} \quad (28)$$

Unlike stage 4 in Pure Synchronization Model, the empty frames in this scenario are both data and SYNC packets dependent; thus the total energy consumed during synchronization is given as follows:

$$E_{sync} = E_{listen\_sync} + E_{period\_sync} + E_{nb\_srch} \quad (29)$$

Given the average neighbors in enclosure of two nodes, the number of neighbors that overhear an RTS/CTS message can be known.<sup>3</sup> We use the  $p_{succ} \times N_{neigh}$  to denote the number of nodes that will join adaptive listening. The energy consumption due to nodes participating in an Adaptive Listening is given by:

$$E_{adapt} = (p_{succ} \times N_{neigh} \times t_{adapt} + t_{idle}) \times P_{idle} \quad (30)$$

In One-phase pull of the Directed Diffusion routing protocol, there are no exploratory and reinforcement overheads. One only needs to calculate the cost of flooding interest and data transmission. In the Interest propagation phase, the sink periodically sends interest to all nodes. The duration of a period is relatively large. The successful transmission and retransmission rate are described in Table 2.

$$\alpha = p_{succ} + (1 - p_{succ}) \times p_{succ} \times 2 + (1 - p_{succ})^2 \times p_{succ} \times 3 + \dots + (1 - p_{succ})^{R_{retry}-1} \times p_{succ} \times R_{retry} \quad (31)$$

Based on the back off behavior of S-MAC, for every transmission/reception pair of one data packet, the idle period can be described as follows.

$$t_{idle} = t_{data\_cw} + t_{backoff} + t_{DIFS} + 3t_{SIFS} \quad (32)$$

<sup>3</sup>There are also nodes that may not be able to hear an RTS/CTS message.

**Table 2** Transmission times of RTS

Number of transmission times	Send/Resend Possibility	Send Times of $M_{RTS}$
1	$p_{succ}$	1
2	$(1 - p_{succ}) \times p_{succ}$	2
3	$(1 - p_{succ})^2 \times p_{succ}$	3
$R_{retry}$	$(1 - p_{succ})^{(R_{retry}-1)} \times p_{succ}$	$R_{retry}$

When a node finishes transmitting/receiving a packet, the remaining time may not always fit to the scheduled active and sleep time of the node, in which case the node has to keep idle until the next active or sleep time arrives. Since we already take the active period in one frame into account, the extra idle time can be estimated by:  $(1 - \text{duty\_cycle}) \times \frac{T_{frame}}{2}$ . Accordingly, the energy consumption of interest propagation can be expressed as:

$$E_{interest\_per\_node} = E_{useful} + E_{waste} \quad (33)$$

where

$$E_{useful} = M_{interest}/R_{data\_rate} \times (P_{trans} + p_{succ} \times (\lambda - 1) \times P_{recv}) \quad (34)$$

And,

$$E_{waste} = (t_{DIFS} + t_{data\_cw} + t_{backoff}) \times P_{idle} + (1 + (\lambda - 1) \times p_{succ}) \times (P_{idle} + P_{sleep}) \times 1 - \text{duty\_cycle} \times \frac{T_{frame}}{2} \quad (35)$$

Every interest packet is successfully transmitted with a probability of  $p_{succ}$ . This holds true for both normal and abnormal conditions.

$$E_{normal\_set} = E_{abnormal\_set} = N \times E_{interest\_per\_node} \quad (36)$$

During a reporting phase, we have either a normal event or an abnormal event. During a normal report, the  $H_2S$  concentration is below 10 ppm. We first calculate the energy consumption of a single event delivery path. Every packet along a single path will be received  $H_{min}$  times. It will be forwarded to the next hop if the concentration is larger than the max value in memory of the current node. The possibility of every intermediate packet being successfully forwarded is assumed to be 0.5. Thus,

$$E_{normal\_report\_one\_path} = H_{min} \times (E_{OH} + E_{trans} + E_{waste}) \quad (37)$$

where

$$E_{OH} = (M_{data} + M_{RTS} + M_{CTS} + M_{ACK}) / R_{data\_rate} \times P_{recv} \quad (38)$$

And,

$$E_{tran} = 0.5 \times (M_{data} + M_{RTS} \times \alpha + M_{CTS} + M_{ACK}) / R_{data\_rate} \times P_{trans} \quad (39)$$

$$E_{waste} = 0.5 \times E_{adapt} \times \alpha + (P_{idle} + P_{sleep}) \times (1 - \text{duty\_cycle}) \times \frac{T_{frame}}{2} \quad (40)$$

The energy consumption during an abnormal case can be calculated in a similar way. Based on the result of the two phases above, we derive the energy consumption by  $N$  nodes during time,  $t$ . This includes the energy consumption of interest propagation phase and reporting phase:

$$E_{routing\_normal} = \frac{t}{I_{normal}} \times E_{normal\_set} + \frac{t}{i_{normal}} \times E_{normal\_report\_one\_path} \times N \quad (41)$$

Similarly, the energy consumption for the abnormal case is expressed as follows:

$$E_{routing\_abnormal} = \frac{t}{I_{abnormal}} \times E_{abnormal\_set} + \frac{d_{report\_abnormal}}{i_{abnormal}} \times E_{abnormal\_report\_one\_path} \times N_{leak} \quad (42)$$

Here  $\frac{d_{report\_abnormal}}{i_{abnormal}}$  refers to the number of messages that a leakage event keeps on reporting.

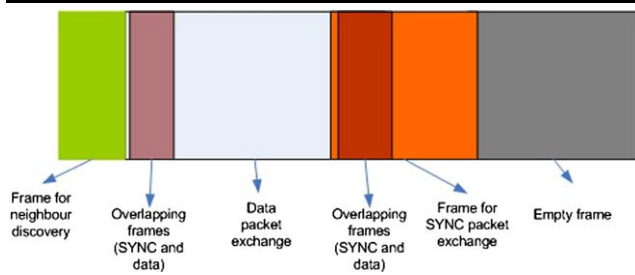
### 5.2.1 Energy for empty frames and missed part

As mentioned before, we now express the energy consumed in idle time of empty frames that are not used during synchronization or data transmission/reception during time  $t$ . To estimate the number of empty frames, we calculate the number of frames occupied in routing and synchronization based on the analysis above. From the interest propagation phase, we get the number of frames for both cases:

$$N_{normal\_set} = N_{abnormal\_set} = N \times \lambda \quad (43)$$

The reporting frame is defined as

$$N_{normal\_one\_path} = (1 + 0.5 \times \alpha) \times H_{min} \quad (44)$$



**Fig. 3** Ratio among number of frames

$$N_{\text{abnormal\_one\_path}} = (1 + \alpha) \times H_{\min} \quad (45)$$

Now with the above intermediate calculation, we derive the number of frames required for data exchange:

$$N_{\text{data}} = F_{\text{comb}} + F_{\text{normal}} + F_{\text{ab}} \quad (46)$$

where,

$$F_{\text{comb}} = \frac{t}{I_{\text{normal}}} \times N_{\text{normal\_set}} + \frac{t}{I_{\text{abnormal}}} \times N_{\text{abnormal\_set}} \quad (47)$$

And,

$$F_{\text{normal}} = \frac{t}{i_{\text{normal}}} \times (N - 1) \times N_{\text{normal\_one\_path}} \quad (48)$$

$$F_{\text{abnormal}} = \frac{d_{\text{report\_abnormal}}}{i_{\text{abnormal}}} \times N_{\text{leak}} \times N_{\text{abnormal\_one\_path}} \quad (49)$$

Then with the number of frames for neighbor discovery and SYNC packets exchanged, we compute at the  $n$  synchronization stage, the number of frames for synchronization:

$$N_{\text{sync\_neighbor\_discovery}} = \left( \text{sync}_p + \frac{t}{(T_{\text{frame}} \times f_{\text{srch\_cycle}})} \times p_{\text{succ}} \right) \times N \quad (50)$$

$$N_{\text{sync\_exchange}} = (N_{\text{sync\_sent\_try}} + N_{\text{sync\_sent}} \times (\lambda - 1)) \times N \quad (51)$$

$$N_{\text{sync}} = N_{\text{sync\_neighbor\_discovery}} + N_{\text{sync\_exchange}} \quad (52)$$

Figure 3 shows the ratio among a number of frames used for synchronization, data, or empty  $\varphi_1$  and  $\varphi_2$  are the number of frames that handle both SYNC and data packets. While  $\varphi_1$  represents the overlap between frames of data and neighbor discovery (the whole frame is in idle state), and  $\varphi_2$  denotes the overlap between frames of data and common SYNC packets exchange.

It is difficult to precisely determine how many frames a node uses for both SYNC and data transmitting/receiving.

We divide the intersection between data and synchronization into sub periods as  $\varphi_1$  and  $\varphi_2$  to decrease the uncertainty. By adding frames on both payload and synchronization, we can estimate the total frames produced by a node. This is expressed as follows:

$$N_{\text{work}} = N_{\text{data}} + N_{\text{sync}} - \varphi_1 - \varphi_2 \quad (53)$$

The total number of frames communicated during  $t$  is:

$$N_{\text{total}} = t / T_{\text{frame}} \times N \quad (54)$$

So the number of empty frames can be calculated by subtracting  $N_{\text{work}}$  from  $N_{\text{total}}$ .

$$N_{\text{empty}} = N_{\text{total}} - N_{\text{data}} - N_{\text{sync\_neighbor\_discovery}} - N_{\text{sync\_exchange}} + \varphi_1 + \varphi_2 \quad (55)$$

Accordingly, we get the energy consumed by empty frames:

$$E_{\text{empty}} = N_{\text{empty}} \times T_{\text{frame}} \times (\text{duty\_cycle} \times P_{\text{idle}} + (1 - \text{duty\_cycle}) \times P_{\text{sleep}}) \quad (56)$$

As we mentioned before, we only calculate the energy for SYNC packets exchange till now, we need to add the missing part here. From the Fig. 3, the  $N_{\text{sync\_miss}}$  can be calculated as follows:

$$N_{\text{sync\_miss}} = N_{\text{sync\_exchange}} - \varphi_2 \quad (57)$$

Thus

$$E_{\text{sync\_miss}} = (N_{\text{sync\_exchange}} - \varphi_2) \times (P_{\text{idle}} \times (T_{\text{frame}} \times \text{duty\_cycle} - M_{\text{sync}} / R_{\text{data\_rate}}) + P_{\text{sleep}} \times T_{\text{frame}} \times (1 - \text{duty\_cycle})) \quad (58)$$

If we add  $E_{\text{empty}}$  and  $E_{\text{sync\_miss}}$ , we can get

$$E_{\text{empty}} + E_{\text{sync\_miss}} = N \times T_{\text{frame}} \times N_{\text{sleep\_idle}} - (N_{\text{sync\_exchange}} - \varphi_2) \times (P_{\text{idle}} \times M_{\text{sync}} / R_{\text{data\_rate}}) \quad (59)$$

With

$$N = N_{\text{total}} - N_{\text{data}} - N_{\text{sync\_neighbor\_discovery}} + \varphi_1 \quad (60)$$

And,

$$N_{\text{sleep\_idle}} = \text{duty\_cycle} \times P_{\text{idle}} + (1 - \text{duty\_cycle}) \times P_{\text{sleep}} \quad (61)$$

$$\varphi_1 \in [0, \min(N_{\text{data}}, N_{\text{sync\_neighbor\_discovery}})] \quad (62)$$

$$\varphi_2 \in [0, \min(N_{\text{data}}, N_{\text{sync\_exchange}})] \quad (63)$$

Based on  $\varphi_1$  and  $\varphi_2$ 's range, we could derive the upper bound and lower bound of the sum of  $E_{empty}$  and  $E_{sync\_miss}$ . The distribution of  $\varphi_1$  and  $\varphi_2$  can be assumed by a Binomial distribution with probability of 0.5, and with a mean value of:

$$\varphi_1 = 0.5 \times \min(N_{data}, N_{sync\_neighbor\_discovery}) \quad (64)$$

$$\varphi_2 = 0.5 \times \min(N_{data}, N_{sync\_exchange}) \quad (65)$$

And finally, taking all intermediate results into consideration, the overall energy consumption of the network can be summed up as follows:

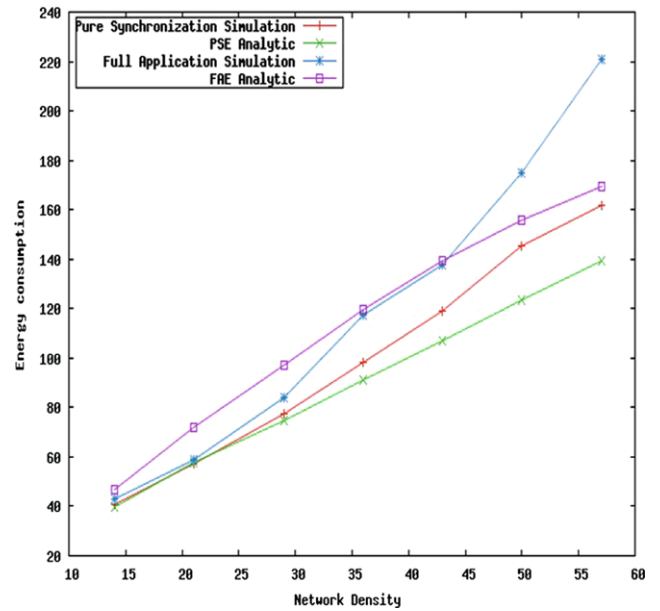
$$E_{total} = E_{routing\_normal} + E_{routing\_abnormal} + E_{sync} + E_{empty} + E_{sync\_miss} \quad (66)$$

## 6 Energy analysis

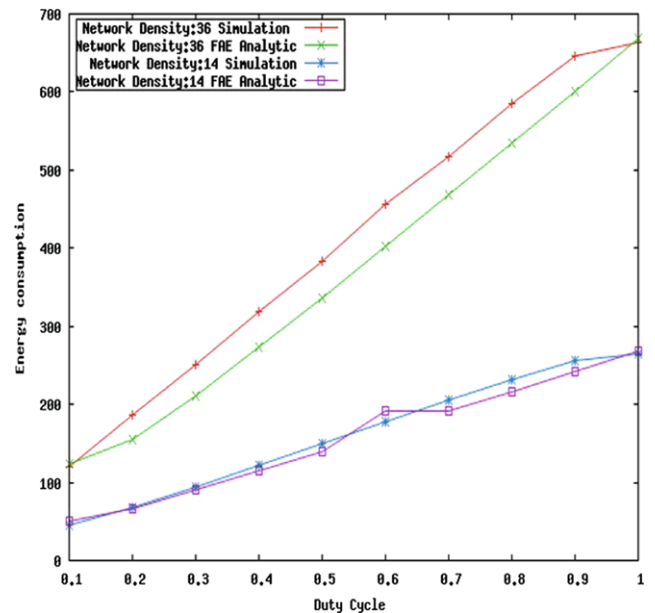
The simulation environment we use is the NS-2 simulator, version 2.31 [9]. Our simulation model combines S-MAC and the one-phase-pull algorithm in Directed Diffusion. In the S-MAC protocol, we enable the adaptive listening and global schedule functionalities. The default duty cycle is set at 10 percent, and the data rate is 2 Mbps for the message sizes we proposed in Sect. 5, the S-MAC frame length will be 1.31 seconds with 10% duty cycle. Error encoding ratio is set at 2, as specified by the default setting in S-MAC. The data message size is 136 bytes and interest size is 96 bytes. We set the interest refresh time as 300 seconds and changed the ping application to report normal data once in 600 seconds, the event generation time is randomly selected. For every abnormal event, it generates 6 abnormal messages repeatedly within 10 s.

We use the topology of randomly distributed nodes in an area of 100 m  $\times$  70 m. One of these nodes is specified as the sink node. The simulation duration is 600 seconds. All the other parameter values are described in Table 3.

We change the network density and compare the energy consumption for both the PSE and FAE models. There is a linear relationship between the density and energy consumption (Fig. 4). The analytic result for both PSE and FAE models is remarkably similar to the simulation results, for density below 45. The small deviation in the energy consumption of the two scenarios illustrates that the synchronization cost is high when S-MAC is used. There are two reasons for this: (1) S-MAC repeatedly uses SYNC packets to synchronize the local timer and discover new neighbors during the entire lifetime; and (2) A node relentlessly attempts to send out a broadcast SYNC packet even if it loses a contention. For a high density networks, efficient packet transmission can be achieved by tuning parameters such as the event generation interval and the interest propagation duration.



**Fig. 4** Energy consumption increases when network density becomes higher



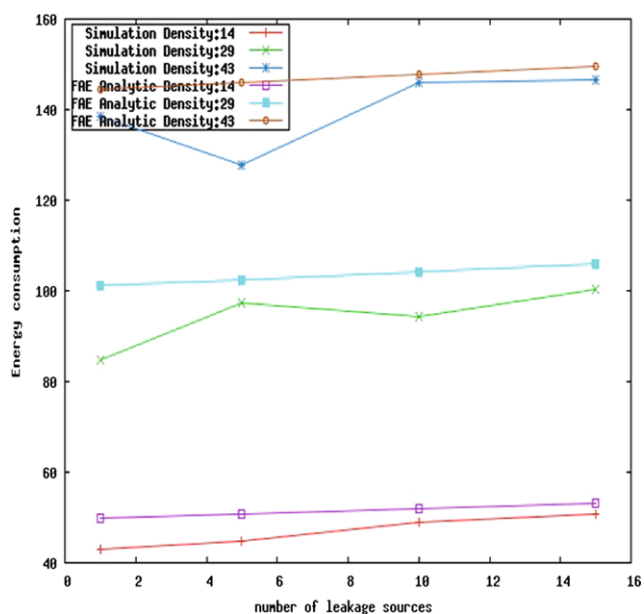
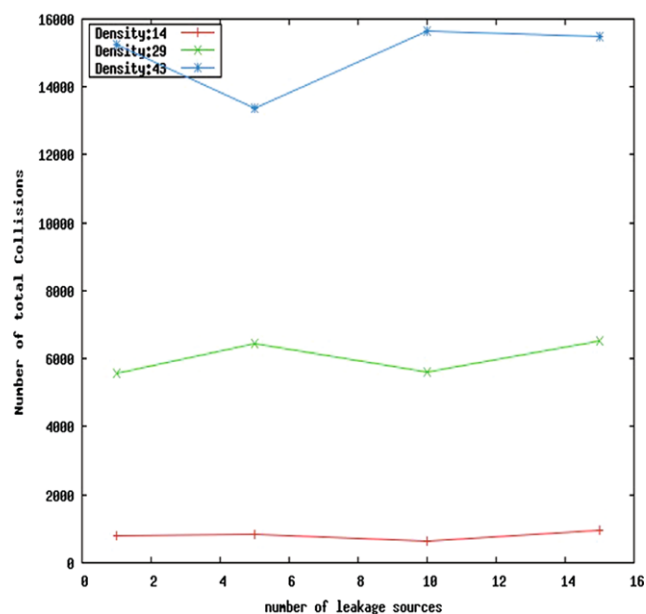
**Fig. 5** Energy consumption as duty cycle changes

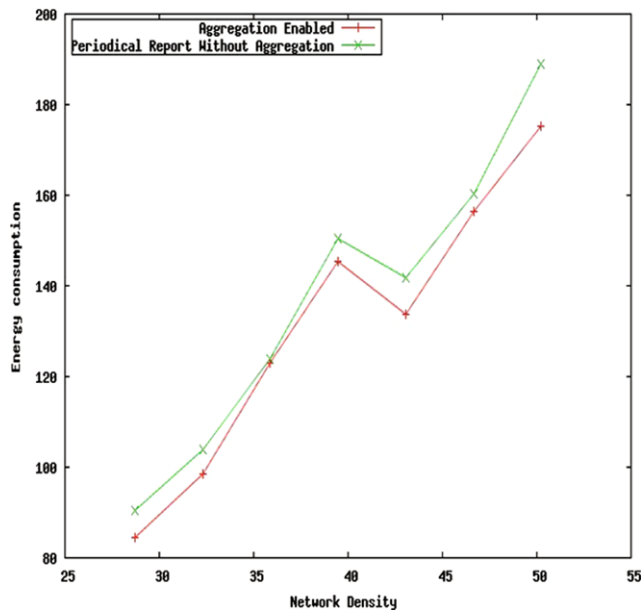
### 6.1 Model validation and duty cycle

Figure 5 shows how energy consumption can be affected by the duty cycle of the MAC protocol. We varied the duty cycle of two different network densities: 14 and 35. The analytic results of the FAE model are similar to the simulation results with the deviation of less than 10% for both densities. The energy consumption increases due to the additional active time as well as collision and synchronization overhead. Because in S-MAC the listen time is fixed, when duty cycle

**Table 3** Simulation parameter list

Basic parameter	Default value
Control message RTS/CTS/ACK	10 bytes
SYNC message	9 bytes
Interest message	96 bytes
Data message	136 bytes
Interest propagation frequency	300 seconds
Normal event report interval	300 seconds
Abnormal event report interval	10 seconds
Abnormal event report period	60 seconds
Abnormal event occurrence ratio	1%
Duty cycle	10%
Bandwidth	2 Mbps
Network density	$\lambda$
Minimum hop counts	Topology dependent
S-MAC Frame length	Message size, duty cycle and Backoff Window
Adaptation time	Frame length dependent
Max retry times	5
Frequency of neighbor Discovery	22
Synchronization period	10
Nominal transmission Range	40 m
Sensing field	7000 m <sup>2</sup> (70 m $\times$ 100 m)
Transmission power	31.2 mW
Receive/idle power	22.2 mW
Radio@sleep status	3 $\mu$ W

**Fig. 6** Energy consumptions varies along with the leakage sources**Fig. 7** Collision varies as number of leakage sources increases



**Fig. 8** Data aggregation impact on energy consumption

varies, the frame length will vary adversely. Therefore when the duty cycle increases, the whole time will be divided into more frames, which will result in SYNC packet overhead increase. This in turn affects SYNC packets broadcasting interval and the neighbor discovery, both of which are frame size dependent.

Figure 6 denotes the relationship between energy consumption and the number of leakage sources. We simulated with three different densities: 14, 29, 43. When the leakage source increases from 1 to 15 with an increasing step of 5, the energy consumption rises in steps, but there is anomalous reduction in the simulation curves. The anomalous reduction becomes more obvious when network density increases. When abnormal events dominate data transmission for a certain period of time, the synchronization as well as neighbor discovery will be delayed, and SYNC packet collisions will be reduced temporarily and eventually results in a transient energy decrease. The relationship between collision and the number of leakage sources is shown in Fig. 7. In Fig. 6 and Fig. 7, these curves reveal similar behavior. In Fig. 6, though our analytical result approaches the simulation result, the analytical energy consumption raises only slightly in a liner fashion, without any anomalous point. This is because energy consumption in the analytical model is more ideally calculated. Though it considers the collision possibility in a statistical way, the collisions with other network behaviors such as synchronization and message queuing were difficult combine.

One way of reducing the data traffic in the network is by forwarding a report from a node only if the maximum  $H_2S$  and  $NH_3$  concentrations it reports is greater than all the other nodes in its neighborhood. This requires data aggregation,

but it reduces the traffic in the entire network significantly. Figure 8 depicts the considerable energy saving under all network densities.

## 7 Conclusions

Both in the analysis and simulation case, as the density of the network increases, the energy utilization of the network increases also. One reason for this is that in a large density networks, the power consumption of each node at the link layer is significantly high due to collision. S-MAC begins applying the sleep schedule for each node only once the nodes have exchanged their schedule. Synchronization claims a significant amount of energy. The disproportional energy distribution even during normal sensing makes S-MAC unsuitable for toxic gas detection. Moreover, during simulation, we have observed that S-MAC's performance deteriorates considerably when the number of nodes in the sensing field exceeded 40. The Bianchi model for computing the energy cost during contention assumes saturation traffic, in which all the nodes have data to send at all times. While this is plausible for normal, periodic reports, it is unsuitable for irregular and bursty traffics. The energy cost of normal and abnormal events propagation decreases exponentially as the interest propagation interval increases. Interest has to be disseminated in the network to update routing paths and to define a new sensing task. Interest dissemination prompts gradient computation and reinforcement. The longer the interval, the lower the energy cost. On the other hand, choosing a long interest propagation interval implies a potential increase in latency of event propagation, since old paths might be broken for a number of reasons, as such is the case when some nodes exhaust their energy more quickly than others. There is a trade-off between latency and energy cost.

In our energy model, we have not considered the energy required for local signal processing, such as the energy consumed by the analog-to-digital (ADC) converter to produce a high resolution sensor data. In reality, however, the ADC consumes a significant amount of power. In the future, we will accommodate this fact to assess the feasibility of using existing off-the-shelf hardware for building wireless sensor networks.

## References

1. Ghosh, B. (1951). Random distances within a rectangle and between two rectangles. *Bulletin of the Calcutta Mathematical Society*, 4.
2. Boukerche, A., Fei, X., & Araujo, R. B. (2006). An energy-efficient sensing coverage protocol for surveillance and monitoring applications using wireless sensors. In *Performance, computing, and communications conference, 2006. IPCCC 2006. 25th IEEE international* (pp. 611–616). April 2006.



3. Bettstetter, C. (2002). On the connectivity of wireless multihop networks with homogeneous and inhomogeneous range assignment. In *IEEE vehicular technology conference, VTC 2002* (pp. 1706–1710).
4. Chao, X., Dargie, W., & Lin, G. (2008). Energy model for h2s monitoring wireless sensor network. In *CSE '08: proceedings of the 2008 11th IEEE international conference on computational science and engineering* (pp. 402–409). Washington: IEEE Computer Society.
5. Dargie, W., Schill, A., Mochaourab, R., & Guan, L. (2009). A topology control protocol for 2d Poisson distributed wireless sensor networks. In *The third international workshop on telecommunication networking, applications and systems*.
6. Feeney, L. M. (2001). An energy consumption model for performance analysis of routing protocols for mobile ad hoc networks. *Mobile Networks and Applications*, 6(3), 239–249.
7. Bianchi, G. (2000). Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE Selected Areas in Communications*, 18, 535–547.
8. Intanagonwiwat, C., Govindan, R., Estrin, D., Heidemann, J., & Silva, F. (2003). Directed diffusion for wireless sensor networking. *IEEE/ACM Transactions on Networking*, 11(1), 2–16.
9. Issariyakul, T., & Hossain, E. (2008). Signal & communication. In *Introduction to network simulator NS2*. Berlin: Springer.
10. Kim, S., Pakzad, S., Culler, D., Demmel, J., Fennes, G., Glaser, S., & Turon, M. (2007). Health monitoring of civil infrastructures using wireless sensor networks. In *IPSN '07: proceedings of the 6th international conference on information processing in sensor networks* (pp. 254–263). New York: ACM.
11. Lee, S.-J., Belding-Royer, E. M., & Perkins, C. E. (2003). Scalability study of the ad hoc on-demand distance vector routing protocol. *International Journal of Network Management*, 13(2), 97–114.
12. Mainwaring, A., Culler, D., Polastre, J., Szewczyk, R., & Anderson, J. (2002). Wireless sensor networks for habitat monitoring. In *ACM international workshop on wireless sensor networks and applications (WSNA 2002)* (pp. 88–97).
13. Nasipuri, A., Casta, N. R., & Das, S. R. (2001). Performance of multipath routing for on-demand protocols in mobile ad hoc networks. *Mobile Networks and Applications*, 6(4), 339–349.
14. Spivak, M. (2006). *Calculus* (3rd ed.). Cambridge: Cambridge University Press.
15. Stoianov, I., Nachman, L., Madden, S., & Tokmouline, T. (2007). Pipenet: a wireless sensor network for pipeline monitoring. In *IPSN '07: Proceedings of the 6th international conference on information processing in sensor networks* (pp. 264–273). New York: ACM.
16. Wei Tseng, H., Yang, S.-H., Chuang, P.-Y., Wu, H.-K., & Chen, G.-H. (2004). An energy consumption analytic model for a wireless sensor mac protocol. In *Vehicular technology conference* (pp. 4533–4537).
17. Werner-Allen, G., Lorincz, K., Welsh, M., Marcillo, O., Johnson, J., Ruiz, M., & Lees, J. (2006). Deploying a wireless sensor network on an active volcano. *IEEE Internet Computing*, 10(2), 18–25.
18. Ye, W., Heidemann, J., & Estrin, D. (2002). An energy-efficient mac protocol for wireless sensor networks. In *Infocom* (pp. 1567–1576).
19. Zhong, L. C. (2004). *A unified data-link energy model for wireless sensor networks*. PhD thesis, Chair-Jan M. Rabaey.
20. Zimmerling, M., Dargie, W., & Reason, J. (2007). Energy-efficient routing in linear wireless sensor networks. In *The fourth IEEE international conference on mobile ad-hoc and sensor systems*.
21. Zimmerling, M., Dargie, W., & Reason, J. M. (2008). Localized power-aware routing in linear wireless sensor networks. In *CASEMANS '08: Proceedings of the 2nd ACM international conference*

on context-awareness for self-managing systems (pp. 24–33). New York: ACM.



**Walteneus Dargie** obtained a Ph.D. in Computer Engineering from the Technical University of Dresden, Germany (2006), a M.Sc. degree in Electrical Engineering from the Technical University of Kaiserslautern, Germany (2002) and a B.Sc. in Electrical and Electronics Technology from the Nazareth Technical College, Ethiopia (1997). Prior to his current position, he has been working as a researcher at the University of Kassel, Germany (2002–2005) and at the Fraunhofer Institute of Experimental Software Engineering, Germany (2002–2003). His research interests include digital signal processing, wireless and mobile networks, wireless sensor networks, and pervasive computing. Presently, Dr. Dargie serves as a Guest Editor to the International Journal of Autonomous and Adaptive Communication Systems (IJAAACS) and to the Journal of Computer and System Science (JCSS). He is also a chair or co-chair of a number of IEEE and ACM workshops, including the CASEMANS 2009 (ACM) and PMECT 2009 (IEEE). Dr. Dargie is the member of IEEE.



**Xiaojuan Chao** Xiaojuan Chao is a graduate of the Technical University of Dresden, from which she obtained a M.Sc. degree in Computational Engineering in 2007. Her research interests include distributed systems, web software architecture, wireless communications and wireless sensor networks. She is now focusing on Internet technologies, Project Management and Architecture. Chao has more than 9 years of experience in software design and system architecture.



**Mieso K. Denko** received his M.Sc. degree from the University of Wales, UK, and his Ph.D. degree from the University of Natal, South Africa, both in Computer Science. Currently, he is with the Department of Computing and Information Science, University of Guelph, Guelph, Ontario, Canada. His current research interests include wireless networks, mobile and pervasive computing, wireless mesh networks, body sensor networks and network security. Dr. Denko is a founder/co-founder of a number of ongoing international workshops and served as program chair/co-chair of a number of IEEE/ACM international conferences. Currently he is serving as guest co-editor of Special Issues for a number of journals including the ACM/Springer Mobile Networks and Applications (MONET) and IEEE Systems Journal. Dr. Denko has co-edited two books in the areas of pervasive computing and wireless networking, and currently co-editing two forthcoming books, autonomic computing and networking

with Springer and Pervasive Computing and Networking with Wiley. He is editorial board member of international journals including, the International Journal of Smart Homes (IJSH), the Journal of Ubiquitous Computing and Communications, (UBICC), and Associate Editor

of the International Journal of Communication Systems (IJCS), Wiley, Security & Communications Network (SCN), Wiley and the Journal of Ambient Intelligence and Humanized Computing, Springer. He is a senior member of the ACM and IEEE.



# A Topology Control Protocol based on Eligibility and Efficiency Metrics

Waltenegus Dargie<sup>a</sup>, Rami Mochaourab<sup>b</sup>, Alexander Schill<sup>a</sup>, Lin Guan<sup>c</sup>

<sup>a</sup>*Chair of Computer Networks, Department of Computer Science, Technical University of Dresden, 01062, Dresden, Germany*

<sup>b</sup>*Chair of Theoretical Communication Engineering, Technical University of Dresden, 01062 Dresden, Germany*

<sup>c</sup>*Department of Computer Science, Loughborough University, Loughborough, Leicestershire, LE11 3TU, UK*

---

## Abstract

The question of fairness in wireless sensor networks is not studied very well. It is not unusual to observe in the literature fairness traded for low latency or reliability. However, a disproportional use of some critical nodes as relaying nodes can cause premature network fragmentation. This paper investigates fairness in multi-hop wireless sensor networks and proposes a topology control protocol that enables nodes to exhaust their energy fairly. Moreover, it demonstrates that whereas the number of neighboring nodes with which a node should cooperate depends on the density of the network, increasing this number beyond a certain amount does not contribute to network connectivity.

---

## 1. Introduction

In wireless sensor networks, communication (receiving as well as transmitting) consumes a significant amount of energy. Since routing involves several nodes, its energy cost outweighs the cost of data processing. As to the exact number of nodes that should participate in a routing task, so far the research community is not in agreement. There are those who argue that multi-hop communication is preferred over single hop communication. One of the premises for this assumption is that as the distance of communication increases, the probability of getting a line-of-sight (LOS) link decreases, in which case the path loss index can no longer be assumed to be 2 but between 2 and 4, and in some cases, even 6. By reducing the distance of communica-

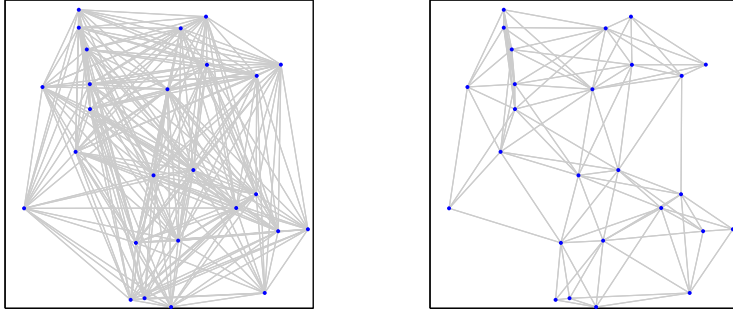


Figure 1: A topology control protocol trims off inefficient links from a disk graph  $\mathcal{G}$  (left) to produce an optimal topology  $\mathcal{T}$  (right).

tion to a shorter length, it is possible to keep a LOS link, which significantly reduces the transmission cost.

On the other hand, there are those (for example Ephremides [7] and Haenggi [9]) who argue that this is an oversimplified analysis that does not take into account the cost of routing overhead, delay, channel coding/decoding, end-to-end reliability, efficiency of transmission power amplifiers, etc., and advocate long-hop routing. For densely and randomly deployed wireless sensor networks (such as in pipelines with several turns in short distances), short-hop routing is quite unsuitable. Apparently, long distance communication has also its disadvantages besides path loss, including interference.

A topology control protocol is necessary to set an upper and lower bound on the number of links that can be active in the network. This ensures that the network remains connected and its lifetime is optimized. Moreover, it guarantees an available link to a higher-level routing protocol that is defined based on an application-specific metric (such as minimum hop, minimum delay, minimum energy consumption, and maximum available power). Figure 1 displays how a topology control protocol can be employed to trim off inefficient links in a wireless (sensor) network.

In wired networks, the way the network elements are physically interconnected directly influences the network's topology. Routing protocols take this fact into account when routes are computed. In wireless networks, however, as long as the communication range suffices, essentially all nodes can establish a link with each other, creating a mesh-topology network [7], which is not energy efficient. Another problem is that during the operation of the net-

work, some nodes may exhaust their energy more rapidly than others while others may become dysfunctional. A topology control protocol deals with all these dynamics and ensures that the network is connected with energy efficient links.

The main challenge is to develop a topology control strategy that is simple, scalable, and less resource intensive. Ideally, it should function based on local information only. In most cases, additional knowledge such as the placement and relative position of a node to the sink node can be obtained from layout information or from blueprints; and can be used to determine relative neighborhood. We propose a localized algorithm that enables nodes to autonomously create and maintain energy-efficient links. The protocol defines proximity and eligibility metrics to ensure network connectivity and to optimize lifetime.

The paper is organized as follows: First, we discuss related work in section 2. In section 3, the network model and the basic assumptions of the model are outlined. In section 4, the theoretical concept of the fair and efficient topology control (FETC) protocol is presented. In section 5, the eligibility criterion for ensuring a fair utilization of energy in wireless sensor networks is discussed. In section 6, the algorithm for executing the topology control protocol is presented. In section 7, a brief summary of the mathematical descriptions and algorithms of the protocols that are used for comparisons are presented. In section 8, the simulation settings and results are discussed. Finally, in section 9, we provide concluding remarks and future work.

## 2. Related Work

Most existing approaches to topology control apply computational geometry techniques and proximity graphs to build sparse, but connected links.

Bhardwaj et al. [18], provide a model for computing the most energy efficient number of hops to relay data from any source in a linear topology network to a fixed base station. The number of hops depends on a *characteristic distance* and the distance of the source to the base station. The *characteristic distance* itself depends on the propagation environment and radio parameters. We extend this approach to support random deployment in a 2-dimensional plain. Jeng et al. [12] use Neighborhood Graphs to compute adjustable neighborhood regions and to optimize the node degree. A similar work that optimizes a node degree is proposed in [20] - their constructed graph is a subgraph of the Relative Neighborhood Graph [11] and

the protocol uses local information (signal strength information). In both cases, fairness in energy dissipation is not addressed.

Wattenhofer et al. [19] propose a topology control protocol to dynamically adjust transmission power based on local decisions. Accordingly, a node increases its transmission power until it finds a neighbor node in every direction. But the question how a node trims off inefficient links in case it discovers several neighbors is not addressed.

The topology control protocol of Kung et al. [13] selects suitable communication nodes, adjusts service loads of critical nodes, and manages sleeping schedules. The protocol principally divides the topology operation into topology formation phase and topology adjustment phases. In the topology formation phase, a link is set up while during the topology adjustment phase, the links are adjusted with an optimal balance of critical nodes in the backbone.

Our strategy is different from the strategies above in the following specific features:

1. It takes the limitations of a node's hardware (the transceiver) and channel characteristics into account to compute the most optimal communication distance;
2. It aims to optimize the energy consumption of any arbitrary multi-hop link based on local knowledge, i.e., knowledge about neighbors; and,
3. Defines an eligibility metric to ensure that relying nodes are fairly selected. This guarantees a fairly uniform energy consumption throughout the network.

### 3. Network Model

Given a flat topology network<sup>1</sup> of  $n$  nodes placed randomly in the Euclidean plane, let  $V$  be the set of vertices representing the nodes and  $E$  be the set of undirected edges representing the communication links between them. The graph of the network is denoted as  $G = (V, E)$ . In addition, let  $g_{digraph}$  represent the digraph<sup>2</sup> of the network with  $E_{digraph}$ , the set of undirected edges.

---

<sup>1</sup>In a flat topology network, all nodes play the same roles, both as sensing and as relaying nodes.

<sup>2</sup>A digraph is a graph with directed edges.

Each node  $i$ ,  $i \in V$ , has a unique identity,  $id_i$ , and is represented in the Euclidian plane with its coordinates. A directed edge between two nodes  $i$  and  $j$  is denoted as  $[i \rightarrow j]$ ,  $[i \rightarrow j] \in E_{digraph}$ , and has a distance of  $d(i, j)$ . An undirected edge between  $i$  and  $j$  is denoted as  $[i \leftrightarrow j]$ ,  $[i \leftrightarrow j] \in E$ . This paper assumes a random distribution of the nodes in a wide rectangular field of deployment. One way to model this type of deployment is by using a 2-dimensional Poisson point process [4]. The points are equally likely to occur anywhere within a bounded region  $A$ , and the probability of finding  $n$  nodes in  $A$  is given as:

$$Pr[n \text{ nodes in } A] = e^{-\lambda} \cdot \frac{(\lambda \cdot A)^n}{n!} \quad (1)$$

where  $\lambda$  is the Poisson process density which is related to the density of the network. The set of neighbors of  $i$ , with which  $i$  is directly connected are denoted as the set  $N(i)$  and defined as  $N(i) : [i \leftrightarrow j] \in E_{digraph}$ . Let  $N_L(i)$  be the neighbor table list in which the state of each  $i$  in  $N(i)$  is stored.  $N_L(i)$  contains the identity, energy reserve, eligibility parameters, and required transmission power to reach each neighbor. Each node has a maximum transmission power of  $P_{t-max}$  and can assign varying transmission powers corresponding to each neighboring node. The transmission power from node  $i$  to  $j$  is denoted as  $P_{t-ij}$ . The residual energy of a node  $i$  at time  $t$  is denoted as  $e_i^t$ . Furthermore, all nodes start with equal initial battery capacity  $E$ .

Communication in the network takes place over a wireless medium in which the transmitted signal experiences an attenuation over distance. Moreover, during propagation, the electromagnetic waves experience losses in the form of reflection, diffraction, and scattering. The received signal power, in general, decays as a power law function of the distance separating the transmitter and the receiver. Thus, the received signal power can be written as:

$$P_{rx} \propto \frac{P_{tx}}{d^\gamma} \quad (2)$$

where  $\gamma$  is the path loss exponent and indicates the rate at which the path loss increases with distance. Depending on the presence or absence of a line of sight link, different values are assigned to  $\gamma$ .

The power consumption model of the radio transceiver used in this paper is adopted from [18, 10], which considers varying transmission powers

to meet minimum receiver sensitivity requirements. This assumption is justified, since most existing transceivers support variable transmission power levels in several discrete steps. An example of such a transceiver is the Texas Instruments Chipcon, CC2420 [1]. Moreover, the model includes the energy consumed in signal reception which, in today's transceivers, is a considerable amount. A transceiver's energy consumption is mainly accounted for digital signal processing (DSP) and the energy consumed by the front end circuit and the power amplifier/voltage amplifier. The power consumed in transmitting a message at  $r$  bits/s over a distance of  $d$  meters can be calculated as [10]:

$$P_T(d) = (\alpha_{11} + \alpha_2 \cdot d^\gamma)r \quad (3)$$

And the power consumed by the receiver to receive this message is given as [10]:

$$P_R = (\alpha_{12}) \cdot r \quad (4)$$

The variables  $\alpha_{11}$  and  $\alpha_{12}$  are constants and depend on several factors, such as the digital coding and decoding mechanisms; modulation and demodulation, and pulse shaping filters.  $\alpha_2$  depends on the antenna characteristics, channel conditions, amplifier efficiency, and receiver sensitivity.

Two widely used propagation models are the Friss Free Space model ( $\gamma = 2$ ) and the Two-Ray Ground propagation model ( $\gamma = 4$ ). Depending on the separation distance between the communicating nodes, the propagation model is chosen. A cross-over distance which determines this selection is defined in [10]. If the distance is below this cross-over distance ( $d_{crossover}$ ), then the free space propagation model is taken, else the two-ray ground propagation model is used. The received signal strength as a function of distance is formulated as [15]:

$$P_{rx}(d) = \frac{P_{tx}G_tG_r\lambda^2}{(4\pi)^2d^2L} \quad (5)$$

where  $d$  is the distance between the transmitter and the receiver in meters;  $P_{tx}$  and  $P_{rx}$  are the transmitted and received power, respectively;  $G_t$  and  $G_r$  are the corresponding gains of the transmitting and receiving antenna;  $h_t$  and  $h_r$  are the height of the transmitting and receiving antenna above ground;  $\lambda$  is the wavelength of the carrier signal; and  $L$  is the system loss factor not related to propagation.

Where there is no line-of-sight link between the transmitter and receiver, the Two-Ray Ground model is more accurate than the Friss Free Space model. The received power at a distance  $d$  from the transmitter can be expressed as [15]:

$$P_{rx}(d) = \frac{P_{tx}G_tG_rh_t^2h_r^2}{d^4} \quad (6)$$

The cross-over distance is formulated as [10]:

$$d_{crossover} = \frac{4\pi\sqrt{L}h_rh_t}{\lambda} \quad (7)$$

#### 4. Fair and Efficient Topology Control

In this section, we establish the basic model of the topology control protocol. The model defines weighted relaying regions in a 2-dimensional plane for any arbitrary node in the network. The weighted regions specify the degree of eligibility of a neighboring node to become a relaying node. The eligibility criteria sets a trade-off between minimizing the overall energy cost of a multi-hop communication; and the minimization of disconnected links that occur due to disproportionate energy consumption by individual nodes. The eligibility of each node is computed by taking only local information into account.

##### 4.1. Background

In [18], Bhardwaj et al. sets a theoretical upper bound on the lifetime of a linear-topology wireless sensor network that supports multi-hop communication. Their model calculates the optimal number of hops based on the notion of a characteristic distance,  $d_{char}$ . This distance is computed by taking the hardware components of a transmitter and a receiver as well as the channel's characteristics. Then, for any arbitrary transmitting node,  $t$ , a receiving node,  $r$ , and a separating distance,  $D$ , between them, there exists an optimal number of hops,  $K_{opt}$ , such that:

$$K_{opt} = \left\lfloor \frac{D}{d_{char}} \right\rfloor \text{ or } \left\lceil \frac{D}{d_{char}} \right\rceil \quad (8)$$

The characteristic distance,  $d_{char}$ , is independent of  $D$  and calculated as:

$$d_{char} = \sqrt[\gamma]{\frac{\alpha_1}{\alpha_2(\gamma - 1)}} \quad (9)$$

where  $\alpha_1 = \alpha_{11} + \alpha_{12}$ .

#### 4.2. Hop Model

For any arbitrary node  $i$ , the position of a neighboring node can be expressed in terms of its deviation from the optimal relaying position. The optimal relaying position is the direct line that connects node  $i$  with the base station and it is a function of the characteristic distance,  $d_{char}$ . The deviation from this line of a neighbor node is illustrated in Figure 2.

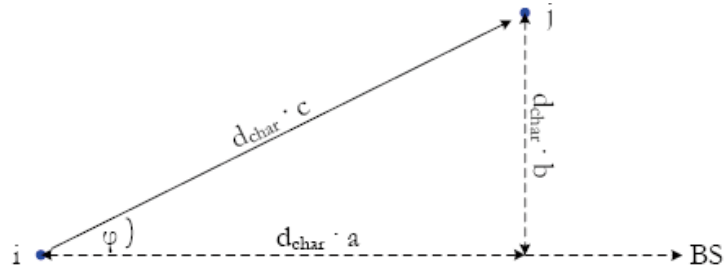


Figure 2: Hop model

Taking  $i$  as the origin of the coordinate system, the  $x$ - and  $y$ -coordinates of node  $i$  are expressed as  $d_{char} \cdot a$  and  $d_{char} \cdot b$  respectively, where  $a, b \in \mathbb{R}$ . The distance from  $i$  to  $j$  is then  $d(i, j) = d_{char} \cdot c$ , where  $c = \sqrt{a^2 + b^2}$ . The  $x$ -coordinate is the progress<sup>3</sup> of the hop.

Since the computation of a multi-hop link is based on local information only, the optimal number of hops,  $K_{opt}$ , can only be an estimation:

$$\begin{aligned} K &= \frac{D}{\bar{a} \cdot d_{char}} \\ &= \frac{K_{opt}}{\bar{a}} \end{aligned} \quad (10)$$

where  $\bar{a}$  is the average value of all  $as$ .

---

<sup>3</sup>Progress is the “effective” distance traversed in one hop.



#### 4.3. Hop Efficiency

In order to develop an efficiency measure for a single hop link, we compare the energy consumption of a theoretically optimal multi-hop link with a link that results from our hop model given in the previous section.

The rate at which energy is consumed by a relaying node can be calculated as the overall power consumed during data reception and transmission over a distance  $d$ . The Power consumption of a single-hop link can be estimated by:

$$P_{relay}(d) = (\alpha_1 + \alpha_2 d^\gamma) r \quad (11)$$

Since the most energy-efficient route between node  $i$  and the BS is the hop-by-hop line of sight link that connects the two nodes, the minimum energy rate  $P_{link-min}$  that should be consumed during a communication is given as:

$$P_{link-min}(D) = K'_{opt} \cdot P_{relay}(d_{char}) \quad (12)$$

where  $D$  is the overall distance to the BS. However, in a randomly deployed sensor network, nodes are not distributed along the optimal link line. Therefore, the power consumed by a link of distance  $D' \geq D$  with intervening nodes deviating from the optimal link line can be expressed as:

$$P_{link}(D') = \sum_{i=1}^K P_{relay}(c_i \cdot d_{char}). \quad (13)$$

Taking  $P_{link-min}(D)$  as a relative measure, building the ratio of  $P_{link-min}(D)$  over  $P_{link}(D')$  gives a measure of the efficiency of a chosen link. In maximizing this ratio, the most energy efficient link can be determined.

$$\frac{P_{link-min}(D)}{P_{link}(D')} = \frac{K'_{opt} \cdot P_{relay}(d_{char})}{\sum_{i=1}^K P_{relay}(c_i \cdot d_{char})} \quad (14)$$

**Theorem 1.** *The overall link efficiency measure,  $\Lambda$ , of a multi-hop link can be formulated as:*

$$\Lambda \leq \frac{\tilde{a} \cdot \gamma}{\bar{c}^\gamma + \gamma - 1} \quad (15)$$

where  $\bar{c}$  is the normalized average link distance over  $d_{char}$ .

*Proof.* Defining the overall link efficiency  $\Lambda = \frac{P_{link-min}(D)}{P_{link}(D')}$ , we can write

$$\begin{aligned}
\Lambda &= \frac{K'_{opt} \cdot P_{relay}(d_{char})}{\sum_{i=1}^K P_{relay}(c_i \cdot d_{char})} \\
&= \frac{K'_{opt}(\alpha_1 + \alpha_2 \cdot d_{char}^\gamma)r}{\sum_{i=1}^K (\alpha_1 + \alpha_2(c_i \cdot d_{char})^\gamma)r} \\
&= \frac{K'_{opt}(\alpha_1 + \alpha_2 \cdot d_{char}^\gamma)}{K \cdot \alpha_1 + \alpha_2 \cdot d_{char}^\gamma \sum_{i=1}^K c_i^\gamma} \\
&= \frac{K'_{opt}(\alpha_1 + \alpha_2 \cdot d_{char}^\gamma)}{K \left( \alpha_1 + \alpha_2 \cdot d_{char}^\gamma \cdot \frac{1}{K} \cdot \sum_{i=1}^K c_i^\gamma \right)}
\end{aligned}$$

Having  $c^\gamma$  a strictly convex function ( $c \in \mathbb{R}^+, 2 < \gamma < 6$ ), we can use Jensen's inequality for convex functions, which states that

$$\forall \{\lambda_i\}, \lambda_i \in \mathbb{R}^+ \text{ such that } \sum_i \lambda_i = 1$$

$$f\left(\sum_i \lambda_i x_i\right) \leq \sum_i \lambda_i f(x_i) \quad (16)$$

with equality if all  $x_i$ s are equal, to get

$$\bar{c}^\gamma \leq \frac{\sum_{i=1}^K (c_i)^\gamma}{K} \quad (17)$$

Using this and the estimated overall-link hop<sup>4</sup>, we can further express:

$$\Lambda \leq \frac{\tilde{a}(\alpha_1 + \alpha_2 \cdot d_{char}^\gamma)}{\alpha_1 + \alpha_2 \cdot d_{char}^\gamma \cdot \bar{c}^\gamma}.$$

Substituting  $d_{char}$  given in Equation 9 in the inequation, we get

---

<sup>4</sup>The *estimated overall-link hop progress* normalized over the characteristic distance and denoted as  $\tilde{a}$ , is the ratio of hops  $K'_{opt}$ , and the number of hops,  $K$ :  $\tilde{a} = \frac{K'_{opt}}{K}$ .

$$\begin{aligned}
\Lambda &\leq \frac{\tilde{a} \left( \alpha_1 + \alpha_2 \left( \frac{\alpha_1}{\alpha_2(\gamma-1)} \right) \right)}{\alpha_1 + \alpha_2 \left( \frac{\alpha_1}{\alpha_2(\gamma-1)} \right) \bar{c}^\gamma} \\
&\leq \frac{\tilde{a}(\alpha_1 + \frac{\alpha_1}{\gamma-1})}{\alpha_1 + (\frac{\alpha_1}{\gamma-1})\bar{c}^\gamma} \\
&\leq \frac{\tilde{a} \cdot (1 + \frac{1}{\gamma-1})}{1 + (\frac{1}{\gamma-1})\bar{c}^\gamma} \\
&\leq \frac{\tilde{a} \cdot \gamma}{\bar{c}^\gamma + \gamma - 1} \quad \square
\end{aligned}$$

A transmitting node's knowledge is limited to its immediate neighbors. Therefore, the efficiency model is applied to enable a node compare and select a neighbor that can participate in building a multi-hop link whose overall energy consumption is minimum. Theorem 1 is employed for the single hop case, substituting the average values with the single hop values. A neighboring node  $j$  in the plane of a searching node is  $\Lambda_j$  efficient for the overall link. Hence, its eligibility of being a neighbor of node  $j$  is determined accordingly.

$$\Lambda_j = \frac{a \cdot \gamma}{c^\gamma + \gamma - 1} = \frac{\cos \varphi \cdot c \cdot \gamma}{c^\gamma + \gamma - 1} \quad (18)$$

If a transmitting node has no knowledge of the direction of message propagation, i.e., the position of the base station is not known, then it cannot estimate the deviation of a node's position from the optimal link. Hence,  $\varphi$  is set to 0 and  $\Lambda$  can be written as:

$$\Lambda_j = \frac{c \cdot \gamma}{c^\gamma + \gamma - 1} \quad (19)$$

In Figure 3,  $\Lambda$  is plotted when no base station direction information is present at the node. The  $x$ -axis is the normalized distance to the neighboring node over  $d_{char}$ .

## 5. Node Eligibility Metric

The eligibility metric  $\Lambda_j$  derived in the previous section, defines an efficiency measure for a position in the region of transmission range of a node.

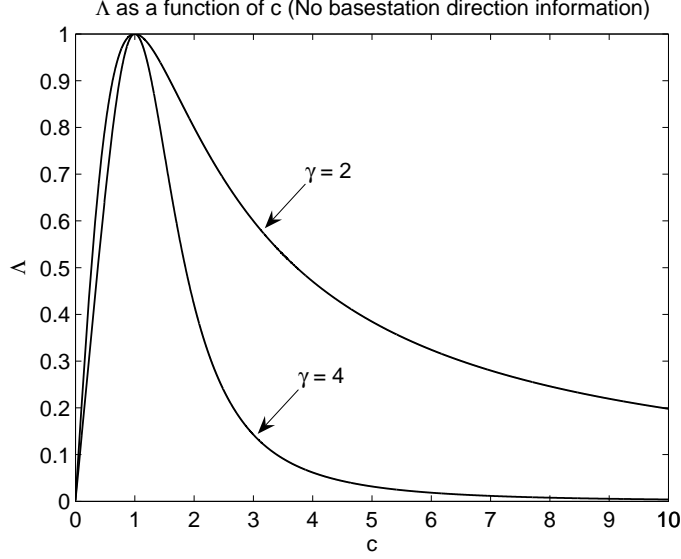


Figure 3: Plot of  $\Lambda$  for path-loss exponents of 2 and 4 where no direction information of the base station exists.

Thus, a node  $j$  within the transmission range of node  $i$  acquires this measure  $\Lambda_j$  as its eligibility to be a neighbor. A link established based on this metric ensures an energy-efficient multi-hop communication. This is one aspect to consider when building the network, as it is essential to reduce the overall energy dissipation due to routing. Another aspect to consider is fairness between the nodes. In order to ensure that nodes exhaust their energy reserves more uniformly, those nodes that have relatively high energy reserves should be chosen as relying nodes. Hence, we define the metric  $\Upsilon_j = \frac{e_j}{E}$ . Similar to the overall efficiency metric,  $\Lambda$ ,  $\Upsilon_j$  is applied to a neighboring node to measure its relative energy reserve with respect to the other nodes.

Combining both metrics, we can achieve overall link efficiency and fairness through a common eligibility measure of a neighboring node. Thus, we define:

$$\Psi_j = \Lambda_j \cdot \Upsilon_j \quad (20)$$

A node  $i$  having node  $j$  in its transmission range calculates  $\Psi_j$ ,  $0 \leq \Psi_j \leq 1$ . This determines a measure for node  $j$ , for which node  $i$  can estimate how eligible it is to be a neighbor.

To accommodate node failure and node mobility, the topology control protocol runs periodically, enabling actual message exchange between the nodes and timely topology adjustment. Moreover, the specific structure of a topology depends on whether nodes have information about the direction of the base station. We denote the graph that is built with the knowledge about the direction of the base station with  $\mathcal{G}_{FETCD}$ . Otherwise, it is denoted as  $\mathcal{G}_{FETC}$ .

## 6. Protocol Description

Topology formation is accomplished in two phases. The first phase is the neighbor discovery phase in which each node selects  $k$  nodes in its neighborhood. The neighbor selection is carried out according to the node eligibility criterion. However, the network graph that is created in this phase is not symmetric. The second phase is concerned with building a symmetric graph from the initial topology that is formed in phase 1. The symmetry is obtained by adding the reverse edge to every asymmetric link.

These phases are described in more detail as follows:

**Phase 1:** Choosing  $k$  Neighboring Nodes (For a generic node  $i$ )

1. Node  $i$  wakes up at time  $t_1$ , and announces its identity ( $id_i$ ) and energy reserve ( $e_i^{t_1}$ ) at the maximum power ( $P_{t-max}$ ).
2. Node  $i$  receives the messages from the neighboring nodes and stores their identities in its neighbor list  $\mathcal{N}(i)$ .
3. Node  $i$  estimates the distance to each node in  $\mathcal{N}(i)$ . Node  $i$  has the energy reserves of the neighboring nodes ( $e_j$ ) as well as the distances to them ( $d(i, j)$ ), where  $j \in \mathcal{N}(i)$ .
4. Node  $i$  calculates  $\Psi_j$ , for each neighbor in its list.
5. Node  $i$  chooses the  $k$  neighbors in its list  $\mathcal{N}(i)$  that have the highest value of  $\Psi$ . If originally node  $i$  has less than  $k$  neighbors, then all nodes are chosen.
6. Node  $i$  updates its neighbor list according to the chosen nodes in step 5.

The developed graph according to phase 1 of the protocol, has directed links and the graph is a directed graph,  $\mathcal{G}_{digraph}$ . Hence, a symmetry phase is necessary to enforce symmetry in the graph. In this phase we build the symmetric super-graph of  $\mathcal{G}_{digraph}$ .

The symmetric super-graph of  $\mathcal{G}_{digraph}$  is defined as the undirected graph  $\mathcal{G}$  obtained from  $\mathcal{G}_{digraph}$  by adding the undirected edge  $[i \leftrightarrow j]$  whenever edge  $[i \rightarrow j]$  or  $[i \leftarrow j]$  is in  $\mathcal{G}_{digraph}$ . That is,  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ , where  $\mathcal{E} = \{[i \leftrightarrow j] | [i \rightarrow j] \in \mathcal{E}_{digraph} \text{ or } [i \leftarrow j] \in \mathcal{E}_{digraph}\}$ .

**Phase 2:** Enforcing Graph Symmetry (For generic node  $i$ )

1. At time  $t_2$ , node  $i$  announces its identity ( $id_i$ ) and list of Neighbors ( $\mathcal{N}(i)$ ) at maximum power ( $P_{t-max}$ ).
2. Node  $i$  receives the neighbor lists, and calculates the set of symmetric neighbors. Node  $i$  checks all neighbor lists and finds if it exists there. When that is the case, it checks if the neighbor list originates from a neighbor in its neighbor list. If not, the corresponding neighbor is added to its list  $\mathcal{N}(i)$ .

After the symmetric graph is constructed, node  $i$  determines for each neighbor in  $\mathcal{N}(i)$  the minimum required transmission power to reach it; then this information is stored in its neighbor table list  $\mathcal{N}_L(i)$ . When communicating with a node in its neighbor list, a node adjusts its transmitter power accordingly. The selected neighbors of a node  $i$  are its logical neighbors. That is, there can be nodes in its maximum transmission range it may not be selected as neighbors. These nodes in  $\mathcal{N}(i)$  are used for the purpose of routing.

## 7. Evaluation Background

Two types of topology control protocols are considered. The first types exploit knowledge of the geometric properties of nodes and graph theories to establish a network's topology. These types of topology control protocols model networks by considering nodes as points in a Euclidean space and communication links as straight lines that connect two of these points. These types of protocols require little knowledge of the deployment setting, but they perform poorly because they consider the channel characteristics as static. The second types employ probabilistic models to capture and take into account network dynamics. Both types of approaches require an initial graph upon which they apply their algorithms. This same graph can also serve as a reference to evaluate the gains and losses of the topology control protocols.

### 7.1. Disk Graph

Most existing topology control protocols use the disk graph shown in Figure 1 as a reference – The Disk Graph (DG) has an edge between two nodes  $u$  and  $v$  if they are at a distance less than  $d_{max}$ ,  $d(u, v) < d_{max}$ , where  $d_{max}$  refers to the maximum transmission range of a node. Table 1 displays the algorithm we used to generate the Disk Graph Topology.

<i>Building the Original Topology Graph <math>\mathcal{G}_{original}</math></i>	
1:	<b>for each</b> $i \in \mathcal{V}$
2:	<b>for each</b> $j \in \mathcal{V}$
3:	<b>if</b> $d(i, j) \leq d_{max}$
4:	$\mathcal{N}(i) \Leftarrow j$
5:	$\mathcal{N}_L(i) \Leftarrow [i \rightarrow j]$

Table 1: Algorithm for determining the original topology.

A network based on the Disk Graph topology will have the highest graph size – which refers to the number of edges (communication links) in the graph. As a result, it is inefficient in terms of its energy consumption. There are different strategies to trim off inefficient links from the DG. For our evaluation, we consider three proximity graphs which are widely referenced in the literature: relative neighborhood graph, Gabriel graph, and K-neighborhood graph. Proximity graphs have the property of being connected if the original graph (i.e., the Disk Graph) is connected. For the sake of completeness, we briefly summarize these strategies and the algorithms we used to construct the corresponding topologies:

### 7.2. Relative Neighborhood Graph

The *Relative Neighborhood Graph* [11, 16] (RNG) of a set  $\mathcal{V}$  is a proximity graph such that there exists an edge between points  $u$  and  $v$  if and only if the lune based region<sup>5</sup> is empty of other points. The RNG graph has an edge between two points  $u$  and  $v$ , if there is no other point  $w$  such that:  $\max\{d(u, w), d(v, w)\} < d(u, v)$ .

---

<sup>5</sup>A lune is the region of intersection made between two circles that have the same radius.

---

```

1: initialize  $\mathcal{G}_{rng} \Leftarrow \mathcal{G}_{gg}$ 
2: for each  $i \in \mathcal{V}$ 
3:   for each  $j \in \mathcal{N}(i)$ 
4:     for each  $k \in \mathcal{N}(i)$ 
5:       if  $\max(d(i, k), d(j, k)) < d(i, j)$ 
6:         remove  $j$  from  $\mathcal{N}(i)$ 
7:         remove  $[i \rightarrow j]$  from  $\mathcal{N}_L(i)$ 

```

---

Table 2: Algorithm for constructing the Relative Neighborhood Graph Topology.

### 7.3. Gabriel Graph

The *Gabriel Graph* (GG) of a set  $\mathcal{V}$  is a proximity graph in which an edge between points  $u$  and  $v$  exists if and only if a disk whose antipodal points are  $u$  and  $v$  does not contain any other points in  $\mathcal{V}$  [8]. Mathematically, the GG graph has an edge between two points  $u$  and  $v$  if and only if there is no other point  $w$  such that  $d^2(u, w) + d^2(v, w) < d^2(u, v)$ . Obviously,  $RNG \subseteq GG \subseteq DG$ . The GG topology algorithm initializes with the disk graph and the RNG topology algorithm initializes with the GG graph, since the RNG is a subgraph of the GG.

#### *Building the Gabriel Graph Topology $\mathcal{G}_{gg}$*

---

```

1: initialize  $\mathcal{G}_{gg} \Leftarrow \mathcal{G}_{original}$ 
2: for each  $i \in \mathcal{V}$ 
3:   for each  $j \in \mathcal{N}(i)$ 
4:     for each  $k \in \mathcal{N}(i)$ 
5:       if  $d(i, k)^2 + d(j, k)^2 < d(i, j)^2$ 
6:         remove  $j$  from  $\mathcal{N}(i)$ 
7:         remove  $[i \rightarrow j]$  from  $\mathcal{N}_L(i)$ 

```

---

Table 3: Algorithmic representation for determining the Gabriel Graph Topology.

### 7.4. KNeigh

The KNeigh Protocol, as described in [2], builds the topology based on the  $k$  nearest neighbors. The preferred value of  $k$  for a large-scale network is derived in the same work and it is set to 9. We represent the algorithm



for building the KNeigh topology based on the algorithm described in Table 5. The algorithm begins on the original topology, each node selecting its neighbors according to its distances to them. The algorithm has two phases: In the first phase, 9 nearest neighbors are chosen based on their proximity to the node. In the second phase (the symmetry phase that begins at step 10), each node ensures that it is listed as a neighbor node by those nodes which it elects as neighbors. This is to ensure the existence of bidirectional links. If this is not the case, a node drops the neighbors in whose list it is not included. The KNeigh protocol applies further pruning to remove links whose transmission cost in a multi-hop communication is inefficient.

### 7.5. FETC and FETCD

The algorithm for constructing FETC and its variant, FETCD, is given previously, in Section 4. The two protocols are different in that in FETCD, all nodes have knowledge of the direction of the base station and take this into account to calculate  $\Psi$ . Table 4 summarizes how eligibility is measured for the two protocols.

Topology	$\Psi_j$
FETC	$\frac{c_{ij} \cdot \gamma_{ij}}{c_{ij}^{\gamma_{ij}} + \gamma_{ij} - 1} \cdot \frac{e_j}{E}$
FETCD	$\frac{\cos \varphi_{ij} \cdot c_{ij} \cdot \gamma_{ij}}{c_{ij}^{\gamma_{ij}} + \gamma_{ij} - 1} \cdot \frac{e_j}{E}$

Table 4: Eligibility metric with and without knowledge of the base station's direction

### 7.6. Routing

The performance of the topology control protocols is best examined when data transmission takes place. For our simulation, we use two different routing protocols: shortest path and energy-aware routing protocols. The first protocol computes a route that has the shortest distance from the source to the destination. The second protocol computes a route that has the maximum overall energy reserves.

*Determining the KNeigh Topology  $\mathcal{G}_{KNeigh}$*

---

```

1: initialize  $\mathcal{G}_{KNeigh} \leftarrow \mathcal{G}_{original}$ 
2: for each  $i \in \mathcal{V}$ 
3:    $L(i) \leftarrow []$ 
4:   for each  $j \in \mathcal{N}(i)$ 
5:      $L(i) \leftarrow d(i, j)$ 
6:   sort  $L(i)$  in ascending order
7:   if number of elements in  $L(i) > 9$ 
8:      $i$  updates  $\mathcal{N}(i)$  and  $\mathcal{N}_L(i)$  to the first 9 elements in
        $L(i)$ 
9:   Symmetry Phase
10:  for each  $i \in \mathcal{V}$ 
11:    for each  $j \in \mathcal{N}(i)$ 
12:      if  $[i \rightarrow j] \in \mathcal{N}_L(i)$  and  $[j \rightarrow i] \notin \mathcal{N}_L(j)$ 
13:        remove  $j$  from  $\mathcal{N}(i)$ 
14:        remove  $[i \rightarrow j]$  from  $\mathcal{N}_L(i)$ 
15:  Pruning Phase
16:  for each  $i \in \mathcal{V}$ 
17:    for each  $j \in \mathcal{N}(i)$ 
18:      if  $d(i, j) \leq d_{crossover}$  then  $\gamma_{ij} = 2$ 
19:      else  $\gamma_{ij} = 4$ 
20:      for each  $k \in \mathcal{N}(i)$ 
21:        if  $d(i, k) \leq d_{crossover}$  then  $\gamma_{ik} = 2$ 
22:        else  $\gamma_{ik} = 4$ 
23:        if  $d(j, k) \leq d_{crossover}$  then  $\gamma_{jk} = 2$ 
24:        else  $\gamma_{jk} = 4$ 
25:        if  $d(i, k)^{\gamma_{ik}} + d(j, k)^{\gamma_{jk}} < d(i, j)^{\gamma_{ij}}$ 
26:          remove  $j$  from  $\mathcal{N}(i)$ 
27:          remove  $[i \rightarrow j]$  from  $\mathcal{N}_L(i)$ 

```

---

Table 5: Algorithm for determining the KNeigh Graph Topology.

Node $i$				
ID	Position	Energy Reserve	Neighbor List	Path to BS
$id_i$	$(x_i, y_i)$	$e_i$	$\mathcal{N}(i)$	$\mathcal{R}(i)$

Table 6: Structure of a node implementation.

## 8. Simulation

We use MATLAB® as a simulation tool. We generate the nodes' random placement – a Poisson point process – using a technique described in [14]. Then, with the position information and the channel characteristic parameters, we built the topologies which define for each node a set of neighbors. The parameters that enable a node to make local decisions are displayed in table 6.

Our simulation is divided into two categories. In the first category, the graphs are investigated from a theoretical point of view, i.e., the graphs' connectivity and node degree are investigated. In the second category, more practical aspects of a wireless sensor network are investigated, namely, the energy cost of the multi-hop links that are built by the different topology control techniques and fairness routing data.

The region of deployment is  $500\text{ m} \times 500\text{ m}$  2-dimensional plane. The number of nodes deployed in this region is: 100, 200, 300, 400, and 500. Accordingly, different deployment densities are examined. The base station is chosen to be the furthest node with the highest  $x$  – *coordinate* in the deployment. The base station is assumed to have an infinite energy reserve, which is true in reality.

We denote a period of time as a time step in which one bit of information is sent to the base station over a multi-hop link. For each event originating from an arbitrary node  $i$ , all eligible relay nodes that are along the path decrease their energy reserve according to our energy model. A relaying node consumes reception power as well as transmission power, depending on its relative distance from a sending node and the next hop. The path-loss exponent  $\gamma$  can be either 2 or 4, according to the required transmission distance. Here we introduce the crossover distance,  $d_{crossover}$ , as in [10]. If the transmission distance is less than  $d_{crossover}$ ,  $\gamma$  is taken as 2. Else,  $\gamma$  is taken as 4. In Table 7, the parameter values used in the simulation are presented.

Description	Parameter	Value
Transmitting antenna Gain	$G_t$	1
Receiving antenna Gain	$G_r$	1
Transmitting antenna height	$h_t$	1.5 m
Receiving antenna height	$h_r$	1.5 m
Maximum transmitting Power	$P_{t-max}$	0 dBm
Receiver Sensitivity	$P_{rx-thresh}$	-85 dBm
Carrier signal wavelength	$\lambda$	0.1224 m (2.45 GHz)
System loss factor	$L$	1
Initial Battery Capacity	$E$	
Transmitter electronics energy	$\alpha_{11}$	26.5 mJ/bit
Receiver electronics energy	$\alpha_{12}$	59.1 mJ/bit
Efficiency	$\eta_{amp}$	0.023
Radio amplifier energy	$\alpha_2$	( $\gamma = 4$ ) ( $\gamma = 4$ )
Path loss exponent	$\gamma$	2 or 4
Relay Rate	$r$	1 bits/s
Characteristic Distance	$d_{char}$	100 m ( $\gamma = 2$ ) 71 m ( $\gamma = 4$ )

Table 7: *Parameters*

### 8.1. Graph Connectivity

Connectivity is one of the essential properties of a network graph. The connectivity of a graph is an expression of its 1-connectivity, i.e., each node in the network has at least one multi-hop path that connects it with the base station. For various network densities, we considered different values of  $k$  that keep the network connected despite trimming off inefficient links. For a network with a 2D Poisson distribution, the probability that a graph is connected is given as [17]:

$$\Pr(1\text{-conn}) \cong \left(1 - e^{-\lambda \cdot \pi \cdot d_{max}^2}\right)^n \quad (21)$$

where  $d_{max}$  is the maximum transmission range of the nodes;  $\lambda$  is related to the network density; and  $n \gg 1$  is the number of nodes. In Figure 4, we plot the probability of graph connectivity as the density of the network varies from 40 to 70 nodes (worst case situation). As can be seen, deploying less

than 70 nodes in an area of  $500 \times 500 \text{ m}^2$  does not result in a connectivity probability of 1 regardless of the node degree – In fact, the connectivity probability does not increase for  $k > 5$ . Comparatively, the GG and RNG graphs can achieve 100% connectivity for the same density. This is one of the reasons why GG and RNG graphs are favored for topology control.

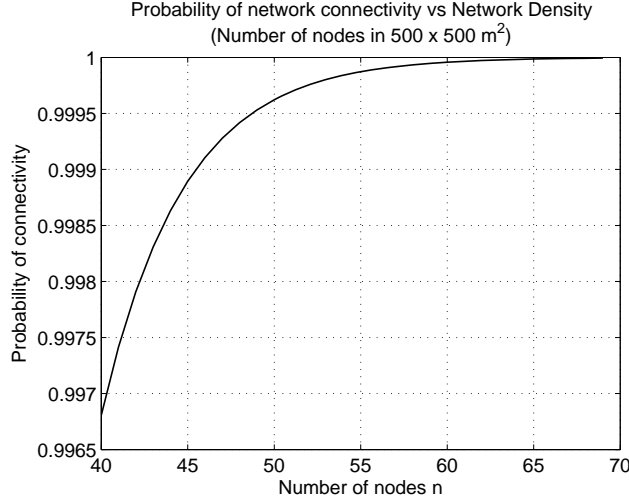


Figure 4: Probability of graph connectivity

### 8.2. Energy Consumption

We investigated two aspects: the overall network energy consumption and the achievable fairness in the distribution of energy reserve. Fairness is measured by quantifying the variances in the energy reserves of the nodes. We run 100 time-steps for five network densities: 100, 200, 300, 400, and 500. In a time step, 100 nodes are randomly chosen, and one bit of information is sent from each of these nodes along a multi-hop link to the base station. For each network density, 10 random deployments are generated and data transmission (event generation) take place. The average variance of the 10 deployments is used to construct the results for the corresponding network density. This is done for the two routing protocols, namely, the shortest path routing and the energy aware routing.

First we studied the rate of energy dissipation in the network. This is the amount of energy dissipated in the overall network as a function of the

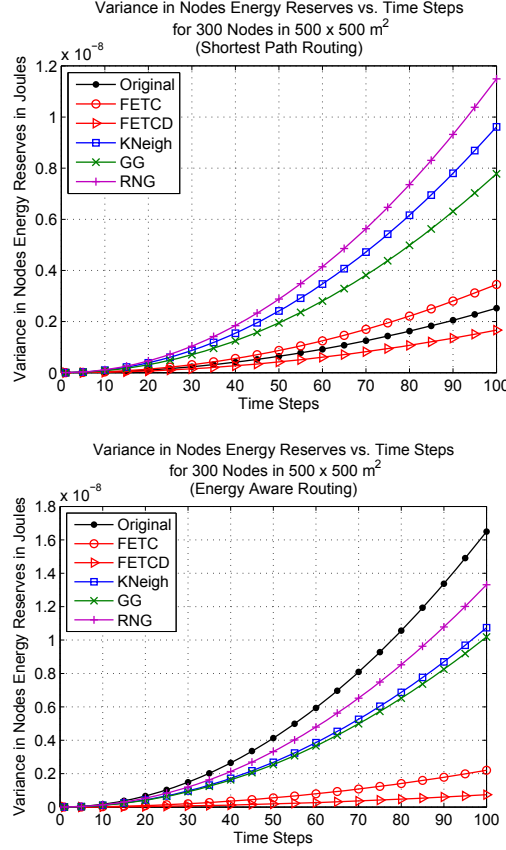


Figure 5: Variance in the nodes' Energy Reserves.

time steps. We normalized the results over the rate of energy consumption of the original topology (the disk graph). Therefore, only relational analysis is displayed.

Figure 5 displays the energy variations as a function of the time stamps. In both graphs, FETC and FETCD, produce the least energy consumption rate in the network. Moreover, for both cases, the curves reveal that the two protocols have gentler slopes, which indicate lower increasing rates in the variance of energy in the network. The graphs which result in high energy variation are the KNeigh, GG, and RNG topologies. These graphs suffer from the increased number of hops, since they attempt to build the topology based on the nearest neighbors. This increases the energy dissipation of

the multi-hop links. The FETC graph has a slight increment in the energy variance as the network density increases. This is expected, as the topology does not exploit knowledge of the direction of the base station to avoid longer routes.

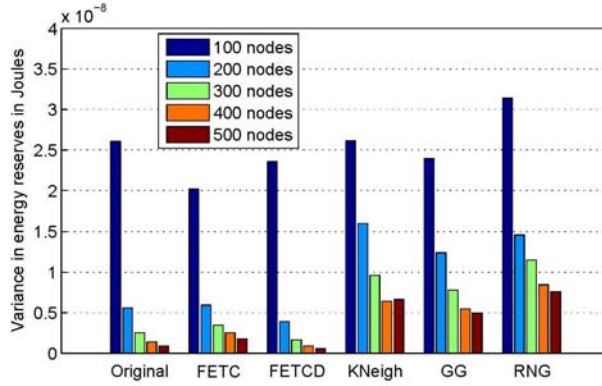


Figure 6: Variance in the nodes' Energy Reserves after 100 time steps.

The second aspect of comparison is in the form of normalized energy dissipation as a measure of fairness between nodes. In Figure 6, the variance of the nodes energy reserves for different node densities after the 100th time step is displayed. The Kneigh, GG, RNG achieve lower fairness compared to the original graph (which has a value of 1).

### 8.3. Qualitative Aspects

So far, the performance of the protocol was discussed quantitatively. In the following subsection, the qualitative aspects are briefly discussed.

#### 8.3.1. Message complexity

The execution of a topology control protocol causes a certain message overhead in the network. The message complexity is an important aspect to mind. Our protocol has a communication complexity<sup>6</sup> of  $\mathcal{O}(n)$ . Each node has to send 2 messages in order to determine the topology of the network.

---

<sup>6</sup>The message complexity is defined as the communication effort in terms of the  $\mathcal{O}$  – *notation* that is necessary to run the topology control protocol [3].

### 8.3.2. Update policy

To deal with node mobility or failures, the topology control protocol is executed periodically. Finding the optimal rate of computation is not a trivial problem [3], and depends on the expected mobility rate. The reconfiguration or re-execution of the topology control protocol can be triggered synchronously or asynchronously. Asynchronous execution is achieved when each node has the choice to determine the time to run the protocol. Synchronous updates, on the other hand, are done when all the nodes execute the topology control protocol at the same time. Our protocol has an asynchronous update policy. Each node initiates the FETC protocol at different times; and has to wait for the replies from the other neighboring nodes for completing the protocol execution. Asynchronous updates are preferred over synchronous updates because of the low retransmission cost at the link layer due to collision.

### 8.3.3. Node degree

Low node degrees are preferable in order to reduce the overhead of route calculations in the network, but this is often at the cost of the probability of connectivity. A study on the network connectivity and the minimum number of neighbors is carried out in [21]. There it has shown that the minimum value of the node degree which guarantees connectivity with high probability is dependent on the number of nodes in the network, such that  $\Theta(\log n)$  neighbors are necessary and sufficient for connectivity of the communication graph.

The FETC protocol does not necessarily take the nearest neighbors as criteria for neighbor selection. Instead, it consider logical linking. The case where the logical links and the physical links are identical is when the density of the node distribution is so small that the nearest neighbor distance is greater than the characteristic distance. In that case, the  $k$  nearest neighbors are selected as the optimal neighbors. We validate the connectivity issue of our topology for different values of  $k$  through intensive simulations; for all the densities we considered, the optimal probability of connectivity can be achieved for  $k = 5$ .

### 8.3.4. Longer hops

A density-independent distance is proposed to determine the neighboring nodes, which in turn may result in higher transmission powers than other topology control protocols. Transmission over long distance is often argued



as being a cause of interference. Since interference is not included in our model, we refer readers to [9] in which convincing reasons in favor of long-hop routings are given.

In [9], it is argued that it is unclear if a single, short duration transmission at high power will bring more interference than multiple short range transmissions. Whereas the former case permits more efficient reuse of the communication channel, the signal to interference ratio (SIR) does not depend on the absolute power levels. Hence, increasing all transmission power levels at the same time does not have a negative impact on any packet reception probability. This indicates that a long-hop transmission does not necessarily cause more interference. Only the signal to interference and noise ratio (SINR) increases.

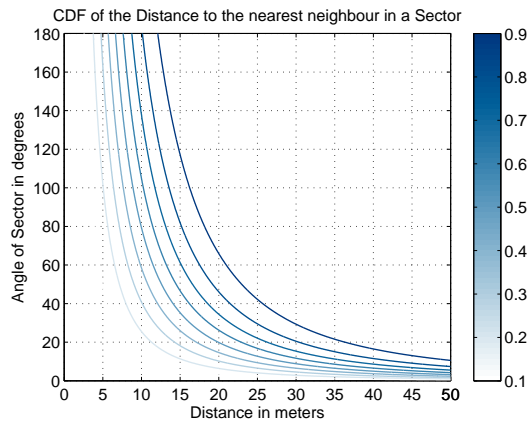


Figure 7: Contour Plot of the distribution function of the distance between a point and its nearest neighbor in a Poisson point process of density  $\lambda = 0.01$ .

Topologies that are formed by considering nearest neighbors cannot avoid the presence of nodes that are near to the base station. These nodes exhaust their energy reserve more quickly than others since they are frequently used. This leads to an energy imbalance in the network. Furthermore, such topologies suffer from traffic accumulation at these particular nodes, making them bottle necks for the network information flow. In [6], the optimal number of relay nodes as a function of the transmission rate is studied. It is shown that as the desired end-to-end rate increases, the optimal number of relay nodes decreases.

Longer hops have higher path efficiencies. The path efficiency is defined

as the ratio of the Euclidean distance of the end nodes and the multi-hop traveled distance. This is the case, since the probability of finding a relaying node that is near to the optimal line of communication is higher. The distance between a point and its nearest neighbor in a sector  $\phi$  for the Poisson point process is given in [5], with the cumulative distribution function given as:

$$F(r) = \Pr(R < r) = 1 - e^{-\lambda \frac{\phi}{2} r^2} \quad (22)$$

where  $r$  is the distance to the node and  $\lambda$  is the Poisson process density. In Figure 7,  $F(r)$  is plotted. For small sectors, the estimated distance to the nearest neighbor increases. Hence nodes near to the optimal link line have higher chances to be elected.

## 9. Conclusion

The energy efficiency of a wireless sensor network and the lifetime maximization problem is tackled by considering two aspects: The overall network energy consumption efficiency and fairness. Based on theoretical work on upper bounds of the network lifetime, we exploited the notion of a characteristic distance,  $d_{char}$ , that is dependent on the radio characteristic and the channel condition. From a node's view point, an estimation is made over the neighboring nodes on their overall link efficiency in relaying a message. This is done according to their positions relative to an optimal relaying position and the position of the base station. The efficiency estimation is made hop by hop. Fairness in energy utilization, and thereby connectivity, is addressed by taking the energy reserves of the nodes in the neighbor selection criteria into account.

The simulation results confirmed that our topology is not as sparse as the RNG, GG, and K-Neighbor topologies. However, with respect to the original topology (the mesh topology), the node degree is slightly increased with network density. Interesting results are obtained concerning the energy dissipation rate in the overall network. Unlike the other topology control protocols, the energy dissipation rates are little affected by increasing network densities. Moreover, concerning the energy reserves between the nodes, contrary to the RNG, GG, and KNeigh topologies, we have minimized imbalance. The results showed that nearest neighbor topologies are energy inefficient for high density networks. The original topology (disk graph), on the other hand, contains inefficient long links which significantly decreased

the energy efficiency of the network. These results show that our network topology suits to prolong the lifetime of the network.

## References

- [1] *Chipcon Product data sheet*: <http://www.chipcon.com/>.
- [2] D. M. Blough, M. Leoncini, G. Resta, and P. Santi. The k-neigh protocol for symmetric topology control in ad hoc networks. In *MobiHoc '03: Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*, pages 141–152, New York, NY, USA, 2003. ACM.
- [3] M. R. Brust and S. Rothkugel. A taxonomic approach to topology control in ad hoc and wireless networks. In *ICN '07: Proceedings of the 6th International Conference on Networking*, page 25, Washington, DC, USA, 2007. IEEE Computer Society.
- [4] X. Chao, W. Dargie, and G. Lin. Energy model for h2s monitoring wireless sensor network. In *CSE '08: Proceedings of the 2008 11th IEEE International Conference on Computational Science and Engineering*, pages 402–409, Washington, DC, USA, 2008. IEEE Computer Society.
- [5] N. Cressie. *Statistics for Spatial Data*. Wiley Series in Probability and Statistics, 1993.
- [6] Z. Dawy and P. Leelapornchai. Optimal number of relay nodes in wireless ad hoc networks with non-cooperative accessing schemes. In *ISITA 2002*, 2002.
- [7] A. Ephremides. Energy concerns in wireless networks. *IEEE Journal of Wireless Communication*, pages 48–59, 2002.
- [8] J. L. Gross and J. Yellen. *Handbook of Graph Theory*. CRC Press, 2004.
- [9] M. Haenggi. Twelve reasons not to route over many short hops. In *60th IEEE Vehicular Technology Conference*, pages 3130–3134, 2004.
- [10] W. B. Heinzelman. *Application-specific protocol architectures for wireless networks*. PhD thesis, 2000. Supervisor-Chandrakasan, Anantha P. and Supervisor-Balakrishnan, Hari.

- [11] J. W. Jaromczyk and G. T. Toussaint. Relative neighborhood graphs and their relatives. *Proc. of the IEEE*, 80(9):1502–1517, 1992.
- [12] A. A.-K. Jeng and R.-H. Jan. The r-neighborhood graph: An adjustable structure for topology control in wireless ad hoc networks. *IEEE Trans. Parallel Distrib. Syst.*, 18(4):536–549, 2007.
- [13] H.-Y. Kung, C.-M. Huang, H.-H. Ku, and Y.-J. Tung. Load sharing topology control protocol for harsh environments in wireless sensor networks. In *AINA '08: Proceedings of the 22nd International Conference on Advanced Information Networking and Applications*, pages 525–530, Washington, DC, USA, 2008. IEEE Computer Society.
- [14] P. A. W. Lewis and G. S. Shedler. Simulation of nonhomogeneous Poisson processes by thinning. *Nav. Res. Logistics Quart.*, 26:403–414, 1979.
- [15] T. Rappaport. *Wireless Communications: Principles and Practice*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2001.
- [16] P. Santi. Topology control in wireless ad hoc and sensor networks. *ACM Comput. Surv.*, 37(2):164–194, 2005.
- [17] P. Santi. Topology control in wireless ad hoc and sensor networks. *ACM Comput. Surv.*, 37(2):164–194, 2005.
- [18] M. B. Timothy, T. Garnett, and A. P. Ch. Upper bounds on the lifetime of sensor networks. In *ICC 2001*, pages 785–790, 2001.
- [19] R. Wattenhofer, L. Li, P. Bahl, and Y. min Wang. Distributed topology control for power efficient operation in multihop wireless ad hoc networks. In *12th Annual Joint Conference of the IEEE Computer and Communications Societies, Infocom*, pages 1388–1397, 2001.
- [20] R. Wattenhofer and A. Zollinger. Xtc: A practical topology control algorithm for ad-hoc networks. *Parallel and Distributed Processing Symposium, International*, 13:216a, 2004.
- [21] F. Xue and P. R. Kumar. The number of neighbors needed for connectivity of wireless networks. *Wirel. Netw.*, 10(2):169–181, 2004.

# Analysis of Error-Agnostic Time- and Frequency-Domain Features Extracted From Measurements of 3-D Accelerometer Sensors

Waltenegus Dargie, *Member, IEEE*, and Mieso K. Denko, *Senior Member, IEEE*

**Abstract**—This paper investigates the expressive power of several time- and frequency-domain features extracted from 3-D accelerometer sensors. The raw data represent movements of humans and cars. The aim is to obtain a quantitative as well as a qualitative expression of the uncertainty associated with random placement of sensors in wireless sensor networks. Random placement causes calibration, location and orientation errors to occur. Different type of movements are considered—slow and fast movements; horizontal, vertical, and lateral movements; smooth and jerky movements, etc. Particular attention is given to the analysis of the existence of correlation between sets of raw data which should represent similar or correlated movements. The investigation demonstrates that while frequency-domain features are generally robust, there are also computationally less intensive time-domain features which have low to moderate uncertainty. Moreover, features extracted from slow movements are generally error prone, regardless of their specific domain.

**Index Terms**—Accelerometer sensors, feature extraction, frequency-domain features, measurement errors, time-domain features, wireless sensor networks.

## I. INTRODUCTION

THIS paper examines an essential assumption based on which a large number of self-organizing and communication protocols in wireless sensor networks are developed: namely, nodes are deployed randomly. The assumption is plausible for some reasons. For example, in a rescue operation, one may not be able to carefully place sensor nodes, given the urgency of the operation. Likewise, in a health care application that monitors the activities of nurses, the nurses may not be able to pay much attention as to how the nodes are placed in some parts of their body (lower arm, upper arm; lower leg, thigh; back, shoulder, etc.). The assumption, however, should not entirely ignore placement and orientation errors and their impact on the quality of sensed data. One way to go around this problem is to have dense deployment, so that by taking measurements from a large number of closely placed nodes,

these errors can be minimized. Dense deployment, however, has its own problem due to cost, management, and computational complexities.

Another approach is to identify error agnostic features that can be extracted from the sensed data. Ideally, these features enable to recognize interesting events in the network with appreciable accuracy regardless of how and where the sensors are placed. Hence, we investigate the robustness of time- and frequency-domain features to calibration, placement, and orientation errors. We focus on cheap and randomly deployed accelerometer sensors that monitor the 3-D movements of humans and cars.

A close scrutiny into existing or proposed applications for wireless sensor networks reveals that movement (vibration) sensors are widely employed. For instance, accelerometer sensors are used to monitor the integrity of structures (bridges and building) [1]; transportation infrastructures [2]; supply-chain management [3]; Healthcare [4]; and active volcano [5]. Almost all of these applications employ model-based digital signal processing to detect interesting events such as defects in structures, abnormal drives, and damage in wheels.

We shall demonstrate that even though frequency-domain features are in general error agnostic, there are also simple and straightforward time-domain features that can be useful to many practical applications. The computational complexity of time domain features is significantly lower and can be carried out locally, on the wireless sensor nodes. Another interesting observation is that slow movements are error prone and difficult to recognize both with time- and frequency-domain features.

The contribution of this paper is summarized as follows.

- 1) Whereas there is a significant body of work on accelerometer sensors, to the best of our knowledge, this is the first comprehensive work that investigates the expressiveness of a large set of time- and frequency-domain features.
- 2) A qualitative metric based on fuzzy-sets and fuzzy membership functions that are defined and employed to examine the robustness of the features considered.

The remaining part of this paper is organized as follows. In Section II, related work is summarized. In Section III, the time- and frequency-domain features that are used in our analysis are discussed. In Section IV, the methodology to acquire the sensor data is presented; and the description of the scenarios for data collection is given. In Section V, a detail account of the analysis as well as its results are discussed. Finally, in Section VI, concluding remarks and outlook to future work are given.

Manuscript received September 02, 2009; revised November 15, 2009; accepted November 15, 2009. Date of publication February 02, 2010; date of current version April 07, 2010.

W. Dargie is with the Chair of Computer Networks, Technical University of Dresden, 01062, Dresden, Germany (e-mail: waltenegus.dargie@tu-dresden.de).

M. K. Denko is with the School of Computer Science, University of Guelph, Guelph, ON N1G 2W1 Canada (e-mail: denko@cis.uoguelph.ca).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/JSYST.2009.2039735

## II. RELATED WORK

A significant body of work exists on energy-efficient protocols for communication and self-organization in wireless sensor networks. Comparatively, the research community has so far focused on three aspects of signal processing, namely, aggregation, sampling, and compression. Dargie *et al.* [6] and Chao *et al.* [7] employ a local maxima technique to reduce the network's traffic. Ganesan *et al.* [8] propose data aggregation and compression mechanisms based on spatial interpolation of data and temporal signal segmentation. Lin proposes a sampling frequency control algorithm and a data compression algorithm [9]—each of them are dependent on the quality of the higher level features extracted from the raw sensor data. The sampling algorithm adjusts the sampling rate based on the features quality. When the sampling frequency cannot be controlled, a data compression algorithm is adopted to reduce the amount of transmitted data. Tang and Raghavendra [10] propose the ESPIHT compression algorithm that uses a distributed source coding and exploits spatio-temporal correlation. Bandyopadhyay *et al.* [11] give analytical results concerning the tradeoffs between sensor density, energy usage, throughput, delay, temporal sampling rates and spatial sampling rates. A more detailed survey concerning the existing data aggregation and information fusion approaches is given by Nakamura *et al.* [12] and [13]. Likewise, Tang *et al.* [10] surveys compression techniques.

As far as modeling and processing measurements of accelerometer sensors are concerned, several techniques and features have been considered. Huynh and Schiele [14] recommend a careful selection of features for different activities. Their experiment result suggests that the choice of a feature and a corresponding window length over which the feature is computed affect a recognition rate. Lukowicz *et al.* [15] investigate the existence of correlation in accelerometer signals to estimate various human activities.

Perhaps the most frequently employed technique in examining accelerometer data is coherence. An interesting work related to this is the one carried out by Engin *et al.* [16] and [17], in which the presence of correlation between different axes of individual accelerometers and between different segments of the same limb (of a human body) is used to study the characteristics of tremor in patients with Parkinson's disease (PD).

Marin-Perianu *et al.* [18] experiment with an incremental correlation algorithm that enables wireless sensor nodes to determine whether they are traveling together (in supply chain management). The algorithm is implemented locally on a sensor node and the data processed is a real-time data series. The scalability of the algorithm is tested with respect to complexities related to communication, energy, memory and speed of execution.

The approaches above identify a set of time- and frequency-domain features and adopt a particular technique to recognize various activities; and to examine the existence of correlation between these activities. Except for Huynh and Schiele, who show how a recognition rate can be affected by the choice of features and their window length, the rest focus rather on the modeling aspect and employ a single technique (usually the coherence function) to recognize activities. These approaches, how-

TABLE I  
TIME- AND FREQUENCY-DOMAIN FEATURES TO ANALYZE  
DATA FROM ACCELEROMETER SENSORS

Domain	Feature
Time	Mean, zero crossing rate, mean-value crossing rate, maxima/minima, autocorrelation, cross correlation, linear correlation coefficient, standard deviation
Frequency	Energy, correlation (for FFT and STFT), spectral roll-off, spectral centroid, spectral flux

ever, do not reveal sufficient insight about the robustness of the features employed. We build upon the existing approaches, but place our focus on investigating the robustness of the features to measurement errors.

## III. MAIN FEATURES

The time- and frequency-domain features we consider are listed in Table I. These features are used by many of the applications listed in Section I to recognize the occurrence of interesting phenomena. The time domain features capture and express temporal aspects, while the frequency-domain features capture and express spectral aspects. The extraction of time domain features does not require intensive pre-processing, but requires that transmission errors (noise and packet-loss) should be accounted for. Moreover, the comparison of two or more time-series measurements requires that the measurements are synchronized in time. On the other hand, the frequency-domain features are robust to transmission errors, but require intensive pre-processing [framing, windowing, filtering, and fast Fourier transformation (FFT)]. Hence, there is a tradeoff between the cost of feature extraction and the robustness of the features.

In the next subsections, a brief summary of the features listed in Table I is given.

### A. Time-Domain Features

1) *Zero-Crossing*: This reveals how often a signal (measurement) crosses a zero-reference line. It is a direct indication of the fundamental frequency of the signal. If the calibration position is known, the zero-crossing rate can be used to estimate the orientation of an accelerometer sensor. For example, if a sensor is calibrated by standing it up (say, along the *y*-axis), then it will produce an acceleration of 1 g if it is laid flat with a displacement of 90° either in the *z*-axis or in the *x*-axis. The zero-crossing rate is expressed as

$$ZCR(s) = \frac{1}{N} \sum_{i=0}^{N-1} F(s[i] \cdot s[i-1] < 0) \quad (1)$$

where *s* is a discrete, time-series sequence and *s*[*i*] and *s*[*i* − 1] are the *i*th and (*i* − 1)th sample values. *F* = 1 if the evaluation is true, *F* = 0 otherwise. Sensors which have different calibration will apparently have different zero-reference line. Therefore, it is difficult to compare their time-series measurements. To avoid this problem, the mean-value crossing rate is used, in which case a calibration-sensitive threshold is defined as  $\bar{S} = 1/N \sum_{i=0}^N s_1[i]$ .

2) *Correlation Coefficient*: This is a measure of the existence of a linear dependency between two time-series measurements. It takes the quotients of the covariance and variance of the individual measurements into account

$$CC(s_1, s_2) = \frac{COV(s_1, s_2)}{\sqrt{VAR(s_1)VAR(s_2)}} \quad (2)$$

where  $COV(s_1, s_2) = (1/N) \sum_{i=1}^N (s_1[i] - \bar{s}_1)(s_2[i] - \bar{s}_2)$  and,  $VAR(s) = (1/N) \sum_{i=1}^N (s[i] - \bar{s})^2$ .

3) *Cross Correlation*: Is an indication of the existence of a correlation between two time series measurements  $s_1[i]$  and  $s_2[i]$ , where  $s_1$  and  $s_2$  may represent either the same type of movement measured at different locations, or a single movement measured at the same location but at different times. In case  $s_2(t)$  represents  $s_1(t + \tau)$ , where  $\tau$  is a specified time lag, the two variables are usually not statistically independent, and large cross correlations between  $s_1$  and  $s_2$  can result. Mathematically, the cross correlation,  $XC$ , is described as follows:

$$XC(n) = \sum_{\tau=-\infty}^{\infty} s_1[\tau]s_2[n + \tau]. \quad (3)$$

Similarly, the autocorrelation is the cross correlation of a time-series measurement with itself.

## B. Frequency-Domain Features

In order to extract frequency-domain features, the FFT is computed. It has a computational complexity of  $n \log(n)$ , where  $n$  is the number of samples. It is the fastest transformation process between time and frequency domain. However, the FFT does not reveal how fast the signal's frequency changes over time. Therefore, it is necessary to divide the time-series measurements into different, short-duration windows. In order to avoid frequency leakages at the two edges of each window, it is customary to overlap neighbor frames, as a rule the overlap is 25% to 50% [6], [19]. Afterwards, FFT is performed on each of these windows. This process is called Short Time Fourier Transformation (STFT). Once the FFT or STFT coefficients are obtained, various more expressive features can be extracted. Below is a short summary of some of them.

1) *Spectral Centroid*: This represents the balancing point of the spectral power distribution

$$C_t = \frac{\sum_{n=1}^N M_t[n] \cdot n}{\sum_{n=1}^N M_t[n]} \quad (4)$$

where  $M_t[n]$  is the magnitude value of the spectrum at position (frequency)  $n$ .

2) *Band Energy*: Expresses the energy of the subbands normalized by the total energy of the signal.

3) *Spectral Roll-Off (SRO)*: Measures the frequency below which a certain amount of spectral energy resides. It measures the "skewness" of the spectral shape [20]. Mathematically, it is expressed as

$$SRO = \sum_{n=1}^{R_t} M_t[n]. \quad (5)$$

The sum of the spectrum up to the roll-off frequency signifies  $\alpha\%$ <sup>1</sup> of the total spectrum.

4) *Spectral Flux*: This is defined as the difference between the magnitude spectra of successive frames [20]

$$SF = \sum_{n=1}^N (N_t[n] - N_{t-1}[n])^2 \quad (6)$$

where  $N_t[n]$  is the normalized magnitude value of the  $n^{th}$  position of the  $t^{th}$  frame.

5) *Maxima*: This measures the similarity in transition of the first  $n$  Maxima of an expressive feature. The measurements should have the same number of samples and these samples will be transformed into the frequency domain. The  $N$  discrete samples produce  $N$  discrete frequency coefficients. Since the Maxima is dependent on the dominant frequency components, they can be considered to contain typical structural characteristics. Equation (7) expresses how the corresponding frequencies can be calculated from the discrete spectrum

$$\omega_n = 2\pi \cdot \frac{n}{(2N + 1)} \cdot \frac{1}{T}. \quad (7)$$

Hence, two sets of measurements exhibit similarity if their first  $n$  Maxima exhibit strong similarities. This applies to the magnitude as well as the position of the Maxima.

## IV. METHODOLOGY

We used SunSpot sensor nodes, each containing three accelerometers that are aligned orthogonally, along the  $x$ -,  $y$ -, and  $z$ -axes. For the detail description of the nodes employed in the experiment, the reader is referred to [21]. The nodes maximum sampling frequency was 350 Hz (i.e., maximum sampling at  $\approx 3$  ms). The average communication delay between the sensor nodes and the remote sink was 4 ms. To minimize packet loss, all data are logged to a remote computer at an average frequency of 150 Hz. Sampling was carried out in a controlled environment to make sure that reading of all sensors takes place in a similar setting. Throughout the measurement, the RAM memory did not overflow so that there was no local congestion.

The measurements were subject to three types of errors (uncertainties): calibration, random placement (some sensors were not placed near the event of interest), and random orientation (alignment).<sup>2</sup> Fig. 1 displays an example of a combined error that arises from orientation and calibration differences between two sensor nodes measuring one and the same movement. The aim of this paper is to quantify and qualitatively describe the degree of indifference of the features considered to these types of errors.

### A. Measurements

In this subsection, we describe the different types of movements we considered for our analysis. Each of these movements

<sup>1</sup>In speech recognition the spectral-roll is expressed as  $SRO_{\text{speech}} = (\alpha/100) \sum_{n=1}^N M_t[n]$ , where  $\alpha$  as a rule equals 85%.

<sup>2</sup>There are, of course, other sources of errors, but we do not consider them here.

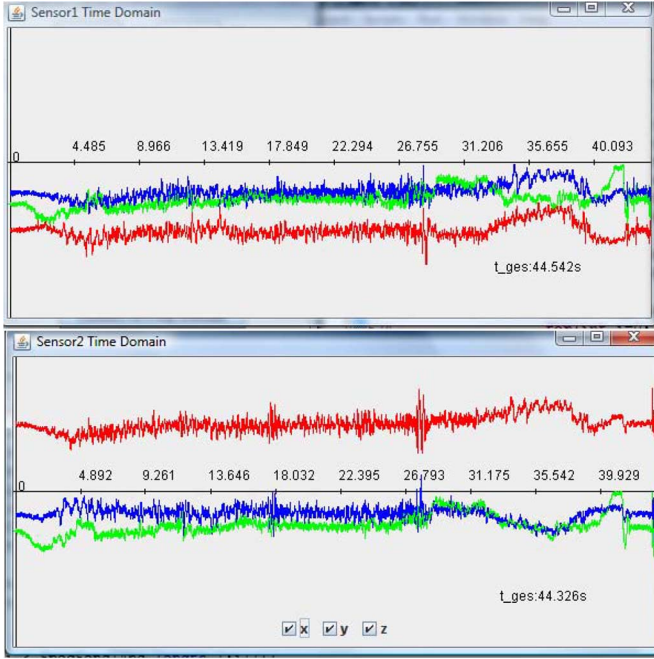


Fig. 1. Snapshot of two time-series measurements taken by two different nodes measuring the same movement. The two sensor nodes have different calibrations and placements. In both graphs, the x-axis represents time, and the y-axis the magnitude of acceleration.

was captured by at least two SunSpot sensor nodes in different placement settings. The nodes directly transmit the raw data to a nearby base station. Each packet was acknowledged; in case of a lost or corrupted packet, retransmission was requested and granted.

#### 1) Movement of People:

- A single person: Two sensor nodes were placed at the opposite thighs and wrists of a walking person. The measurements were taken from calibrated sensors, but each time, they were placed randomly and they had random orientation.<sup>3</sup>
- Two persons climbed up and down a staircase, side by side (without synchronization of steps): Two sensor nodes were randomly placed in the thighs and wrists of each test person. The persons climbed a staircase of 18 steps side by side. After the climb was over, the persons turned back and climbed down the staircase. The two types of movements were labeled independently.
- Two persons climbed up and down a staircase one after another (without synchronization of steps): Two sensors were attached to the test persons as described in the previous setting, but this time, one person was walking in front of the other.
- Dancing: Two sensor nodes were attached to the thighs of the test persons. They danced for about a minute. The dance was a free and uncoordinated movement (with no premeditated pattern), but the occurrence of certain body contacts signifying some distinct movements were labeled.
- Couch: Two nodes were attached at the thighs of two test persons. The experiment measured actions and reaction

<sup>3</sup>Orientation errors were kept below 90° throughout the experiment, however.

TABLE II  
OVERVIEW OF THE MEASUREMENTS DURATION AND DATA SIZE

Measurement	Duration [ms]	Size [Kb]
Two persons: side by side	45826	365
Two persons: one after another	39222	329
Dancing	48586	426
Couch	45848	418
Highway drive: back & front seats	26776	235
Highway drive: with seat belts	44542	411
Highway drive: (acceleration lane)	62849	575
City drive	64172	572

movements whenever the test persons sat down on and got up from a couch. During the analysis of the measurements, other types of movements (drinking a coffee from a mug; writing, etc.) were deliberately filtered out.

#### 2) Movement of Car:

- Highway drive: a series of measurements were taken from two accelerometer sensor nodes that were placed on the back and front seats of a car during a highway drive, with an average speed of 120 kmph. The sensors themselves were untethered. Measurements from these sensors were taken to investigate the existence of correlation between the different parts of a car even though these parts react differently to accelerations and brakes as well as to the irregular surface on which the car drives.
- Highway drive: Two sensors were attached to the seat belts of the driver and the front passenger and were very close to the center of gravity. However, exact alignment to the center of gravity was not made.
- Free drive: In this setting, the sensors were placed in the front cabin of the car; they were calibrated and aligned to the seats of the driver and the front passenger. This measurement was used as a reference to the measurements taken in the previous settings.
- City drive: the sensors were placed inside the glove box at the front cabin of the car, untethered.

Table II summarizes the durations and size of the measurements obtained.

## V. ANALYSIS AND RESULTS

### A. Time-Domain Features

1) *Zero-Crossing Rate*: Comparison of the zero-crossing rates of the correlated and uncorrelated movements is a straightforward and inexpensive process. However, a high zero-crossing rate may indicate a high frequency measurement as well as a measurement which is highly corrupted by noise. As expected, all the measurements we took exhibit strong dissimilarities due to calibration error. Even those sensors which had similar orientation and placement and measured the same movement resulted in  $\pm 40/s$  zero-crossing rate due to calibration error. Substituting the zero-crossing rate by a mean value-crossing rate resulted in a deviation ranging between 2 and 15/s for individual axes; and only 0–2/s for the absolute acceleration values of the individual measurements.<sup>4</sup> Cross test

<sup>4</sup>The absolute acceleration value,  $Abs$  is given as:  $Abs = \sqrt{x^2 + y^2 + z^2}$ .



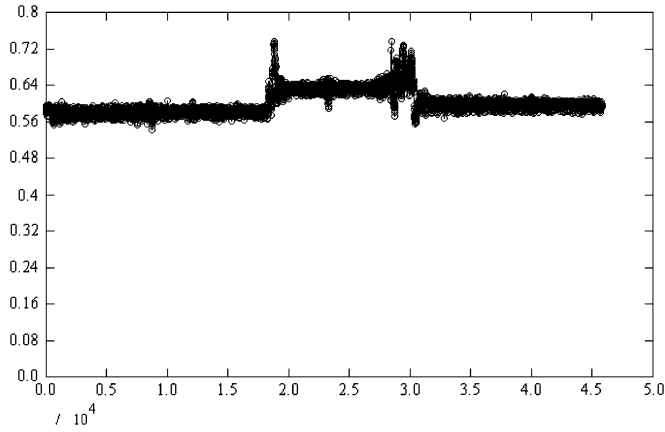


Fig. 2. Autocorrelation of the absolute acceleration values of the measurements taken from the first person.

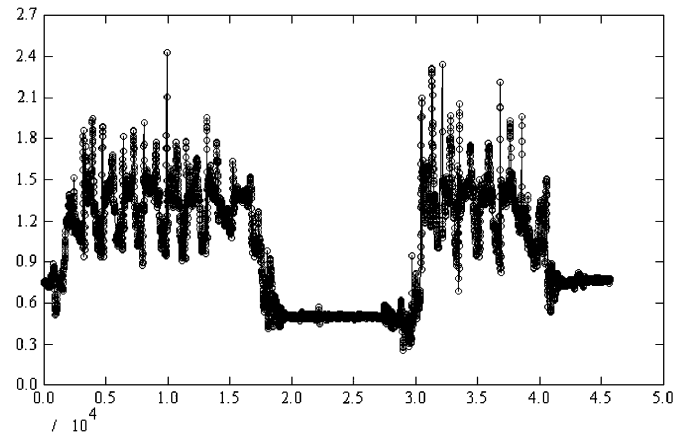


Fig. 3. Autocorrelation of the absolute acceleration values of the measurements taken from the second person.

of uncorrelated measurements, on the other hand, resulted in an overall deviation that ranges from 7 to 50/s for the individual axes; and 5–30/s for the absolute acceleration value.

2) *Mean Value*: The mean value is perhaps the simplest and the least computationally intensive feature. The deviation in value between similar (correlated) fast movements (car) for individual axes was  $\pm 0.5$  g. The absolute mean value of these movements, even for randomly oriented measurements, has a markedly small deviation, i.e.,  $\pm 0.05$  g. On the contrary, the deviation in human movements was high. For example, the average deviation in the absolute acceleration value of similar (correlated) measurements was  $\pm 0.4$  g.

3) *Correlation Coefficient*: The correlation coefficient in this context is a measure of the degree of similarity in movement patterns, i.e., how similar are the temporal structure of two sets of measurements. Once again, the measurements that are taken from the different parts of a car during the same drive yield a correlation between 0.4 and 0.9. On the other hand, the correlation between dissimilar measurements was  $\{0.2 \pm 0.1\}$ . Exceptions to this were the measurements taken from the sensors which laid fixed on the front and back seats of the car. In which case, it was not at all possible to establish correlation. The correlation coefficients related to human movements are markedly small, i.e., in the ranges of 0.2 and 0.3 for measurements representing similar movements. The correlation coefficients of the absolute acceleration values of all the measurements are notably better. The scenario that produced the highest correlation was the movements of people on the couch. Figs. 2 and 3 show the absolute values of the correlation of the readings taken from the thighs of the two people as described in Section IV-A. As can be seen, the movement of one person produced a reaction movement in the other person.

The problem with autocorrelation is its requirement of time synchronization. In the absence of time synchronization, the correlation coefficients of all movements were very small.

4) *Cross Correlation*: The cross correlation is used to measure the magnitude of the time offset between two time-series measurements. This is particularly useful to model correlated movements that cannot be compared piecewise. A typical example is the correlation between the movements of people climbing up and down a staircase without synchronizing their

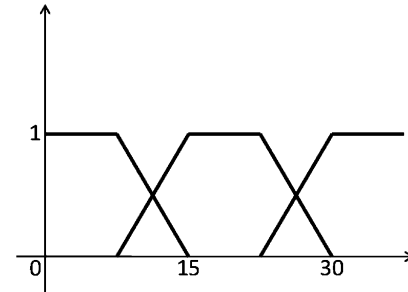


Fig. 4. Membership function for establishing the fuzzy set of the MCR feature.

steps. Intuitively, the movements should demonstrate strong correlations. However, due to the anatomy of the persons and the relative distance between the two people (back and forth), a sample-by-sample correlation was irrelevant. With the help of the cross correlation, we were able to detect and correct a mean offset value of  $\pm 1.4$  s over all the three axes of the accelerometer sensors.

5) *Autocorrelation*: Another approach to deal with measurements that cannot be compared piecewise is to test the linear correlation between two autocorrelation functions. This feature performs very well to test uncorrelated movements instead of correlated movements. For example, comparison of the autocorrelation of the measurements taken during the staircase movements revealed that the structure of the autocorrelation functions was almost identical even though their magnitude at any given location was different.

Table V summarizes the different time-domain features we considered. The features test both the presence and the absence of correlation between different measurements. In the second and the third column, the boundary signifies the width of the variance of similarity.

For a qualitative analysis, we defined fuzzy sets to model the uncertainty associated with each feature. We used empirical observations to define the membership function of the fuzzy sets for each feature. In each case, the membership function is defined as a trapezoid function in which the beginning and the end of the trapezoids were taken from the experiment results. The trapezoid function for the MCR is shown in Fig. 4. Equation (8)

TABLE III  
MEMBERSHIP FUNCTION DEFINITION FOR THE MCR

Description	Membership Function
Low	$\mu_l : T(x, 0, 4, 1, 11), m_a = 0; m_b = 15$
Midium	$\mu_m : T(x, 17, 22, 14, 14), m_a = 10; m_b = 30$
High	$\mu_h : T(x, 30, \infty, 14, 1), m_a = 25; m_b = \infty$

TABLE IV  
MEMBERSHIP FUNCTION DEFINITION FOR THE  
LINEAR CORRELATION COEFFICIENTS

Description	Membership Function
Low	$\mu_l : T(x, 0.15, 0.25, 0.05, 0.05), m_a = 0.1; m_b = 0.3$
Midium	$\mu_m : T(x, 0.35, 0.45, 0.1, 1.5), m_a = 0.25; m_b = 0.6$
High	$\mu_h : T(x, 0.65, 1.1, 0.19, 1.0), m_a = 0.5; m_b = 1.1$

TABLE V  
SUMMARY OF TIME DOMAIN FEATURES

Feature	Deviation (Correlated)	Deviation (Uncorrelated)	Uncertainty
ZCR/MCR (s.T)	0.0 - 15.0/s	7 - 50/s	Medium
ZCR/MCR (Abs.)	0.0 - 2.0/s	5.0 - 30.0/s	Low
Mean value (s.T)	$\pm 0.5g$	$\pm 0.5g$	Unsuitable
Mean value (Abs.)	$\pm 0.05g$	$\pm 0.5g$	Medium
Corr. coeffi. (s.T. m)	0.2 - 1	< 0.2	Medium/High
Corr. coeffi. (s.T. c)	0.4 - 1.0	< 0.4	Low
Legend s.T: Individual time series measurements (x, y, z) Abs.: The absolute values of the raw sensor measurements m: Human movements c: Car movements			

shows the expression of the membership function for this feature. Tables III and IV display the membership definitions of the MCR and the linear correlation coefficients

$$T(x, m_1, m_2, \alpha, \beta) = \begin{cases} 0 & x < m_a, x \geq m_b \\ 1 & m_1 \leq x < m_2 \\ \frac{(x-m_a)}{\alpha} & m_a \leq x < m_1 \\ \frac{(m_b-x)}{\beta} & m_2 < x < m_b \end{cases} \quad (8)$$

### B. Frequency-Domain Features

The frequency-domain features were extracted after transforming the time-series measurements *in their entirety* into frequency domain using the FFT. Exception to this is the short-time Fourier transform (STFT), in which case, the time series measurements were divided into several overlapping short frames before they were transformed into frequency-domain features.

All the frequency-domain features require preprocessing. The resource consumption of these steps is discussed in more detail in [6] and [19]. Additional to these processes, we carry out frequency normalization using a hamming window, so that the frequency resolution for each measurement is optimized.

1) *Maxima*: The  $n$ -maxima of a frequency spectrum was used to compare the dominant frequencies of different measurements. To obtain a significant size of representative frequency samples, first the  $i$ -th Maxima were summed up and divided by the total Maxima. Once this was done, comparison was made by selecting the  $n$ -th Maxima and observing the deviation from the average Maxima.

For human movements,  $n = 100$  was sufficient, while for car  $n$  should be in the order of 1000. This is because human movements contain low frequency components. The average distance between the first 100 Maxima of the individual axis for similar movements lied between 0.7 and 3 Hz, while it was between 0.7 and 3.5 Hz for the absolute value of the individual axes. On the other hand, the same distance for uncorrelated (dissimilar) movements was between 1.4 and 3.9 Hz for the individual axes and between 1.6 and 4.8 Hz for the absolute value. For car movement and with the distance between the first 1000 Maxima, the corresponding results were between 5–13 Hz for strong correlation and 20–30 Hz, for uncorrelated (individual dimensions); and 7–18 Hz and 11–25 Hz for absolute values.

2) *Energy*: The spectrum energy of a set of sensor readings reveals the spectrum's structure. In this context, the spectrum energy refers to the overall energy of the two readings being analyzed. To start with, the spectrum was divided into  $n$  subbands and the portion of energy in each band was normalized by the overall energy of the spectrum. Correlation test was performed subband by subband comparison on different measurements—the stronger the correlation between the measurements, the lesser the difference between the subband energies. As a result the average difference of the subband energies was used to measure correlation. In our analysis, the energy mass of similar movements of objects was between 0.006 and 0.1, while for uncorrelated, dissimilar movements, the mass was between 0.2 and 0.9. Human movement was very difficult to categorize with the energy mass as the range was not strikingly different for similar and dissimilar movements.

3) *Linear Correlation Coefficient*: Ideally, if the spectral structures of two sets of measurements are similar, then there is a strong correlation between them. The frequency-domain linear correlation coefficients examine this hypothesis. Indeed the measurements taken from different places during the same type of movement revealed the existence of a strong correlation (between 0.6 and 0.99). Unfortunately, we also observed that correlation coefficients of unrelated movements frequently yielded values above 0.6. The best explanation for this is that all types of movements have high frequency components which undermine the significance of the band-pass frequencies, which are distinct from movement to movement. As a result, a large portion of the curves are similar and can wrongly be interpreted as being correlated.

We attempted to reduce this effect by quantizing the measurements. Even though there was some improvement, linear correlation coefficients in the frequency domain are the feeblest features. Moreover, the quantization level was very much dependent on the measurements being compared or tested—the bigger the pick to pick individual amplitudes, the larger should be the quantization level.

We investigated the change of frequencies over time as a measure of correlation. For this, we used the STFT, which is computed by dividing the sensor measurements into several overlapping frames. Each frame is then Fourier transformed, and the complex result is added to a matrix, recording the magnitude and phase of each point in time and frequency domain. As a scaling factor, we summed up the correlation coefficients and divided them by the size of the frequencies being considered. The

test showed that a marked difference between the movements of people and the movements of cars. For human movements, the results were on the average between 0.1 and 0.3, which confirms the results we got in the time domain for the same data set. Cross tests of uncorrelated movements resulted in weighted linear correlation coefficients that ranged from 0.03 to 0.09. This much could not be achieved in the time domain for the same data sets.

The result of car related movements were even better. While the correlation coefficients for similar movements were between 0.2 and 0.6, for unrelated movements, these were between 0.02 and 0.1. This observation suggests that other frequency-domain analysis, such as coherence, can be more expressive if they consider STFT instead of the FFT.

4) *Spectral Roll-Off*: The spectral roll-off is another structural feature in the frequency domain in which only the Fourier transformation of the acceleration vectors was taken into account. In most cases, such as in speech recognition, it is usually customary to consider  $\alpha = 85\%$ . This, however, did not produce any significant difference between correlated and uncorrelated movements. Not unexpectedly, a significant portion of the energy of most movements was contained within the lower frequency components. Subsequently, we lowered down the value of  $\alpha$  to 60%. As a result, similar movements revealed a roll-off distance that ranged from 0–4 Hz while the roll-off distance for uncorrelated movements were between 2.5 and 10 Hz.

5) *Spectral Centroid*: The spectral centroid is similar to the “first  $n$ -Maxima” and indicates the relative location of the “center of gravity of the spectrum. It is computed as the weighted mean of the frequencies—the magnitudes of the frequencies being taken as weights. This scheme requires a precise knowledge of the movements being considered. The analysis was carried out by dividing the main frame into several subbands and the centroid of each subbands was independently computed and piecewise comparison was performed. The mean difference of the subband centroids,  $dis$ , is summarized by (9) as

$$dis = \frac{1}{N} \cdot \sum_{i=0}^N |SC_1(i) - SC_2(i)| \quad (9)$$

where  $SC_1$  and  $SC_2$  refers to the spectral centroid of sensor 1 and 2, respectively; and  $N$  is the number of subbands.

Regardless of the movement types, division of the entire spectrum into five equal segments resulted in a centroid distance that ranged from 0.2 to 0.5 Hz for related movements; and from 0.3 to 1.7 Hz for unrelated movements. By taking the absolute values of the spectrum, the related movements yielded a better correlation, the centroid being tighter than the previous, i.e., from 0.2 to 0.3 Hz.

6) *Spectral Flux*: The spectral flux is a measure of how quickly the power spectral changes. Ideally, similar movements should have a deviation of 0 flow. To compute the spectral flux of two measurements, both should have the same time duration. In the simplest case of considering the entire duration, we obtained a spectral flux that ranged from 0.0026 to 0.25 for similar movements—0.25 is rather the worst case. Otherwise, the spectral flux of similar movements was between 0 and 0.1. On the contrary, the spectral flux of unrelated movements varied from 0.2 to 1.0.

TABLE VI  
SUMMARY OF THE FREQUENCY-DOMAIN FEATURES

Feature	Deviation (Correlated)	Deviation (Uncorrelated)	Uncertainty
Maxima (s.T. m)	0.7 - 3Hz	1.4 - 3.9Hz	High
Maxima (Abs. m)	0.7 - 3.5Hz	1.6 - 4.8Hz	High
Maxima (s.T. c)	5 - 13Hz	20 - 30Hz	Medium
Maxima (Abs. c.)	7 - 18Hz	11 - 25Hz	Medium
Energy (Abs. m.)	0.0 - 0.02	0.02 - 0.9	Medium
Energy (Abs. c.)	0.0 - 0.1	0.2 - 0.9	Low
Corr. (FFT) s.T. Abs.	0.6 - 0.9	0.6 - 0.9	Very high
Corr. (STFT) Abs. m.	0.1 - 0.3	0.03 - 0.09	Low
Corr. (STFT) Abs. c.	0.2 - 0.6	0.02 - 0.1	Low
Spectral roll-off (Abs.)	0 - 4Hz	2.5 - 10.0Hz	Medium
Spectral centroid (s.T)	0.2 - 0.5Hz	0.3 - 1.7Hz	Low
Spectral Centroid (Abs.)	0.2 - 0.3Hz	0.3 - 1.7Hz	Low
Spectral flux (Abs.)	0.0 - 0.25	0.2 - 1.0	Low
Legend s.T: Individual time-series measurements (x, y, z) Abs.: The absolute values of the raw sensor measurements m: Human movements c: Car movements			

Table VI summarizes our observation for the frequency-domain analysis.

## VI. CONCLUSIONS

We investigated the expression power of several time- and frequency-domain features in the presence of calibration, placement and orientation errors. The measurements from which the features are extracted represent movements of humans (slow movements) and cars (fast movements) for various scenarios, both in calibrated and un-calibrated conditions. The time-domain features we considered were zero-crossing rate (mean-value crossing rate), correlation coefficients, and cross correlations. The frequency-domain features were Maxima and energy; correlation coefficients of FFT and STFT, spectral roll-off, spectral centroids, and spectral flux. We observed that the features extracted from the absolute values of the raw measurements were more robust to noise and calibration errors than the features extracted directly from the raw measurements of individual axes. The frequency-domain features that were least vulnerable to noise and exhibit the strongest expression power were the correlation coefficients of the absolute values of the STFTs. The features extracted from slow movements were in general prone to measurement errors.

In the future, we aim to extend our studies by employing different types of sensors. We have already collected a large amount of data with MicaZ sensors. This will enable us to evaluate how feature extraction can be affected by node architecture.

## REFERENCES

- [1] S. Kim, S. Pakzad, D. Culler, J. Demmel, G. Fennes, S. Glaser, and M. Turon, “Health monitoring of civil infrastructures using wireless sensor networks,” in *Proc. 6th ACM Int. Conf. Information Processing in Sensor Networks*, New York, 2007, pp. 254–263.
- [2] P. Mohan, V. N. Padmanabhan, and R. Ramjee, “Nericell: Rich monitoring of road and traffic conditions using mobile smartphones,” in *Proc. 6th ACM Conf. Embedded Network Sensor Systems*, New York, 2008, pp. 323–336.

- [3] M. Malinowski, M. Moskwa, M. Feldmeier, M. Laibowitz, and J. A. Paradiso, "Cargonet: A low-cost micropower sensor node exploiting quasi-passive wakeup for adaptive asynchronous monitoring of exceptional events," in *Proc. 5th ACM Int. Conf. Embedded Networked Sensor Systems*, New York, 2007, pp. 145–159.
- [4] S. A. Ballegaard, T. R. Hansen, and M. Kyng, "Healthcare in everyday life: Designing healthcare services for daily life," in *Proc. 26th Annu. SIGCHI Conf. Human Factors in Computing Systems*, New York, 2008, pp. 1807–1816.
- [5] G. Werner-Allen, K. Lorincz, M. Welsh, O. Marcillo, J. Johnson, M. Ruiz, and J. Lees, "Deploying a wireless sensor network on an active volcano," *IEEE Internet Comput.*, vol. 10, no. 2, pp. 18–25, Mar./Apr. 2006.
- [6] W. Dargie, "Adaptive audio-based context recognition," *IEEE Trans. Syst., Man, Cybern. A: Syst., Humans*, vol. 39, no. 4, pp. 715–725, Jul. 2009.
- [7] X. Chao, W. Dargie, and G. Lin, "Energy model for H2S monitoring wireless sensor network," in *Proc. 2008 11th IEEE Int. Conf. Computational Science and Engineering*, Washington, DC, 2008, pp. 402–409.
- [8] D. Ganesan, S. Ratnasamy, H. Wang, and D. Estrin, "Coping with irregular spatio-temporal sampling in sensor networks," *SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 1, pp. 125–130, 2004.
- [9] S. Lin, "Data Management and Data Analysis Techniques for Wireless Sensor Networks," Ph.D. dissertation, Univ. California, Riverside, CA, 2007.
- [10] C. Tang and C. S. Raghavendra, "Compression Techniques for Wireless Sensor Networks," pp. 207–231, 2004.
- [11] S. Bandyopadhyay, Q. Tian, and E. J. Coyle, "Spatio-temporal sampling rates and energy efficiency in wireless sensor networks," *IEEE/ACM Trans. Netw.*, vol. 13, no. 6, pp. 1339–1352, Dec. 2005.
- [12] E. F. Nakamura, A. A. F. Loureiro, and A. C. Frery, "Information fusion for wireless sensor networks: Methods, models, and classifications," *ACM Comput. Surv.*, vol. 39, no. 3, p. 9, 2007.
- [13] X. Dai, F. Xia, Z. Wang, and Y. Sun, "A survey of intelligent information processing in wireless sensor network," *Mobile Ad-Hoc Sensor Netw.*, pp. 123–132, 2005.
- [14] T. Huynh and B. Schiele, "Analyzing features for activity recognition," in *Proc. ACM 2005 Joint Conf. Smart Objects and Ambient Intelligence*, New York, 2005, pp. 159–163.
- [15] J. A. Ward, P. Lukowicz, G. Troster, and T. E. Starner, "Activity recognition of assembly tasks using body-worn microphones and accelerometers," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 10, pp. 1553–1567, Oct. 2006.
- [16] M. Engin, S. Demirağ, E. Z. Engin, G. Çelebi, F. Ersan, E. Asena, and Z. Çolakoğlu, "The classification of human tremor signals using artificial neural network," *Expert Syst. Appl.*, vol. 33, no. 3, pp. 754–761, 2007.
- [17] N. L. W. Keijsers, M. W. I. M. Horstink, and S. C. A. M. Gielen, "Ambulatory motor assessment in parkinson's disease," *Movement Disorder Soc.*, vol. 21, no. 1, pp. 34–44, 2006.
- [18] R. Marin-Perianu, J. Hurink, and P. Hartel, "A generalized clustering algorithm for dynamic wireless sensor networks," in *Proc. 2008 IEEE Int. Symp. Parallel and Distributed Processing with Applications*, Washington, DC, 2008, pp. 863–870.
- [19] W. Dargie and T. Tersch, "Recognition of complex settings by aggregating atomic scenes," *IEEE Intell. Syst.*, vol. 23, no. 5, pp. 58–65, Sep. 2008.
- [20] A. J. Eronen, V. T. Peltonen, J. T. Tuomi, A. P. Klapuri, S. Fagerlund, T. Sorsa, G. Lorho, and J. Huopaniemi, "Audio-based context recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 14, no. 1, pp. 321–329, Jan. 2006.
- [21] W. Dargie, "Analysis of time and frequency domain features of accelerometer measurements," in *Proc. 3rd IEEE Workshop on Performance Modeling and Evaluation of Computer and Telecommunication Networks (PMECT 2009)*, 2009.



**Walteneus Dargie** (M'08) received the B.Sc. degree from the Nazareth Technical College, Nazareth, Ethiopia, in 1997, and the M.Sc. degree from the Technical University of Kaiserslautern, Kaiserslautern, Germany, in 2002, both in electrical engineering, and the Ph.D. degree in computer engineering from the Technical University of Dresden, Dresden, Germany, in 2006.

He is a Researcher/Lecturer at the Technical University of Dresden. Previously, he was a Researcher with both the Department of Electrical Engineering and Computer Science, University of Kassel, Kassel, Germany, between 2002 and 2005 and the Fraunhofer Institute of Experimental Software Engineering, Kaiserslautern, Germany, between 2002 and 2003. His research interests include autonomous computing, context-aware computing, wireless networks, and digital signal processing.



**Mieso K. Denko** (SM'07) received the M.Sc. degree from the University of Wales, U.K., and the Ph.D. degree from the University of Natal, Natal, South Africa, both in computer science.

Currently, he is with the School of Computer Science, University of Guelph, Guelph, ON, Canada. His current research interests include wireless networks, mobile and pervasive computing, wireless mesh networks, body area sensor networks and network security. He has co-edited three books in the areas of pervasive computing, autonomic computing and wireless networking.

Dr. Denko is a founder/co-founder of a number of ongoing international workshops & symposia and served as Program Chair/Co-Chair for several IEEE/ACM/IFIP international conferences. Currently, he is Guest Co-Editor of Special Issues for several journals, including the ACM/Springer *Mobile Networks and Applications* (MONET) and the IEEE *SYSTEMS JOURNAL*. He is an Editorial Board Member of international journals including, the *International Journal of Smart Homes* (IJSH), the *Journal of Ubiquitous Computing and Communications* (UBICC), and is an Associate Editor of the *International Journal of Communication Systems* (IJCS), Wiley, *Security & Communications Network* (SCN), Wiley, and the *Journal of Ambient Intelligence and Humanized Computing* (JAHC), Springer. He is a Senior Member of the ACM.

# Adaptive Audio-Based Context Recognition

Waltenegus Dargie, *Member, IEEE*

**Abstract**—Context recognition is an essential aspect of intelligent systems and environments. In most cases, the recognition of a context of interest cannot be achieved in a single step. Between measuring a physical phenomenon and the estimation or recognition of what this phenomenon represents, there are several intermediate stages which require a significant computation. Understanding the resource requirements of these steps is vital to determine the feasibility of context recognition on a given device. In this paper, we propose an adaptive context-recognition architecture that accommodates uncertain knowledge to deal with sensed data. The architecture consists of an adaptation component that monitors the capability and workload of a device and dynamically adapts recognition accuracy and processing time. The architecture is implemented for an audio-based context recognition. A detail account of the tradeoff between recognition time and recognition accuracy is provided.

**Index Terms**—Audio-signal processing, context awareness, context reasoning, context recognition, context-recognition accuracy, context-recognition time.

## I. INTRODUCTION

CONTEXT-AWARENESS is an essential aspect of intelligent computing systems. It deals with the collection of raw data from various sensors which are placed in various places and devices and the processing of these data to extract meaningful higher level activities and social situations (higher level contexts). The successful recognition or estimation of higher level contexts enables intelligent computing systems to perform the following functions: 1) seamlessly adapt to a perceived change; 2) augment human perception; and 3) provide relevant services in a proactive manner.

Several smart systems and collaborative environments have been proposed in the recent past. For example, the iBadge [1] wearable system monitors the social and individual activities of children in a nursery school. It incorporates sensing, processing, communication, and actuating units. The sensing unit includes a magnetic sensor, a dual-axis accelerometer, a temperature sensor, a humidity sensor, a pressure sensor, and a light sensor. It includes also an ultrasound transceiver and an RF transceiver for position and distance estimations. The processing unit includes speech and sensor data processing. A server side application assists a teacher by receiving and processing location, orientation, ambient, and audio contexts from the iBadge to determine the social and learning status of a child. The location and orientation contexts are used to

determine whether a child is isolated or associates with other children. The audio context is used to determine whether a child is sociable or aggressive.

Dargie and Tersch [2] use acoustic signals and audio digital-signal processing to determine more than 20 different activities in a university campus. The context-recognition process involves modeling frequency-domain audio features and building a Bayesian network. Similarly, Dargie and Hammann [3] use Bayesian networks to estimate the whereabouts of a mobile user. The Bayesian network models stochastic features of data taken from various sensors (humidity, temperature, and light).

Likewise, Magee *et al.* [4] introduce the nonintrusive communication-interface system called EyeKeys. It runs on an ordinary computer with a video input from an inexpensive Universal Serial Bus camera and works without special lighting. EyeKeys detects and tracks the person's face using multiscale template correlation. The symmetry between left and right eyes is exploited to detect if the person is looking at the camera or to the left or to the right side. The detected eye direction is used to control applications such as spelling programs or games.

Several system architectures have been proposed to develop intelligent systems and environments. The architecture of Dargie and Tersch [2] consists of raw-data extraction, atomic-feature extraction, atomic-scene recognition, and context recognition. The architecture employs a knowledge base to model various everyday human activities. The iBadge system discussed earlier is built with the Sylph architecture [1], which consists of sensor modules, a proxy core, and a service-discovery module. Wang *et al.* [5] propose the semantic space framework that consists of context wrappers, a knowledge base, aggregators, a context-query engine, and a context reasoner. Chen *et al.* [6] propose the CoBra middleware, which consists of a knowledge base, a context-reasoning engine, a context-acquisition module, and a policy-management module. Similarly, Korpipää *et al.* [7] propose a distributed architecture for context recognition and management. The architecture consists of a context manager, a resource server, a context-recognition service, a change-detection service, and a security service. The context manager shields applications from the concern of context acquisition by functioning as a central server. All the other services post their output to it.

There is a remarkable similarity between the proposed architectures. First of all, they all support the separation of context acquisition from context usage. This is done by providing context widgets, context wrappers, context-acquisition modules, and context-query engines. Second, they support the presentation of a context at various abstraction levels—this is the task of interpreters, aggregators, and reasoning engines. Third, they support the dynamic binding of context sources by introducing service-discovery mechanisms.

While these architectures identify the conceptual steps of a context recognition, they are, however, higher level, in that

Manuscript received January 8, 2008; revised May 8, 2008 and October 1, 2008. First published April 24, 2009; current version published June 19, 2009. This paper was recommended by Associate Editor Y. Lin.

The author is with the Technical University of Dresden, 01062 Dresden, Germany (e-mail: waltenegus.dargie@tu-dresden.de).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TSMCA.2009.2015676

they rarely address the associated system complexity. System complexity is rather a crucial issue for several practical reasons. For example, the scope and usefulness of a context of interest depends on its timeliness and accuracy—both metrics being application-specific. Furthermore, the workload of the intelligent system, which is a dynamic aspect, influences both metrics, and itself depends on the available resources (such as the energy reserve, CPU speed, communication bandwidth, active memory, and storage). Variable system resources, in turn, directly affect the way sensed data can be obtained, processed, and communicated. Subsequently, a context-recognition architecture should take these dynamic aspects into account.

We provide an adaptive context-recognition architecture which has two essential aspects.

- 1) It takes the workload of a device into account and adapts the accuracy and duration of a context-recognition process.
- 2) It takes the capability of a device into account to select a suitable complexity class. This enables us to recognize a context of interest on heterogeneous devices.

The implementation of the architecture will be demonstrated for auditory-based context recognition. In earlier reports [3], [8], a part of the architecture (i.e., without the adaptation component) was implemented to recognize the whereabouts of a mobile user (rooms, corridors, or outdoor). The raw data were obtained from temperature, humidity, and light sensors.

The rest of this paper is organized as follows. In Section II, we present related work. In Section III, we present the conceptual architecture for computing context as an abstraction of real-world settings. In Section IV, a scenario is given for which the architecture is implemented. In Section V, we report the implementation details of the conceptual architecture. In Section VI, we provide a detailed account of the adaptation aspect of our architecture. In Section VII, we discuss our experience and provide comparisons of our result with previous results and close this paper with concluding remarks.

## II. RELATED WORK

### A. Adaptive Architecture

As far as adaptive context recognition is concerned, to the best of our knowledge, previous contribution is limited in this area. The adaptive context-recognition approach of Nam *et al.* [9] and Young *et al.* [10] focus on filter fusion to support a robust face recognition under uneven illumination (image processing). The system's working environment is learned, and the environmental context is identified (bright, normal, or poor illumination). Based on the initial context, a group of classifiers that are most likely to produce accurate output is generated for each environmental context. A combination of the results of multiple classifiers is determined using a t-test decision model.

Laasonen *et al.* [11] propose an adaptive framework for identifying the whereabouts of a mobile user from cellular-network data. Adaptation is defined as a dynamic thread-off between accuracy and resource consumption. The authors define three concepts (bases, areas, and routes) on the basis of which the complexity of a context-recognition process is estimated. Past and present locations, as well as mobility information, are used to reduce the resource consumption of a context-recognition task.

Stäger *et al.* [12] provide an empirical design process for audio-based context recognition. The process is a result of examining the tradeoff between power consumption and context-recognition accuracy. Given a hardware and its nominal power-consumption profile, the process tunes audio parameters (sampling rate, frame size, size of feature vector, etc.) to satisfy the power-consumption constraint. Based on this design guideline, they developed a wearable context-aware system that recognizes the activity of a user in a kitchen. The sources of audio data are a microwave, a coffee maker, a hot-water nozzle, a coffee grinder, and a water tap.

### B. Acoustic Context Recognition

Most existing or proposed auditory-based context-recognition schemes focus mainly on computational aspects, namely, on the accuracy, processing time, and power consumption of a context-recognition task.

Even though auditory-based context-recognition shares several similarities with speech recognition, there are also notable differences. For example, in speech recognition, knowledge of human perception (tone, pitch, loudness, etc.) is useful to disambiguate an uttered speech. This is possible because of the following conditions: 1) the speaker is not far from the microphone and speaks sufficiently clearly and loudly, and 2) there is no significant hindrance between the speaker and the microphone. This is not the case in auditory-based context recognition.

To begin with, the amplitude of the audio signal representing a user's surrounding is not appreciably large, since the audio sources are usually far away from the user (microphone). Second, the device in which the microphone is embedded can be hidden inside a suitcase or a pocket. Third, whereas the frequency of interest in speech recognition is well below 4 kHz, the signal collected from a user's surrounding may incorporate frequencies that are up to and above 10 kHz. All these facts should therefore be taken into account when designing an auditory-based recognition system.

Eronen *et al.* [13] identify time- and frequency-domain features, as well as stochastic features, to classify various everyday outdoor and indoor scenes (streets, restaurants, offices, homes, cars). They report that, by using Mel-frequency cepstral coefficients (MFCCs) and hidden Markov Models (HMMs), they were able to achieve a recognition accuracy of up to 88%. The recognition accuracy as a function of the test-sequence length appears to converge after about 30–60 s. Interestingly, they report that human's recognition accuracy of the same data set was 82% with an average reaction time of 14 s.

Korpipää *et al.* [14] employ a naive Bayesian classifier and an extensive set of audio features derived partly from the algorithms of the MPEG-7 standard. The classification is based mainly on audio features measured in a home scenario. With a resolution of 1 s in segments of 5–30 s and using leave-one-out cross validation, they achieve a recognition rate of 87% of true positives and 95% of true negatives. The result is averaged over nine 8-min scenarios containing 17 segments of different lengths and nine different contexts. The reference accuracies measured by testing training data are 88% (true positive) and 95% (true negative), suggesting that the model is capable of covering the variability introduced in the data on purpose.

Reference recognition accuracy in controlled conditions is 96% and 100%, respectively.

Ma *et al.* [15] also employ HMMs on MFCCs to recognize ten auditory scenes. By varying the hidden states of the HMMs, they achieve different recognition rates. For example, with just three states, the classifier achieves a context-recognition accuracy of 78%, while with 15 states, the recognition accuracy reaches 91.5%. Remarkably, the authors observe a decline in context recognition for higher hidden states. Smith *et al.* [16] extend the work of Ma *et al.* [15] by introducing a belief-revision mechanism to improve the recognition rate (92.27%) and to increase the number of contexts that can be recognized (namely, 12).

### C. Summary of Related Work

This paper is more similar to the work of Eronen *et al.*, Korpipää *et al.*, and Stäger *et al.* There are, however, significant differences. First, while they offer no reusable and extensible system architecture, we provide one that can be used beyond audio-based context recognition. Second, even though they extensively investigate the impact of spectral parameters on recognition accuracy, they do not exploit knowledge of the dynamic workload of a device to support adaptation (Stäger *et al.*, for instance, demonstrate the influence of power consumption on context recognition, but runtime power-aware context recognition is not supported). Third, they do not take recognition time into account, which is rather a vital aspect, as the relevance of a piece of context is determined by its timeliness.

This paper complements these approaches by providing an adaptation component. Depending on the capability and workload of a device it performs the following tasks: 1) determines how much resource should be made available for a context-recognition task, and 2) informs the user about the expected processing time and accuracy.

## III. ARCHITECTURE

Recognition of the social and conceptual settings in which computing devices operate cannot be captured in a single step. Between the measuring of audio signals and the recognition of what these signals represent, there are normally several intermediate stages. These stages consume significant resources. Defining different abstract stages enables rapid prototyping and reuse of components. If required, this approach enables also gradual implementation. Subsequently, we provide a conceptual architecture for a context recognition. Its basic differences from existing or proposed architectures can be summarized as follows.

- 1) The provision of a belief (uncertain knowledge) modeling component.
- 2) The provision of an adaptation component that exploits knowledge of the resource profile and dynamic workload of a device when computing a context.

To better present the architecture, we separate the recognition components from the adaptation component. The recognition component is shown in Fig. 1, while the adaptation component is shown in Fig. 2. The recognition part of the architecture consists of a set of primitive context servers (PCSs), an aggregator, an empirical ambient knowledge (EAK) component, and a composer.

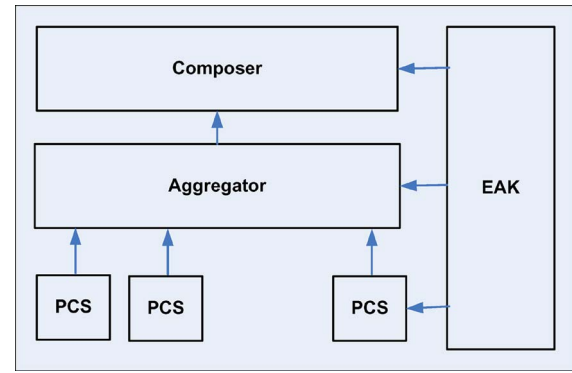


Fig. 1. Conceptual architecture for computing a context.

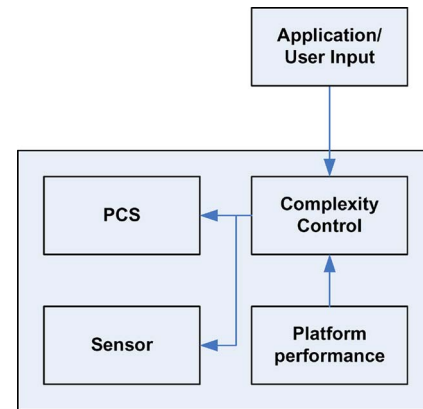


Fig. 2. Adaptation component that ensures the recognition of a context on several devices.

### A. Primitive Context Server (PCS)

A PCS abstracts from other components (such as an aggregator) the details and complexities of extracting a meaningful feature or an atomic context from a physical sensor. This feature is not application-specific or situation-specific and can be shared by multiple applications or situations. It refers to a single aspect of a certain phenomenon or a real-world object.

### B. Aggregator

A piece of context obtained from a single source may not be sufficient to appropriately model a real-world situation [17]. The real world is far too complex to be captured in complete detail in this way. Aggregation deals with the association, correlation, and combination of data from single or multiple sources to achieve a refined estimation. Subsequently, an aggregator gathers and processes data from multiple sensors which are spatially and temporally related. The outcome is a feature or a set of features that is (are) meaningful for a single application or a set of applications.

### C. Empirical Ambient Knowledge (EAK)

A prior knowledge of entities (places, devices, persons, etc.) is useful both for modeling a situation of interest and for appropriately interpreting sensor measurements. In most cases, this knowledge is taken as a fact however incomplete. However, as far as dealing with physical sensors is concerned, whatever

knowledge we have cannot be taken as fact but as an uncertain knowledge or belief. Here are some examples of beliefs.

- 1) Human thermal and humidity perception varies from season to season. A temperature ranging from 20 °C to 23.6 °C is perceived as comfortable in winter, while in summer, the range from 22.8 °C to 26 °C is perceived as comfortable. Likewise, a relative humidity ranging from 30% to 60% is perceived as comfortable in winter, while in summer, the range from 40% to 60% is perceived as comfortable.
- 2) In winter, a person is more sensitive to drought than in summer. Hence, the acceptable air velocity inside a room should be below 0.15 m/s, while in summer, it could be up to 0.25 m/s.

The above beliefs can be modeled as uncertain knowledge because they may hold true for most places, but they must not be taken as fact. For these reasons, our architecture separates facts from beliefs. The EAK associates conditional probabilities (or fuzzy membership functions, basic probability mass functions, etc.) to describe the degree of truthfulness of the uncertain knowledge it stores.

#### D. Composer

The composer computes a higher level context (setting) by classifying freshly acquired evidence from sensors according to the facts and beliefs stored in the EAK.

#### E. Adaptation

The problem with the recognition architecture is that, in itself, it does not guarantee equal support of all users independent of their end devices, i.e., devices such as laptops, mobile phones, and PDAs. It also does not take into account the dynamic workload of a device. To address these concerns, we have introduced an additional adaptation component that can be plugged into the architecture. It consists of two subcomponents, namely, the platform-performance monitor and the complexity control. This is shown in Fig. 2.

1) *Platform-Performance Monitor*: A platform has a static and a dynamic aspect. Its static aspect refers to its nominal resource profile, maximum available power, processor speed, memory, networking capability, storage, etc. Its dynamic aspect refers to its present workload and remaining resources. Both aspects should be taken into account to perform context recognition, since the accuracy, as well as the processing time, depends on these aspects. The platform-performance monitor provides the complexity control with an updated information regarding the platform's present workload and available resources.

2) *Complexity Control*: Context recognition is a tradeoff between recognition accuracy and processing time. The complexity control receives from the consumers of a context (a user or an application) an upper and lower threshold on these two recognition metrics and allocates resources for the recognition of the context accordingly. If a context recognition takes a processing time below a lower threshold, it increases the complexity level of the process to improve accuracy (i.e., by sacrificing more resources). If, on the other hand, a context-recognition process exceeds the upper threshold time, it reduces the complexity in favor of reducing the processing time below the specified upper bound. The complexity itself is determined

by a set of parameters. The complexity control dynamically adjusts the accuracy and processing time according to the actual workload of the device.

In Section VI, we will give a detailed explanation of an audio-based context recognition and the parameters that can be tuned to adjust recognition accuracy and processing time. Parameter tuning will be made according to the user's requirement and the device's capability and present workload.

### IV. SCENARIO

We implemented the conceptual architecture for audio-based context recognition. The implementation automates the recognition of various human activities in two different settings.

The first setting is a university campus. At the Faculty of Computer Science (Technical University of Dresden), a Chair occupies several rooms and uses them for different purposes. Some of these rooms are offices, laboratories, a conference room, a library, and a kitchen. There are also other rooms which are shared with other Chairs. The employees may be interested in some of these rooms for various activities including project meetings, thesis presentation, student consultations, impromptu chats among each other, brief discussions with visitors, celebration of a birthday party (graduation), etc.

Some Chairs have online room-reservation systems, but there are times when these systems are not flexible enough. This is because employees should reserve a room well ahead of the intended purpose. Our aim is to complement a room-reservation system and not to replace it. For any impromptu activity, an employee creates a query request in order to find out which rooms are free or occupied by less imperative activities (such as bilateral discussions or a casual chat). The system responds to the query by returning the names and locations of the rooms that satisfy the request without compromising privacy, i.e., without actually providing specific contents pertaining to the activities.

The approach can also be useful in other places. For example, in several big companies, a significant number of employees are mobile and hold universal keys. The keys give them access into their company buildings anywhere in the world. A context recognition can be useful to dynamically locate meetings and seminars; kitchens, special occasions, unoccupied rooms, etc. Recognition of the activities discussed earlier can also be helpful to mobile devices to dynamically adapt to the environments wherein they operate. For example, a mobile phone can dynamically switch from a ringing mode to a vibration or silence mode when an employee enters into a meeting room or holds a presentation.

In the second setting, the activities in trains and trams are recognized. The intention is to help and simplify the task of a human controller. Given the size and the number of carriages in a train, a limited number of controllers may not be able to effectively monitor what is taking place in a passenger train. Dynamic activity recognition will simplify this task.

For the first setting, the contexts of interest are casual talk, party, group discussion, lecture (presentation), and quietness.<sup>1</sup> For the second setting, the contexts of interest are fighting (aggression), loudness (such as drunken sport fans shouting), casual talk, and quiet.

<sup>1</sup> A quiet room is assumed to be empty.



For the two scenarios, there are two different deployment strategies. In the first setting, microphones are carefully placed in different rooms, and the signals from these microphones are processed centrally by a resource-rich computer.<sup>2</sup> Other devices place a higher level query request to this computer in order to learn what is taking place in certain places or where certain activities are taking place. In the second deployment setting, the microphones embedded in mobile devices are used to gather data, and the mobile devices themselves process the data. The second deployment setting does not rely on the existence of any established sensing infrastructure. For both settings, however, adaptation is useful, as the processing time and accuracy are dependent on the capability of the device as well as its present or anticipated workload.

## V. IMPLEMENTATION

This section reports the implementation details of the conceptual architecture discussed in Section III for the scenario presented in Section IV.

### A. PCS

The PCS is implemented for abstracting the acquisition of audio signals from ordinary microphones which were embedded in ordinary laptop computers. The PCS can be configured or reconfigured at any time to determine the beginning, duration, sampling rate, and resolution of the audio signal being sampled. Once acoustic signals are sampled, it extracts time- and frequency-domain parameters.

In order to extract suitable features, the PCS divides the audio data stream into small time frames. This is useful to model a nonstationary signal as quasi-stationary. There should be a 25%–50% overlap between adjacent frames to compensate the loss of information due to frequency leakage. Frequency leakage occurs due to the abrupt separation of neighboring frames; as a result of which, high-frequency components will emerge at the edges of each frame. This should be removed by a process called windowing, i.e., each frame is multiplied by a window function that decays rapidly toward the edges.

The length of a frame is mostly between 10 and 50 ms, and it influences the recognition accuracy as well as the computation time. To extract temporal features, further processing is not necessary. To extract spectral features, however, at least two additional steps are necessary: frequency-leakage correction and fast Fourier transformation (FFT). After windowing, an FFT is applied on each frame to obtain the magnitude of the power spectrum of each frame.

### B. Aggregator

The aggregator extracts application- and domain-specific features from the time- and frequency-domain properties of each frame. The time-domain features include a frame's zero-crossing rate, bandwidth, band energy, and average energy. The frequency-domain features include spectral centroid, spectral roll-off, linear spectral energy, log-spectral energy, and

<sup>2</sup>Krysander and Frisk [18] propose an algorithm for optimal selection and placement of sensors to meet a diagnosis requirement specification concerning fault detectability and fault isolability.

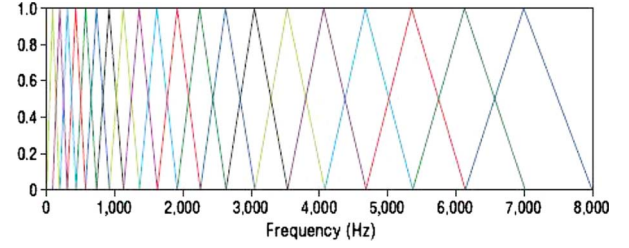


Fig. 3. Triangular filter bank for extracting the MFCCs.

spectral flux. For us, however, the most interesting features are the MFCCs. These are representations of the frequency bands which are Mel-scaled to approximate the human auditory perception more accurately than the linearly spaced frequency bands obtained directly from the FFT. The MFCCs enable context-recognition schemes to “perceive” their surroundings as humans would perceive theirs.

Therefore, the main task of the aggregator is to transform the linear-frequency spectrum obtained by the PCS into MFCCs. This is achieved by first scaling the frequency spectrum logarithmically using the so-called Mel filter bank  $H(k, m)$

$$X'(m) = \ln \left( \sum_{k=0}^{N-1} |X(k)| \times H(k, m) \right) \quad (1)$$

for  $m = 1, 2, \dots, M$ , where  $M$  is the number of filter banks and  $M \ll N$ . The Mel filter bank is a collection of triangular filters defined by the center frequencies  $f_c(m)$ , given as

$$H(k, m) = \begin{cases} 0, & \text{for } f(k) < f_c(m-1) \\ \frac{f(k) - f_c(m-1)}{f_c(m) - f_c(m-1)}, & \text{for } f_c(m-1) \leq f(k) < f_c(m) \\ \frac{f_c(m+1) - f(k)}{f_c(m+1) - f_c(m)}, & \text{for } f_c(m) \leq f(k) < f_c(m+1) \\ 0, & \text{for } f(k) \geq f_c(m+1) \end{cases} \quad (2)$$

The size of the triangular filter bank is variable. These filters are equidistant in the Mel-frequency domain, and there is a 50% overlap between adjacent filters. Fig. 3 shows 20 triangular Mel filters.

The center frequencies of the filter bank are computed by approximating the Mel scale with

$$\phi = 2595 \times \log_{10} \left( \frac{f}{700} + 1 \right). \quad (3)$$

The center frequencies on the Mel scale are given by

$$c(m) = m \Delta_\phi \quad (4)$$

where  $m = 1, 2, \dots, M$  and  $M$  is the number of filter banks.  $\Delta_\phi$  is described as

$$\Delta_\phi = \frac{\phi_{\max} - \phi_{\min}}{M + 1} \quad (5)$$

where  $\phi_{\max}$  is the highest frequency of the filter bank on the Mel scale and  $\phi_{\min}$  is the lowest frequency in the Mel scale.

Finally, the MFCCs are obtained by computing the DCT of  $X'(m)$  using

$$c(l) = \sum_{m=1}^M X'(m) \cos \left( l \frac{\pi}{M} \left( m - \frac{1}{2} \right) \right) \quad (6)$$

for  $l = 1, 2, \dots, M$ , where  $c(l)$  is the  $l$ th MFCC.

Finally, to reduce the effect of very low and very high MFCC components (at both edges of the Mel spectrum), the so-called *liftering* process<sup>3</sup> is performed. Equation (7) shows a typical liftering function.

$$C'(l) = \left( 1 + \frac{L}{2} \times \sin \frac{\pi l}{L} \right) \times C(l) \quad (7)$$

where  $L$  is a constant.

### C. Separation of Concern

There are two essential reasons for separating feature extraction (by aggregators) from preprocessing (by PCS).

- 1) Signal processing is independent of any feature, i.e., the outcome can be shared by multiple aggregators that are interested in extracting different features. For example, an aggregator can extract MPEG-7 features instead of MFCCs for an entirely different context-recognition assignment.
- 2) A significant amount of resource is consumed during signal processing—this will be demonstrated in Section VI. Therefore, it is more efficient for the adaptation component to deal with a single component (a PCS) instead of two.

It must be noted, however, that there is a tradeoff between flexibility and efficiency. Obviously, combining the two processes as a single monolithic process avoids the extra cost of running two independent components. But then, if one is interested in using the same lower level features for different recognition schemes, the monolithic approach does not work. The separation avoids repeating the preprocessing, thereby saving significant computational resources.

### D. Composer

The composer receives the most representative and independent higher level audio features from the aggregator and performs estimation or recognition. This is done by computing the likelihood probabilities of individual context types in a well-defined context space. We have experimented with various techniques, including Bayesian networks and HMMs. We choose HMM because it is most convenient to train and requires little prior knowledge.

An HMM is a deterministic, stochastic, and finite-state machine. A Markov chain or process is a sequence of events (called states) of which the probability of each is entirely dependent on the event immediately preceding it. An HMM represents stochastic sequences as Markov chains; the states are not directly observed, but are associated with observable symbols (or evidences), called emissions, and their occurrence probabilities depend on the hidden states. The generation of

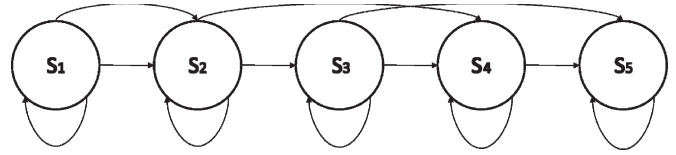


Fig. 4. Left-to-right HMM with five states.

a random sequence is the result of a random transition in the chain. In order to model a process with an HMM, the following elements should be available.

- 1) The number of states in the model  $N$ .
- 2) The number of observation symbols  $M$ , as well as a probability distribution matrix  $B$ , in each of the states describing the occurrence of observable symbols.
- 3) The state-transition probability matrix  $A$ .

Given the number of states in the model,  $N = (S_1, S_2, \dots, S_n)$ , the state-transition probability matrix, and the current state (at time  $t$ ) of the model, it is possible to predict the model's state at time  $t + 1$ .

The state-transition matrix is a square matrix in which each element describes the probability of the model being in state  $S_j$  at time  $t + 1$  given its immediate preceding state

$$a_{ij} = P(q_{t+1} = S_j | q_t = S_i), \quad 1 \leq i, j \leq N \quad (8)$$

where  $q$  refers to a state. The transition probabilities between all states build a state-transition matrix  $A$  of size  $N \times N$ .

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N1} & a_{N2} & \cdots & a_{NN} \end{bmatrix}. \quad (9)$$

The probability of observing the symbol  $O_i$  from a set of symbols  $(O_1, O_2, \dots, O_M)$  when the state machine is in state  $S_j$  at time  $t$  is given as follows:

$$b_{ij} = P(o_t = O_j | q_t = S_i), \quad 1 \leq i \leq N, 1 \leq j \leq M. \quad (10)$$

The probability of observing a symbol in state  $S_j$  is independent of all previous states and observed symbols. Subsequently, the  $N \times M$  observation matrix can be described as follows:

$$B = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1M} \\ b_{21} & b_{22} & \cdots & b_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ b_{N1} & b_{N2} & \cdots & b_{NM} \end{bmatrix}. \quad (11)$$

To complete the description of an HMM, knowledge of the model's initial state is required:  $\pi = (\pi_1, \pi_2, \dots, \pi_N)$ . The initial state of the model, the transition probability matrix, and the observation matrix together make up an HMM, signified by  $\lambda$ .

$$\lambda = \{A, B, \pi\}. \quad (12)$$

The topology of an HMM depends on the process model. Fig. 4 shows the so-called "left-to-right" model which sets a restriction on the way state transitions should be observed—at any given time, either there will be no transition at all or transition should occur in a forward direction only. Even though this restriction is not applicable for audio-based context

<sup>3</sup>The linear-frequency-domain equivalent process is bandpass filtering, while the time-domain equivalent process is smoothing [19].

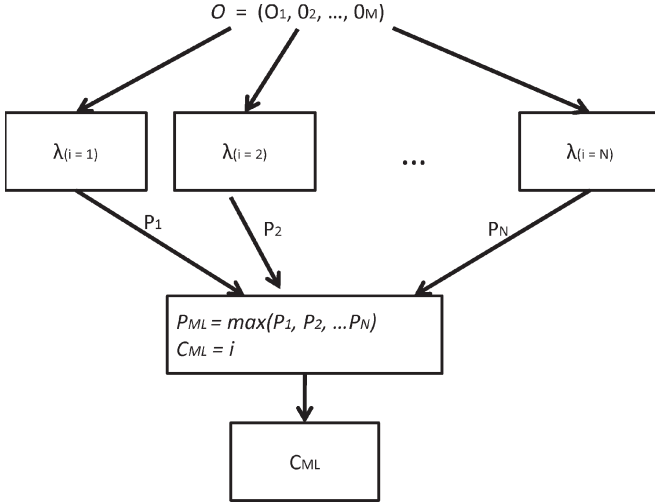


Fig. 5. Recognition of a context using HMMs.

recognition (for example, an audio data may reveal events that repeat themselves), the “left-to-right” topology is useful in modeling time dependences between a sequence of events. For context recognition, the frame duration is normally very short (in the range of milliseconds), and compared to this duration, a meaningful audio scene lasts over several audio frames. Therefore, it is possible to describe an audio scene as a time sequence of several frames (events).

1) *Training*: An HMM can be trained to configure its model parameters  $\lambda = \{A, B, \pi\}$  to a sequence of observed symbols  $O = (o_1, o_2, \dots, o_M)$ . In other words, given an initial description of the model parameters<sup>4</sup>  $\lambda_0$  and a sequence of observation symbols  $O$ , the model parameters should be optimized such that the probability  $P(O|\lambda)$  is maximum

$$P(O|\lambda) = \sum_q \pi_1 b_{q1}(O_1) \cdot \prod_{t=2}^M a_{qt-1qt} b_{qt}(O_t). \quad (13)$$

This is a formidable challenge, since there is no analytic approach to tackle it. However, there are a plethora of estimation algorithms that produce satisfactory results for most real-world situations. We adopt the *Baum–Welch* algorithm and maximize  $P(O|\lambda)$  locally.

2) *Recognition*: During the recognition process, the task of the composer is to determine which of the HMMs (one for each higher level context) the audio data best fit. In other words, given an HMM,  $\lambda$ , and a sequence of observable symbols  $O$ , it computes a sequence of states  $Q$  that maximize  $P(Q|O, \lambda)$ .  $P(Q|O, \lambda)$  is called the likelihood probability.

We employed the *Viterbi* algorithm to compute the log-likelihood probability distribution because, as it is,  $P(Q|O, \lambda)$  is very small.

3) *Vector Quantization*: It is not possible to directly feed the continuous  $n$ -dimensional feature vectors (MFCC) to an HMM, since the model is made up of discrete observation symbols and states. Therefore, the composer maps the  $n$ -dimensional feature vectors into a single vector or codebook. The size of the codebook can be tuned between 64 and 256. Fig. 5 shows the

<sup>4</sup>This can be an approximation based on a prior knowledge of the context type or an arbitrary assignment of parameter values.

TABLE I  
RELATIVE TIME DISTRIBUTION OF A CONTEXT-RECOGNITION PROCESS

Recognition process	Time [ms]	Relative time [%]
Pre-processing	65	20.2
FFT	192	59.8
13 MFCC	25	7.8
VQ code-book size 256	5	1.6
HMM classification	34	10.6
Total	321	100.0

sequence of symbols being provided to different HMMs, each of which computes the likelihood probability of the sequence according to its model parameters. The context with the highest likelihood probability is chosen to be the one that is best represented by the audio signal.

### E. EAK

The EAK manages the features extracted from the training data set, establishes conditional dependences between the contexts and the features, and revises its beliefs whenever new data sets are available. Moreover, it stores the structure and the model parameters of the HMMs.

## VI. ADAPTATION

### A. Defining Complexity Classes

The accuracy and processing time of a context recognition depends on the available resources in a device. It is possible to tune various parameters according to the resource profile and workload of the device. As a result, the context-recognition accuracy and time can be adapted. Parameter tuning requires knowledge of each parameter and its contribution to a context recognition. Table I is an overview of the intermediate stages of the auditory-based context recognition and the relative processing time required by each stage. To produce Table I, we recorded several audio scenes with a nominal duration of 10 s. The audio data were sampled at the rate of 48 kHz, and the context recognition was performed on a laptop computer with a processor speed of 700 MHz and a random-access memory of 256 MB; it had a 4% average workload over a period of 30 min. The frame size was 512 samples with an overlapping percentage of 50%. Each frame had a width of 23.21 ms.

The preprocessing time includes offset-compensation, pre-emphasis, framing, and windowing. Apparently, a large portion of the device’s resources is consumed by signal processing (including the FFT) and not by feature extraction or classification. We varied the hidden states of the HMMs and the size of the codebook but could not observe any appreciable change during the entire processing time. This was an essential observation. Since signal processing is done inside the PCS, the adaptation component should deal with it.

Dealing with PCS means tuning various signal-processing parameters. For example, if time instead of accuracy is more important to an application, the PCS reduces the raw auditory data by reducing the sampling rate. The minimum sampling rate is 8 kHz. A sampling rate below this does not fulfill Shannon’s sampling requirement and does deteriorate the recognition accuracy significantly. At the same time, increasing the sampling

TABLE II  
THREE CLASSES OF RESOURCE PROFILES

Audio Features	Class Low	Class Middle	Class High
Sampling rate [S/s]	8000	11025	22050
Frame Size [samples]	256	256	512
Frame Duration [ms]	46.4	23.2	23.2
Overlapping [%]	25	50	50
Record Duration [s]	1	5	10
Distance between measurements [s]	9	5	0

rate above 22.05 kHz does not improve the recognition accuracy appreciably. Additional parameters that can be tuned by the PCS are the frame size and percentage of frame overlapping. Frame overlapping offers a greater flexibility to adjust processing time and can be varied between 50% and 25%. A 25% overlap can reduce the raw auditory data by more than 30%.

Another possibility of adapting recognition and processing time is by defining different complexity classes (profiles). This classes are taken as static templates into which the computation profiles of mobile devices should fit.

The main challenge here is to define a quantitative relationship between recognition time and accuracy, on the one hand, and resource consumption and CPU workload, on the other hand. We performed an extensive experiment to model relationship between these parameters. This turned out to be a very difficult task. Instead, we applied a heuristic judgment to define three complexity classes that represent the computational capability of different mobile devices. We defined the three complexity classes by assuming that, at any given time, a mobile user may own either a laptop, a PDA, or a mobile phone. This same assumption also enables us to model the different workload of a single device, such as a laptop. According to the classification, we set the upper and lower bounds on recognition time, which can be used to appropriate computing resources for a recognition task. The three classes of profiles are defined as Class Low, Class Medium, and Class High.

Table II provides the description of the three complexity classes for the context types identified in the two scenarios, eight context types in all.

### B. Upper and Lower Bounds on the Recognition Time

The time required for computing a context determines the usefulness of the context. We used benchmarking to estimate the upper and lower bounds of the recognition time that can be achieved by each complexity class. A benchmarking can be explained as follows: A device will be given a task of known complexity, and the time required for accomplishing the task is measured. Usually, the chosen task is similar in complexity to the one the device should carry out afterward. If the computation time is not acceptable by the application developer or the end-user, a runtime reconfiguration (for example, reducing the sampling rate) is made so that the complexity of the task is reduced. This implies that the duration of the audio signal representing the audio scene, the preprocessing, the size of the auditory features, the size of the codebook, etc., can be dynamically adjusted.

Table III shows the lower and upper bounds of the recognition time we computed for the three complexity classes.

TABLE III  
UPPER AND LOWER BOUNDS OF A CONTEXT-RECOGNITION TIME

Device Class	Lower bound	Upper bound
Class Low	50ms	-
Class Middle	100ms	200ms
Class High	-	400ms

TABLE IV  
OVERVIEW OF THE AUDITORY SIGNAL USED FOR TRAINING AND RECOGNITION

Context Type	Total Length of Record Duration
Office	3m 30s
Fight	15m
Party	11m 20s
Loudness	27m 50s
Tram	11m 40s
Train	30m 40s
Presentation/lecture	30m
Casual talk	18m 20s

TABLE V  
EFFECT OF FRAME LENGTH ON ACCURACY

Sample Size [samples]	128	256	512	1024	2048
Length[ms]	5.80	11.61	23.22	46.44	91.02
Recognition Accuracy [%]	58.75	80.17	83.46	80.49	77.20

### C. Accuracy

Accuracy in the context of this paper should be understood as the number of correct decisions the context-recognition scheme makes. This depends on the following factors:

- 1) the number of contending context types;
- 2) the degree of similarity between these contending context types;
- 3) the amount of auditory data to be processed (by implication, the duration of context-recognition time).

The remaining part of this paper focuses on the third factor. To compute the accuracy of context recognition and to attribute the result to the various recognition parameters, we made repeated experiments. It should be noted that we used ordinary microphones and did not pay much attention to the position or orientation of the laptop in which the microphone was embedded. Our intention was to appropriately represent the way ordinary users handle their mobile devices.

Table IV describes the training scenario. The audio signals were processed according to the three profile classes, namely, the sampling rates were set to 22 050, 11 025, and 8000 Hz, while the record length was adjusted for each context to 10, 5, and 1 s. We used the recorded auditory signals to train and test the HMMs. As mentioned before, during the testing phase, the HMMs computed a likelihood probability distribution for all the contexts of interest.

1) *Frame Length*: An HMM attempts to recognize a context of interest by constructing a time sequence of audio frames and by creating conditional dependences between these frames. An audio frame represents the smallest unit of information. If the frame duration is too short, an auditory event of longer duration can be divided into many frames, represented falsely as several events; on the contrary, if the frame is too long, several evanescent events can be mistaken for a single long-duration event. Table V displays the effect of frame length on the accuracy of context recognition. As can be seen from the table, the optimal frame duration which yields the highest



TABLE VI  
EFFECT OF FRAME OVERLAPPING ON  
CONTEXT-RECOGNITION ACCURACY

Overlapping [%]	0	12.5	25	50
Number of Audio Frames	43	49	57	86
Recognition Accuracy [%]	78.37	79.85	83.46	82.12

TABLE VII  
EFFECT OF AUDIO FEATURES ON ACCURACY

Audio Features	Recognition Accuracy [%]
14 MFCC	79.85
12 MFCC	81.55
10 MFCC	79.68
8 MFCC	78.77
12 MFCC without Liftering	69.78
12 MFCC + log-Energy	83.46
12 MFCC + ZCR	79.43
12 MFCC + spectral centroid	78.05

accuracy is achieved at the sampling frequency of 22 050 Hz, namely, 23.22 ms, with the sample size of 512.

2) *Overlapping*: Through the windowing process (to attenuate the effect of high-frequency components during the abrupt separation of frames), some information is lost at the two edges of a frame. That is the reason why a frame overlapping is required. In the literature (speech recognition), a 50% overlap is recommended, but surprisingly, for context recognition, the optimal frame overlapping was achieved at 25%.

This can be explained as follows: For speech recognition, the spectral bandwidth is below 4 kHz, whereas for context recognition, the bandwidth is significantly larger (ca. 10 kHz). As a result, a 25% frame overlapping was sufficient to compensate for the lost information at the edges of a frame. Table VI summarizes recognition accuracy as a function of frame overlapping.

3) *Audio Features*: The selection of the right type and amount of audio features depends on the spectral aspect of the audio scene being processed. For example, for audio events that are made up of frequently changing scenes having both low- and high-frequency components, a large number of MFCCs may be necessary. This way, it is possible to ensure the inclusion of a wide range of frequencies in the extracted features. However, merely increasing the spectral features above a certain threshold may not have any impact on the recognition accuracy.

Our context types involve events that have both slowly and quickly changing scenes as well as low- and high-frequency components. Therefore, we varied the MFCCs from 8 to 14. The result is presented in Table VII.

As can be seen in the table, the number of MFCCs that achieved the highest recognition accuracy is 12 (and not 14!). Perhaps this number would be different for a different record (of the same scenes) or had we employed a different microphone. The best way to explain this is by relating the number of MFCCs with the number of Mel filters [see (6)]. Increasing the Mel filters directly affects the way a frame is subdivided in the frequency domain. This subdivision may result in fragmenting an audio scene into different subbands, as if they were independent events. This can cause an erroneous conclusion.

TABLE VIII  
CODEBOOK SIZE VERSUS RECOGNITION ACCURACY

Codebook Size	Recognition Accuracy [%]
32	65.54
64	80.37
128	81.55
256	83.46
512	79.32

Apart from pure MFCCs, we experimented also with additional time- and frequency-domain features<sup>5</sup> to study their combined effect on recognition accuracy.

4) *Size of the Codebook*: The vector quantization reduces the infinite range of values of the feature vectors into a limited size of code vectors. During this process, if the size of the codebook is significantly small, then many features will be represented by a single vector, and the quantization error will become significant. Subsequently, information which can be vital to the recognition of a context can get lost. On the other hand, making the codebook size considerably large implies the need for a large amount of training data, since only code vectors that appear often in the HMMs during the training phase can be correctly classified. Moreover, the duration of the recognition time will increase significantly. Table VIII summarizes the effect the codebook size on the context-recognition accuracy.

#### D. Implementing the Adaptation Components

The platform-performance monitoring component stores the resource profile of the device in which a context recognition is performed. Moreover, it periodically or randomly (depending on the configuration) samples the CPU workload and considers the past  $n$  samples to estimate device capability. Likewise, the application that binds the recognition system specifies the minimum desired accuracy and processing time, which is the basis for parameter tuning. The complexity control periodically queries the platform-monitoring component to determine how much resource is available and whether the required accuracy and processing time can be achieved. Once again benchmarking is used for the estimation. If the desired accuracy and processing time cannot be achieved, the control component prompts the user to stop some running processes in favor of the context-recognition task or to accept a reduced accuracy or an increased processing time. Accordingly, the complexity control component selects a complexity profile (Class High, Class Medium, or Class Low) or adjust the preprocessing parameters.

## VII. DISCUSSION

We evaluated the context-recognition accuracies of the three complexity classes by establishing a confusion matrix for each class. Table IX displays a confusion matrix for the complexity Class *High*. Table X summarizes the result of two confusion matrices for Classes *Middle* and *Low*. In Table IX, the rows represent the actual context types while the columns represent the recognized context types. At the end of each row is given the context-recognition accuracy for each context type. The last

<sup>5</sup>Including spectral centroid, which is the balancing point of the spectral power distribution, and a zero crossing, which is the number of times a time-domain signal crosses the zero reference [20].

TABLE IX  
PERFORMANCE OF PROFILE CLASS *HIGH*

Context Type	Office	Fight	Loudness	Party	Casual Talk	Tram	Lecture/Pres.	Train	Correct Decision
Office	44	0	0	0	0	0	0	0	100.0
Fight	0	161	0	0	0	0	0	0	100.0
Loudness	0	0	143	0	0	0	0	0	100.0
Party	0	0	28	21	0	0	0	0	42.8
Casual Talk	0	9	19	1	102	13	0	5	68.4
Tram	0	14	0	0	0	72	1	13	72.0
Lecture/pres.	0	1	2	0	0	9	131	4	89.1
Train	0	0	0	0	0	1	38	111	74.0
Average									83.24%

TABLE X  
PERFORMANCE OF PROFILE CLASSES *MIDDLE* AND *LOW*

Context Type	Office	Fight	Loudness	Party	Casual Talk	Tram	Lecture/Pres.	Train	Average [%]
Class L	85	99.7	90.0	17.5	49.3	71.0	40.4	59.7	64.33
Class M	73.3	95.3	68.6	35.1	25.3	79.0	41.3	63.0	60.96

TABLE XI  
AVERAGE RECOGNITION TIME FOR THE THREE PROFILE CLASSES

Profile Class	Average computation time [ms]
Class <i>High</i>	359
Class <i>Middle</i>	109
Class <i>Low</i>	16

row provides the average overall accuracy that can be achieved. The average computation time for the three classes of profiles is summarized in Table XI.

Tables IX and X demonstrate that, while some context types (such as “loudness”) can be recognized by all-complexity classes with appreciable accuracy, others are not. Similarly, other context types (such as “party,” which is starkly mistaken for “loudness”) is recognized poorly, regardless of the class types. Of all the parameters we tuned, the sampling rate has a significant impact on the accuracy and time of context recognition. This is not unexpected. Due to the difference in sampling rate, each MFCC represents a different spectral domain. Therefore, for each complexity class, a different codebook is generated in which the code vectors are distributed in different vector spaces.

The profile Class *Low* performs very poorly for all context types containing predominantly spectral components above 4 kHz. One may be led to conclude that Class *Low* devices may not be useful for auditory-based context recognition. However, this is not the case. As can be seen from the tables, even Class *Low* performs well in recognizing “train,” “tram,” “loudness,” and “aggression.” This implies that the suitability of a device for a context recognition is partly decided by the type of contexts.

#### A. Comparison

The preceding sections demonstrate that the time for context recognition and the associated accuracy depend on several factors. While it is essential to make quantitative comparisons between our result and the results of previous work, some reports conceal a wealth of information, making justifiable comparisons a difficult task.

Eronen *et al.* extensively experimented with different types of recognition schemes, changing the topology of HMMs and varying the test-sequence length of audio signals. In summary,

they are able to recognize 24 everyday context types with an average recognition accuracy of 58% and 6 higher level contexts with an average recognition accuracy of 82%. With the profile Class *High*, ours is better by 1.24%, but this is a rather negligible figure. Moreover, we consider eight context types while they recognize nine.

Similarly, Korpipää *et al.* achieve a context-recognition accuracy of 96% true positives and 100% true negatives under controlled conditions—nine higher level contexts are considered which are in many respects similar to ours. They offer additional insight regarding uncontrolled environments, in which context transitions are not known beforehand and there are disturbances and undefined phenomena. Their result shows that the overall recognition accuracy falls to 87% true positives and 95% true negatives.

We employed ordinary microphones embedded in ordinary laptop computers, both during the training and the test phases. Furthermore, we recorded the audio signals without much preparation, to imitate the way users handle their mobile devices while moving or carrying out other more important activities. In contrast, Eronen *et al.* considered various configurations for their experiment: a binaural setup (Brüel & Kjaer 4128 head and torso simulator), a stereo setup (AKG C460B microphones), and a B-format setup (SoundField MkV microphone). The acoustic material was recorded into a digital multitask recorder in 16-bit and 48-kHz sampling-rate format and a Sony (TCD-D10) digital audio tape recorder in 16-bit and 48-kHz sampling-rate format. Likewise, the measurement system hardware of Korpipää *et al.* consists of an extra small sensor box attached to a shoulder strap of a backpack containing a laptop. When collecting scenario data, the backpack was carried around. The measurement system was controlled with a cordless mouse to mark the scenario phases. The microphone was a small omnidirectional AKG C 417/B.

Unlike Eronen *et al.* and Korpipää *et al.*, our evaluation goes beyond recognition accuracy and addresses the relationship and the tradeoff between recognition accuracy and processing time. We defined also three complexity classes to support adaptive context recognition. Moreover, the adaptation aspect of our system enables a user (or an application) to define quality metrics for a context-recognition task. When the specified quality metrics are not achievable because there are not enough resources for the task, the system prompts the user to stop

some running processes. Otherwise, it offers the user a reduced accuracy or an increased processing time.

### B. Future Work

In this paper, we used laptop computers for context recognition. While we have tested and demonstrated that three different complexity classes can be emulated with a single device, all the three complexity classes satisfy the minimum resource requirements to process acoustic signals that are below 10 kHz. For example, the minimum sampling rate was 8 kHz. In the future, our aim is to deploy resource-efficient signal-processing algorithms on wireless sensor nodes, which are resource-constrained. The networks that can be established by these nodes promise several applications, but the scope and usefulness of the applications are defined and limited by the energy consumption of the networks. At present, extracting raw sensor data claims a large portion of the energy consumption of wireless sensor networks. Compact and efficient signal-processing algorithms can significantly reduce the data traffic in the networks, either by enabling efficient sampling and data compression or by supporting the extraction of higher level features locally.

### ACKNOWLEDGMENT

The author would like to thank Dipl.-Ing. D. Hofmann for his contribution to this paper. During the writing of his Diploma thesis, Dipl.-Ing. Hofmann has performed an extensive laboratory experiment to test the adaptation subsystem of the architecture presented in this paper.

### REFERENCES

- [1] A. Chen, R. Muntz, S. Yuen, I. Locher, S. Park, and M. Srivastava, "A support infrastructure for the smart kindergarten," *Pervasive Comput.*, vol. 1, no. 2, pp. 49–57, Apr.–Jun. 2002.
- [2] W. Dargie and T. Tersch, "Recognition of complex settings by aggregating atomic scenes," *IEEE Intell. Syst.*, vol. 23, no. 5, pp. 58–65, Sep./Oct. 2008.
- [3] W. Dargie and T. Hamann, "A distributed architecture for reasoning about a higher-level context," in *Proc. IEEE Int. Conf. Wireless Mobile Comput., Netw. Commun. WIMOB*, 2006, pp. 268–275.
- [4] J. J. Magee, M. Betke, J. Gips, M. R. Scott, and B. N. Waber, "A human-computer interface using symmetry between eyes to detect gaze direction," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 38, no. 6, pp. 1247–1261, Nov. 2008.
- [5] X. Wang, J. S. Dong, C. Chin, S. Hettiarachchi, and D. Zhang, "Semantic space: An infrastructure for smart spaces," *Pervasive Comput.*, vol. 3, no. 3, pp. 32–39, Jul.–Sep. 2004.
- [6] H. Chen, F. Perich, D. Chakraborty, T. Finin, and A. Joshi, "Intelligent agents meet semantic web in a smart meeting room," in *Proc. 3rd Int. Joint Conf. Auton. Agents Multiagent Syst.*, 2004, pp. 854–861.
- [7] P. Korpipää, J. Mäntyjärvi, J. Kela, H. Kernén, and E.-J. Malm, "Managing context information in mobile devices," *Pervasive Comput.*, vol. 2, no. 3, pp. 42–51, Jul.–Sep. 2003.
- [8] W. Dargie and T. Springer, "Integrating facts and beliefs to model and reason about context," in *Proc. 7th IFIP Int. Conf. Distrib. Appl. Interoperable Syst.*, 2007, pp. 17–31.
- [9] M. Y. Nam, M. R. Bashar, and P.-K. Rhee, "Adaptive feature representation for robust face recognition using context-aware approach," *Neurocomputing*, vol. 70, no. 4–6, pp. 648–656, Jan. 2007.
- [10] N. Young, M. Bashar, and P. Rhee, "Adaptive context-aware filter fusion for face recognition on bad illumination," in *Knowledge-Based Intelligent Information and Engineering Systems*. Berlin, Germany: Springer-Verlag, 2006, pp. 532–541.
- [11] K. Laasonen, M. Raento, and H. Toivonen, "Adaptive on-device location recognition," in *Proc. Pervasive*, 2004, pp. 287–304.
- [12] M. Stäger, P. Lukowicz, and G. Tröster, "Power and accuracy trade-offs in sound-based context recognition systems," *Pervasive Mob. Comput.*, vol. 3, no. 3, pp. 300–327, Jun. 2007.
- [13] A. J. Eronen, V. T. Peltonen, J. T. Tuomi, A. P. Klapuri, S. Fagerlund, T. Sorsa, G. Lorho, and J. Huopaniemi, "Audio-based context recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 14, no. 1, pp. 321–329, Jan. 2006.
- [14] P. Korpipää, M. Koskinen, J. Peltola, S.-M. Mäkelä, and T. Seppänen, "Bayesian approach to sensor-based context awareness," *Pers. Ubiquitous Comput.*, vol. 7, no. 2, pp. 113–124, Jul. 2003.
- [15] L. Ma, D. Smith, and B. Milner, "Context-awareness using environmental noise classification," in *Proc. Eurospeech*, 2003, pp. 2237–2240.
- [16] D. Smith, L. Ma, and N. Ryan, "Acoustic environment as an indicator of social and physical context," *Pers. Ubiquitous Comput.*, vol. 10, no. 4, pp. 241–254, Mar. 2006.
- [17] A. Padovitz, S. W. Loke, and A. Zaslavsky, "Multiple-agent perspectives in reasoning about situations for context-aware pervasive computing systems," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 38, no. 4, pp. 729–742, Jul. 2008.
- [18] M. Krysanter and E. Frisk, "Sensor placement for fault diagnosis," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 38, no. 6, pp. 1397–1410, Nov. 2008.
- [19] M. Benzeghiba, R. De Mori, O. Deroo, S. Dupont, T. Erbes, D. Jouviet, L. Fissore, P. Laface, A. Mertins, C. Ris, R. Rose, V. Tyagi, and C. Wellekens, "Automatic speech recognition and speech variability: A review," *Speech Commun.*, vol. 49, no. 10/11, pp. 763–786, Oct./Nov. 2007.
- [20] M. Döller and H. Kosch, *The MPEG-7 Multimedia Database System (MPEG-7 MMDb)*, vol. 81. New York: Elsevier, 2008, pp. 1559–1580.



**Waltenegus Dargie** (M'08) received the B.Sc. degree in electrical engineering from the Nazareth Technical College, Adama, Ethiopia, in 1997, the M.Sc. degree in electrical engineering from the Technical University of Kaiserslautern, Kaiserslautern, Germany, in 2002, and the Ph.D. degree in computer engineering from the Technical University of Dresden, Dresden, Germany, in 2006.

He was a Researcher with the Fraunhofer Institute of Experimental Software Engineering, Kaiserslautern, between 2002 and 2003, and with the Department of Electrical Engineering and Computer Science, University of Kassel, Kassel, Germany, between 2002 and 2005. He is currently a Researcher with the Technical University of Dresden. His research interests include autonomous computing, context-aware computing, wireless networks, and digital signal processing.

# Recognition of Complex Settings by Aggregating Atomic Scenes

Waltenegus Dargie and Tobias Tersch, *Technical University of Dresden*

**O**ne important aspect of ubiquitous computing is context awareness, which aims to establish a shared understanding of the user's social and conceptual settings (contexts). Establishing such a shared understanding can be simple or complex. By simple, we mean that you can easily obtain the necessary sensors and can map the sensed

data to a meaningful setting. In most cases, however, the process is complex and the end result is uncertain. In the latter situation, context acquisition involves modeling and reasoning about the characteristics of and relationships between several entities.

Most approaches to context reasoning model complex settings (higher-level contexts) as monolithic scenes rather than aggregations of distinct scenes. For example, recognition of a street setting on the basis of features extracted from an audio signal requires an existing model of a street. To produce the model, such approaches will take audio signals from various streets, analyze these signals' stochastic properties, and extract the most representative (and independent) features. However, these approaches don't separate the signals according to the scenes that make up the complex setting (cars, pedestrians, street bands, and so forth).

In reality, a street setting isn't a result of a stationary mix of different events but rather a complex mix of time-variant events. For example, the frequency and types of cars passing by change continuously. A recognition scheme can deal with this type of dynamic only if it can separate the street's stationary scenes from the transient scenes. More-

over, by modeling the scenes independently and establishing a relationship between them, we can define a higher-level context declaratively.

We model complex settings as an aggregation of distinct atomic scenes. To support declarative context aggregation, we provide a conceptual architecture that enables a systematic modeling and gradual reasoning of complex settings. Applying our architecture to auditory-based context recognition, we've modeled seven everyday situations with more than 20 atomic scenes, achieving high recognition rates for both the atomic scenes and complex settings.

## A conceptual architecture for context recognition

Humans recognize complex settings by perceiving individual settings and examining the relationships between them. Their certainty of the perceived setting depends on how well they have gathered and interpreted data from their surroundings. It also depends on the presence or absence of some vital scenes that constitute the setting. A complex setting consists of individual settings unfolding in a certain order. Moreover, by combining individual settings from their memory, humans can imagine set-

*This approach imitates human reasoning to enable flexible context recognition. Its usefulness is demonstrated by employing audio-signal processing to recognize several everyday situations.*



tings they have never experienced. For example, a person who never watched a symphony orchestra playing Beethoven can imagine it by combining pictures of individual scenes of an orchestra from his or her experience.

Our aim is to imitate human-like reasoning. Proper imitation will lead us to

- improve context recognition accuracy and
- declaratively define an entirely new setting by aggregating known individual scenes.

To this end, we propose the four-layered conceptual architecture in Figure 1.

The architecture's bottom layer (the raw-sensor-data layer) consists of an array of physical sensors embedded in mobile devices or carefully placed in physical environments.

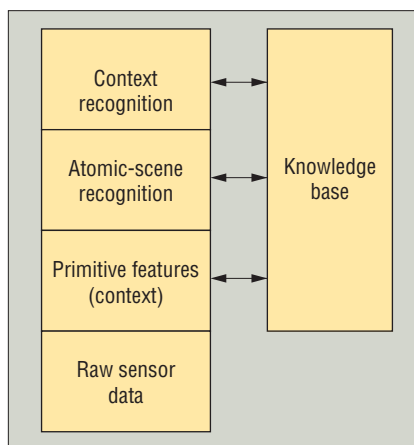
The second layer extracts primitive features (contexts). A primitive context represents a single, indivisible aspect of a certain phenomenon or physical entity (device, place, person, and so forth). It's a meaningful interpretation of raw sensed data. Because it's primitive, it's extracted either from a single sensor or from multiple sensors representing the same aspect. Unlike a higher-level context, whose meaning is application-specific, a primitive context can be useful for recognizing several higher-level contexts.

The third layer constructs atomic scenes. The premise for this layer is that most everyday settings consist of distinct scenes, and multiple settings can have several scenes in common. If the system recognizes these scenes and stores their models separately in a knowledge base, it can reuse them to declaratively define complex settings for which it hasn't previously been trained.

For example, we can describe a meeting setting by the flipping of papers, conversations, and occasional whispers. We can describe a lecture by a monotonous oration, flipping of papers, occasional coughs, sporadic whispers, the sound of writing with chalk on a blackboard, and so forth. These two settings share the flipping of papers and whispering. A context-recognition system can therefore exploit this knowledge to accommodate the definition of a meeting or a lecture even though it has never been trained to recognize either of these two contexts.

The fourth layer handles context recognition. It employs a deterministic or probabilistic reasoning scheme or a combination of both (for more on these approaches, see the "Related Work in Context Recognition" sidebar on p. 60). It aggregates evidence from the third layer, establishes logical or probabilistic relations between the atomic scenes the system has already recognized, and computes a higher-level context. The layer takes into account domain knowledge of the mutual occurrence of the atomic scenes.

Common to all layers except the raw-sensor-data layer is the knowledge base. It comprises facts that constitute an application domain's vocabulary and a list of assertions about individual named entities in terms of this vocabulary. The vocabulary consists of concepts, which denote sets of entities, and relations, which denote binary relationships between these entities. The knowledge base also allows the building of complex descriptions of concepts and relations. The system uses this knowledge to extract meaningful features from sensors, classify atomic scenes, and model relationships between the atomic scenes to recognize higher-level contexts.



**Figure 1. A conceptual architecture for recognition of complex settings. This architecture allows a recognition system to mimic human reasoning.**

### Auditory-based context recognition

We chose auditory signals for three reasons. First, among the human senses, hearing is second only to vision in recognizing social and conceptual settings; this is due partly to the richness in information of audio signals. Second, you can embed cheap but practical microphones in almost all types of places or mobile devices, including PDAs and mobile phones. Finally, auditory-based context recognition consumes significantly fewer computing resources than camera-based context recognition.

To better explain the implementation of our architecture for auditory-based context recognition, we offer here a summary of digital audio-signal processing.

Even though auditory-based context recognition is similar to speech recognition, there are several differences. For example, in speech recognition, knowledge of human perception (tone, pitch, loudness, and so forth) is useful to disambiguate an uttered speech. This is possible because

- the speaker isn't far from the microphone and speaks sufficiently loud, and
- no significant hindrance exists between the speaker and the microphone.

This isn't the case with auditory-based context recognition. First, the audio-signal amplitude representing a user's surrounding isn't appreciably large because the audio sources might be farther from the user (the microphone). Moreover, the device with the embedded microphone might be hidden in a suitcase or pocket. So, auditory-based context recognition can't achieve the same accuracy as speech recognition.

### Extracting audio features

The statistical properties of audio signals representing most everyday settings aren't stationary. To extract features that represent temporal and spectral aspects, audio-based recognition systems divide the audio data stream into small time frames that can then be considered quasi-stationary. Some overlap between the frames is desirable; typically, the overlap is between 25 and 50 percent. A frame's duration is usually between 10 and 50 milliseconds, depending on the desired recognition accuracy and computation time. Further processing isn't necessary to extract temporal features, but at least two additional steps are necessary to extract spectral properties.

## Related Work in Context Recognition

A context recognition (reasoning) process can be deterministic, probabilistic, or both. Deterministic context reasoning classifies sensed data into distinct states and produces a distinct output that can't be uncertain or disputable. Probabilistic reasoning, on the other hand, considers sensed data to be uncertain input and thus outputs multiple contextual states with associated degrees of truthfulness.

Several researchers have proposed probabilistic-reasoning techniques for context reasoning. These techniques differ according to the type of context they recognize and the types of sensors they employ.

Nicolas Moeënne-Loccoz, François Brémond, and Monique Thonnat proposed Bayesian networks to recognize various human activities on a street (aggressive behavior, casual talk, and play); they obtained sensed data from a camera.<sup>1</sup> Huadong Wu employed a camera and several microphones to reason about the attention of people during a meeting session.<sup>2</sup> He applied the Dempster-Shafer theory of evidence to combine data from microphones with data from an omnidirectional camera.

Jani Mäntyjärvi, Johan Himberg, and Pertti Huuskonen proposed *k*-means clustering and minimum-variance segmentation algorithms to process data from a skin conductance sensor, a microphone, a light sensor, an accelerometer, and a temperature sensor, to recognize a mobile device's status and its user's activity.<sup>3</sup> Device status refers to whether the device is in the user's hands, on a table, or inside a suitcase; user activity refers to walking, running, or going up or down a staircase.

Some researchers have focused particularly on processing audio signals to recognize various everyday human situations. Vesa Peltonen and his colleagues classified auditory scenes into predefined classes by employing two classification schemes: a 1-NN (1-nearest neighbor) classifier and Mel-frequency cepstral coefficients (MFCCs) with Gaussian mixture models.<sup>4</sup> The auditory scenes comprised several everyday outdoor and indoor situations (streets, restaurants, offices, homes, cars, and so forth). The features extracted from audio signals for classification were time and frequency domain features and linear prediction coefficients. Altogether, the classification systems classified 17 indoor and outdoor scenes with an accuracy of 68.4 percent.

For their experiment, Peltonen and his colleagues con-

sidered various configurations: a binaural setup (a Brüel & Kjaer 4128 head and torso simulator), a stereo setup (AKG C460B microphones), and a B-format setup, which contains 3D information of the audio event being recorded (Sound-Field MkV microphone). They recorded the sounds on a digital multitask recorder with a 16-bit, 48-kHz sampling rate and on a Sony (TCD-D10) digital audio tape recorder with a 16-bit, 48-kHz sampling rate.

Antti Eronen replaced the two classifiers that Peltonen and his colleagues used with hidden Markov models (HMMs) to imitate human hearing sensitivity and to increase recognition accuracy up to 88 percent.<sup>5</sup>

Ling Ma, Dan Smith, and Ben Milner also employed HMMs and MFCCs to recognize 10 auditory scenes.<sup>6</sup> By varying the hidden states of the Markov models, they achieved different recognition rates. With only three hidden states, the classifier achieved 78 percent context recognition; with 15 hidden states, it achieved 91.5 percent recognition. Remarkably, context recognition declined for more than 15 hidden states. Dan Smith, Ling Ma, and Nick Ryan extended this research by introducing a belief revision mechanism that increased the recognition rate to 92.27 percent and the number of recognized contexts to 12.<sup>7</sup>

Panu Korpipää and his colleagues employed a naive Bayesian classifier and an extensive set of audio features derived partly from the algorithms of the MPEG-7 standard.<sup>8</sup> They based the classification mainly on audio features measured in a home scenario. To collect the data, Korpipää and his colleagues used an extra-small sensor box attached to a shoulder strap of a backpack containing a laptop. When collecting scenario data, researchers wore the backpack. A cordless mouse controlled the measurement system to mark the scenario phases. The microphone was a small, omnidirectional AKG C 417/B.

With a resolution of 1 second in segments of 5–30 seconds and using leave-one-out cross-validation, Korpipää and his colleagues achieved a recognition rate of 87 percent of true positives and 95 percent of true negatives, averaged over nine 8-minute scenarios containing 17 segments of different lengths and nine different contexts. The reference accuracies measured by testing with training data were 88 percent (true positive) and 95 percent (true negative), suggesting that the model can cover the variability introduced in the data on purpose. Reference recognition accuracy in con-

Owing to the abrupt separation of neighboring frames, high-frequency components will emerge at both edges of each frame. This *frequency leakage* should be removed (or at least its effect should be minimized) through a *windowing* operation, a filtering process that multiplies each frame with a window function that decays rapidly toward the edges. Before this process, however, we want to smooth the spectrum and enhance the high-frequency components by passing the frames through a first-order, finite-impulse-response preemphasis high-pass filter:

$$s_{sp}(n) = s(n) - \mu s(n-1)$$

In this equation,  $s_{sp}(n)$  is the improved  $n$ th sample of a frame,  $s(n)$  is the original  $n$ th sample,  $s(n-1)$  is the original  $n-1$ th sample,

and  $\mu$  is a unitless quantity, which normally ranges between 0.90 and 0.98. For the windowing operation, we use a standard Hemming window, which we can describe as

$$s_w(n) = \{0.54 - 0.46 \times \cos(2\pi(n-1)/(N-1))\} \times s_{sp}(n)$$

where  $s_w(n)$  refers to the  $n$ th sample of a frame that has passed through a Hemming window and  $N$  is the number of samples in a frame.

Mel-frequency cepstral coefficients (MFCCs) are the most frequently used features for classifying auditory data. They represent frequency bands that are Mel-scaled to approximate the human auditory system's response more accurately than linearly spaced frequency bands obtained directly from a fast Fourier transform

trolled conditions was 96 percent (true positive) and 100 percent (true negative).

Filip Bonnevier employed Bayesian networks to recognize 25 different contexts from 21 MPEG-7 features with a 69 percent recognition rate.<sup>9</sup> Interestingly, the context recognition ran on a pocket PC.

Table A summarizes the audio-based context-recognition schemes, their recognized contexts, and their recognition accuracies.

## References

1. N. Moeënne-Loccoz, F. Brémond, and M. Thonnat, "Recurrent Bayesian Network for the Recognition of Human Behaviours from Video," *Proc. 3rd Int'l Conf. Computer Vision Systems (ICVS 03)*, Springer, 2003, pp. 66–77.
2. H. Wu, "Sensor Data Fusion for Context-Aware Computing Using Dempster-Shafer Theory," PhD thesis, Dept. of Computer Science, Carnegie Mellon Univ., 2003.
3. J. Mäntyjärvi, J. Himberg, and P. Huuskonen, "Collaborative Context Recognition for Handheld Devices," *Proc. 1st IEEE Int'l Conf. Pervasive Computing and Communications (PerCom 03)*, IEEE Press, 2003, pp. 161–168.
4. V. Peltonen et al., "Computational Auditory Scene Recognition," *Proc. Int'l Conf. Acoustic Speech and Signal Processing (ICASSP 02)*, IEEE Press, 2002, pp. 1941–1944.
5. A. Eronen, "Automatic Musical Instrument Recognition," master's thesis, Dept. of Information Technology, Tampere Univ. of Technology, 2001.
6. L. Ma, D. Smith, and B. Milner, "Context-Awareness Using Environmental Noise Classification," *Proc. 8th European Conf. Speech Communication and Technology (Eurospeech 03)*, 2003, pp. 2237–2240.
7. D. Smith, L. Ma, and N. Ryan, "Acoustic Environment as an Indicator of Social and Physical Context," *Personal Ubiquitous Computing*, vol. 10, no. 4, 2006, pp. 241–254.
8. P. Korpipää et al., "Managing Context Information in Mobile Devices," *IEEE Pervasive Computing*, vol. 2, no. 3, 2003, pp. 42–51.
9. F. Bonnevier, "Audio Based Context-Awareness on a Pocket PC," master's thesis, Dept. of Electrical Eng., Stockholm Inst. of Technology, 2006.

**Table A. Audio-based context-recognition schemes.**

Authors	Primitive features (contexts)	Classifier	Recognition accuracy (%)	Context
Peltonen et al. <sup>4</sup>	Temporal, spectral, Mel-frequency cepstral coefficient (MFCC)	k-nearest neighbor (k-NN) and Gaussian mixture model (GMM)	68.4 (17 of 26 contexts)	Bathroom, street, church, car, supermarket, office
Eronen et al. <sup>5</sup>	MFCC	Hidden Markov model (HMM)	Between 61 and 85 (18 contexts)	Library, office, lecture, train, bus
Ma, Smith, and Milner <sup>6</sup>	MFCC	HMM	91.5 (10 contexts)	Bar, beach, bus, lecture, office, street, launderette
Smith, Ma, and Ryan <sup>7</sup>	MFCC	HMM	92.27 (12 contexts)	Bus, car, presentation, supermarket, train, office
Korpipää et al. <sup>8</sup>	MPEG-7	Naïve Bayesian classifier	88 (9 contexts)	Running, walking, music, speech, elevator, tap water, car
Bonnevier <sup>9</sup>	Spectral, temporal, MPEG-7	Bayesian network	69 (25 contexts)	Street, car, bus, cooking, TV, kitchen, living room

(FFT) or a discrete cosine transformation (DCT). Such representations allow context-recognition schemes to "perceive" their surroundings as humans would perceive theirs.

To obtain MFCCs, we perform an FFT; the result passes through a bank of triangular filters called Mel-filters (see Figure 2 on p. 62) to produce the Mel-spectrum. The number of filters can vary, but as a rule, speech recognition uses 23 filters. These filters are equidistant in the Mel-frequency domain, with a 50 percent overlap between adjacent filters. The following equation computes the center of each triangular filter:

$$c(x) = 2,595 \left( x \cdot \log_{10} \left( 1 + \frac{x}{700} \right) \right)$$

where  $c(x)$  is the center of a triangular filter (in the frequency domain) that has taken the amplitude spectrum,  $x$ , of the FFT as its input.

Finally, we perform an inverse DCT on the natural logarithm of the filters' output to obtain the MFCCs:

$$c_i = \sum_{j=1}^N \ln|f_i| \cos \left( \frac{\pi i}{N} \left( j - \frac{1}{2} \right) \right),$$

$$0 \leq i \leq M$$

where  $c_i$  is the  $i$ th cepstral coefficient,  $f_i$  is the  $i$ th frequency component,  $N$  is the number of the triangular filters, and  $M$  is

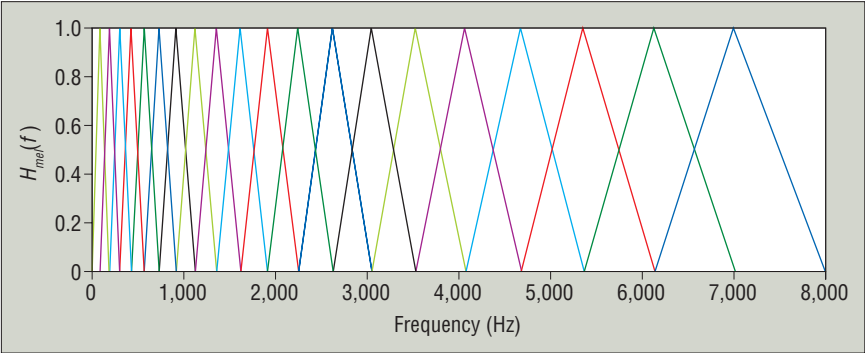


Figure 2. A triangular filter bank. The filters’ response has a linear frequency spacing below 1,000 Hz and a logarithmic spacing above 1,000 Hz.  $H_{Mel}(f)$  denotes a normalized magnitude spectrum.

Table 1. Higher-level contexts defined declaratively as aggregations of atomic scenes.

Higher-level context	Individual scenes
Office	Clacking of keyboard Conversation Mouse clicking Telephone conversation Telephone ringing
Cafeteria	Background noise Chair movement Clacking of cash register keys Clattering dishes Conversation
Library	Chair movement Clacking of keyboard Coughing Door opening and closing Flipping pages Mouse clicking Whispering
Tram	Station announcement signal Background noise Door-closing warning People getting on and off
Street	Moving cars People walking and talking Background noise
Lecture	Background noise Chair movement Coughing Flipping pages Oration Whispering Writing on a chalkboard
Train	Background noise People getting on and off Conversation

the number of the extracted MFCCs. In this way, we set the dimension of the feature vectors; the typical dimension is 13. To weaken the effect of very low and high orders of the cepstral coefficients, we need to subject the MFCCs to a “band-pass filtering” process called *liftering*. The following equation

displays a typical liftering function:

$$c_i = \left( 1 + \frac{L}{2} \sin \left( \frac{\pi i}{L} \right) \right) c$$

where  $c_i$  is the corrected cepstral coefficient,  $c$  is the  $i$ th uncorrected cepstral coefficient, and  $L$  is a liftering factor.

Recognition

Feature extraction quantizes the audio signal and transforms it into various characteristic features. This results in an  $n$ -dimensional feature vector representing each audio frame. A classifier then takes this feature vector and determines what it represents—that is, it determines an auditory scene.

Several recognition techniques are readily available, most of which we mention in the sidebar. The three most common are  $k$ -nearest neighbor ( $k$ -NN) classifiers, hidden Markov models (HMMs), and Bayesian networks.

Implementation

We selected these seven higher-level contexts: office, cafeteria, library, tram, street, lecture, and train. Table 1 lists these settings along with the associated atomic scenes.

We chose the atomic scenes on the basis of how well they represented the higher-level settings and how accurately they could be recognized.

The raw-sensor-data layer consisted of commonplace microphones embedded in ordinary laptop PCs during the training and test phases. Moreover, we recorded the audio signals without much preparation to imitate how users handle their mobile devices while moving or carrying out other more important activities.

We implemented the second layer by adopting the OC-volume framework (<http://ocvolume.sourceforge.net>). Even though the framework was initially intended for speech recognition, we could reuse it for extracting MFCCs and for vector quantization, using the LBG (Linde, Buzo, and Gray) algorithm.<sup>1</sup> However, we had to modify the algorithm to

- model time dependency in the audio signals and
  - increase the signals’ bandwidth to accommodate the surrounding noise’s dominant frequencies.
- As a result, we could consider ranges of frequencies between 30 and

**Table 2. Recognition accuracy of atomic scenes.**

Atomic scene	Recognition rate (%)	Deviation (%), with incorrect classifications
Car	100	0
Flipping pages	100	0
Door opening and closing	91	Background noise, Tram: 9
Chair movement	80	Background noise, Tram: 20
Door-closing warning	100	0
Clattering dishes	50	Coughing: 50
Coughing	100	0
Background noise, Tram	87.5	Oration: 12.5
Writing on a chalkboard	80	Whispering: 20
Whispering	56	Background noise, Lecture: 44
Oration	75	Background noise, Cafeteria: 10 Conversation, Office: 5 Background noise, Lecture: 5
Conversation	72	Oration: 18 Background noise, Train: 10
Background noise, Street	58	Writing on a chalkboard: 23 Background noise, Cafeteria: 19
Background noise, Lecture	40	Whispering: 60
Mouse clicking	94	Flipping pages: 4 Clacking of keyboard: 2
Background noise, Train	100	0
Station announcement signal	0	Poorly captured audio signal
Clacking of keyboard	79	Mouse clicking: 10 Background noise, Library: 6 Door opening and closing: 3 Coughing: 1 Background noise, Train: 1
Conversation, Telephone	100	0
Background noise, Cafeteria	65	Conversation: 25 Station announcement signal: 5 Chair movement: 5
Background noise, Library	62	Clacking of keyboard: 32 Mouse clicking: 6
<b>Overall recognition rate</b>	<b>69.92</b>	

10,000 Hz. For speech recognition, the frequency of interest is below 3,400 Hz.

To realize the atomic-scene layer, we chose a  $k$ -NN classifier because of its simplicity. It also classifies a large number of scenes in an acceptable recognition time. The classifier performs a class vote among the  $k$ -nearest neighbors on a point to be classified. A Euclidean distance,  $d$ , between the points determines which atomic scenes are represented by the extracted MFCCs. We set  $k = 1$ . Vesa Peltonen and his colleagues demonstrated that classification with  $k$  greater than 1 yields no significant improvement in recognition accuracy.<sup>2</sup>

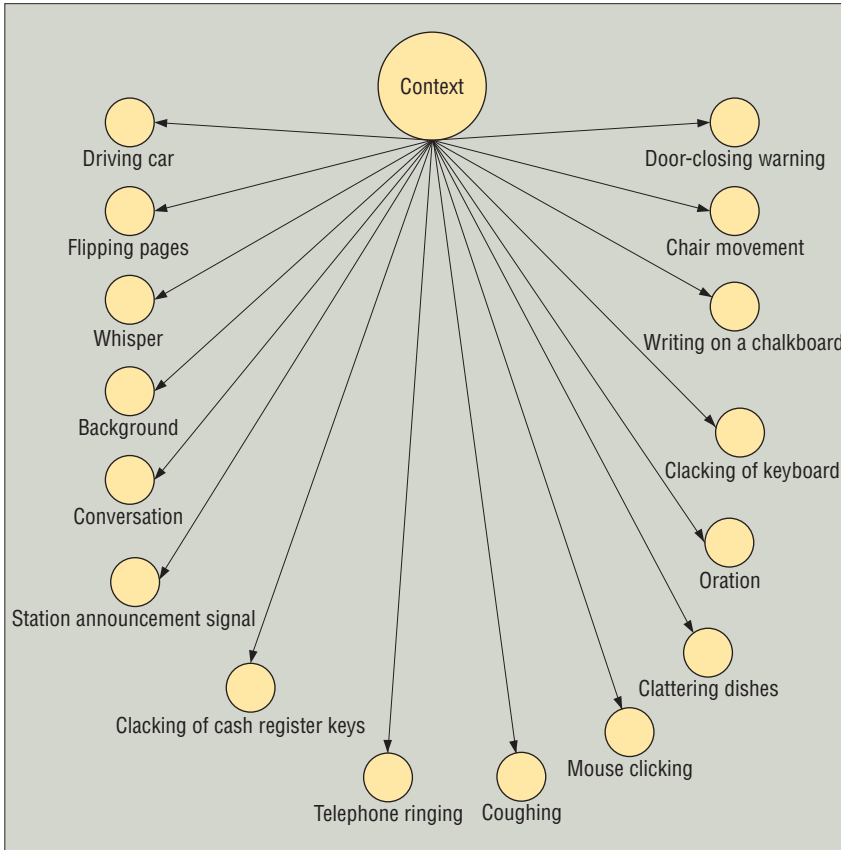
Table 2 displays the atomic scenes we could recognize, the recognition accuracy, and the deviation, with the atomic scenes that

were wrongly recognized. The quality of the recognized atomic scenes depended on how distinct they were from other atomic scenes. It also depended on the recorded audio signal's quality.

We chose a Bayesian network to model relationships between the higher-level contexts and the atomic scenes and to recognize a higher-level context. We used the JavaBayes framework ([www.cs.cmu.edu/~javabayes/index.html](http://www.cs.cmu.edu/~javabayes/index.html)) to implement the knowledge base and the context-recognition layer. The knowledge base stores models of the Bayesian network structure as well as conditional-probability distributions.

The Bayesian classifier establishes a network based on the atomic scenes recognized in the lower layer. We applied heuristic





**Figure 3. A Bayesian network for establishing conditional dependencies between higher-level settings and atomic scenes. The child nodes represent the atomic scenes, and the parent node represents the higher-level contexts. Altogether, the parent node can have seven different values depending on the network configuration.**

observations for establishing the conditional dependencies between the atomic scenes and the higher-level contexts. Figure 3 shows our Bayesian network.

Bayesian networks apply Bayes's theorem to model probabilistic relationships among distinctions of interest in uncertain reasoning. The networks are directed acyclic graphs (DAGs) in which nodes represent random variables and a directed arrow represents a conditional dependency between the variables. A particular configuration of a Bayesian network refers to an instantiation of the random variables with values from a 2D value vector. A particular configuration's likelihood is determined by the sum of the products of the associated conditional probabilities.

A Bayesian network obeys the Markov condition for mathematical and computational tractability. So, a node is conditionally independent of its nondescendants given its parent in  $G$ , the network's graph topology. Mathematically, we express this as

$$p(n_1, n_2, \dots, n_j) = \prod_{j=1}^N p(n_j | \text{parent}(n_j))$$

where  $n_1, n_2, \dots, n_j$  are the possible values of the network's random variables, and  $p$  refers to probability.

Once we establish a Bayesian network and define the degree

of independence between random variables, even partially, it's possible to carry out three essential tasks.<sup>3</sup> First, because the model encodes dependencies among all variables, it can readily reason about situations where some data entries are missing. Second, we can train the network to learn causal relationships and hence use it to understand a problem domain and to predict the consequences of intervention. Finally, because the model has both causal and probabilistic semantics, it's ideal for combining prior knowledge (which often comes in causal form) and data.

### Discussion

Table 3 lists our higher-level contexts and the corresponding atomic scenes that contribute to their recognition. The last column lists the normalized percentage of each atomic scene's contribution. The percentage doesn't add up to 100 percent because the list doesn't include erroneous atomic scenes.

The higher-level context with the lowest recognition rate is a street (37 percent). In fact, the spurious cafeteria context had higher recognition accuracy—47 percent. Interestingly, the Bayesian classifier could recognize a cafeteria with 100 percent accuracy without mistaking it for a street or another contending setting. This implies that context recognition is asymmetric—a context's recognition accuracy depends on not only how well it's represented by the atomic scenes but also whether the captured auditory test signal typically represents the setting. For our case, for example,

the test signal came twice from a street with little activity, and the activities at a nearby cafeteria dominated the recording. Predictably, this led to a wrong conclusion.

On the other hand, page flipping might seem difficult to recognize because it isn't loud. We could, however, recognize it with 100 percent accuracy. This is because the atomic scene was associated with a lecture and a library, where the background noise and other atomic scenes could be distinctly discerned. Moreover, as we trained and tested our system, we placed a laptop with a microphone near the user who was reading and flipping pages.

The least-recognized scene—in fact, the system didn't recognize it at all—was chair movement in a library. The system sometimes mistook an oration in a lecture room for a conversation in an office or cafeteria, which is understandable.

We were interested in comparing our results with others', but this wasn't easy. Some research reports conceal a wealth of information. Maybe this is because recognition accuracy depends on not only the particular schemes or features employed but also many other factors. To begin with, it depends on the types of contexts to be recognized. The larger and more similar the context types, the harder it is to distinguish between them. Recognition accuracy also depends on the test signal's length, the audio signal's sampling rate, the

**Table 3. Recognition of complex settings by aggregating atomic scenes.**

Higher-level context	Correctly and incorrectly recognized contexts (%)	Distribution of the correctly recognized atomic scenes (%)
Library	Library: 67 Office: 33	Clacking of keyboard: 25.17 Mouse clicking: 23.87 Door opening and closing: 20.86 Chair movement: 7
Office	Office: 90 Train: 10	Conversation, Telephone: 70.7 Clacking of keyboard: 25.17 Mouse clicking: 3.7
Cafeteria	Cafeteria: 100	Background noise, Cafeteria: 51.86 Conversation: 32.8 Clattering dishes: 7.4
Street	Street: 37 Cafeteria: 47 Lecture: 10 Train: 5	Moving cars: 51.49 Chair movement: 10.89 Background noise, Street: 10 Conversation: 8.49 Background noise, Cafeteria: 7.4
Tram	Tram: 75 Cafeteria: 25	Background noise, Tram: 70 Station announcement signal: 7
Train	Train: 40 Library: 40 Lecture: 15 Office: 5	Background noise, Train: 87.67 Conversation: 9.4
Lecture	Lecture: 100	Oration: 36.53 Chair movement, Lecture: 26.03 Background noise, Lecture: 15.19 Writing on chalkboard: 8.22 Whispering: 6.4
<b>Overall recognition rate</b>	<b>72.71</b>	

MFCCs' size, and the size of the code book of the vector quantization process. Subsequently, a trade-off always exists between recognition time and recognition accuracy.

More important, the recording devices used and the audio signal's length and duration influence context-recognition accuracy. Using expensive, bulky, and power-hungry audio devices might yield remarkable accuracy, but using them in everyday situations, particularly in mobile environments, isn't feasible.

The research that comes closest to ours is that of Peltonen and his colleagues and Antti Eronen (see the sidebar). Our atomic-scene-recognition accuracy is similar to theirs, but we achieved recognition accuracy through commonplace microphones and ordinary laptop computers as compared to the sophisticated devices they used to record audio signals. Moreover, our approach can be generalized to accommodate sensors other than microphones, while their approaches are limited to audio-based context recognition.

**O**ur experience demonstrates the difficulty of context recognition using a single context source—namely, an audio signal. Humans aptly apply other faculties besides hearing to appropriately perceive their surroundings. This justifies the need for heterogeneous sensing.

We're interested in investigating the possibility of deploying—at least in part—audio-signal-processing algorithms on wireless sen-

## The Authors

**Waltenegus Dargie** is a researcher at the Technical University of Dresden. His research interests include ubiquitous computing, context-aware computing, wireless networks, and digital signal processing. Dargie received his PhD in computer engineering from the Technical University of Dresden. Contact him at [waltenegus.dargie@tu-dresden.de](mailto:waltenegus.dargie@tu-dresden.de).

**Tobias Tersch** is a software developer at the Sidon Software and Engineering Service-Providing Company. His research interests include context awareness and smart systems. Tersch received his diploma in computer science from the Technical University of Dresden. Contact him at [tobias.tersch@web.de](mailto:tobias.tersch@web.de).

sor nodes. This will enable us to gather and process surrounding acoustic information and to better interface the physical world with the virtual world. ■

## References

1. H.-Y. Chang et al., "Performance Improvement of Vector Quantization by Using Threshold," *Advances in Multimedia Information Processing—PCM 2004*, LNCS 3333, Springer, 2005, pp. 647–654.
2. V. Peltonen et al., "Computational Auditory Scene Recognition," *Proc. Int'l Conf. Acoustic Speech and Signal Processing*, 2002.
3. D. Heckerman, "A Tutorial on Learning with Bayesian Networks," *Learning in Graphical Models*, M.I. Jordan, ed., MIT Press, 1999, pp. 301–354.