Traffic Analysis, Modeling and Their Applications in Energy-Constrained Wireless Sensor Networks

- On Network Optimization and Anomaly Detection

Qinghua Wang

Mittuniversitetet
MID SWEDEN UNIVERSITY

Department of Information Technology and Media
Mid Sweden University

Doctoral Thesis No. 78
Sundsvall, Sweden
2010

Mittuniversitetet
Informationsteknologi och medier
SE-851 70 Sundsvall
SWEDEN

ISBN 978-91-86073-64-0
ISSN 1652-893X

Akademisk avhandling som med tillstånd av Mittuniversitetet framlägges till offentlig granskning för avläggande av teknologie doktorsexamen onsdagen den 3 februari 2010 i L111, Mittuniversitetet, Holmgatan 10, Sundsvall.

©Qinghua Wang, januari 2010

Tryck: Tryckeriet Mittuniversitetet

To My Wife
To My Parents

Abstract

Wireless sensor network (WSN) has emerged as a promising technology thanks to the recent advances in electronics, networking, and information processing. A wide range of WSN applications have been proposed such as habitat monitoring, environmental observations and forecasting systems, health monitoring, etc. In these applications, many low power and inexpensive sensor nodes are deployed in a vast space to cooperate as a network.

Although WSN is a promising technology, there is still a great deal of additional research required before it finally becomes a mature technology. This dissertation concentrates on three factors which are holding back the development of WSNs. Firstly, there is a lack of traffic analysis & modeling for WSNs. Secondly, network optimization for WSNs needs more investigation. Thirdly, the development of anomaly detection techniques for WSNs remains a seldomly touched area.

In the field of traffic analysis & modeling for WSNs, this dissertation presents several ways of modeling different aspects relating to WSN traffic, including the modeling of sequence relations among arriving packets, the modeling of a data traffic arrival process for an event-driven WSN, and the modeling of a traffic load distribution for a symmetric dense WSN. These research results enrich the current understanding regarding the traffic dynamics within WSNs, and provide a basis for further work on network optimization and anomaly detection for WSNs.

In the field of network optimization for WSNs, this dissertation presents network optimization models from which network performance bounds can be derived. This dissertation also investigates network performances constrained by the energy resources available in an indentified bottleneck zone. For a symmetric dense WSN, an optimal energy allocation scheme is proposed to minimize the energy waste due to the uneven energy drain among sensor nodes. By modeling the interrelationships among communication traffic, energy consumption and WSN performances, these presented results have efficiently integrated the knowledge on WSN traffic dynamics into the field of network optimization for WSNs.

Finally, in the field of anomaly detection for WSNs, this dissertation uses two examples to demonstrate the feasibility and the ease of detecting sensor network anomalies through the analysis of network traffic. The presented results will serve as an inspiration for the research community to develop more secure and more fault-tolerant WSNs.

Acknowledgements

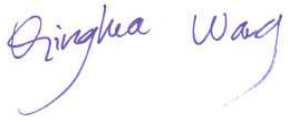
Firstly I would like to thank Prof. Tingting Zhang and Prof. Youzhi Xu, who have given me the opportunity to perform my Ph.D. studies at Mid Sweden University. One more thank to Prof. Tingting Zhang for her excellent supervision and fruitful cooperation. I also thank Dr. Stefan Pettersson who has been my co-supervisor and who has provided me with many constructive suggestions on my research work.

I would like to also thank all my previous and present colleagues working in the Department of Information Technology and Media. There are too many of you to mention, but rest assured, you are not forgotten. However, I would particularly like to thank Annika Berggren, Ulf Reiman, Ulf Jennehag, Linda Karlsson, Patrik Österberg, Roger Olsson, Cao Cao, etc. for their assistance on my work. I would particularly like to thank Theo Kanter, Jamie Walters, Xin Huang, Peng Cheng, Victor Kardeby, Stefan Forsström, Felix Dobslaw, Bengt Oelmann, Johanna Sefyrin, Zimic Sheila, Magnus Eriksson, Lena Höijer, Rahim Rahmani, Oliver Popov, Martin Kjellqvist, Aron Larsson, etc. for their kindness and friendship. Many thanks also go to the previous and present Chinese visiting reseachers, postdoctoral fellows, PhD students and exchange students for bringing happiness to my life.

Financial support from Mid Sweden University, the Knowledge Foundation (KK-stiftelsen), and the ARTES++ graduate school is also gratefully acknowledged.

But most of all I would like to thank my family for all their love and support. I would like to thank my kindly grandmother who passed away in November 2008. I would like to thank my parents, Ximo and Yinzhu, for their dedication and sacrifice to give me the opportunities that they never had. I would like to also thank my wife Miaomiao for her unconditional love and support.

Sundsvall, December 2009



Qinghua Wang

Contents

List of Papers and Contributions

This dissertation is mainly based on the following papers, herein referred to by their Roman numerals:

I Q. Wang, T. Zhang, and S. Pettersson, "Bounding the information collection performance of wireless sensor network routing," in *Proc. of the 5th Annual Conference on Communication Networks and Services Research (CNSR'07)*, pp. 55–62, Fredericton, Canada, May 2007.

II Q. Wang, and T. Zhang, "Detecting anomaly node behavior in wireless sensor networks," in *Proc. of the IEEE 21st International Conference on Advanced Information Networking and Applications Workshops (AINAW'07)*, vol. 1, pp. 451–456, Niagara Falls, Canada, May 2007.

III Q. Wang, T. Zhang, and S. Pettersson, "An effort to understand the optimal routing performance in wireless sensor network," in *Proc. of the IEEE 22nd International Conference on Advanced Information Networking and Applications (AINA'08)*, pp. 279–286, Okinawa, Japan, March 2008.

IV Q. Wang, and T. Zhang, "Source traffic modeling in wireless sensor networks for target tracking," in *Proc. of the 5th ACM International Symposium on Performance Evaluation of Wireless Ad-Hoc, Sensor, and Ubiquitous Networks (PE-WASUN'08)*, pp. 96–100, Vancouver, Canada, October 2008.

V Q. Wang and T. Zhang, "Bottleneck zone analysis in energy-constrained wireless sensor networks," *IEEE Communications Letters*, vol. 13, no. 6, pp. 423–425, June 2009.

VI Q. Wang, and T. Zhang, "Characterizing the traffic load distribution in dense sensor networks," to appear in *Proc. of the 2nd International Workshop on Wireless Sensor Networks: theory and practice (WSN'09)*, in conjunction with IFIP NTMS 2009, Cairo, Egypt, December 2009.

VII Q. Wang, and T. Zhang, "Fair energy allocation in large-scale and dense sensor networks," submitted to *IEEE Global Communications Conference (GLOBECOM'10)*, 2010.

Related papers by the author, but not explicitly included in this dissertation:

1 Q. Wang, "Malicious code detection in LRD networks," in *Proc. of Regional Inter-University Postgraduate Electrical and Electronic Engineering Conference (RIUPEEEC'05)*, Hongkong, China, July 2005.

2 Q. Wang, and T. Zhang, "Sec-SNMP: policy-based security management for sensor networks," in *Proc. of the International Conference on Security and Cryptography (SECRYPT'08)*, in conjunction with ICETE 2008, Porto, Portugal, July 2008.

3 Q. Wang, and T. Zhang, "Traffic load analysis in large-scale and dense wireless sensor networks," in *Proc. of the 6th Swedish National Computer Networking Workshop and 9th Scandinavian Workshop on Wireless Adhoc Networks (SNCNW+Adhoc'09)*, Uppsala, Sweden, May 2009.

4 Q. Wang, and T. Zhang, "Simulating wireless multihomed node in ns-2," in *Proc. of the 6th Swedish National Computer Networking Workshop and 9th Scandinavian Workshop on Wireless Adhoc Networks (SNCNW+Adhoc'09)*, Uppsala, Sweden, May 2009.

5 Q. Wang, and T. Zhang, "A multihoming extension of wireless node implementation in ns-2," in *Proc. of the 4th International Conference on Communications and Networking in China (ChinaCom'09)*, Xi'an, China, August 2009.

6 Q. Wang and T. Zhang, "A survey on security in wireless sensor networks," in *Security in RFID and Sensor Networks*, Y. Zhang and P. Kitsos, Eds. CRC Press, Taylor & Francis Group, 2009, Chapter 14, pp. 293–320.

7 Q. Wang, and T. Zhang, "Traffic load distribution in large-scale and dense wireless sensor networks," to appear in *Proc. of the 5th Annual International Wireless Internet Conference (WICON'10)*, Singapore, March 2010.

Other related contributions by the author:

Q. Wang, NS2-MIUN, MIUN's Sensor Network Extension to Ns-2, available at *http://apachepersonal.miun.se/∼qinwan/resources.htm*.

List of Figures

List of Tables

Terminology

Abbreviations and Acronyms

AODV	Ad hoc On-Demand Distance Vector Routing
ASCII	American Standard Code for Information Interchange
ASIC	Application-Specific Integrated Circuit
BSN	Body Sensor Network
CDF	Cumulative Distribution Function
DARPA	Defense Advanced Research Projects Agency
Dest	Destination
DSN	Distributed Sensor Network
DSP	Digital Signal Processor
DSR	Dynamic Source Routing
ECG	Electrocardiograph
FA	Flow Augmentation Routing
FN	False Negative
FP	False Positive
FPGA	Field Programmable Gate Array
FSM	Finite State Machine
ID	Identification
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IP	Internet Protocol
LEACH	Low Energy Adaptive Cluster Hierarchy Routing
MAC	Medium Access Control
MANET	Mobile Ad hoc Networking
MH	Minimum Hop Routing
MREP	Maximum Residual Energy Path Routing
MSN	Mobile Sensor Network
MTE	Minimum Total Energy Routing
MTPR	Minimum Total Transmission Power Routing
NS	Network Simulator

OS	Operating System
PEGASIS	Power Efficient Gathering in Sensor Information System Routing
QoS	Quality of Service
RAM	Random Access Memory
RF	Radio Frequency
RIFES	Routing Independent Fair Energy-Allocation Scheme
Src	Source
TDMA	Time Division Multiple Access
WSN	Wireless Sensor Network

Mathematical Notations Related to Traffic Analysis & Modeling

h	mean routing hop length
k	shape parameter of the generalized Pareto distribution
$Pt\_$	transmit power
r	a general node's distance from the sink
r'	a node's unified distance from the sink
r_S	node S's distance from the sink
R	the radius of the considered circular sensing field
$RXThresh\_$	lower bound on the receive power of any packet that can be successfully received
S_S	the proximity region of node S
$Traffic(r)$	The traffic load for a node which has a distance r from the sink
$\triangle r$	average distance between neighboring nodes
σ	scale parameter of the generalized Pareto distribution
θ	threshold parameter of the generalized Pareto distribution

Mathematical Notations Related to Network Optimization

A	the area of the sensing area
B_t	the bottleneck zone and its size
c	the total energy spent by the sensor nodes inside the bottleneck zone in order to relay a one-bit message from outside of the bottleneck zone to the sink
d_m	the characteristic distance which is the optimal routing hop length
$data_u$	the upper bound of the total information collected during network lifetime
D	the maximum transmission range of a node

e^b	the initial energy reserve of those nodes inside an identified bottleneck zone
e^R	the constant receiving power for receiving a packet
e^T	the energy/packet consumed by the transmitter electronics
e^r_{ij}	receiving power at node j to receive a packet from node i
e^t_{ij}	transmitting power at node i to send a packet to node j
$E(r)$	the part of energy allocated to a node which has a distance r from the sink
E^0_i	the initial energy reserve of node i
$E_{in,t}$	the total energy spent on sensing and relaying messages originated inside the bottleneck zone till time t
$E_{out,t}$	the total energy spent on relaying messages originated outside of the bottleneck zone by nodes inside the bottleneck zone till time t
E_{pkt}	the energy consumed by a packet transmission operation
E_{total}	the total amount of available energy
$Energy_{budget}$	the energy budget for energy allocation
$Energy_{final}$	the final allocated energy
f_{ij}	directional flow rate (packets per second) from i to j
F_{ij}	the total number of packets transmitted from i to j till the end of network operation
h	the mean routing hop length
l_{ij}	directional link state from i to j (1 for connectable, and 0 for non-connectable)
n	the total number of sensor nodes
o_i	pre-assigned sensing rate at node i
o'_i	sensing rate at node i after adjustment
$\bar{o}$	the average sensing rate
O_i	the total number of packets sampled by node i till the end of network operation
r_{ij}	the distance between node i and node j
S_{sensor}	the set of sensor nodes
S_{sink}	the set of sink nodes
t_u	the upper bound of network lifetime
$T_{(sys)}$	network lifetime
T_i	the runtime till the energy reserve of node i is exhausted or node i is disconnected from all sinks
Tw_i	useful working time of node i, $Tw_i = min(T_i, T_{(sys)})$
$Traffic(r)$	The traffic load for a node which has a distance r from the sink
$Traffic_{expected}$	a node's expected traffic load
$Traffic_{experienced}$	a node's experienced traffic load
$Traffic_{total}$	the total traffic load over all the deployed nodes
α	this parameter means there is a r^α loss due to channel transmission

α_{11}	the energy/bit consumed by the transmitter electronics
α_{12}	the energy/bit consumed by the receiver electronics
α_2	the energy dissipated in the transmit op-amp
α_1	$\alpha_1 = \alpha_{11} + \alpha_{12}$
β	the energy consumed by each packet sensing operation (Section 4.1)
β	the energy cost in sensing a data bit (Section 4.2)
σ_i	information weight of node i
ζ_{amp}	the energy dissipated in the transmit op-amp
ρ	node density
∂	the guaranteed normalized network information-collecting ability

Mathematical Notations Related to Anomaly Detection

$d(a, b)$	the difference between a and b
$d_{min}(u)$	the minimum distance between u and all the entries in a pattern table
k	the length of the scanning window (Section 5.1)
k	shape parameter of the generalized Pareto distribution (Section 5.2)
$Mean$	the mean value
$x\|F(x) = 0.99$	the corresponding value of x when $F(x) = 0.99$

Chapter 1

Introduction

Recent advances in wireless communications and electronics have enabled the development of low cost, low power, small size, yet reasonably efficient wireless sensor nodes. These tiny sensor nodes, which consist of sensing, data processing, communicating and power source components, make a new technological vision possible: Wireless sensor networks (WSNs).

WSNs combine short-range wireless communication, minimal computation facilities, and some kinds of sensing functions into a new form of network that can be deeply embedded in our physical environment. They involve deploying a large number of tiny sensor nodes in either hostile or non-hostile environments. The nodes then sense environmental changes and report them to other nodes (usually these are sink nodes connected to the end-user) over a flexible network architecture. Because there is no, or only a limited, infrastructure, WSNs are usually self-organized.

Based on the vision of WSNs, new types of applications become possible. Applications [1] include environmental controls such as fire fighting or marine ground floor erosion but also installing sensors on bridges or buildings to understand earthquake vibration patterns; many types of surveillance tasks such as intruder surveillance in premises; deeply embedding sensing into machinery where wired sensors would not be feasible, e.g., because wiring would be too costly, could not reach the deeply embedded points, limits flexibility, represents a maintenance problem, or disallows mobility of devices; tagging mobile items such as containers or goods in a factory floor automation system or smart price tags for foods that can communicate with the fridge; etc. In addition there are classes of applications including car-to-car or in-car communication.

1.1 Motivation

Although WSN is a promising technology which can be used in many applications, there are still a few obstacles to overcome before it finally becomes a mature technol-

ogy. One of the key obstacles is the energy constraint suffered by the most inexpensive sensor nodes, where batteries are the main source of power supply. Given this obstacle cannot be removed in the near future, optimizing the design of WSNs thus the minimum energy will be consumed is very important.

In WSNs, communication is believed to dominate the energy consumption [2]. Energy expenditure is less for sensing and computation. The energy cost of transmitting 1 Kb a distance of 100 meters is approximately the same as that for the execution of 3 million instructions by using a general-purpose processor [3]. Thus, minimizing the energy consumption due to communication is the key for the relief of the energy constraint in WSNs.

Currently, the knowledge about the communication in WSNs is still partial and vague, especially for traffic characteristics and communication patterns. Obviously, the knowledge about the traffic characteristics and communication patterns can aid in the understanding of the energy consumption and its distribution in WSNs. Thus, the investigation of traffic characteristics and communication patterns is a good starting point in the search for more energy-efficient WSNs. Following on from this it will be possible to propose new solutions for the design of WSNs in order to optimize the energy consumption.

Another concern for WSN technology involves security. WSNs will not be successfully deployed if the security issue is not addressed adequately. Security becomes more important because WSNs are usually used for very critical applications. Furthermore, WSNs are very vulnerable and thus attractive to malicious attacks because of their cheap prices, human-unattended deployment and the nature of wireless communication. The existing solutions to the security in WSNs include using key management and authentication [4]. However, these preventive mechanisms alone can not deter all possible attacks (e.g. insider attacks possessing the key). Actually, malicious attacks may exhibit anomalous behaviours in WSNs. With regard to communication, malicious attacks can trigger arbitrary communications, while a normal communication must follow protocol specifications and application scenarios. Thus, it should be interesting to investigate the possibility of detecting malicious attacks by identifying the anomalies exhibited within the WSNs' communication traffic.

Because sensor nodes are cheap devices and they can be deployed in harsh environments (e.g. battlefield, forest), they are prone to fail either by themselves or by means of others (e.g. enemies, animals, fire). Further, it is also common for battery-supported sensor nodes to fail because of energy exhaustion. To provide efficient maintenance for WSNs, those performing this maintenance require instant notifications about the sensor node failures. Because a failed sensor node cannot maintain efficient communication with the other nodes, sensor node failures have the possibility to be instantly noted by observing the degraded or lost communication with the failed nodes. This strategy has a similarity with the detection of traffic anomalies caused by malicious attacks. Both of them require comprehensive knowledge about the communication traffic in WSNs before they can identify any traffic anomaly.

1.2 Overall Aim

The aim of this dissertation is to investigate the communication traffic dynamics and patterns in WSNs and find their applications with reference to network optimization and anomaly detection.

1.3 Scope

The applications of WSNs are abundant. Because the communication traffic in WSNs is very dependent on the application scenario, only those selected typical WSN scenarios (e.g. surveillance, target tracking) will be investigated. Additionally different types of communication traffic exist, including data traffic, routing discovery traffic, link layer feedback and hello message, etc. This dissertation mainly focuses on data traffic and there is a limited involvement of other traffic types. The author agrees that aggregation can be very useful in energy-constrained WSNs. However, there is a significant variety within the different aggregation scenarios and they do in fact fall outside the scope of this dissertation.

1.4 Concrete and Verifiable Goals

This dissertation has three concrete goals: understanding the communication traffic dynamics and patterns in WSNs, optimizing WSNs based on the understanding of communication traffic, and detecting WSN anomalies based on the understanding of communication traffic.

1.4.1 Understanding the Communication Traffic Dynamics and Patterns in WSNs

A typical WSN consists of a large number of sensor nodes and a few sink nodes. The communication mainly occurs when a sensor node reports its data to a sink node, or when a query is sent out from a sink node to the interested sensor nodes. The ad hoc communication among sensor nodes is maintained at a minimum. Thus, the communication in WSNs is less dynamic than those in the traditional networks (e.g. Internet, ad hoc networks) and this makes it possible to precisely model the different aspects of communication behaviors in WSNs.

1.4.2 Optimizing WSNs Based on the Understanding of Communication Traffic

There are three main sources of energy consumption in WSNs: sensing, computing, and communicating, but communicating is believed to consume much more energy

than the other two. Thus, understanding the dynamics and patterns of communication traffic will provide a clue in relation to the means by which the energy is consumed in WSNs. In addition, at the time of writing this dissertation, the limited energy resources have been the main constraint on WSN performances. This means that the WSN performances can be optimized by optimizing the distribution of energy consumption and minimizing the energy waste. Because changing the communication also changes the distribution of energy consumption and the energy waste can be minimized if the distribution of energy consumption is known, the optimization of WSNs will benefit from an understanding of the communication traffic.

1.4.3 Detecting WSN Anomalies Based on the Understanding of Communication Traffic

WSNs consist of a large number of tiny sensor nodes which communicate with each other to form self-organized networks. Because of the nature of wireless communication, the behaviours of those tiny sensor nodes can be best observed in their communications. It has been mentioned that the communication in WSNs is less dynamic compared to other networks and precise traffic profiles can be built for those nodes of interest. Once the precise traffic profiles are built, they can be used to detect WSN anomalies. This is because a normal communication will obey the built traffic profiles while an abnormal communication will tend to violate the built traffic profiles. WSN anomalies can thus be detected by identifying the traffic profile change over time.

1.5 Outline

The work presented in this dissertation is divided according to the three goals presented in Section 1.4. Following this introduction, Chapter 2 provides a more detailed background relating to WSNs. There is particular importance given to existing works relating to traffic analysis & modeling, network optimization and anomaly detection for WSNs. The introduction of the author's own works are also integrated into the background presentation. Chapter 3 presents the author's contributions in the field of traffic analysis & modeling in WSNs and these contributions will assist the community to understand the communication traffic dynamics and patterns in WSNs. The contents of Chapter 3 are based on the included papers (Paper II, IV, VI). Chapter 4 presents the author's contributions within the field of WSN optimization and the contents there have been presented in the included papers (Paper I, III, V, VII). Chapter 5 presents the author's contributions within the field of anomaly detection in WSNs, and the contents there have been partially presented in the included papers (Paper II, IV). The results acquired in both Chapter 4 and Chapter 5 have benefited from the author's prior research and understanding of the communication traffic dynamics and patterns in WSNs. Finally, Chapter 6 concludes this dissertation and offers suggestions for future work.

1.6 Contributions

The author's contributions to the research community are mainly provided in the included publications - in which the author has performed the greater part of the development, simulation, evaluation, analysis and presentation. The main contributions are stated here.

Traffic analysis & modeling has been among the least developed areas related to WSNs. The knowledge relating to the detailed traffic characteristics can aid in understanding the network, its devices, services and vulnerabilities. The author contributes to this field by presenting several ways of modeling different aspects of WSN traffic, and by signaling the importance of this field to the research community. More specifically, the author suggests in Paper II and Section 3.2 of this dissertation that the packet arriving sequence at the place of individual sensor nodes can be modelled, and this can be conducted by extracting all given length unique subsequences observed during a sufficiently long time span and constructing a pattern database using these extracted subsequences. Following on from this, the pattern database can be used as the traffic profile for the corresponding node. For an event-driven sensor network, bursty traffic can be triggered upon detection of an interesting event. Traditional Poisson processes have been shown to be inappropriate in modeling the bursty traffic [5, 6]. The author proposes in Paper IV and Section 3.3 of this dissertation the use of an ON/OFF model to capture the burstiness of communication traffic in event-driven WSNs. Furthermore, the ON/OFF period distributions and their properties are studied. Usually, traffic load is not evenly distributed over the nodes in a WSN. Understanding the traffic load distribution can guide the network-wide energy allocation, direct the design of routing protocols, and optimize the node deployment. In Paper VI and Section 3.4 of this dissertation, the expected traffic load at a node of interest is estimated by counting the number of one-hop and multi-hop child nodes of the node of interest. Finally, the traffic load distribution over the nodes in a considered dense sensor network scenario is determined as a function of the sensor nodes' distances from the sink. An initial version concerning the traffic load distribution has been presented in the related but not included Paper 3. In the related but not included Paper 7, a similar problem is addressed based on a hypothesized routing algorithm which is not used in Paper VI and this dissertation.

WSN optimization has been a hot research topic. The optimization of WSNs include efforts relating to resource allocation, protocol design, and cross-layer optimization, etc. Because energy resource has been the main constraint in WSNs and it usually limits the network lifetime acquired, it has been the main concern in many WSN optimization problems. The author has noticed that there are interrelationships within the communication traffic, energy consumption and WSN performances. By analyzing and modeling the energy consumed by the communication traffic, different WSN optimization problems are formed and insights are then provided. In Paper I, Paper III and Section 4.1 of this dissertation, WSN optimization is formulated as a non-linear optimization problem and then relaxed to a linear programming (LP) problem. Being different to other proposed optimization models [7, 8, 9] where a confining and absolute definition for network lifetime is used, a new definition of

network lifetime is used in our model. The contribution of this definition is that it incorporates the consideration that different application scenarios can have variable tolerances in relation to the number of dead nodes. In Paper V and Section 4.2 of this dissertation, the maximum energy consuming rate due to the communication traffic within a defined bottleneck zone is analyzed. Utilizing the knowledge that there are limited energy resources available in this zone, the author presents the analytical results concerning the performance upper bounds which are limited by the bottleneck zone. In Paper VII and Section 4.3 of this dissertation, the distribution of energy consumption is mapped to the distribution of traffic load in a dense sensor network. By allocating the precious energy resources over the individual sensor nodes according to their expected traffic loads, the author's proposed energy allocation scheme is shown to be capable of minimizing the energy waste and thus maxmizing the network performances.

Identifying misbehaviours is an important challenge for monitoring, fault diagnosis and intrusion detection in WSNs. Usually, such a task is fulfilled by anomaly detection which detects interesting changes from the normal observed behaviour. In this dissertation, the author investigates the feasibility of detecting network anomalies by identifying the traffic profile change over time. Another popular area of anomaly detection, which deals with the detection of incorrect sensor measurements, will not be dealt with in this dissertation. In the related but not included Paper 6, the author has made a comprehensive survey on WSN security including anomaly detection in WSNs. In Paper II and Section 5.1 of this dissertation, the author proposes a method of detecting malicious attacks by identifying those anomalies exhibited in the packet arriving sequences. The examples given show that it is easy for malicious attacks to violate the normal traffic profile represented by those learned packet arriving sequences. In Paper IV and Section 5.2 of this dissertation, the sensor nodes which have been active in participating in communication for an unusually long time span or have been silent for an unusually long time span can be signs of an energy-consuming attack or a node failure. The author shows that those highly anomalous active/silent sensor nodes can be detected quickly in the author's considered WSN scenario.

Except for the above stated contributions, the author has also developed an ns2 [10] extension called NS2-MIUN [11] which supports the simulation of WSNs. The simulations in the author's Paper IV and Paper 5 are based on NS2-MIUN. NS2-MIUN is made available publicly and has been recommended by ns2 forums [12]. According to the queries received by the author, NS2-MIUN has been used by students worldwide in their thesis projects.

1.6.1 Abstracts of the Selected Papers

Paper I: Bounding the information collection performance of wireless sensor network routing

Wireless sensor networks have mainly been designed for information-collecting purposes, such as habitat monitoring, product process tracing, battlefield surveillance,

etc. In order to support efficient communications for such networks, many routing protocols have been proposed. However, protocol designs are out-pacing formal analysis. We propose an optimization model in this paper to bound the routing performance in terms of network information collection. We first argue that a network can only be given a death sentence when it fails to satisfy the application's requirement and we propose the adoption of a more reasonable network lifetime definition. Then, the optimization model concerning maximizing information collection routing is presented based on this new network lifetime definition. Existing typical routing algorithms: MH, MTE, FA and MREP are simulated as references in order to validate the model proposed. Results show our model can provide a tight upper bound and thus can be used to evaluate existing and up-coming routing algorithms.

Paper II: Detecting anomaly node behavior in wireless sensor networks

Wireless sensor networks are usually deployed in a way "once deployed, never changed". The actions of sensor nodes are either pre-scheduled inside chips or triggered to respond to outside events in the predefined way. This relatively predictable working flow makes it easy to build accurate node profiles and detect any violation of normal profiles. In this paper, observed traffic patterns are used to model node behaviour in wireless sensor networks. Firstly, selected traffic related features are used to translate observed packets into different events. Following this, unique patterns based on the arriving order of different packet events are extracted to form the normal profile for each sensor node during the profile learning stage. Finally, real time anomaly detection can be achieved based on the profile matching.

Paper III: An effort to understand the optimal routing performance in wireless sensor network

Wireless sensor network is remarkable for its promising use in relation to human-unattended information collection, such as forest fire monitoring. In order to support efficient communication, many specially designed routing algorithms for such networks have been proposed. However, at the present time it remains unclear as to whether these proposed routing algorithms are already good enough or still have a long way to go to perfect, since there is currently a lack of understanding about the optimal routing performance.

This paper makes some progress in the understanding of the optimal routing performance. The metrics used in this case to measure the routing performance are the final acquired network lifetime and the final collection of the total information. The condition used to judge the network's death is defined by the user's requirement for the guaranteed network information collecting ability. Optimization models based on the metrics and death condition mentioned above are proposed. Experiments show that some existing routing proposals are already working in a satisfactory manner when the user's requirement is strict, but few are equally satisfactory when the user's requirement is loose.

Paper IV: Source traffic modeling in wireless sensor networks for target tracking

Researches around wireless sensor network (WSN) have recently been particularly prolific. However, traffic modeling related WSN research has been rather less satisfactory. In this paper, source traffic dynamics in a simulated target tracking WSN scenario are explored. We find that the source traffic arrival process does not follow the usually considered Poisson model. Instead, an ON/OFF model is found to be capable of capturing the burst nature of the source traffic arrival. In addition, we find the measured ON/OFF periods perfectly follow the generalized Pareto distribution. Mathematical analysis also shows a surprising fact: all ON/OFF period distributions in the experiment exhibit a short-tail property, which is a nice property that could be exploited by applications such as anomaly detection and node failure detection.

Paper V: Bottleneck Zone Analysis in Energy-Constrained Wireless Sensor Networks

In a typical sensor network, nodes around the sink consume more energy than those further away. It is not unusual for the limited energy resources available at the nodes around the sink to become the bottleneck which confines the performance of the whole network. In this paper, we firstly present our considered bottleneck zone in a general sensor network scenario. Then, the effect of the bottleneck zone on network performance is investigated by deducing the performance bounds imposed by the energy resources available inside the bottleneck zone. Both the performance bound in terms of the network lifetime and the performance bound in terms of the information collection are explored. Finally, the ways by which network deployment variables may affect the performance bounds are analyzed.

Paper VI: Characterizing the Traffic Load Distribution in Dense Sensor Networks

Traffic load is not evenly distributed over the nodes in a wireless sensor network (WSN). Understanding the traffic load distribution can guide the network-wide energy allocation, direct the design of routing algorithms, and optimize the node deployment in WSNs. In this paper, we consider a dense WSN with nodes uniformly distributed in a disk sensing area, and find the traffic load distribution over the nodes as a function of their distance from the sink. Additionally, the effects of network scale and routing strategy on traffic load are also investigated. The traffic loads on individual nodes are found to be in direct proportion to the radius of the network and in inverse proportion to the routing hop length, while independent of network density. The results presented in this paper are verified by means of extensive simulation experiments.

Paper VII: Fair Energy Allocation in Large-Scale and Dense Sensor Networks

Individual sensor nodes in a wireless sensor network usually suffer from energy constraints. The multi-hop transmission mode in a large-scale sensor network causes a deterioration in this constraint by consuming much more energy on the neighboring nodes to a sink node. Usually the energy allocation in wireless sensor networks is not good at considering individual energy consuming rates. An ill-considered energy allocation scheme will result in one part of the sensor nodes dying at an earlier stage than the others and the network performance will thus be degraded. In this paper, we present an efficient fair energy allocation scheme which allocates energy to a sensor node according to its expected traffic load. In concrete terms, we consider a large-scale and dense sensor network with a large number of sensor nodes evenly deployed in a disk area and a sink node located at the center. We firstly determine the traffic load distribution over the sensor nodes as a function of their distance from the sink. Afterwards, a traffic load based energy allocation scheme is proposed given the total amount of energy available and the total number of deployed sensor nodes. The proposed energy allocation scheme is considered to be fair as it is the case that the allocated energy on each individual sensor node is matched to the expected traffic load for this node and thus all sensor nodes have the same energy exhaustion time. Further, the underlying routing strategy (differentiated by the mean routing hop length) is found to have no effect on the proposed energy allocation scheme. Thus, the fair energy allocation scheme proposed is also considered to be generally applicable.

Chapter 2

Background

This chapter provides a detailed introduction to the history and current state of the art with regard to wireless sensor networks (WSNs). An elaboration regarding the existing efforts on the areas of traffic analysis & modeling, network optimization and anomaly detection for WSNs is also given.

2.1 Wireless Sensor Networks (WSNs)

2.1.1 History

The origins of the research on WSNs can be traced back to the Distributed Sensor Networks (DSN) program at the Defense Advanced Research Projects Agency (DARPA) at around 1980 [13]. By this time, the Arpanet (predecessor of the Internet) had been operational for a number of years, with about 200 hosts at universities and research institutes. DSNs were assumed to have many spatially distributed low-cost sensing nodes that collaborated with each other but operated autonomously, with information being routed to whichever node was best able to use the information. It was an ambitious program given the state of the art at that time. This was prior to the era of personal computers and workstations; processing was mostly performed on minicomputers and the Ethernet was just becoming popular. Technology components for a DSN were identified in a Distributed Sensor Nets wrokshop in 1978 [14]. These included sensors (acoustic), communication, processing techniques and algorithms, and distributed software. Researchers at Carnegie Mellon University (CMU) even developed a communication-oriented operating system called Accent [15], which allowed flexible, transparent accesss to distributed resources required for a fault-tolerant DSN. A demonstrative application of DSN was a helicopter tracking system [16], using a distributed array of acoustic microphones by means of signal abstractions and matching techniques, developed at the Massachusetts Institute of Technology (MIT).

Even though early researchers on sensor networks had in mind the vision of a DSN, the technology was not quite ready. More specifically, the sensors were rather large (i.e. shoe box and up) which limited the number of potential applications. Further, the earliest DSNs were not tightly associated with wireless connectivity. Recent advances in computing, communication and microelectromechanical technology have caused a significant shift in WSN research and brought it closer to achieving the original vision. The new wave of research in WSNs started in around 1998 and has been attracting more and more attention and international involvement. In the new wave of sensor network research, networking techniques and networked information processing suitable for highly dynamic ad hoc environments and resource-constrained sensor nodes have been the focus. Further, the sensor nodes have been much smaller in size (i.e. pack of cards to dust particle) and much cheaper in price, and thus many new civilian applications of sensor networks such as environment monitoring, vehicular sensor network and body sensor network have emerged. Again, DARPA acted as a pioneer in the new wave of sensor network research by launching an initiative research program called SensIT [17] which provided the present sensor networks with new capabilities such as ad hoc networking, dynamic querying and tasking, reprogramming and multi-tasking. At the same time, the IEEE noticed the low expense and high capabilities that sensor networks offer. The organization has defined the IEEE 802.15.4 standard [18] for low data rate wireless personal area networks. Based on IEEE 802.15.4, ZigBee Alliance [19] has published the ZigBee standard which specifies a suite of high level communication protocols which can be used by WSNs. Currently, WSNs has been viewed as one of the most important technologies for the 21st century [20]. Countries such as China have involved WSNs in their national strategic research programmes [21]. The commercialization of WSNs are also being accelerated by new formed companies like Crossbow Technology [22] and Dust Networks [23].

2.1.2 Hardware Platform

A WSN consists of spatially distributed sensor nodes. A sensor node, also known as a 'mote', is a node that is capable of performing some processing, gathering sensory information and communicating with other connected nodes in the network. Fig. 2.1 shows a photograph of Mica2 Mote developed by Crossbow Technology [22]. The usual hardware components of a sensor node is shown in Fig. 2.2.

As seen from Fig. 2.2, the main components of a sensor node are the embedded processor, transceiver, memory, power source and one or more sensors.

Embedded Processor

The embedded processor performs tasks, processes data and controls the functionality of other components in the sensor node. The alternatives that can be used as an embedded processor include Microcontroller, Digital Signal Processor (DSP), Field Programmable Gate Array (FPGA) and Application-Specific Integrated Circuit (ASIC). Among all these alternatives, the Microcontroller has been the most used

Figure 2.1: Mica2 Mote

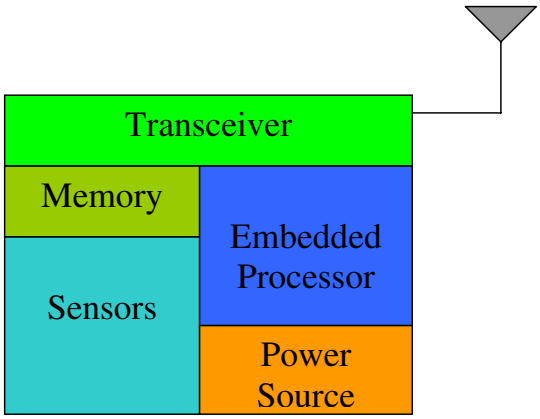


Figure 2.2: The components of a sensor node

embedded processor for sensor nodes because of its flexibility to connect to other devices and its cheap price. For example, the newest CC2531 development board provided by Chipcon (acquired by Texas Instruments) uses 8051 microcontroller, and the Mica2 Mote platform provided by Crossbow uses ATMega128L microcontroller.

Transceiver

A transceiver is responsible for the wireless communication of a sensor node. The various choices of wireless transmission media include Radio Frequency (RF), Laser and Infrared. RF based communication fits to most of WSN applications. The operational states of a transceiver are Transmit, Receive, Idle and Sleep. Mica2 Mote uses two kinds of RF radios: RFM TR1000 and Chipcon CC1000. The outdoor transmission range of Mica2 Mote is about 150 meters.

Memory

Memories in a sensor node include in-chip flash memory and RAM of a microcontroller and external flash memory. For example, the ATMega128L microcontroller running on Mica2 Mote has 128-Kbyte flash program memory and 4-Kbyte static RAM. Further, a 4-Mbit Atemel AT45DB041B serial flash chip can provide external memories for Mica and Mica2 Motes [24].

Power Source

In a sensor node, power is consumed by sensing, communication and data processing. More energy is required for data communication than for sensing and data processing. Power can be stored in batteries or capacitors. Batteries are the main source of power supply for sensor nodes. For example, Mica2 Mote runs on 2 AA batteries. Due to the limited capacity of batteries, minimizing the energy consumption is always a key concern during WSN operations. To remove the energy constraint, some preliminary research working on energy-harvesting techniques for WSNs has also been conducted. Energy-harvesting techniques convert ambient energy (e.g. solar, wind) to electrical energy and the aim is to revolutionize the power supply on sensor nodes. A survey about the energy-harvesting sensor nodes is provided by [25].

Sensors

Sensors are hardware devices that produce a measurable response to a change in a physical condition such as temperature, pressure and humidity. The continual analog signal sensed by the sensors is digitized by an analog-to-digital converter and sent to the embedded processor for further processing. Because a sensor node is a micro-electronic device powered by a limited power source, the attached sensors should also be small in size and consume extremely low energy.

2.1.3 Operating System

The role of any operating system (OS) is to promote the development of reliable application software by providing a convenient and safe abstraction of hardware resources. OSs for WSN nodes are typically less complex than general-purpose OSs both because of the special requirements of WSN applications and because of the resource constraints in WSN hardware platforms.

TinyOS [26] is perhaps the first operating system specifically designed for WSNs. It features a component-based architecture which enables rapid innovation and implementation while minimizing code size as required by the severe memory constraints inherent in WSNs. TinyOS's component library includes network protocols, distributed services, sensor drivers, and data acquisition tools - all of which can be further refined for a custom application. Unlike most other OSs, TinyOS is based

on an event-driven programming model instead of multithreading. TinyOS programs are composed into event handlers and tasks with run-to-completion semantics. When an external event occurs, such as an incoming data packet or a sensor reading, TinyOS calls the appropriate event handler to handle the event. Event handlers can post tasks that are scheduled by the TinyOS kernel at a later stage. Both the TinyOS system and programs written for TinyOS are written in a special programming language called nesC which is an extension of the C programming language. NesC is designed to detect race conditions between tasks and event handlers. Currently, TinyOS has been ported to over a dozen platforms and numerous sensor boards. A wide community uses it in simulation to develop and test various algorithms and protocols. According to the figure published on TinyOS forum, over 500 research groups and companies are using TinyOS on the Berkeley/Crossbow Motes. Because TinyOS is open source, numerous groups are actively contributing code to the development of TinyOS and thus making it even more competitive.

Contiki [27] is another open source OS specifically designed for WSNs. The Contiki kernel is event-driven, like TinyOS, but the system supports multithreading on a per-application basis. Furthermore, Contiki includes protothreads that provide a thread-like programming abstraction but with a very small memory overhead. Contiki provides IP communication, both for IPv4 and IPv6. Many key mechanisms and ideas from Contiki have been widely adopted within the industry. The uIP embedded IP stack, originally released in 2001, is today used by hundreds of companies in systems such as freighter ships, satellites and oil drilling equipment. Contiki's protothreads, first released in 2005, have been used in many different embedded systems, ranging from digital TV decoders to wireless vibration sensors. Contiki's idea of using IP communication in low-power WSNs has lead to an IETF standard and an international industry alliance - IP for Smart Objects (IPSO) Alliance [28].

There are also other OSs that can be used by WSNs. For example, SOS [29] is an event-driven OS for mote-class sensor nodes that adopts a more dynamic point on the design spectrum. The prime feature of SOS is its support for loadable modules. A complete system is built from smaller modules, possibly at run-time. to support the inherent dynamism in its module interface, SOS also focuses on supporting dynamic memory management. Unfortunately, SOS is no longer under active development due to the graduation of the core developers. LiteOS [30] is an open source, interactive, UNIX-like operating system designed for WSNs. With the tools that come from LiteOS, it is possible to operate one or more WSNs in a Unix-like manner. It is also possible to develop programs for nodes, and wirelessly distribute such programs to sensor nodes.

2.1.4 Networking

Network Architecture

A WSN is a network consisting of numerous sensor nodes with sensing, wireless communications and computing capabilities. These sensor nodes are scattered in an unattended environment (i.e. sensing field) to sense the physical world. The

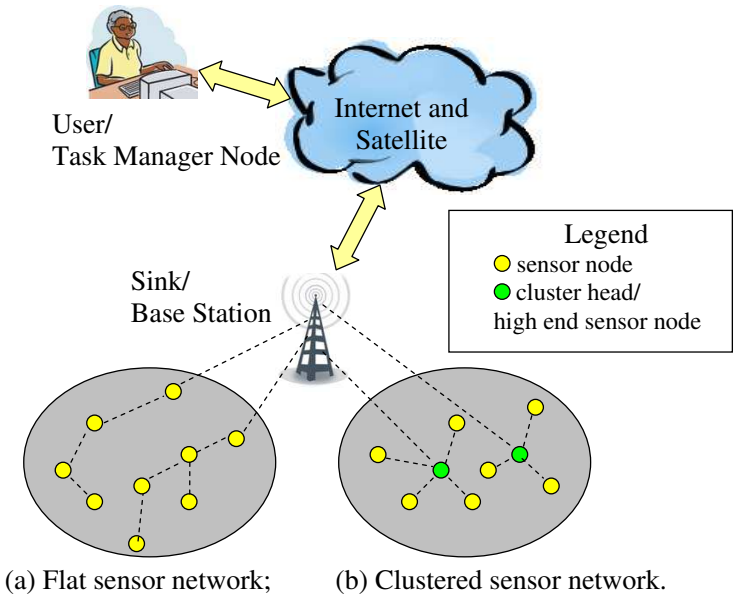


Figure 2.3: The Operation of WSNs

sensed data can be collected by a few sink nodes which have access to infrastructured networks like the Internet. Finally, an end user can remotely fetch the sensed data by accessing infrastructured networks. Fig. 2.3 shows the operation sketch map of WSNs.

In Fig. 2.3, two kinds of network topologies are shown. The sensor nodes either form a flat network topology where sensor nodes also act as routers and trasfer data to a sink through multi-hop routing, or a hierarchical network topology where more powerful fixed or mobile relays are used to collect and route the sensor data to a sink.

Protocol Stack of WSNs

The protocol stack used by the sink, cluster head and sensor nodes are shown in Fig. 2.4. According to [31], the sensor network protocol stack is much like the traditional protocol stack, with the following layers: application, transport, network, data link, and physical. The physical layer is responsible for frequency selection, carrier frequency generation, signal detection, modulation and data encryption. The data link layer is responsible for the multiplexing of data streams, data frame detection, medium access and error control. It ensures reliable point-to-point and point-to-multipoint connections in a communication network. The network layer takes care of routing the data supplied by the transport layer. The network layer design in WSNs must consider the power efficiency, data-centric communication, data aggregation, etc. The transportation layer helps to maintain the data flow and may be

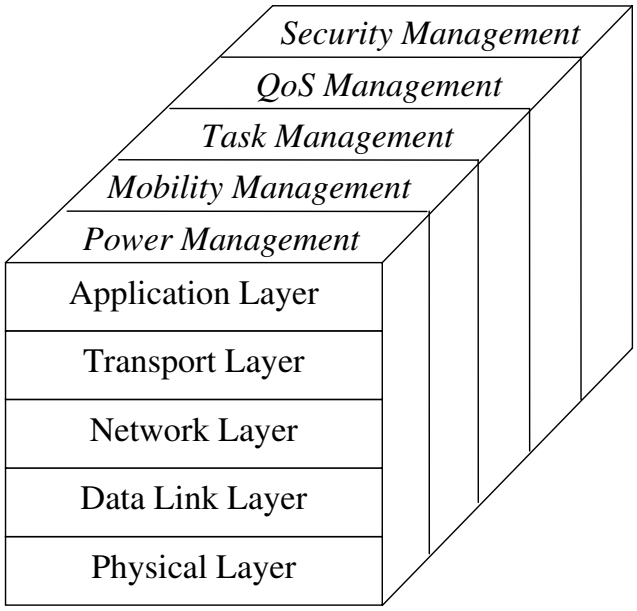


Figure 2.4: The Protocol Stack of WSNs

important if WSNs are planned to be accessed through the Internet or other external networks. Depending on the sensing tasks, different types of application software can be set up and used on the application layer.

WSNs must also be aware of the following management planes in order to function efficiently: mobility, power, task, quality of service (QoS) and security management planes. The power management plane is responsible for minimizing power consumption and may turn off functionality in order to preserve energy. The mobility management plane detects and registers movement of nodes so a data route to the sink is always maintained. The task management plane balances and schedules the sensing tasks assigned to the sensing field and thus only the necessary nodes are assigned with sensing tasks and the remainder are able to focus on routing and data aggregation. QoS management in WSNs [32] can be very important if there is a real-time requirement with regard to the data services. QoS management also deals with fault tolerance, error control and performance optimization in terms of certain QoS metrics. Security management is the process of managing, monitoring, and controlling the security related behavior of a network. The primary function of security management is in controlling access points to critical or sensitive data. Security management also includes the seamless integration of different security function modules, including encryption, authentication and intrusion detection. Please refer to the author's publications [4, 33] for more information about security management in WSNs. It is obvious that networking protocols developed for WSNs must address all five of these management planes.

2.1.5 Applications

The original motivation behind the research into WSNs was military application. Examples of military sensor networks include large-scale acoustic ocean surveillance systems for the detection of submarines, self-organized and randomly deployed WSNs for battlefield surveillance and attaching microsensors to weapons for stockpile surveillance [34]. As the costs for sensor nodes and communication networks have been reduced, many other potential applications including those for civilian purposes have emerged. The following are a few examples.

Environmental Monitoring

Environmental monitoring [35] can be used for animal tracking, forest surveillance, flood detection, and weather forecasting. It is a natural candidate for applying WSNs [13], because the variables to be monitored, e.g. temperature, are usually distributed over a large region. One example is that researchers from the University of Southampton have built a glacial environment monitoring system using WSNs in Norway [36]. They collect data from sensor nodes installed within the ice and the sub-glacial sediment without the use of wires which could disturb the environment. Another example is that researchers from EPFL have performed outdoor WSN deployments on a rugged high mountain pass located between Switzerland and Italy [37]. Their WSN deployment is used to provide spatially dense measures to the Swiss authorities in charge of risk management, and the resulting model will assist in the prevention of avalanches and accidental deaths.

Health Monitoring

WSNs can be embedded into a hospital building to track and monitor patients and all medical resources. Special kinds of sensors which can measure blood pressure, body temperature and electrocardiograph (ECG) can even be knitted into clothes to provide remote nursing for the elderly. When the sensors are worn or implanted for healthcare purposes, they form a special kind of sensor network called a body sensor network (BSN). BSN is a rich interdisciplinary area which revolutionizes the healthcare system by allowing inexpensive, continuous and ambulatory health monitoring with real-time updates of medical records via the Internet. The earliest research on BSNs was conducted in Imperial College London, where a specialized BSN sensor node and BSN Development Kit have been developed [38].

Traffic Control

Sensor networks have been used for vehicle traffic monitoring and control for some time. At many crossroads, there are either overhead or buried sensors to detect vehicles and to control the traffic lights. Furthermore, video cameras are also frequently used to monitor road segments with heavy traffic. However, the traditional commu-

nication networks used to connect these sensors are costly, and thus traffic monitoring is usually only available at a few critical points in a city. WSNs will completely change the landscape of traffic monitoring and control by installing cheap sensor nodes in the car, at the parking lots, along the roadside, etc. Streetline, Inc. [39] is a company which uses sensor network technology to help drivers find unoccupied parking places and avoid traffic jams. The solutions provided by Streetline can significantly improve the city traffic management and reduce the emission of carbon dioxide.

Industrial Sensing

As plant infrastructure ages, equipment failures cause more and more unplanned downtime. The ARC Advisory Group estimates that 5% of production in North America is lost to unplanned downtime. Because sensor nodes can be deeply embedded into machines and there is no infrastructure, WSNs make it economically feasible to monitor the "health" of machines and to ensure safe operation. Aging pipelines and tanks have become a major problem in the oil and gas industry. Monitoring corrosion using manual processes is extremely costly, time consuming, and unreliable. A network of wireless corrosion sensors can be economically deployed to reliably identify issues before they become catastrophic failures. Rohrback Cosasco Systems (RCS) [40] is the world leader in corrosion monitoring technology and is applying WSNs in their corrosion monitoring. WSNs have also been suggested for use in the food industry to prevent the incidents of contaminating the food supply chain [41].

Infrastructure Security

WSNs can be used for infrastructure security and counterterrorism applications. Critical buildings and facilities such as power plants, airports, and military bases have to be protected from potential invasions. Networks of video, acoustic, and other sensors can be deployed around these facilities. An initiative in Shanghai Pudong International Airport has involved the installation of a WSN-aided intrusion prevention system on its periphery to deter any unexpected intrusions. The Expo 2010 Shanghai China [42] is also going to secure its expo sites with the same intrusion prevension system.

2.2 Traffic Analysis & Modeling for WSNs

WSNs consist of a large number of tiny and cheap sensor nodes that cooperatively sense a physical phenomenon. Existing research results and products have provided the possibility to build effective WSNs for many applications. If the traffic features inside WSNs were better understood then the WSNs could be made to be even more effective. For example, better routing protocols and sensor deployment strategy could be designed if the traffic burden among the sensors was better understood.

Better fault and security management could be applied if normal and abnormal traffic could be kept apart according to traffic features.

The traffic dynamics for different types of traditional networks, both wired and wireless, have been investigated in the literature. However, the specialty of WSNs makes a reinvestigation of traffic dynamics necessary. Constructing accurate and analytically tractable models for sensor network traffic will provide a basis for future work on network design, optimization and security. Unfortunately, at the time that this dissertation was written, research regarding traffic modeling and analysis in WSNs was still rather limited. The few studies that do exist include work focusing on data traffic arrival process, sequence relations among general kinds of packets, and data traffic load distribution.

2.2.1 Data Traffic Arrival Process

Because the data traffic dynamics in different WSN scenarios are quite different, the data traffic modeling and analysis in WSNs will be quite application dependent. Ref. [43] suggests that WSN applications can be categorized as *event-driven* or *periodic data generation*. For periodic data generation scenarios, constant bit rate (CBR) can be used to model the data traffic arrival process when the bit rate is constant [44]. When the bit rate is variable, a Poisson process can be used to model the data traffic arrival process as long as the data traffic is not bursty [45]. For event-driven scenarios such as *target detection* and *target tracking*, bursty traffic can arise from any corner of the sensing area if an event is detected by the local sensors. A Poisson process has also been used to model the traffic arrival process in an event-driven WSN [46]. However, there is no solid ground to support the use of a Poisson process in this case. Actually, the widely used Poisson processes are quite limited in their burstiness [5, 6]. Instead of using Poisson processes, the author of this dissertation proposes to use an ON/OFF model to capture the burst phenomenon in the source data traffic of an event-driven WSN [47]. Further, the distributions of ON/OFF periods are found to follow the generalized Pareto distribution in his considered WSN scenario. Ref. [48] studies a different WSN scenario - a mobile sensor network (MSN). In an MSN, the node mobility introduces new dynamics to network traffic. The authors of [48] find that the mobility variability of humans (in this case, sensor nodes are attached to humans) and the spatial correlation of the collected information lead to the pseudo-LRD (i.e. long range dependent) traffic, which exhibits characteristics significantly different to that of Markovian traffic.

2.2.2 Sequence Relations among General Kinds of Packets

Sequence relations exist in some kinds of packets. For example, a Routing Reply message always comes after a Routing Request message and that is specified by any ordinary routing protocol. In [49], the authors propose to use a finite state machine (FSM) to specify correct routing behavior for the ad hoc on demand distance vector (AODV) routing [50]. The rationale behind this is that the AODV protocol

has specified the sequence relations among different kinds of routing messages and such sequence relations can be depicted by an FSM. The authors in [51] also use FSM to model the correct routing behavior for the dynamic source routing (DSR) [52]. Because the routing protocols AODV and DSR have clearly specified the routing operations, the sequence relations among different kinds of routing packets can be manually abstracted into an FSM. In both [49] and [51], the authors have used their FSMs to validate real-time routing behaviors and detect possible malicious attacks.

In addition to that the sequence relations among some special kinds of packets (e.g. routing messages) are possible to be specified according to protocol specifications, the author of this dissertation suggests that the sequence relations among general kinds of packets can also be learned automatically by on-line training. In [53], the author of this dissertation firstly classifies the arriving packets according to their attributes (e.g. packet type, addresses) and then maps the packet arriving sequence to an infinite character string. Afterwards, the on-line learning of the packet sequence relations are conducted by extracting every unique character substring encountered during the window-based scanning process. The learned packet sequence relations can be used to build the normal traffic profile for the node of interest in a static WSN. In a dynamic WSN in which some of the nodes are mobile, the traffic profile learned in this manner will evolve quickly over time and will thus be less meaningful.

2.2.3 Data Traffic Load Distribution

In a WSN, the data traffic load is not evenly distributed over the nodes. For example, the sensors which are one hop away from the sink, relay the entire network's data traffic. This imbalanced data traffic load distribution can degrade the network's lifetime and functionality. Hence, efforts have been devoted to characterizing the data traffic load distribution in WSNs. Ref. [54] proposes an analytical analysis on the data traffic load distribution over a randomly deployed linear WSN. It has been shown that the data traffic load over a node increases the closer it is to the sink, however, a reduction in the data traffic load is expected for sensors that are very close to the sink. In [55, 56, 57], data traffic load is formularized as a function of the distance to the sink in dense planar WSNs. In a similar manner to that in a linear WSN, the data traffic load over a node in planar WSNs also increases as the node moves closer to the sink. For a symmetric sensor network (i.e. all nodes of the same distance from the center of the network are similar) with nodes evenly distributed in the sensing field, the author of this dissertation concludes that the expected data traffic load over a node is in direct proportion to the network radius, in inverse proportion to the mean routing hop length, and independent of the node density [56, 57].

Because the data traffic load distribution is closely related to the energy consumption and the latter has a significant impact on the performance of WSNs, the research results concerning the data traffic load distribution can be used to optimize the performance of WSNs. For example, the author of this dissertation has proposed an optimal energy allocation scheme for WSNs based on the understanding of the data traffic related energy consumption in the network [58]. More details relating to this

will be provided in Section 2.3. Since this dissertation will not consider traffic load other than data traffic load, "traffic load" and "traffic load distribution" will be respectively used to refer to "data traffic load" and "data traffic load distribution" in the following sections of this dissertation unless explicitly stated.

2.3 Network Optimization for WSNs

There are many network optimization problems to be solved in WSNs, such as rate control, flow control, congestion control, medium access control, queue management, power control and topology control, etc. [59]. It is difficult to provide a complete overview in relation to all issues relating to network optimization in WSNs. However, it is worthwhile, none the less, to aim for a fairly comprehensive summary of important topics, with particular emphasis on energy optimization.

2.3.1 Energy-Efficient Routing Design

Because communication dominates the critical energy consumption, routing design is usually considered to be the core of sensor network design. Many routing algorithms have been proposed in prior research. The shortest path is the typical and fundamental consideration for network flow routing problems. A simple translation of this consideration in sensor network routing is the minimum hop (MH) routing. The AODV routing is an example of using the number of link hops as its routing metric. However as the limitation of battery power is one of the most fundamental aspects of sensor networks, routing algorithms for sensor networks generally attempt to minimize the utilization of this valuable resource. Many researchers have proposed shortest path algorithms in order to minimize the utilization of energy. For example, the minimum total transmission power routing (MTPR) proposed in [60] and the minimum total energy (MTE) routing introduced in [7, 61] attempt to reduce the total transmission energy per data bit, where the path length is the sum of energy expended per data bit during its transmission over each link in the forwarding path.

It was realized by the sensor network research community that improving the ratio of packets transmitted to energy consumed by the network is, by itself, not a good measure of the efficiency of the network [62]. Ref. [61] proposes an algorithm which attempts to minimize the variation in node energy levels. This metric ensures that all the nodes in the network remain up and running together for as long as possible. A flow augmentation (FA) [7, 63] algorithm incorporates MH, MTE, and other residual energy considered routing algorithms together with adjustable parameters. The maximum residual energy path (MREP) routing [8, 63] is an algorithm based on similar considerations which attempts to postpone the death of the first node by using the maximum remaining energy path.

To provide more insights into the energy-efficient routing design, a theoretical analysis concerning the optimal routing performance has also been conducted. In [8], the authors consider the problem of choosing routes between a set of source

nodes and a set of sink nodes of an ad-hoc network so that the time until the first battery expires, is maximized. The authors note that choosing a route that results in minimum total energy expenditure is not always desirable because some of the nodes may have an excessive relaying burden, and hence these nodes may expire too soon. This in turn could lead to a loss of connectivity. To overcome this problem, the authors suggest that the routes should be chosen with the ultimate objective of maximizing the time until the first battery expires. In order to achieve this objective, the minimum energy paths are not necessarily the best choices. In [8], such an energy-efficient routing problem reduces to a linear programming problem which is described as the following:

$$\begin{aligned} max \;& \textit{Lifetime} \\ s.t.\;& \textit{1. Energy Constraint} \\ & \textit{2. Flow Conservation Constraint} \end{aligned} \tag{2.1}$$

where *Lifetime* is the network operational time till the first battery expires, *Energy Constraint* specifies that the energy expended by sensing, communication and other operations cannot surpass the initial energy reserves, and *Flow Conservation Constraint* specifies that the number of outgoing data flows of each node should be equal to the sum of the number of incoming data flows of that node plus the number of data flows originating at that node. Obviously, the data flows which maximize the *Lifetime* correspond to the optimal routing strategy.

However, the fact that the routing strategy is designed in such a way that all nodes die simultaneously (by attempting to postpone the death of the first node) does not automatically imply that the energy utilization is optimal [62]. In practice, the energy possessed by a normal sensor node is very easily exhausted and thus the node fails. For many sensor network applications such as military surveillance, full or guaranteed sensing coverage can still be provided in the case of sensor failures, by leveraging the redundant deployment of sensor nodes. Given that the network can still be useful even after some of sensor nodes have died, the metric attempting to postpone the death of the first node is unable to offer an optimal solution. The authors of [62] have exhibited similar thinking and define the network lifetime as the time taken for some fraction of nodes in the network to die, which is more practical than earlier definitions which use the time to the death of the first node as the network's lifetime. Unfortunately, as the network lifetime definition changed, the fundamental performance bound or the reference to the optimal solution also became unclear.

In [64, 65], the author of this dissertation uses a new concept called application-tolerable network run-time information-collecting ability in judging the lifetime of a network, which is more information oriented compared to the definition used in [62]. With this new network lifetime definition, nodes are allowed to die during the network's operational lifetime, which means that the network topology could change during the network operational lifetime and the data transmission between any two nodes could become unstable. All these make it difficult to give a linear programming optimization model similar to those proposed in [7, 8, 9]. Thus, a relaxed linear programming optimization model which can give a tight upper bound

is instead proposed in [64, 65]. In a similar manner to (2.1), the optimziation problem formulated in [64, 65] is described as the following:

$$
\begin{aligned}
max\ & \textit{Lifetime or Total Information Collected} \\
s.t.\ & \textit{1. Energy Constraint} \\
& \textit{2. Flow Conservation Constraint} \\
& \textit{3. Application-Dependent Requirement on Network Information-Collecting Ability}
\end{aligned}
\tag{2.2}
$$

In the above, *Lifetime* is the network operational time till the network's information collecting ability falls below the application-dependent requirement. *Total Information Collected* is a performance metric which could be more suitable for information-collecting purpose WSNs. It represents the total information collected by the entire network throughout its lifetime. *Energy Constraint* and *Flow Conservation Constraint* have the same meaning as those in (2.1). *Application-Dependent Requirement on Network Information-Collecting Ability* specifies the worst network information collecting ability which can be tolerated by the application. It can be also viewed as a translation of the network lifetime definition. The details of this network optimization problem are also available in Chapter 4 of this dissertation. The results acquired offer insights for future routing design and can also be used as benchmarks in the evaluation of energy-efficient routing algorithms designed for WSNs.

2.3.2 Energy-Efficient MAC Design

Compared to routing protocols, medium access control (MAC) protocols provide more direct influence over the utilization of the transceiver which is the largest energy consumer in most sensor nodes. Traditionally, MAC protocols are designed to maximize packet throughput, minimize latency and provide fairness. However, the design of MAC protocols for WSNs focuses on minimizing energy consumption.

It has been identified that the idle mode energy expenditure may spend a considerable amount of energy in WSNs [66]. Because many WSN applications possess a low message rate characteristic, most energy will be wasted by *idle listening* when traditional MAC protocols are used for WSNs: since a node does not know when it will be the receiver of a message from one of its neighbors, it must maintain its radio in receive mode at all times. If nodes exchange short messages with their neighbors at an average rate of one per second and both the transmitting and the receiving of a short message take 5 milliseconds, then the radio will spend 99% of the time on idle listening [67].

There are several solutions addressing the problem of energy waste due to idle listening. In general, some kind of duty cycle is involved, which allows each node to sleep periodically. TDMA-based protocols are naturally energy preserving. However, allocating TDMA slots is a complex problem that requires coordination. Another way of energy saving is to use an extra radio, which operates on a different frequency to that of the radio used for communication [68]. However, this approach is not appropriate for most wireless sensor nodes currently in use where only a single

radio is available on each node. S-MAC [69] is a single-frequency contention-based protocol specially designed for WSNs. It divides the time into fairly large frames. Each frame consists of two parts: an active period and a sleeping period. During the sleeping period, a node turns off its radio in order to preserve energy. During the active period, a node communicates with its neighbors and sends any message queued during the sleeping period. In order to synchronize, the sensor nodes periodically transmit SYNC messages at the beginning of the active period. The SYNC messages allow the sensor nodes to learn of their neighbors' schedules so that they can wake up at the appropriate time. Each sensor node performs a simple contention avoidance algorithm based on a random backoff to limit the number of SYNC message collisions. The T-MAC [67] protocol extends S-MAC by using a timer to indicate the end of the active period instead of relying on a fixed duty cycle schedule. By adaptively ending the active period, T-MAC nodes may save energy by lowering the amount of time they spend on idle listening and also by adapting to changes in traffic conditions.

2.3.3 In-Network Processing

WSNs are capable of collecting an enormous amount of data over space and time. Often, raw data is transmitted from each sensor node to a central processing location. This may cause a significant drain on communication and energy resources. However, in many applications, the ultimate objective is not merely the collection of "raw" data, but rather an estimate of certain environmental parameters or functions of interest (e.g., source locations, spatial distributions) [70]. Distributed in-network processing, which eliminates the need to transmit raw data to a central point, may significantly reduce the communication and energy resources consumed.

There have been many existing in-network processing approaches many of which are combined with routing algorithms. If the ultimate objective is to compute the average or other quadratic cost functions of all the measurements, the estimate of the objective parameter can be passed and updated along a routing path which passes through all the nodes and visits each node just once [70]. Each node updates the estimate by adjusting the previous value to improve or reduce its local cost and then passes the update to the next node. In the case of a quadratic cost function, one pass through the network is sufficient to achieve the objective. In more general cases, several "cycles" through the network are required in order to obtain a solution. The LEACH protocol presented in [71] is an elegant solution to the data aggregation problem in which clusters are formed in a self-organized manner to fuse data before transmitting it to the base station or sink. In LEACH, a designated node in each cluster, called the clusterhead, is responsible for collecting and aggregating the data from sensors in its cluster and eventually transmitting the result to the base station or sink. In [72], the authors propose a new chain-based protocol called PEGASIS that minimizes the energy consumption at each sensor node. The key idea is that nodes organize to form a chain and each node take turns in being the leader for communication to the base station or sink. The data is collected by starting from each endpoint of the chain and aggregated along the path to the designated head node.

Unlike LEACH, PEGASIS uses a flat topology thereby eliminating the overhead of dynamic cluster formation.

2.3.4 Load Balancing

In WSNs, the dominating communication pattern is that a large number of sensor nodes deliver their sensed information to one or a few data sinks through multi-hop transmission [64, 65]. This kind of communication pattern causes a drastic imbalance to the traffic load distribution across the network in which the nodes close to a sink experience heavy traffic loads. Since communication is believed to dominate the energy consumption of a sensor node [71] and sensor nodes are usually provided with limited energy resources, the imbalanced traffic load distribution is very harmful and it could cause the nodes close to a sink to die at an earlier stage which thus renders the remainder of the network to be useless.

To counter or alleviate the harm resulting from an uneven traffic load distribution, many researchers have turned their attention to the problem of load balancing. The authors of [63, 73, 74] realize that the imbalanced traffic load distribution can cause one part of nodes to die earlier than the others, thus degrading the network performance. To counter the negative effect of the imbalanced traffic load distribution on network performance, new routing algorithms which resort to the measure of the remaining energy reserves and other kinds of path capacity measurements are proposed. The authors of [75] consider the load balancing problem of uniformly distributed traffic demands in a unit disk. By deliberately routing traffic along slightly longer paths instead of the shortest paths, the highly congested links are avoided and a particularly flat traffic load distribution is achieved. The authors of [76] address the problem of balancing the traffic load in multi-hop wireless networks with uniformly distributed point-to-point communication. They develop a routing algorithm called Curveball Routing which can avoid the crowded center and provide a performance which is not significantly worse than that of the optimum. The authors of [77] propose an algorithm that makes a decision at each step as to whether to propagate data one-hop towards the sink, or to send data directly to the sink in order to balance the energy consumption over the nodes. If appropriate, data aggregation and in-network processing techniques are also methods for balancing traffic distribution. The adoption of data aggregation not only reduces the total amount of packets being transmitted but also yields a more even traffic distribution.

2.3.5 Resource Allocation

Fair resource allocation is another approach to counter the harm resulting from uneven traffic load distribution as explained in Section 2.3.4.

In the category of fair resource allocation, resources (e.g. energy, bandwidth, nodes) are allocated to an object (e.g. a node or an application) according to the workload of that object. The authors of [78] present an optimal energy allocation criterion and thus all clusters have the same exhausion time in a cluster based WSN.

The author of this dissertation has discovered that the performance upper bounds in a WSN linearly increase with the energy reservation in an identified bottleneck zone; thus assigning more available energy resources to the important bottleneck zone can effectively alleviate the bolttleneck effect [79]. Radio range adjustment is also proposed to save the energy consumption on a routing path [78, 80, 81]. However, this must be conducted with caution since assigning shorter relaying ranges for nodes closer to the sink adds more imbalance to the already imbalanced traffic load distribution. The authors of [82] propose efficient node placement and topology control protocols to balance the power consumption of sensor nodes. More specifically, they propose the allocation of more sensor nodes to the zone closer to the sink and also to assign a smaller packet transmission power to them. The author of this dissertation proposes a fair energy allocation scheme such that the initial energy resource allocated to a node is proportional to its expected traffic load [58]. Because traffic load is an indicator of the energy consuming rate, the proposed fair energy allocation scheme maximizes the network lifetime by equalizing the expected lifetime of each individual sensor node.

2.4 Anomaly Detection for WSNs

2.4.1 The Necessity of Anomaly Detection in WSNs

WSNs consist of a large number of tiny sensor devices that have limited power and limited sensing, computation, and wireless communication capabilities. Sensor nodes usually operate in unattended and even harsh environments, and as a result, sensor nodes are prone to failures and are vulnerable to malicious attacks. Since it is not possible to avoid the appearance of failures and malicious attacks, it will be essential that these failures and malicious attacks are detected immediately after their appearance. Thus, emergency responses can be made accordingly in order to alleviate the harm caused by the failures and malicious attacks.

One common point between failures and malicious attacks is that they both cause errors inside the system. Therefore, the system can malfunction due to the errors caused. The difference is that failures cause errors randomly, but malicious attacks are usually done deliberately and will preferentially target the most important component in the system. In addition, failures can exist everywhere in the system and can happen at anytime, but the scope of malicious attacks is subject to the abilities of attackers. In terms of available techniques, there are similarities between the detection of failures and malicious attacks. Because errors caused by failures and malicious attacks are abnormal events in the system, it should be possible to detect such an event by realizing that there has been a deviation of a system's state to that considered normal. The technique of detecting a system's abnormal events or behaviors by comparing a system's run-time profile to its normal profile is called *anomaly detection*. Of course, the technique of anomaly detection can also be used to detect anomalies other than failures and malicious attacks. For example, the target's behavior change in a target-tracking system can also be detected by anomaly detection.

Although anomaly detection usually suffers from a high false alarm rate in traditional systems, the anomaly detection in WSNs is expected to perform well because the operations of WSNs are less dynamic in comparison to those in traditional counterpart systems like the Internet.

2.4.2 Packet Traffic in WSNs Serves as the Data Source of Anomaly Detection

Packet traffic has been the most used data source in the anomaly detection for WSNs. The authors of [83] propose that an anomaly in WSNs could violate one of the following rules applied to packet traffic: 1) Interval rule: A failure is raised if the time which passes between the reception of two consecutive messages is larger or smaller than the allowed limits. 2) Retransmission rule: The monitor listens to a message, pertaining to one of its neighbors as its next hop, and expects that this node will forward the received message, which does not happen. 3) Integrity rule: The message payload must be the same along the path from its origin to a destination, considering that in the retransmission process there is no data aggregation by other sensor nodes. 4) Delay rule: The retransmission of a message by a monitor's neighbor must occur before a defined timeout. 5) Repetition rule: The same message can be retransmitted by the same neighbor only a limited number of times. 6) Radio transmission range: All messages listened to by the monitor must have originated (previous hop) from one of its neighbors. 7) Jamming rule: The number of collisions associated with a message sent by the monitor must be lower than the expected number in the network. By regularly monitoring the violations of the listed rules, network anomalies will be detected.

Ref.[49] proposes a specification-based anomaly detection to detect malicious attacks on AODV routing. In this approach, the authors use an FSM to specify correct AODV routing behavior and use distributed network monitors to detect run-time violation of the AODV specifications. The rationale behind this is that the AODV protocol has specified the sequence relations among different kinds of routing messages, and such sequence relations can be depicted by an FSM. Any violation of the protocol specification will trigger an alert. Ref.[51] also proposes a specification-based anomaly detection to detect routing attacks. In their approach, they use an FSM to specify the DSR routing behavior, instead of the AODV routing behavior.

In addition to the ability to specify the sequence relations among some special kinds of packets (e.g. routing messages) according to protocol specifications, the author of this dissertation suggests that the sequence relations among general kinds of packets can also be learned automatically by on-line training, thus anomalies can be detected by comparing run-time traffic patterns with learned historical traffic patterns [53].

Ref.[84] uses another traffic feature instead of packet sequence relations. It records the arrival time of each observed packet and checks the mean and the standard deviation of the interarrival times of the packets in a long term receive buffer and a short term intrusion buffer. An arrival is considered anomalous if the statistics in these

two buffers deviate significantly.

Ref.[85, 86] introduce a new data mining method that uses "cross-feature analysis" to capture the inter-feature correlation patterns in normal packet traffic, thus it can make decisions based on multiple traffic features. These patterns can be used as normal profiles to detect deviations (or anomalies) caused by malicious attacks and network failures. More specifically, this approach computes a classifier C_i for each feature f_i using $\{f_1, f_2, ..., f_{i-1}, f_{i+1}, ..., f_L\}$, where $\{f_1, f_2, ..., f_L\}$ is the feature set. C_i can be learned from a set of training data. It predicts the most likely value of f_i based on the values of other features. Based on a set of rules presented, this approach can identify the attack type of several well-known attacks. In some cases the rules can also identify the attacking or misbehaving nodes.

2.4.3 Evaluating Anomaly Detection Strategies for WSNs

The two commonly used measurements for evaluating the performance of an anomaly detection strategy are the false positive rate (FP) and the false negative rate (FN). FP is defined as the proportion of normal events that are erroneously classified as abnormal. FN is defined as the proportion of abnormal events that are erroneously classified as normal. Obviously, a good anomaly detection strategy should have both a low FP and a low FN. However, a tradeoff is usually to be made between FP and FN, given that these two measurements are usually influenced in opposing ways, by adjusting the threshold parameters used in many anomaly detection strategies. In addition to FP and FN, the overhead introduced by an anomaly detection strategy is also a concern. Considering the extreme resource-constrained specialties of WSNs, a good anomaly detection strategy should introduce as little overhead as possible. Although WSNs are designed for low rate communication, a broad range of real-time applications, such as health care, highway traffic coordination and even multimedia transmission have also been proposed. When an anomaly detection strategy is designed for such real-time applications, it should also fulfill a real-time requirement such that it will not cause performance degradation to the underlying application(s).

2.5 Chapter Summary

WSNs have been identified as one of the most important technologies for the 21st century. This chapter provides information concerning both its history and the current state of the art with regard to this important technology. The author provides a summary of the current works involved in traffic analysis & modeling, network optimization and anomaly detection for WSNs.

As WSNs are still a young research field, much activity is still on-going in order to solve many open issues. For example, traffic dynamics in WSNs are application dependent. For many WSN application scenarios, the traffic dynamics are still very obscure. Network optimization continues to be the prime important research area for WSNs given the constraint of the very limited resources which are unable to be

removed in the near future. As more and more WSNs become available for practical deployment, the problems relating to sensor failures and mailicious attacks will attract more and more attention. Anomaly detection, which is a promising technique for the immediate detection of any network anomaly (e.g. sensor failure, malicious attack), has as yet been touched upon only rarely.

In the following chapters, the author will present his research results and contributions to the areas of traffic analysis & modeling, network optimization and anomaly detection for WSNs. Through the background introduction of this chapter, the readers can already see that many of the works involved in network optimization and anomaly detection are based on the research results from traffic analysis & modeling. In Chapter 4, network traffic and its associated energy consumption has been considered to play a key role in all the author's presented works relating to network optimization for WSNs. In Chapter 5, the author uses two examples to demonstrate the feasibility and the goodness of detecting sensor network anomalies through the analysis of network traffic.

Chapter 3

Traffic Analysis & Modeling on Selected WSN Scenarios

In ordinary WSNs, there are, in the main, two kinds of nodes: the low-power sensor nodes and the powerful sink nodes. In some tiered sensor networks, it is also possible for there to be a third type, namely the cluster heads. The cluster heads usually have abilities which are between those of the ordinary sensor nodes and those of the sink nodes, and they are used to relay and aggregate the data packets received from the ordinary sensor nodes. In this dissertation, we mainly consider the flat sensor network scenario when there is no cluster head. Unless it is specified, the network scenario discussed below is flat sensor network.

3.1 The Dominating Traffic Pattern

The mission of WSNs is to collect data from their deployed environments and report them to the base station, which is represented by one or a few sink nodes. Thus, the traffic flows from the distributed sensor nodes to the base station (many-to-one) and this will dominate the communication in a properly designed WSN. In addition, the base station may need to proactively collect data by sending queries to the distributed sensor nodes, and it may also be necessary to conduct some global configuration on the distributed sensor nodes. In this case, the traffic flows from the base station to the distributed sensor nodes (one-to-many) also exist. Depending on the specific network protocols used and the concrete application, there could also be some other types of traffic which may include the routing traffic, the link layer "hello message" and the application-specific traffic.

3.2 Packet Sequence Modeling

Sequence relations exist between some types of packets. For example, a Routing Reply message always comes after a Routing Request message which is specified by an ordinary routing protocol. In ref. [49], the authors propose to use a finite state machine (FSM) to specify the correct AODV [50] routing behavior. The authors in ref. [51] also use an FSM to model the correct routing behavior for another routing protocol DSR [52].

In addition to that the sequence relations among some special kinds of packets (e.g., routing messages) can be specified according to protocol specifications, the sequence relations among general packets can also be learned automatically by online training. In the following, the automatic learning of the general packet relations will be discussed in greater detail.

3.2.1 Packet Classification

To learn the sequence relations among general packets, those general packets must firstly be classified properly. Otherwise, either the class set has an unmanageable size or the learned sequence relations have no practical use. We propose to classify the packets in such a way that the whole set of packet categories can be mapped to a set of single byte ASCII characters and the sequence relations learned based on the classified packets can reflect the unique behavior of the node of interest.

Actually, protocol specifications have specified the arrival order of different packet types. For example, the AODV routing protocol has specified that a Routing Reply can only follow a Routing Request. Thus, the feature *Packet Type* can act as a natural base to classify different packets and there is the promise that patterns learned based on this have ability to detect the violation of protocol specifications.

In wireless sensor networks, a node does not need to talk with all the other nodes in order to finish its assigned task. For example, it is only necessary for the sensed data to be reported to the sink, or a cluster head if there is one, in a sensor network for surveillance. HELLO messages which are used to maintain link states are only sent to the neighborhood. Thus, a sensor node only participates in communication with a limited number of the other nodes, and *{Src, Dest}* (the abbreviation for the pair of source and destination addresses) pairs of a node's observed packets reflects this node's traffic profile.

In this case we are going to classify packets according to the different combinations of *Packet Type* and *{Src, Dest}* pairs. In order to control the number of the packet categories and make the packet classification scheme scalable, we further map the general address space to an abstracted address space. The abstracted address space has only five entries: {me; neighbor; local; unlocal; sink/cluster head}, which are classified from the point of view of the node of interest. In concrete terms, "me" is the node of interest, "neighbor" represents all those nodes within one hop distance of the node of interest, and "local" represents all those nodes that are already known by the node of interest through learning of the source and destination nodes

of all its previous observed packets. During the packet sequence learning, no node is classified as "unlocal". Once a stable set of all learned packet sequence relationships is acquired, the observation of a packet with its source or destination node to be "unlocal" is usually a signal of anomaly.

3.2.2 Packet Translation

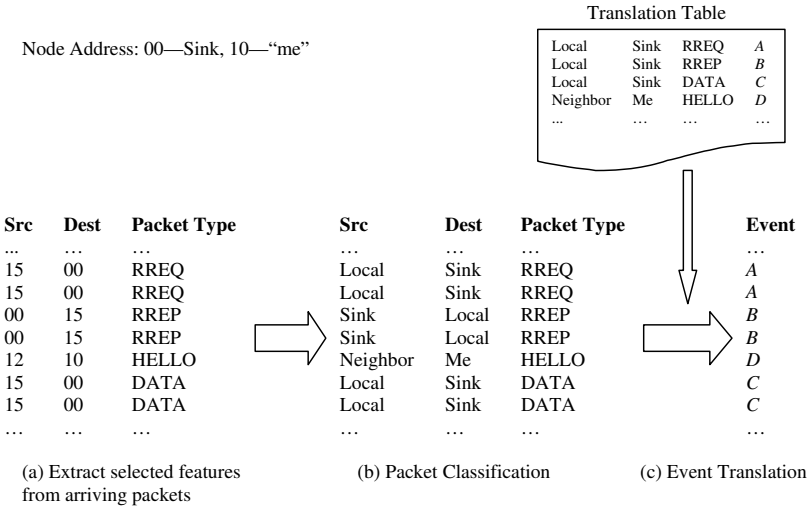


Figure 3.1: Translation of packet arriving events to characters

For simplicity, the classified packets can be further mapped to a set of single byte ASCII characters. Fig. 3.1 shows the process of packet classification and the process of mapping the classified packets to a character set. Finally, the sequence of packets arriving at a node of interest can be viewed as a large (or asymptotically infinite) string of characters.

3.2.3 Pattern Extraction

To learn the sequence relationships among the arriving packets, we must extract patterns from the large (or asymptotically infinite) string of characters. The pattern extraction algorithm which was firstly proposed by Forrest et al. in [87, 88] for the intrusion detection in a Unix system is used in this case. During the pattern extraction, the arriving sequence of the abstracted packet events (i.e. the character string) is scanned for all given length, k, unique subsequences. Simultaneously, a database of all such unique subsequences which have been found, i.e. patterns, is built. Once a stable database has been constructed, the process of pattern extraction has been completed.

The construction of the pattern database is best illustrated with an example. For

$k = 4$ and the sample sequence AABBDCC, we obtain the following pattern table: AABB, ABBD, BBDC, BDCC.

If one pattern is encountered more than once, it shows up only once in the pattern database.

Because WSNs are usually deployed in a way "once deployed, never changed", the actions of the sensor nodes are either pre-scheduled inside chips or triggered to respond to outside events in the predefined way. The communication traffic in WSNs evolves quite slowly and it is possible to acquire a stable pattern database after observing the packet arriving sequence for a sufficiently long time. Once a stable pattern database has been acquired, it can be used as the traffic profile of the observing node.

3.2.4 Summary of Packet Sequence Modeling

This section presents a method of learning sequence relations among packets arriving at an observing sensor node. Firstly, selected traffic related features are used to translate the observed packets into different events. Following this, unique patterns based on the arriving order of different packet events are extracted. As long as there is a sufficiently long observation time, the probability of observing new patterns is minimized. Thus, a stable set of the learned patterns can be acquried and used to build the traffic profile for the observing sensor node finally.

3.3 Modeling The Bursty Traffic Arrival Process in Event-Driven WSNs

Event-driven data collection and processing architecture has been used by many proposed WSN scenarios. In an event-driven WSN, bursty traffic can arise from any corner of the sensing area if an event is detected by the local sensors. A Poisson process has been used to model the traffic arrival process in an event-driven WSN [46]. However, there are no solid grounds to support the use of a Poisson process. Actually, the widely used Poisson processes are quite limited in their burstiness [5, 6]. Instead of using Poisson processes, we propose the use of an ON/OFF model to capture the burst phenomenon in the source traffic of an event-driven WSN. Concretely, we consider a special event-driven WSN scenario: a sensor network deployed for target tracking purposes. Based on simulation experiments, we analyze the appropriateness of using the ON/OFF model. Further, the properties of the ON/OFF period distributions are extracted.

3.3.1 WSNs for Target Tracking

A typical WSN for target tracking consists of spatially distributed sensor nodes monitoring a mobile target collaboratively. When a target enters into the surveillance

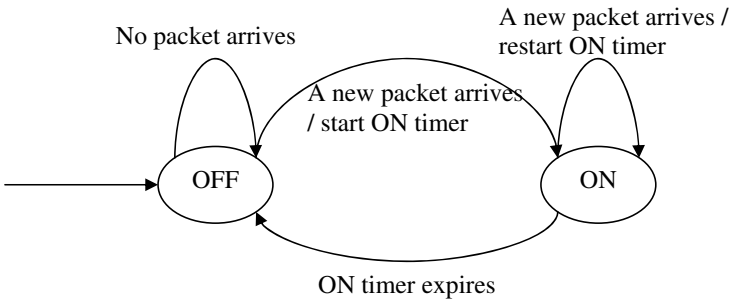


Figure 3.2: ON/OFF state transition diagram

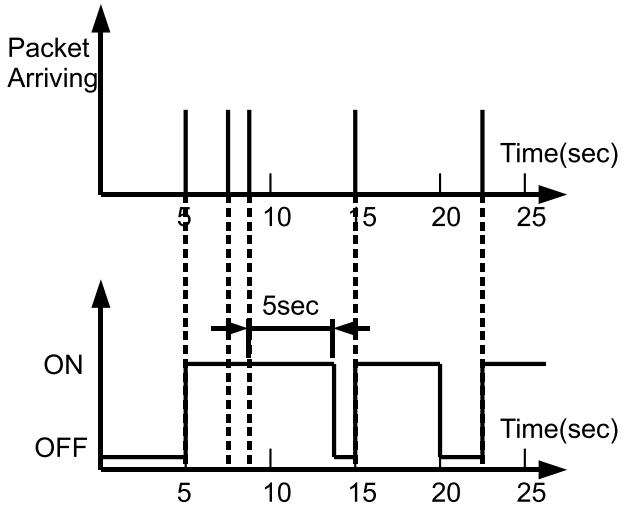


Figure 3.3: ON/OFF state transitions with an ON timer of 5 seconds

area, any sensor node with sensing ability will discover this target if the target is within its sensing range. As long as the target remains in the sensing range, the alarmed sensor node will keep reporting its observation about the target to a base station by means of a multi-hop routing. Thus, this kind of event-driven working manner will generate bursty source traffic at the sensor nodes.

3.3.2 ON/OFF Model

We use the ON/OFF model [89] to capture the burst phenomenon of source traffic. In the ON/OFF model, each ON interval corresponds to a time span when the target appears within the sensing range and each OFF interval represents a time span when the target is out of the sensing range. Fig. 3.2 shows the state transition diagram of the ON/OFF model, and Fig. 3.3 shows an example regarding on how we can build on the ON/OFF traffic model. The ON/OFF traffic model includes two states: ON

and OFF. The default state is OFF when the model starts training. Each time a source packet is observed, the state is turned ON if it is currently OFF, and an ON timer is started. The state is kept ON until the ON timer expires. If another source packet arrives before the ON timer expires, the ON timer is restarted. If no source packet arrives during the ON timer window, the state is turned OFF. The total amount of time when the state remains ON is designated as the ON period. Similarly, the time duration when the state remains OFF is designated as the OFF period. Each ON period indicates an event during which the neighborhood of the considered sensor node is visited by the target, while the length of an ON period states the duration of a visit.

3.3.3 Experimental Results

We made a simulation to explore the dynamics of source traffic in a target tracking scenario. The simulation was conducted using our modified ns-2 simulator [11], which was a modified ns-2 version based on a standard ns-2.27 release. 100 sensor nodes were deployed in a 1000m by 1000m grid surveillance area. A base station was placed at the center of the surveillance area. A mobile target was simulated by a phenomenon node [90], which simulated the sensing process by regularly emanating phenomenon packets through a special sensing channel. The radio related ns-2 parameters were set as $RXThresh\_ = 3.65262e-10, Pt\_ = 0.281838$, and the radio propagation model used was two-ray ground. With these parameter values, the signal transmission range for both the target node and the sensor nodes (i.e. both sensing range and communication range) were about 250m. The target node was placed at a random location in the field at the beginning of the simulation. Then it started a random movement. The random movement model used was the default model used by a mobile node in ns-2.27. The basic concept for this mobility model is: for every $POSITION\_UPDATE\_INTERVAL$, the target node randomly selects a (new) destination location in the field and moves there at a speed randomly distributed between $\langle MIN\_SPEED, MAX\_SPEED \rangle$. In our simulation, $POSITION\_UPDATE\_INTERVAL$ was set to be 5 seconds, MIN_SPEED was 0, and MAX_SPEED was 100m/s. When the target node was moving, every sensor node that was within distance of the sensing range detected the target and reported to the base station by means of a multi-hop routing. Since the sensing frequency is usually high (the frequency we used was 10 sensing operations per second), an event reporting rate upper limit 1 packet/s was imposed on each sensor node when the target remained in the sensing range[1] to mitigate the intensity of source traffic burst.

The simulation was run for 50,000 seconds. For each arriving source packet (i.e. target event report), the base station recorded its timestamp and source address. Packets with the same source address were placed into the same source traffic group. The source traffic modeling provided below was based on an individual source traffic group.

[1]The target was considered to be remaining within the sensing range if it was repeatedly sensed within a given time interval, which was 0.2 seconds in our simulation.

Table 3.1: The count of ON periods in the case of different ON timers

ON timer's length	*node-edge*	*node-center*
1.5s	159	1998
5s	157	1485
20s	152	322

For ease of narration, two representative nodes are selected to present the results. The two selected nodes are *node-edge* at the position $\langle 950m, 50m\rangle$ and *node-center* at the position $\langle 450m, 550m\rangle$. Due to the target mobility model used in the simulation, the two selected nodes have experienced a significant difference in their probabilities of observing the target. In this experiment, the target observation probability is 0.03 for *node-edge* and 0.70 for *node-center*. In addition, the length of the ON timer may have a significant influence on the results presented. Given the maximum event reporting rate is 1 packet/s in this experiment, a reasonable ON timer length would fall within the range from several seconds to dozens of seconds. To investigate the influence of the ON timer's length on the traffic modeling results, three different ON timer lengths: 1.5s, 5s, 20s, have been explored in our modeling.

Count of ON Periods

In the ON/OFF model, each ON period roughly represents a traffic burst and thus a target visit at the source node. The count of ON periods provides information relating to the number of times the target has visited the considered source node. For *node-edge* and *node-center*, the count of the ON periods in the case of different ON timers is shown in Table 3.1.

Obviously, Table 3.1 shows that the observation of the number of target visits at a source node can be affected by the choice of ON timer. This is because a long ON timer could merge neighboring traffic bursts into one observed ON period. In the scenario given in this dissertation, a node experiencing more frequent target visits, such as *node-center*, is more sensitive to the choice of ON timer.

Since every ON period is followed by an OFF period, the count of OFF periods displays the same statistics as that shown in Table 3.1.

ON Period Distribution

The ON period distribution provides a general idea regarding how long a target visit would last.

Fig. 3.4 shows the statistical distributions of ON periods at the place of *node-edge*. It can be seen that the ON period distributions observed by different ON timers appear to be similar. If the ON period distribution observed by the length l_1 ON timer is shifted to the right along the x-axis by the distance $|l_2 - l_1|$, then the ON period distribution observed by the length l_2 ON timer is almost acquired. This

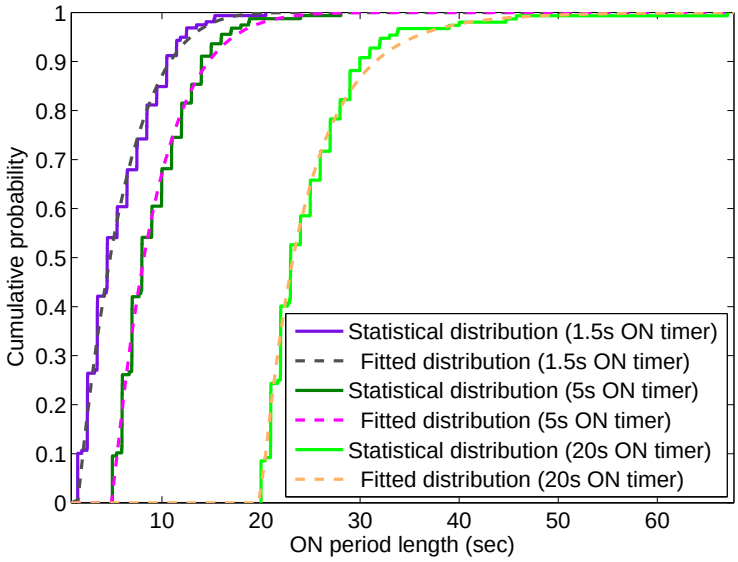


Figure 3.4: CDF plot of ON period distribution for *node-edge*

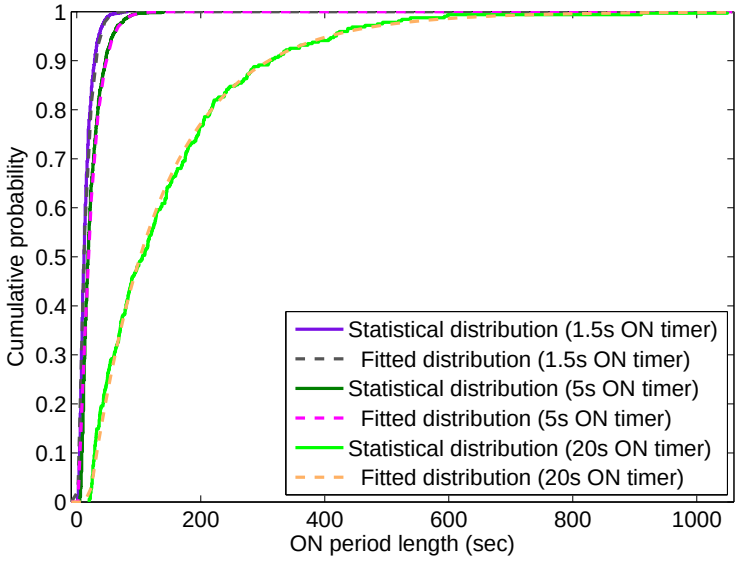


Figure 3.5: CDF plot of ON period distribution for *node-center*

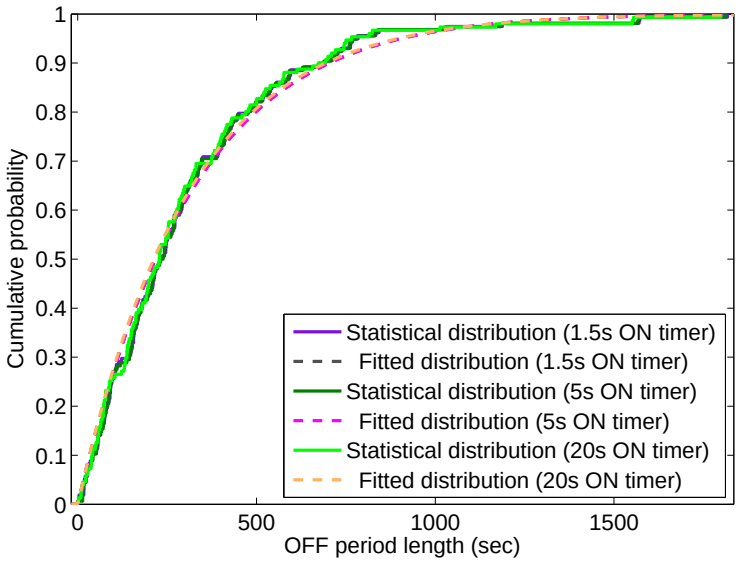


Figure 3.6: CDF plot of OFF period distribution for *node-edge*

shows: For *node-edge*, the only difference involved in using a different length ON timer is simply artificially adding the observed ON periods with a different time length (in the case where the length of ON timer is 5 seconds, the artificially added length on each observed ON period is also 5 seconds), and there is no significant influence on the shape of the observed ON period distribution.

Fig. 3.5 shows the statistical distributions of ON periods for *node-center* in the case of different ON timers. It can be seen for *node-center* that the shape of the ON period distribution is quite dependent on the choice of ON timer. This is because: For *node-center*, an ON period observed with a long ON timer is quite possibly the union of several neighboring ON periods observed with a short ON timer, as shown in Table 3.1. Thus, an average observed length of an ON period in the case of a long ON timer is more than the sum of that in the case of a short ON timer and the ON timer's length difference. Intuitively, the length of an ON period observed with a long ON timer is the addition of the lengths of several ON and OFF periods observed with a short ON timer, plus the artificially added ON timers' length difference.

OFF Period Distribution

The OFF period distribution provides a general idea about how long the silence between two neighboring target visits would last.

Fig. 3.6 shows the statistical distributions of OFF periods at the place of *node-edge*. The OFF period distributions observed by different ON timers are indistinguishable from each other. This is because the extremely long OFF periods (far longer than a reasonable length ON timer) experienced by *node-edge* are not sensitive to the choice

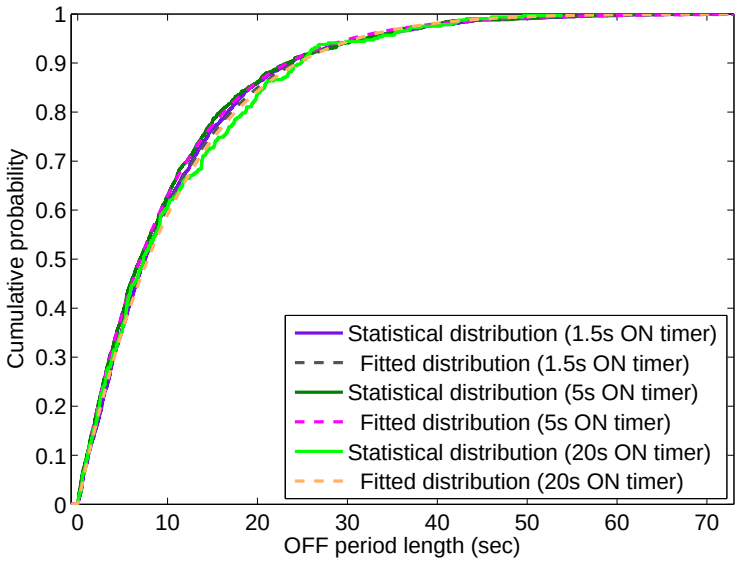


Figure 3.7: CDF plot of OFF period distribution for *node-center*

of ON timer.

Fig. 3.7 shows the statistical distributions of OFF periods at the place of *node-center*. Despite the high sensitivity of the observed count of ON/OFF periods for *node-center* to the choice of ON timer as that shown in Section 3.3.3, the OFF period distribution observed is found to be insensitive to the choice of ON timer in Fig. 3.7. The relatively stable OFF period distribution observed is a good indication with regards to the potential use of this distribution, however the reason behind this is still unclear at the time of writing this dissertation.

3.3.4 ON/OFF Distribution Fitting

To investigate the mathematical essence behind the simulation results, distributions have been fitted to the acquired statistical data. The fitted distributions are also shown in Fig. 3.4-3.7.

We find that the generalized Pareto distribution is the most appropriate distribution type to fit ON and OFF periods. The generalized Pareto distribution allows a continuous range of possible shapes that includes both the exponential and Pareto distributions as special cases. The probability density function for the generalized Pareto distribution with shape parameter $k \neq 0$, scale parameter σ, and threshold parameter θ, is

$$y = f(x|k,\sigma,\theta) = \left(\frac{1}{\sigma}\right)\left(1 + k\frac{(x-\theta)}{\sigma}\right)^{-1-\frac{1}{k}}$$

for $\theta < x$, when $k > 0$, or for $\theta < x < -\frac{\sigma}{k}$ when $k < 0$. In the limit for $k = 0$, the

Table 3.2: Parameters used when fitting the generalized Pareto distribution to statistical ON period distributions

	node-edge			*node-center*		
	1.5s ON timer	5s ON timer	20s ON timer	1.5s ON timer	5s ON timer	20s ON timer
k	-0.200	-0.109	0.080	-0.148	-0.060	0.074
σ	5.054	4.768	4.570	14.802	19.654	115.23
θ	1.49s	4.99s	19.99s	1.49s	4.99s	19.99s

Table 3.3: Parameters used when fitting the generalized Pareto distribution to statistical OFF period distributions

	node-edge			*node-center*		
	1.5s ON timer	5s ON timer	20s ON timer	1.5s ON timer	5s ON timer	20s ON timer
k	-0.033	-0.034	-0.022	-0.030	0.009	-0.073
σ	317.08	317.91	308.89	10.861	10.026	11.559
θ	0.001s	0.001s	0.001s	0.001s	0.001s	0.001s

density is

$$y = f(x|0, \sigma, \theta) = \left(\frac{1}{\sigma}\right) e^{-\frac{(x-\theta)}{\sigma}}$$

for $\theta < x$. If $k = 0$ and $\theta = 0$, the generalized Pareto distribution is equivalent to the exponential distribution. If $k > 0$ and $\theta = \sigma$, the generalized Pareto distribution is equivalent to the Pareto distribution.

Figs. 3.4-3.7 show that the generalized Pareto distribution can fit all statistical distributions very well. The parameter values used in the fitted distributions are shown in Table 3.2 and Table 3.3. More analysis relating to the fitted distributions will be presented in Chapter 5.

Packet Arriving within an ON Period

The packet arriving within an ON period does not follow a Poisson distribution. However, it is found that the distribution of the traffic rate within an ON period roughly follows a truncated normal distribution, with the maximum rate being the rate upper limit imposed and the minimum rate being the reciprocal of the ON timer's length.

3.3.5 Summary of modeling the bursty source traffic in event-driven WSNs

Realistic traffic models for WSNs have not been sufficiently developed. In this case the ON/OFF model is used to model the source traffic characteristics in a target tracking WSN scenario. For the source traffic generated by each single sensor node, the traffic bursts are captured by ON periods, and the silence intervals between traffic bursts are captured by OFF periods. The source traffic arrival process is thus viewed as the interchanging of ON and OFF periods. The experiments show that the ON/OFF model is suitable for the modeling of the source traffic in the event-driven WSNs, while a Poisson process does not prove to be satisfactory. In addition, both the ON and OFF period distributions are found to follow the generalized Pareto distribution very well in our experiment, and this observation exists independently of the target observation probability and is also resilient to the length of the ON timer.

Based on the ON/OFF source traffic model and the ON/OFF period distributions presented in this dissertation, a source traffic generator can also be designed. To generate a trace of source traffic, it is firstly possible to generate a sequence of ON and OFF periods according to the generalized Pareto distribution. Then the packet arriving within ON periods could be simulated according to a truncated normal distribution.

3.4 Traffic Load Distribution in Dense Sensor Networks

Usually, traffic load is not evenly distributed over the nodes in a wireless sensor network (WSN). Understanding the traffic load distribution can guide the network-wide energy allocation, direct the design of routing algorithms, and optimize the node deployment in WSNs. We consider a dense WSN with nodes evenly deployed in a disk area, and determine the traffic load distribution over the nodes as a function of their distance from the sink. Further, the effects of network scale and routing strategy on traffic load are also investigated. The traffic loads on individual nodes are found to be in direct proportion to the radius of the network and in inverse proportion to the routing hop length, while independent of network density. The results presented here are verified through simulation experiments.

3.4.1 Network Scenario

We consider a dense WSN of n sensor nodes evenly deployed in a disk area of radius R. There is a sink node located at the center of the deployed disk area. All the deployed sensor nodes regularly sense the physical phenomenon around and forward that information to the sink directly or by multi-hop transmission.

In a dense sensor network, a routing path is along the line segment connecting the source node and the destination node. In addition, the number of routing hops traversed can be assumed to be linearly proportional to the distance between the

source and the destination, i.e. the routing hop length varies around a mean value along the routing path. This assumption is realistic in the sense that it is consistent with many well-accepted routing algorithms and protocols. For example, the shortest path routing algorithm looks for a routing path where all hops have a length close to the maximum communication range. The closed formula for the mean hop length between two arbitrary nodes has been given in [91] where the shortest path routing algorithm is used in a one dimensional network. In the same paper, the closed formula for the mean hop length between two arbitrary nodes is also given for the case in which a simple greedy progress-based routing algorithm is used in a two dimensional network. In both scenarios, the mean hop length is found to be independent of the Euclidian distance between the source and the desitination, which means that the mean hop distance (in terms of the number of hops) linearly increases with the distance between the source and the destiantion. Actually, many research works [92, 93] have pointed out, that in order to minimize the path cost, the optimal routing policy should choose relay nodes on an equidistant basis along the line that connects the source and the destination and that there exists an optimal hop length which is decided by the link cost model. With regards to the popular reactive routing protocols such as AODV [50], it has been found in [94] that except for the first hop, the average number of hops grows linearly as the distance between the source and the destination increases in one demensional MANETs.

In this section, only those source-initiated routing algorithms which exhibit a mean hop length in their building routing paths when the nodes are uniformly deployed in an Euclidian space and the value of the mean hop length is independent of the source or destination are considered. Given that a dense network is considered in this case, the deviation of an ordinary hop length to the mean hop length can be ignored. Thus, the value of a hop length as well as the mean hop length is denoted h in the following narration.

In this evenly distributed network, all sensor nodes have the same average sensing rate (i.e. the same amount of packets originate during a unit time period), and this average sensing rate is represented by $\bar{o}$. Furthermore, investigations will only be performed on the "stable-state" traffic load distribution when no node fails because of the energy constraint.

3.4.2 Traffic Load Analysis

We define the traffic load at a given node as the total amount of packets handled on that node during a unit time period. Obviously, the total amount of packets handled at a node includes the amount of packets which originate at that node and the amount of packets relayed by that node. For any node, $\bar{o}$ should be the expected amount of packets which originate at that node during a unit time period, and this conclusion is straight-forward. However, to determine the amount of packets relayed by a given node, the number of child nodes the given node has must firstly be determined. As for the definition of child node, we say node b is a child node of node a or node a is a parent node of node b if node b needs node a to act as a relay when forwarding its packets to the sink.

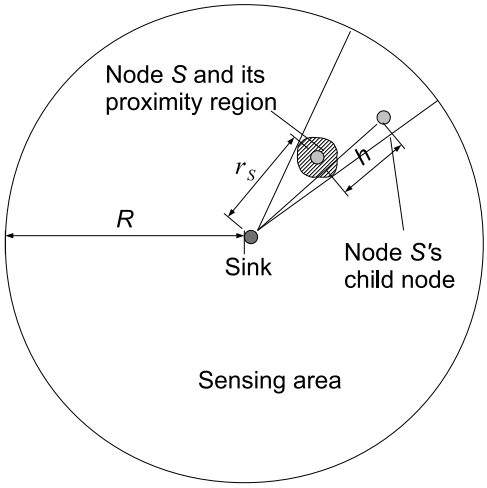


Figure 3.8: Sketch map for a sensor network deployed in a disk area. The definition of a node's child node is shown.

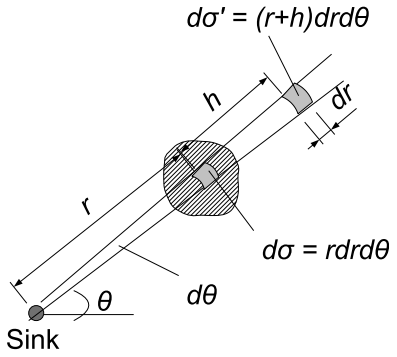


Figure 3.9: An infinitesimal region in a node's proximity region.

Consider node S shown in Fig. 3.8. Node S has a distance of r_S from the sink. Because the network is dense, node S must lie along the line segments connecting its immediate child nodes and the sink. Further, an immediate child node's distance to the sink must be one hop farther than that of node S. We say a node lies along a line segment if the proximity region of this node is traversed by the line sgement. Additionally it is said that a point is within a node's proximity region if this node is the closest deployed node to the point.

Let S_S be the proximity region of node S in Fig. 3.8. For each infinitesimal $d\sigma$ in S_S which is defined by $d\sigma = r\,dr\,d\theta$ in polar coordinates (see Fig. 3.9), all the points which lie on the straight line connecting a point in $d\sigma$ and the sink and also one hop farther away from the sink, together form a region $d\sigma'$ which is defined by $d\sigma' = (r+h)\,dr\,d\theta$. Because the summation of $d\sigma$ forms S_S and the expected relay locations of node S's immediate child nodes reside in S_S, the summation of $d\sigma'$ forms a region

which is the complete set of all the potential locations for node S's immediate child nodes. Using the denseness assumption, we can approximate the size of the region located by node S's immediate child nodes as $\sum d\sigma' = \iint_{S_S}(r+h)\,dr\,d\theta \approx (1 + h/r_S)\iint_{S_S} r\,dr\,d\theta = (1+h/r_S)\sum d\sigma$. In the considered evenly distributed network, every node is expected to occupy the same size of proximity region. Because there is only one node located in the region S_S of size $\sum d\sigma$, we can estimate that the expected number of node S's immmediate child nodes is $1 + h/r_S$.

With the knowledge of the expected number of node S's immediate child nodes, the expected number of node S's i-th hop child nodes can be inferred by using iteration. The expected number of node S's i-th hop child nodes is $1 + i\,h/r_S$. Suppose the m-th hop child nodes are the farthest child nodes of node S, the expected total number of child nodes for node S would be $\sum_{i=1}^{m}(1 + i\,h/r_S)$. Because each child node and node S itself periodically generate packets at the average rate $\bar{o}$ packets per unit time, the expected traffic load on node S during a unit time period can be expressed as:

$$Traffic(r_S) = \sum_{i=0}^{m}(1 + i\,\frac{h}{r_S})\bar{o} \tag{3.1}$$

Note that node S is only a representation of those nodes deployed in the sensing area. We can generalize the traffic load expression for any node which has a distance r from the sink, and that is:

$$Traffic(r) = (m+1)(1 + \frac{m\,h}{2r})\bar{o} \tag{3.2}$$

Given that the deployed area has a finite radius R, we have $m = \lfloor\frac{R-r}{h}\rfloor$ in the above formula.

It can be seen that the number of deployed nodes n is not present in (3.2). This is contradictory to the intuitive notion that increasing the number of deployed nodes may simultaneously aggravate the expected traffic load situation at each single node. Actually, the only factor that influences the expected traffic load is a node's distance from the sink, when the routing and the network radius are fixed. However this is only an approximate conclusion drawn for load-balanced networks. The heaviest experienced traffic load in a non-load-balanced network will still increase as the number of deployed nodes is increased.

It would also be interesting to determine the influences of the network radius R and the routing hop length h when these two parameters are not fixed. These influences are not clearly expressed by (3.2). In the following, an approximated traffic load expression is derived, from which the relations between the network radius R, the routing hop length h and the traffic load, can be clearly observed.

Let $m = \frac{R-r}{h} - \epsilon,\ \epsilon \in [0,\ 1)$ in (3.2). We have

$$\begin{aligned} Traffic(r) &= (\frac{R-r}{h} - \epsilon + 1)(1 + \frac{R-r-\epsilon\, h}{2r})\bar{o} \\ &= \frac{R-r+(1-\epsilon)h}{h}\frac{R+r-\epsilon\, h}{2r}\bar{o} \\ &= \frac{(R^2-r^2)+(R+r-2R\epsilon)h-\epsilon(1-\epsilon)h^2}{2r\, h}\bar{o} \end{aligned}$$

Let $r' = r/R$ as a node's unified distance from the sink. We further have

$$Traffic(r') = \frac{(1-r'^2)+(1+r'-2\epsilon)\frac{h}{R}-\epsilon(1-\epsilon)(\frac{h}{R})^2}{2r'\frac{h}{R}}\bar{o}$$

Using the knowledge that $\frac{h}{R}$ is usually a small number, the following can be obtained as an approximation:

$$Traffic(r') \approx \frac{1-r'^2}{2r'}\frac{R}{h}\bar{o} \tag{3.3}$$

(3.3) clearly states that the traffic loads experienced by individual sensor nodes are in direct proportion to the radius R of the network, and in inverse proportion to the routing hop length h. For networks where the network radius and the routing hop length are fixed, the traffic load only varies with the node's distance from the sink. It can even be seen from (3.3) that the traffic load is a monotonically decreasing convex function for $r' \in (0,\ 1]$. This means that the traffic load has a blowout when $r' \to 0$, which is consistent with the conventional acquaintance relating to the bottleneck issue in WSNs [79].

3.4.3 Simulation Results

A dense sensor network was firstly simulated in which 20,000 sensor nodes are deployed in a grid within a disk area of radius 100 meters. Thus, the average distance between sensor nodes is around 1.8 meters. The shortest path routing algorithm has been used in our simulation, and the maximum communication range is set to be 10 meters. In addition, all sensor nodes are configured with the same packet generation rate which is 1 packet per time unit. Because of the grid topology used, the nodes can experience quite different traffic loads even if they have similar distances from the sink. To counter the effect of the traffic asymmetry in the radial direction, the average traffic loads experienced by all nodes located within the annulus region of inner radius $r - \triangle r/2$ and outer radius $r + \triangle r/2$ are taken as the estimate of the traffic load for nodes which are r away from the sink. Here, $\triangle r$ is the edge length of the grid and $\triangle r = \sqrt{\frac{\pi R^2}{n}}$. Fig. 3.10 shows the result of the estimated traffic load as a function of r with the theoretical result (calculated from (3.2) with $h = 10m$) shown as a comparison. From Fig. 3.10, it can be seen that the experienced traffic load oscillate around the theoretical traffic load, and the experimental traffic load converges

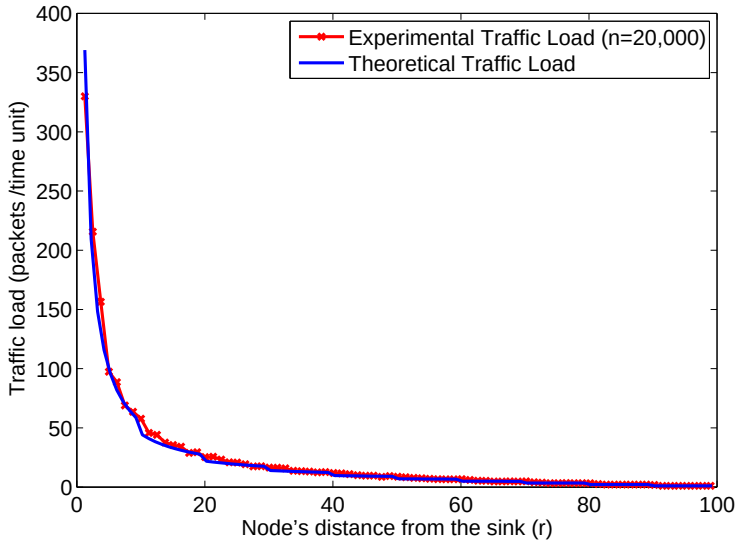


Figure 3.10: Traffic load distribution in a network where $n = 20,000$ nodes are gridly deployed in a disk area of radius $R = 100m$ and the routing hop length is $h = 10m$.

to the theoretical value when $r \to R$. The oscillation observed for small r is due to the accumulation of the asymmetric traffic flow along its path to the sink.

Although the theoretical result about the traffic load distribution in section 3.4.2 is achieved for dense sensor networks, it may be interesting to check how different this is when the network is of low density. Thus, a new experiment is made in which the traffic load distribution in low density networks is simulated. In concrete terms, there are only 250 sensor nodes which are evenly deployed along the grid points within a disk area of radius 50 meters. The shortest path routing algorithm is also used in this simulation. The maximum communication range for all the nodes is set to be 15 meters, and the packet generation rate is still 1 packet per time unit. Fig. 3.11 shows the results in which the average has been constructed to be similar to that for the dense network experiment. It can be seen that the theoretical traffic load distribution (calculated from (3.2) with $h = 15m$) even coincides with the experimental traffic load distribution when the network is of small size and low density. This enhances the practical significance of the presented theoretical traffic load distribution.

Another experiment is also conducted to show that the traffic load expressed by (3.3) is a good approximation to that expressed by (3.2), thus all the corollaries based on (3.3) are reasonable. This experiment is based on calculation. The parameters $R = 100m$ and $h = 10m$ in (3.2) and (3.3) are used. The result is shown in Fig. 3.12.

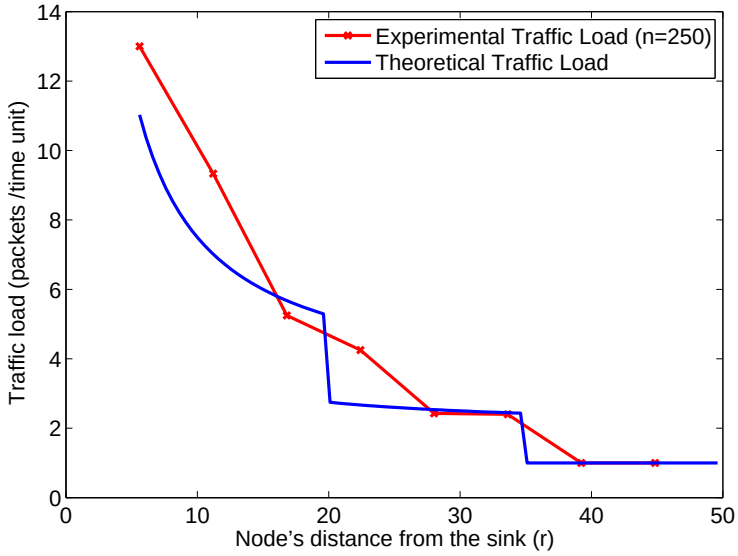


Figure 3.11: Traffic load distribution in a network where $n = 250$ nodes are gridly deployed in a disk area of radius $R = 50m$ and the routing hop length is $h = 15m$.

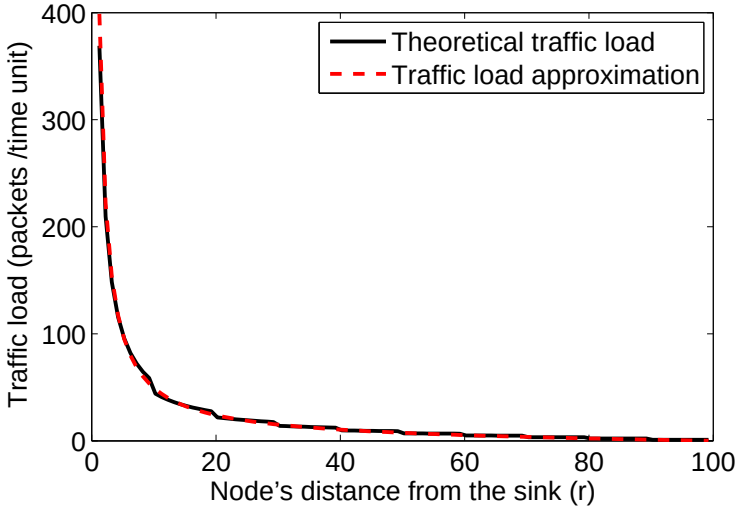


Figure 3.12: Comparison between the precise theoretical traffic load and the approximated traffic load ($R = 100m$, $h = 10m$).

3.4.4 Summary of The Traffic Load Distribution in Dense Sensor Networks

The special communication pattern in WSNs leads to the conclusion that the traffic accumulates on its routing path to the sink. In this section, the traffic load distribu-

tion over the deployed nodes was determined as a function of their distance from the sink in the evenly distributed dense sensor networks. It was shown that the traffic load over a node increases as the node becomes closer to the sink, and a traffic load blowout is expected in the proximity of the sink. In a simulated dense sensor network, the theoretical traffic load distribution presented had a near-optimum fit to the experimental traffic load distribution for almost all deployed nodes except those which are too close to the sink. Additionally, the theoretical traffic load distribution presented offered a good fit to the experiment relating to that for a sensor network of low node density. This enhances the practical significance of the presented theoretical traffic load distribution.

3.5 Chapter Summary

Currently, only limited knowledge exists regarding the detailed traffic characteristics in WSNs. The knowledge concerning the detailed traffic characteristics can assist in the understanding of the network, its devices, services and vulnerabilities. Thus, different enhancement methods will be able to be developed in order to optimize WSNs. However, the traffic characteristics of WSNs can vary greatly in different application scenarios. This causes the work of extracting the traffic characteristics in WSNs to be very troublesome.

In this chapter, the author has proposed new methods of extracting traffic characteristics in WSNs. The author believes that there are patterns in the packet arriving sequences at the place of distributed sensor nodes, and proposes an automatic method of extracting packet sequence patterns from the runtime traffic. The author also analyzes the feasibility of using an ON/OFF Model to model the traffic characteristics in a target tracking WSN scenairo, which is an example of the more generalized event-driven WSNs. For very dense sensor networks, the author formulizes the traffic load distribution over the nodes which are evenly deployed in a planar disk area. The simulation experiments even show that the formulized traffic load expression is also a good approximation for non-dense WSNs.

Chapter 4

Optimizing the Design and the Operation of Energy-Constrained WSNs

Given that the energy constraint for an ordinary WSN cannot be eliminated in the near future, optimizing the design and the operation of energy-constrained WSNs so the best performance will be achieved is vital. Because communication dominates the energy consumption, the knowledge concerning the traffic characteristics is essential for the optimization of WSNs. In Chapter 3, the traffic characteristics in selected WSN scenarios have been researched. In this chapter, the study of traffic characteristics for energy-constrained WSNs is continued. However, the focus of the study will be turned to the application of the learned traffic knowledge on network optimization.

In the following, the maximizing network performance problem is formulated as a relaxed linear programming problem, in which the traffic flow constraint and the energy consumption, mainly dominated by communication, are considered. The special many-to-one multi-hop communication pattern actually forms a bottleneck zone in a WSN, a performance upper bound determined by this bottleneck zone is also analyzed. Based on the results of traffic load distribution in dense WSNs presented in Chapter 3, an efficient optimal energy allocation scheme called RIFES is proposed for such networks. RIFES maximizes the network lifetime when all sensor nodes are contributing.

4.1 Optimization Models for Maximizing the Information Extraction & the Network Lifetime

This dissertation makes progress in the understanding of the optimal network performance. Metrics used to measure the network performance in this case are the network lifetime finally acquired and the total information finally collected. The condition used to judge the network's death is defined by the user's requirement on the guaranteed network information collecting ability. Optimization models based on the performance metrics and death condition mentioned above are proposed.

4.1.1 Problem Setting

A data-centric network, deployed for information-collecting purposes, typically consists of one or more sinks (coordinators), and a large number of different functional sensors (end devices). There may or may not be any dedicated routers in the network. If there is no dedicated router, it is usual for some sensors to also be responsible for packets forwarding [18, 95].

Consider such a network deployed in a sensing field for periodic information-collecting purposes. All sensor nodes have predefined sensing rates and will periodically report their sensed data to the closest sink. There are a negligible number of messages originating from a sink. Each sensor node is constrained by the limitation of the available battery power and, in addition, has power control to expend the minimum required energy to reach the intended recipients and to be turned off to avoid receiving unintended transmissions. The transmission range of each node is limited by a distance D. If the distance r_{ij} between node i and j is less than D, the transmitting power e_{ij}^t from node i to j can be adjusted to the minimum $e_{ij}^t = e^T + \zeta_{amp} \cdot r_{ij}^\alpha$ [7, 96, 97], where e^T is the energy/packet consumed by the transmitter electronics, ζ_{amp} is the energy dissipated in the transmit op-amp, and parameter α means there is an r_{ij}^α energy loss due to channel transmission. We assume that the receiving power is a constant e^R in the experiments given in Section 4.1.4. Sink nodes are assumed to have unlimited power to receive and further process data collected from the entire network.

For each sensor node i, there is an information weight σ_i, where σ_i reflects the importance of the data sensed by node i, and this weight value is decided according to the position of node i in the sensed field. Generally, if the interested phenomenon can be observed clearly by node i, then data originating from node i would be assigned a high importance. However, exceptions do exist. If node i has many neighbors or a high sensing frequency, then the sensed data is less important because of data correlation. The total information contributed by node i is measured by the multiplication of its weight σ_i to the total data packets sensed during its lifetime. The total information extracted from the entire network is the sum of the information contributed by all sensor nodes.

In sensor network literature, several different definitions have been proposed

concerning the "lifetime" of a sensor network. Ref. [7, 97] defines "lifetime" as the passage of time to the point when the first sensor "dies". Ref. [96] considers the lifetime as the passage of time until all the sensors die. As nodes continue to die, the total amount of information delivered from the network to the sinks is constantly reduced. The definition of "lifetime" should obviously depend on the nature of the application [62]. For instance, for applications such as surveillance, it may be crucial that all sensors be alive in order to guarantee full coverage. The death of only one sensor may terminate the "useful" lifetime of the whole network, while for other cases this may not be true. Usually, when the network ability in information collecting falls below a given threshold, it may not be worthwhile for the network to continue to operate. Here, we define the information-collecting ability for a single sensor node i as the multiplication of its information weight σ_i to its sensing rate o_i. Additionally, the information-collecting ability for the entire network is the sum of each currently live node's ability in information collecting. A sensor is thought to be dead if its energy is exhausted or there is no available path for connection to any sink. The information-collecting ability of a dead sensor cannot be counted in the total network information-collecting ability.

A normalized network information-collecting ability is the ratio of the residual network information-collecting ability to the initial network information-collecting ability at the network's start-up. In the next subsection, a parameter ∂ with the value belonging to $[0, 1]$ is used to represent the tolerable run-time network information-collecting ability of an application. When the normalized network information-collecting ability falls beneath this value, it is considered not to be worthwhile continuing to operate the network. Obviously, this network lifetime definition is more information oriented compared to that in [62], where the fraction of live (or dead) nodes instead of the residual information-collecting ability is used to judge the network's death.

We have now defined the "information" and, in addition, the network's "lifetime". The important concerns for the user involve the network lifetime and the total information delivered by the network throughout its lifetime. In the following section, an optimization model will be presented in order to maximize network information extraction, and also to maximize network lifetime as a comparison, from which the theoretical optimal network performance can be extracted. For ease of reading the following analysis, all the related notations which have been presented and are to be presented are listed in Table 4.1.

4.1.2 Non-Linear Optimization Model

A wireless sensor network generally consists of two kinds of elements: nodes and links. All nodes are classified as either sensor nodes or sink nodes. We use S_{sensor} and S_{sink} to separately represent the sets of these two different nodes. A link exists if and only if the physical distance between two nodes is less than the maximum radio transmission range. The network works in such a way that all sensor nodes sense the physical field in the predefined sensing rates and then report their data to the closest sink. Packets that arrive at any sink are considered to have been received by users.

Table 4.1: Notation definition

Notation	Definition
S_{sensor}	The set of sensor nodes
S_{sink}	The set of sink nodes
$T_{(sys)}$	Network lifetime
T_i	The runtime till the energy reserve of node i is exhausted or node i is disconnected from all sinks
Tw_i	Useful working time, $Tw_i = min(T_i, T_{(sys)})$
E_i^0	Initial energy reserve of node i
l_{ij}	Directional link state from i to j (1 for connectable, and 0 for non-connectable)
f_{ij}	Directional flow rate (packets per second) from i to j
e_{ij}^t	Transmitting power at node i to send a packet to node j
e_{ij}^r	Receiving power at node j to receive a packet from node i
β	Energy consumed by each packet sensing operation
σ_i	Information weight of node i
o_i	Pre-assigned sensing rate at node i
o_i'	Sensing rate at node i after adjustment
O_i	Total packets sampled by node i till the end of network operation
F_{ij}	Total packets transmitted from i to j till the end of network operation
∂	Guaranteed normalized network information-collecting ability

A pure or dedicated router is a sensor node with a zero sensing rate. Each packet is assumed to have the same packet length. Depending on the routing protocols, sensed data can have a packet head added to it for the transmission. In that case, the packet in transmission would have a longer length than that initially sensed.

σ_i is used to represent the information carried by each packet originating from sensor node i, and o_i is the predefined sensing rate of node i, T_i is the lifetime of node i, and $T_{(sys)}$ is the lifetime of the whole network. A sensor node is thought to be dead when its energy is depleted or it loses connection with all sinks because of network partition. When a sensor node dies, the link states are updated. The whole network is considered dead when it cannot fulfill the information-collecting ability requirement of the application.

Each sensor node continuously contributes sensed information until its death or until the whole network dies. The useful working time for sensor node i can be denoted as $Tw_i = min(T_i, T_{(sys)})$. The total information contributed by node i during its useful working time would be $\sigma_i \cdot o_i \cdot Tw_i$. In addition, the total amount of information extracted during the entire network's lifetime would be the sum of the information contributed by each single sensor node. If the goal of network optimization is to maximize the total information extracted throughout the network lifetime, the objective function of the optimization model would be:

$$Maximize \sum_{i \in S_{sensor}} \sigma_i \cdot o_i \cdot Tw_i \tag{4.1}$$

The assumption is that neither data aggregation nor packet drop exist in the network. In this case, the data packets received by any sensor node plus those originating from it should be equal to the data packets forwarded by it. This is called flow conservation. Given the constant sensing rate, the total packets sensed by node i during its lifetime is $o_i \cdot T_i$. Use $f_{ij}(t)$ to represent the flow rate from i to j at time t, and $l_{ij}(t)$ to represent the link state (1 for connectable, and 0 for non-connectable) of the directional link $(i \rightarrow j)$ at time t. Then the total number of packets received from node j during node i's lifetime is $\int_0^{T_i} f_{ji}(t) \cdot l_{ji}(t) \cdot dt$. Additionally, the total number of packets sent to node j during node i's lifetime is $\int_0^{T_i} f_{ij}(t) \cdot l_{ij}(t) \cdot dt$. Given each sensor node can receive and send packets, whereas a sink only receives packets, the total number of packets received from and sent to the entire network during node i's lifetime would separately be $\sum_{j \in S_{sensor}} \int_0^{T_i} f_{ji}(t) \cdot l_{ji}(t) \cdot dt$ and $\sum_{j \in S_{sensor} \cup S_{sink}} \int_0^{T_i} f_{ij}(t) \cdot l_{ij}(t) \cdot dt$. Thus the flow conservation constraint can be written as:

$$o_i \cdot T_i + \sum_{j \in S_{sensor}} \int_0^{T_i} f_{ji}(t) \cdot l_{ji}(t) \cdot dt = \sum_{j \in S_{sensor} \cup S_{sink}} \int_0^{T_i} f_{ij}(t) \cdot l_{ij}(t) \cdot dt, \ \forall i \in S_{sensor} \tag{4.2}$$

In the case of the sinks, they are considered to have sufficient energy to run throughout the lifetime of the network. However, for each sensor node i, there is a limited initial energy storage E_i^0. This limited energy storage is usually spent on three parts: data sensing, data receiving and data transmitting.

If symbol β is used to represent the energy consumed by each packet sensing, then the energy spent on data sensing during node i's lifetime can be written as $\beta \cdot o_i \cdot T_i$. If e_{ji}^r is used as the energy required for node i to receive one packet from node j, then the energy spent on data receiving at node i should be the sum $\sum_{j \in S_{sensor}} e_{ji}^r \cdot \int_0^{T_i} f_{ji}(t) \cdot l_{ji}(t) \cdot dt$. If e_{ij}^t is used as the energy required for node i to transmit one packet to node j, then the energy spent on data transmitting at node i should be the sum $\sum_{j \in S_{sensor} \cup S_{sink}} e_{ij}^t \cdot \int_0^{T_i} f_{ij}(t) \cdot l_{ij}(t) \cdot dt$. Data sensing, receiving and transmitting would all compete for the limited initial energy storage at each sensor node. The energy constraint is:

$$\beta \cdot o_i \cdot T_i + \sum_{j \in S_{sensor}} e_{ji}^r \cdot \int_0^{T_i} f_{ji}(t) \cdot l_{ji}(t) \cdot dt + \sum_{j \in S_{sensor} \cup S_{sink}} e_{ij}^t \cdot \int_0^{T_i} f_{ij}(t) \cdot l_{ij}(t) \cdot dt \leq E_i^0, \forall i \in S_{sensor} \tag{4.3}$$

It has been mentioned in Section 4.1.1 that users may still accept degraded network information-collecting ability depending on the application. However, this does not mean that an information-collecting ability which is arbitrarily poor is acceptable. A definition concerning normalized network information-collecting ability has been given in Section 4.1.1. Here this concept is further used to describe the guaranteed network information-collecting ability required by the user.

According to the definition given in Section 4.1.1, the normalized residual net-

work information-collecting ability at time t can be written as:

$$\frac{\sum_{i \in S_{sensor}} \sigma_i \cdot o_i \cdot H(T_i, t)}{\sum_{i \in S_{sensor}} \sigma_i \cdot o_i},$$

where $H(T_i, t) = \begin{cases} 0, & t \geq T_i \\ 1, & t < T_i \end{cases}$, $\forall i \in S_{sensor}$. It is simple to prove that the normalized network information-collecting ability always degrades with time.

To describe the user requirement, a parameter $\partial \in [0, 1]$ is used to represent the worst normalized network information-collecting ability that can be tolerated by the application. The system should definitely fulfill the user requirement during its runtime:

$$\frac{\sum_{i \in S_{sensor}} \sigma_i \cdot o_i \cdot H(T_i, t)}{\sum_{i \in S_{sensor}} \sigma_i \cdot o_i} \geq \partial \tag{4.4}$$

The above formula also limits the longest possible network lifetime and can be viewed as the death condition for the system.

Finally, the objective function (4.1), together with constraints (4.2) (4.3) (4.4), give a non-linear mathematical depiction of the maximizing information extraction problem.

Unfortunately, the link states $l(t)$ and flow rates $f(t)$ stated above change with time, and their values change when the topology changes. As each sensor node fails, the topology also changes. This is an extremely difficult problem to solve.

4.1.3 A Relaxed Linear Optimization Model

In the real situation, topology will change each time a sensor node fails. This has implications for the link states and routing flow rates which are topology dependent, and causes the modeling problem to be extremely difficult. If it proves possible to delay the death of each sensor node until the network dies, while still retaining the same amount of total information contribution and the same network operational time, the modeling problem is simplified and benefits from a stable topology.

In the following, it is assumed that the sensing rate of each sensor node can be adjusted to an appropriate value, so that no sensor node will fail before the whole network ceases to function and that it is possible to achieve one stable topology throughout the whole network lifetime.[1] To retain an equal amount of information contribution for each sensor node, the adjusted sensing rate should fulfill the relation: $o_i \cdot Tw_i = o'_i \cdot Tw'_i$, $\forall i \in S_{sensor}$, where o_i and Tw_i are separately the original sensing rate and the original useful working time for node i, and o'_i and Tw'_i are separately the sensing rate and the useful working time for node i after adjustment. Since

[1] Note that this assumption here is given to simplify the mathematical calculation, without considering its physical feasibility.

each node is assumed not to fail before the entire network ceases to function, the useful working time for each sensor node exactly equals the network lifetime (which is unchanged after the sensing rate adjustment) and thus $Tw'_i = T_{(sys)}$, $\forall i \in S_{sensor}$.

If $o_i \cdot Tw_i$ is replaced by $o'_i \cdot T_{(sys)}$ in (4.1), then the new objective function is:

$$Maximize \sum_{i \in S_{sensor}} \sigma_i \cdot o'_i \cdot T_{(sys)} \tag{4.5}$$

For those nodes which originally could fail before the failure of the entire network, their working times are prolonged to the network lifetime. In order to still retain the same information contribution with the prolonged working time, the sensing rate for each sensor must be decreased and becomes

$$o'_i \leq o_i, \ \forall i \in S_{sensor} \tag{4.6}$$

As the topology has now become stable, the flow rates and link states will no longer change with time. Thus, the new flow conservation constraint is obtained as:

$$o'_i + \sum_{j \in S_{sensor}} f_{ji} \cdot l_{ji} = \sum_{j \in S_{sensor} \cup S_{sink}} f_{ij} \cdot l_{ij}, \ \forall i \in S_{sensor} \tag{4.7}$$

Similarly, the energy constraint in (4.3) can be rewritten as:

$$(\beta \cdot o'_i + \sum_{j \in S_{sensor}} e^r_{ji} \cdot f_{ji} \cdot l_{ji} + \sum_{j \in S_{sensor} \cup S_{sink}} e^t_{ij} \cdot f_{ij} \cdot l_{ij}) \cdot T_{(sys)} \leq E^0_i, \ \forall i \in S_{sensor} \tag{4.8}$$

The number $\partial \in [0, 1]$ in constraint (4.4) guarantees the normalized network information-collecting ability during the network operational time. After the sensing rates have been adjusted, the network information-collecting ability should still be guaranteed. Thus

$$\frac{\sum_{i \in S_{sensor}} \sigma_i \cdot o'_i}{\sum_{i \in S_{sensor}} \sigma_i \cdot o_i} \geq \partial \tag{4.9}$$

With the new objective function (4.5) and constraints (4.6) (4.7) (4.8) (4.9), the relaxed optimization model is obtained. Further, $O_i = o'_i \cdot T_{(sys)}$ and $F_{ij} = f_{ij} \cdot T_{(sys)}$ are set and used to substitute o'_i and f_{ij} in (4.5) (4.6) (4.7) (4.8) (4.9). The optimization model can then be depicted using the following mixed integer linear programming problem, with σ_i, o_i, β, e^r_{ji}, e^t_{ij}, l_{ij} and E^0_i acting as inputs, O_i, F_{ij} acting as independent integer variables and $T_{(sys)}$ acting as an independent real variable.

$$Maximize \sum_{i \in S_{sensor}} \sigma_i \cdot O_i$$

$$s.t. \begin{cases} O_i \leq o_i \cdot T_{(sys)}, \ \forall i \in S_{sensor} \\ O_i + \sum_{j \in S_{sensor}} F_{ji} \cdot l_{ji} = \sum_{j \in S_{sensor} \cup S_{sink}} F_{ij} \cdot l_{ij}, \ \forall i \in S_{sensor} \\ \beta \cdot O_i + \sum_{j \in S_{sensor}} e^r_{ji} \cdot F_{ji} \cdot l_{ji} + \sum_{j \in S_{sensor} \cup S_{sink}} e^t_{ij} \cdot F_{ij} \cdot l_{ij} \leq E^0_i, \ \forall i \in S_{sensor} \\ \frac{\sum_{i \in S_{sensor}} \sigma_i \cdot O_i}{\sum_{i \in S_{sensor}} \sigma_i \cdot o_i \cdot T_{(sys)}} \geq \partial \\ O_i \in I^+, F_{ij} \in I^+, T_{(sys)} \in R^+, \ \forall i \in S_{sensor}, \forall j \in S_{sensor} \cup S_{sink} \end{cases}$$

An optimization model whose goal is to maximize network lifetime can be easily acquired by substituting the above objective function with the new objective function: $Maximize \ T_{(sys)}$.

4.1.4 Experiments

Experiments were performed in order to validate the optimization model presented. In the experiments, radio parameters (see Section 4.1.1 for definitions) were set using $D = 25m$, $e^T = 8\mu J/packet$, $e^R = 24\mu J/packet$, $\zeta_{amp} = 0.016\mu J/packet/m^4$ and $\alpha = 4$. In addition, a sensing power $\beta = 8\mu J/packet$ was assumed. Each node was assigned with an initial energy 0.1J. For all sources (i.e. those nodes with sensing responsibility), a constant sensing rate 1 $packet/s$ was assumed. Additionally all sensed packets were assumed to carry unit weight information.

Consideration is given to the appearance of the model-generated optimal performance when the number of sources changes and when the requirement for the guaranteed normalized network information-collecting ability ∂ changes. We consider a network with 100 sensor nodes randomly deployed in a 100m by 100m square field and with one sink located at the center of the field. In calculating the optimal performances, the linear programming solver LPSolve 5.5 is called from MATLAB. The average results over 100 simulated instances for maximizing information extraction and for maximizing network lifetime are separately shown in Fig. 4.1 and Fig. 4.2. It can be seen that for any given number of sources, the optimal network performance is a monotonically decreasing function of the ∂ in both cases. Since a smaller ∂ means a degraded requirement for the application-tolerable network information-collecting ability, a direct result is a prolonged network lifetime and thus an increased amount of information collected. It can also be seen that for any given ∂, the optimal network performance increases with the number of sources when the optimization goal is to maximize information extraction, but decreases with the number of sources when the optimization goal is to maximize network lifetime. This is easy to understand since more sources generally generate more information. However, more sources also means that the limited network energy resources are required to be consumed more intensely, and thus this involves sacrifices in relation to the network lifetime.

Actually, the presented optimization model is shown to provide tight upper bounds for network performances. In [65], several existing routing algorithms are evaluated

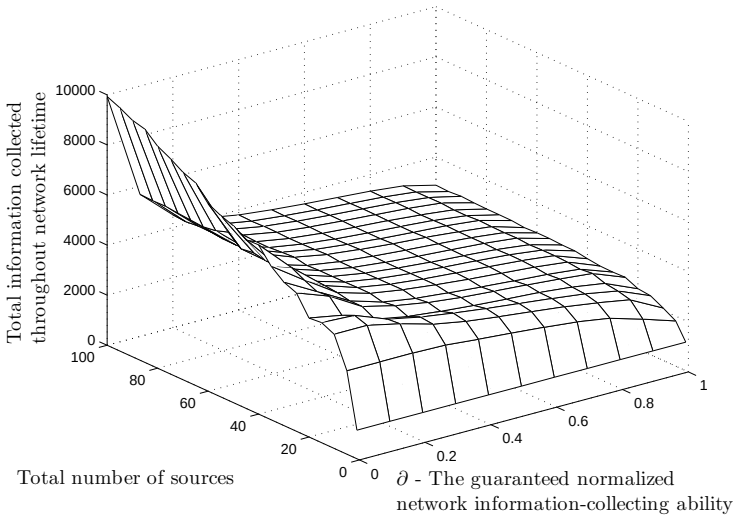


Figure 4.1: Optimal *Information collection* for different number of source nodes and different guarantees

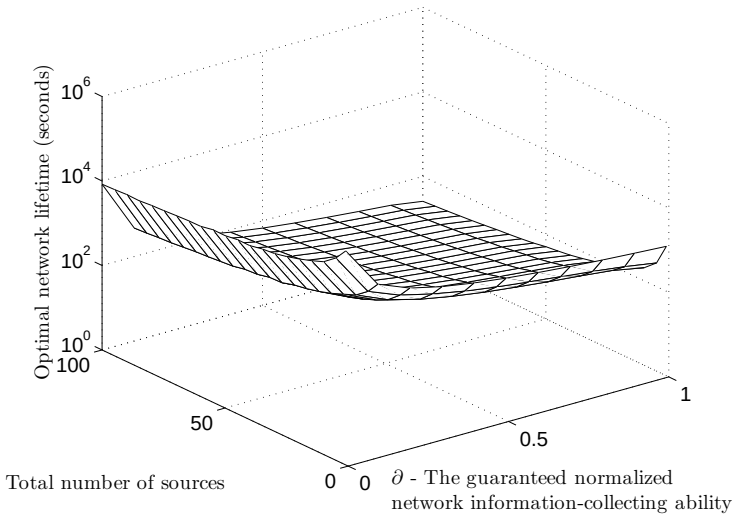


Figure 4.2: Optimal *network lifetime* for different number of source nodes and different guarantees

using the optimization model presented in this case. Some routing algorithms are found to yield nearly optimal performances, especially when there is a high requirement on the guaranteed information-collecting ability (i.e. when ∂ is close to 1).

4.1.5 Summary of Section 4.1

We argue that the lifetime of a wireless sensor network should be determined neither by the death of the first node nor by the death of the last node. It should be judged by the condition when the residual network information-collecting ability fails to satisfy the user requirement, which is usually dependent on the underlying application. For a general purpose wireless sensor network, maximizing the total information extracted throughout its entire network lifetime or simply maximizing network lifetime could be two natural performance metrics. Based on these considerations, a non-linear optimization model is firstly proposed. Due to the intractability of the proposed non-linear optimization model, a relaxed linear optimization model is given afterwards. Experiments show that the relaxed linear optimization model provides tight upper bounds for network performances and thus can be used to evaluate the existing and up-coming network protocols.

4.2 Acquiring Performance Upper Bounds Based on Bottleneck Zone Analysis

In a typical sensor network, nodes around the sink consume more energy than those which are further away. It is not unusual for limited energy resources which are available at the nodes around the sink to become the bottleneck which confines the performance of the whole network. In this section, we firstly present our considered bottleneck zone in a general sensor network scenario. Then, the effect of the bottleneck zone on network performance is investigated by deducing performance bounds imposed by the energy resources available inside the bottleneck zone. In this section, both the performance bound in terms of network lifetime and the performance bound in terms of information collection are explored. Finally, the ways by which network deployment variables may affect the performance bounds are analyzed.

4.2.1 Energy-Constrained Wireless Sensor Networks

Wireless sensor networks consist of a large number of sensor nodes which are supposed to be battery-supported. Because the energy provided by a battery is very limited and usually non-renewable, the energy constraint dominates the underlying network performance.

It has been found that the energy consumed in a sensor network mainly comes from the sensing operation and communication. For a sensing operation, the energy cost in sensing a data bit is usually a constant, and we use β to represent it. In relation to communication, the energy spent can be represented by the first order radio model [93]. In this model, the energy cost for transmitting one data bit across a distance of r without relay can be expressed as

$$E = \alpha_{11} + \alpha_2 r^{\alpha} + \alpha_{12} = \alpha_1 + \alpha_2 r^{\alpha} \tag{4.10}$$

where α_{11} is the energy/bit consumed by the transmitter electronics, α_{12} is the energy/bit consumed by the receiver electronics, α_2 is the energy dissipated in the transmit op-amp, and parameter α means that there is an r^{α} energy loss due to channel transmission. There is also a relation $\alpha_1 = \alpha_{11} + \alpha_{12}$. In reality, the maximum transmitting power for a given radio is limited, and that power corresponds to the maximum radio transmission range. In this case we assume that all sensor radios have the same maximum transmitting power and use D to represent the corresponding maximum radio transmission range.

Since the energy constraint is usually a significant problem for a wireless sensor network and the network often fails to function because of its exhausted energy, the operational lifetime of a wireless sensor network is widely used to measure the network performance. Given that an important goal of deploying a wireless sensor network is collecting interesting information from its environment, the total amount of collectable information after a network is deployed also serves as an important performance metric (see Section 4.1 in this chapter).

4.2.2 Bottleneck Zone in a Wireless Sensor Network

Multi-hop routing strategies are widely used in wireless sensor network scenarios. Benefiting from a multi-hop routing strategy, it is possible for a sensor node which is deployed far away from the sink to communicate with the sink through the relaying of those sensor nodes deployed in between. Because the dominating communication pattern in a sensor network involves dispersed sensor nodes forwarding their sensed information to the sink, multi-hop routing strategies inevitably cause the nodes closer to the sink to suffer more energy consumption than those further away. The consequence is that the nodes around the sink will fail earlier and this also causes the nodes further away from the sink to be disconnected from the sink. Actually, the nodes around the sink form a bottleneck zone which limits the network performance such as lifetime.

In this dissertation, the bottleneck zone in a sensor network is considered as the intersection area between the sensor deployment area and a round surface centered at the sink. The round surface has a radius of D, which represents the maximum radio transmission range used by the sensor nodes. A sketch map of the bottleneck zone is shown in Fig. 4.3.

Obviously, any message which has originated at a node outside of the bottleneck zone cannot reach the sink without utilizing the relay of the nodes inside the bottleneck zone.

4.2.3 Performance Upper Bounds Imposed by the Bottleneck Zone

In a sensor network deployment, the whole network relies on the nodes inside the bottleneck zone to relay messages. Thus, the functioning of nodes inside the bottleneck zone is essential, and it actually imposes upper bounds on network performances.

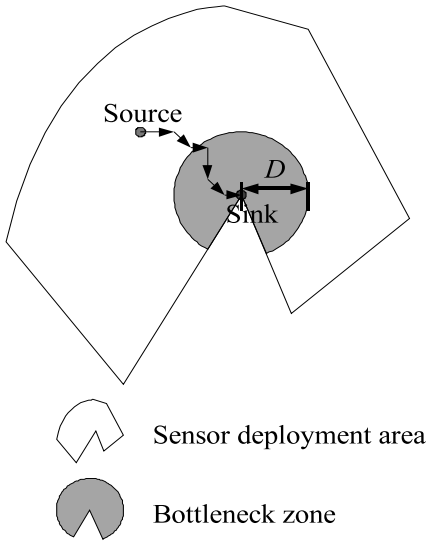


Figure 4.3: The Bottleneck Zone in a Sensor Deployment Area

Sensor Network Scenario

Without loss of generality, consider a wireless sensor network with n sensor nodes uniformly deployed in a full-connected area of size A. There is a data sink located in or close to the sensor deployment area. Each sensor node senses its physical environment according to its configured sensing rate, and all sensed data are directly forwarded to the sink without delay. At the start of a network operation, each sensor node has an initial energy reserve. In this section, all the nodes inside the bottleneck zone are assumed to have the same amount of initial energy reserve, which is represented by e^b. Further, the network is considered to follow the first order radio model as introduced in Section 4.2.1.

Energy Cost in the Bottleneck Zone for Relaying a One-Bit Message

Let d_m be the characteristic distance which is the optimal routing hop length in terms of the minimum energy consumption along the relay path from the sender to the receiver. According to ref. [93], $d_m = \sqrt[\alpha]{\frac{\alpha_1}{\alpha_2(\alpha-1)}}$ holds in the case in which the first order radio model is used.

Any sensor node outside of the bottleneck zone cannot directly forward its messages to the sink. Instead it must firstly forward its messages to at least one relay node located within the bottleneck zone. Let c be the total energy spent by the sensor nodes inside the bottleneck zone in order to relay a one-bit message from outside of the bottleneck zone to the sink. With the proper assumption that the length of a

normal routing hop is far less than the bottleneck zone radius D, we can show that

$$c \geq \frac{D(\alpha_1 + \alpha_2 d_m^{\alpha})}{d_m} = \alpha_1 \frac{\alpha}{\alpha - 1} \frac{D}{d_m} \tag{4.11}$$

Total Energy Cost Within the Bottleneck Zone till Time t

Use B_t to represent both the bottleneck zone and its size. Since sensor nodes are evenly deployed in the sensing area, the number of nodes outside of the bottleneck zone would be $n\frac{A-B_t}{A}$. Until time t (time units), the total number of message bits originating outside of the bottleneck zone and then being relayed through the bottleneck zone would be $n\frac{A-B_t}{A}\bar{o}t$, where $\bar{o}$ is the average sensing rate (bits/unit time) of sensor nodes.

Let $E_{out,t}$ be the total energy spent on relaying messages which have originated outside of the bottleneck zone by nodes inside the bottleneck zone until time t. We have

$$E_{out,t} \approx \sum_{i=1}^{\lfloor n\frac{A-B_t}{A}\bar{o}t \rceil} c_i \tag{4.12}$$

where c_i is the total energy spent inside the bottleneck zone in order to relay the i-th bit coming from the outside.

Incorporating (4.11) into (4.12) yields

$$E_{out,t} \gtrsim \alpha_1 \frac{\alpha}{\alpha - 1} \frac{D}{d_m} \frac{A - B_t}{A} n\bar{o}t \tag{4.13}$$

In addition to spending energy on relaying messages which have originated outside of the bottleneck zone, the nodes inside the bottleneck zone must also spend energy on sensing and relaying the messages which have originated inside the bottleneck zone. Let $E_{in,t}$ be the total energy spent on sensing and relaying the messages which have originated inside the bottleneck zone until time t. We have

$$E_{in,t} = n\frac{B_t}{A}\bar{o}\beta t + \iint_{B_t} f(r)\frac{n}{A}\bar{o}t\, d\sigma \tag{4.14}$$

(4.14) consists of two parts: the energy spent on sensing which is $n\frac{B_t}{A}\bar{o}\beta t$, and the energy spent on forwarding the internal sensed data to the sink which is $\iint_{B_t} f(r)\frac{n}{A}\bar{o}t\, d\sigma$. $f(r)$ here represents the amount of energy spent on forwarding a one-bit message which originates at a sensor node of a distance r to the sink. According to ref. [2], we have $f(r) \geq \alpha_1 \frac{\alpha}{\alpha-1}\frac{r}{d_m} - \alpha_{12}$. The $-\alpha_{12}$ term accounts for the fact that the node generating the message does not spend any energy on receiving. Applying the expression of $f(r)$ to (4.14), we have

$$E_{in,t} \geq (\beta - \alpha_{12})n\bar{o}t\frac{B_t}{A} + \alpha_1 \frac{\alpha}{\alpha - 1}\frac{n\bar{o}t}{Ad_m}\iint_{B_t} r\, d\sigma \tag{4.15}$$

Obviously, the total energy cost within the bottleneck zone until time t is the sum of $E_{out,t}$ and $E_{in,t}$.

Performance Upper Bound in Terms of Network Lifetime

To develop the performance upper bound in terms of network lifetime, or the network lifetime bound for ease of narration, we use the condition that the total energy consumption within the bottleneck zone cannot exceed the initial energy reserve in the zone. We then have

$$E_{out,t} + E_{in,t} \leq ne^b \frac{B_t}{A} \tag{4.16}$$

Let t_u be the network lifetime bound. Incorporating (4.13) and (4.15) into (4.16) yields

$$t \lesssim \frac{d_m B_t e^b}{\bar{o}\alpha_1 \frac{\alpha}{\alpha-1}[D(A-B_t) + \iint_{B_t} r\, d\sigma] + \bar{o}(\beta - \alpha_{12}) d_m B_t} = t_u \tag{4.17}$$

(4.17) gives the network lifetime bound. In the special case in which the bottleneck zone is a complete round surface of radius D, we have

$$t_u = \frac{d_m \pi D^2 e^b}{\bar{o}\alpha_1 \frac{\alpha}{\alpha-1}[D(A-\pi D^2) + \frac{2}{3}\pi D^3] + \bar{o}(\beta - \alpha_{12}) d_m \pi D^2} \tag{4.18}$$

Performance Upper Bound in Terms of the Total Collectable Information

The performance upper bound in terms of the total information collected throughout the network lifetime, or the information bound for ease of narration, can be easily derived from the network lifetime bound which has been acquired above.

Until the optimal network lifetime t_u, the sink can receive at most $n\bar{o}t_u$ data bits in the best case in which no sensor node fails. Let $data_u$ be the information bound, we have the following relationship:

$$data_u = \bar{\sigma} n \bar{o} t_u \tag{4.19}$$

where $\bar{\sigma}$ is the average information weight for the sensed data bits.

Incorporating (4.17) into (4.19) yields the information bound as

$$data_u = \frac{\bar{\sigma} n d_m B_t e^b}{\alpha_1 \frac{\alpha}{\alpha-1}[D(A-B_t) + \iint_{B_t} r\, d\sigma] + (\beta - \alpha_{12}) d_m B_t} \tag{4.20}$$

Similarly, we can acquire the information bound for the special case in which the bottleneck zone is a complete round surface by incorporating (4.18) into (4.19).

Discussion on Performance Upper Bounds

In addition to the inflexible radio parameters and the bottleneck zone which is preferred to be a round surface wherever possible, the network deployment variables that can affect the performance bounds t_u and $data_u$ only include the bottleneck

nodes' initial energy reserve e^b, the total number of deployed nodes n and the deployment area size A. The relations between the network performance bounds and these deployment variables are analyzed as the following.

Scenario 1: Fixed n and A, variable e^b. (4.17) and (4.20) show that both t_u and $data_u$ linearly increase with e^b. By assigning more available energy resources to the important bottleneck zone nodes instead of the peripheral nodes can effectively alleviate the bottleneck effect.

Scenario 2: Fixed e^b and A, variable n. It can be seen that n disappears in (4.17), thus $t_u = constant$ as n varies. However, $data_u$ linearly increases with n. In this dissertation, we do not touch upon the situation when the network becomes oversaturated as n grows.

Scenario 3: Fixed e^b and n, variable A. We can prove that t_u decreases by the order of $\frac{1}{A}$ as $A \to \infty$. Since n is fixed in this scenario, $data_u$ also decreases by the order of $\frac{1}{A}$ as $A \to \infty$.

Scenario 4: Fixed e^b and $\frac{n}{A}$, variable n and A. In this scenario, the network size varies while the network density remains the same. We can prove that t_u decreases by the order of $\frac{1}{A}$ as $A \to \infty$, while $data_u$ remains constant as $A \to \infty$.

4.2.4 Summary of Section 4.2

In this section, the bottleneck zone in an energy-constrained wireless sensor network is identified. Further, the influence of the bottleneck zone on the entire network's performance is analyzed by deducing the performance upper bounds limited based on the available energy resources inside the bottleneck zone. Finally, the means by which network deployment variables (e.g. the number of nodes) may affect the performance bounds are discussed.

The acquired results provide some insights for the proper deployment of wireless sensor networks. Further, the performance upper bounds given in this section can be used as benchmarks in the performance evaluation of protocols or algorithms developed for energy-constrained wireless sensor networks.

4.3 RIFES: An Optimal Energy Allocation Scheme for Dense Sensor Networks

Individual sensor nodes in a wireless sensor network usually suffer from energy constraints. The multi-hop transmission even causes a deterioration in this constraint by consuming much more energy on the neighboring nodes to a sink node. Usually the energy allocation in wireless sensor networks is not good at considering individual energy consuming rates. An ill-considered energy allocation scheme will result in one part of the sensor nodes dying at an earlier stage than the others and the network performance will thus be degraded. In this section, we will present an efficient

fair energy allocation scheme which allocates energy to a sensor node according to its expected traffic load. In concrete terms, we consider a dense multihop sensor network with a large number of sensor nodes evenly and densely deployed in a disk area and a sink node located at the center. The traffic load distribution for the considered network scenario has already been formulated in Section 3.4. Based on the knowledge of the traffic load distribution, an optimal energy allocation scheme called RIFES is proposed, given the total amount of energy available and the total number of deployed sensor nodes. RIFES is considered to be optimal as it is the case in which the allocated energy on each individual sensor node is matched to the expected traffic load for this node. Because communication dominates the energy consumption of a sensor node, all sensor nodes will have the same expected energy exhaustion time and the network performance is maximized.

4.3.1 Network Scenario & Traffic Load Distribution

Consider a dense multihop sensor network scenario with n sensor nodes evenly and densely deployed in a disk area of radius R. There is a sink node located at the center of the area. The dominating communication pattern is that the distributed sensor nodes regularly sense the physical phenomenon around and forward that information to the sink directly or by multi-hop transmission. In the considered dense and symmetric network, a routing path will be along the straight line connecting the sender and the receiver (i.e. the sink). Further, we consider a source-initiated routing algorithm whose generated routing path has the number of hops linearly increasing with the Euclidian distance between the source and the destination in a uniformly deployed Euclidian space. In the considered dense network, all routing hops will have a length which is fairly close to their mean hop length denoted by h.

The traffic load distribution for the considered network scenario has already been deduced in Section 3.4. For any node which is r away from the sink in this symmetric network, its expected traffic load can be expressed as:

$$Traffic(r) \approx \frac{R^2 - r^2}{2rh}\bar{o} \tag{4.21}$$

(4.21) generally applies to all nodes deployed in the disk area and having a distance $r \in (0,\, R]$ from the sink. However, (4.21) has an unreasonable extreme value zero when $r \to R$. To remove this problem, we can use the knowledge that the edge nodes which lie within one hop distance from the area's boundary only experience the sensing traffic load which is $\bar{o}$ packets per unit time. Thus, the revised approximated traffic load function becomes:

$$Traffic(r) = \begin{cases} \frac{R^2-r^2}{2rh}\bar{o}, & \text{if } r \in (0,\, R-h]; \\ \bar{o}, & \text{if } r \in (R-h,\, R]. \end{cases} \tag{4.22}$$

4.3.2 Optimal Energy Allocation

In this section, we are interested in finding an optimal energy allocation scheme thus all deployed nodes have the same exhaustion time. Note that the uneven traffic load distribution means that those nodes closer to the sink will transfer more traffic and consume more energy. If an equal amount of energy is allocated to each deployed node, the nodes closer to the sink will always die at an earlier stage because of their heavier traffic loads. This will, in turn, lead to the failure of the whole network by disconnecting those nodes furthest from the sink. However, at this time the disconnected furthest nodes still have some residual energy which leads to an inefficiency in utilizing the allocated energy. Naturally, the ideal means of allocating energy is to ensure that all the nodes have the same exhaustion time and thus this maximizes the network lifetime as all the nodes are contributing.

In this section, the fair energy allocation problem is formulated as the problem of finding the exact amount of energy allocated to all deployed nodes with preknowledge of their Euclidean distances from the sink in the considered network scenario, given the total amount of available energy; thus all deployed nodes have the same exhaustion time.

Because communication dominates the energy consumption of a sensor node [71], the energy consuming rate at a node can be considered to be proportional to this node's traffic load. Further, the energy consuming rate of the whole network is accordingly proportional to the total traffic load over all the deployed nodes. In Section 4.3.1, the closed form expression of the expected traffic load experienced by a node which has a distance r from the sink has already been presented, from which the total traffic load over all the deployed nodes can be estimated as:

$$\begin{aligned} Traffic_{total} &= \iint Traffic(r)\rho\, d\sigma \\ &= \int_0^{2\pi} d\theta \int_0^{R-h} \frac{R^2-r^2}{2rh}\bar{o}\frac{n}{\pi R^2} r\, dr + \int_0^{2\pi} d\theta \int_{R-h}^{R} \bar{o}\frac{n}{\pi R^2} r\, dr \\ &= \frac{n\bar{o}}{hR^2}\int_0^{R-h}(R^2-r^2)\, dr + \frac{2n\bar{o}}{R^2}\int_{R-h}^{R} r\, dr \\ &= n\bar{o}[\frac{2R}{3h} + \frac{h}{R}(1-\frac{2h}{3R})] \\ &\approx \frac{2R}{3h}n\bar{o} \end{aligned} \tag{4.23}$$

where $\rho = \frac{n}{\pi R^2}$ is the node density in the deployed area. The final approximation uses the knowledge that $\frac{h}{R}$ is a small number. In the above derivation, (4.22) is used as the traffic load expression. The same result can be acquired by using (4.21).

As explained above, the traffic load is an indicator of the energy consuming rate. An optimal energy allocation scheme will allocate a proportion of the total amount of available energy to a node according to the proportion of this node's expected traffic load. Let E_{total} be the total amount of available energy, and let $E(r)$ be the part of the energy allocated to a node which has a distance r from the sink. We have

the following energy allocation formula:

$$E(r) = \frac{Traffic(r)}{Traffic_{total}} E_{total} \tag{4.24}$$

By incorporating (4.22) and (4.23) into (4.24), we obtain

$$E(r) = \begin{cases} \frac{3(R^2-r^2)}{4Rrn} E_{total}, & \text{if } r \in (0,\ R-h]; \\ \frac{3h}{2Rn} E_{total}, & \text{if } r \in (R-h,\ R]. \end{cases} \tag{4.25}$$

The optimal energy allocation scheme is thus acquired by repeatedly applying (4.25) to all the deployed nodes with known distances from the sink. In addition, each different node has an independent energy allocation providing all the nodes are going to be uniformly deployed in the field. The allocated energy associated with this scheme is matched to every node's expected traffic load, thus the resulting network lifetime or battery replenishment period is maximized.

From (4.25), we observe that except for the edge nodes whose energy allocation requires the knowledge of the mean routing hop length h, the energy allocation for the majority of the deployed nodes does not require an estimation of the routing hop length. Thus, we consider the presented energy allocation scheme to be routing independent and call it RIFES (Routing Independent Fair Energy-Allocation Scheme) in the following sections.

It should be noted that the total energy finally allocated by using RIFES may have a small discrepancy to the energy budget E_{total}, because $Traffic_{total}$ in (4.23) is not acquired by the summation of the individual nodes' expected traffic loads, and is thus only an approximation of the total expected traffic load. The benefits of using (4.23) are that it is possible to allocate energy to individual nodes seperately without the necessity of considering the precise locations of the other deployed nodes, and as well as obtaining a neater expression for (4.25).

4.3.3 Simulation Results

In this section, we use simulations to validate the optimal energy allocation scheme RIFES presented above. We simulate a network with $n = 10,000$ sensor nodes uniformly distributed along grid points in a 2-dimensional disk sensing area of radius $R = 100$ meters (that is to say, the average distance between the deployed nodes is about 1.8 meters). A sink node is located at the center of the sensing area. The routing algorithm used in the simulations is the source-initiated shortest path routing, which means a forwarding node will select one of its neighboring nodes as its next hop relay and the selected neighboring node should have the shortest Euclidean distance to the sink among all available neighboring nodes (which are within the maximum communication range of the forwarding node). The maximum communication range for all sensor nodes is configured as 10 meters. Due to the high density of the considered network, this maximum communication range is also viewed as

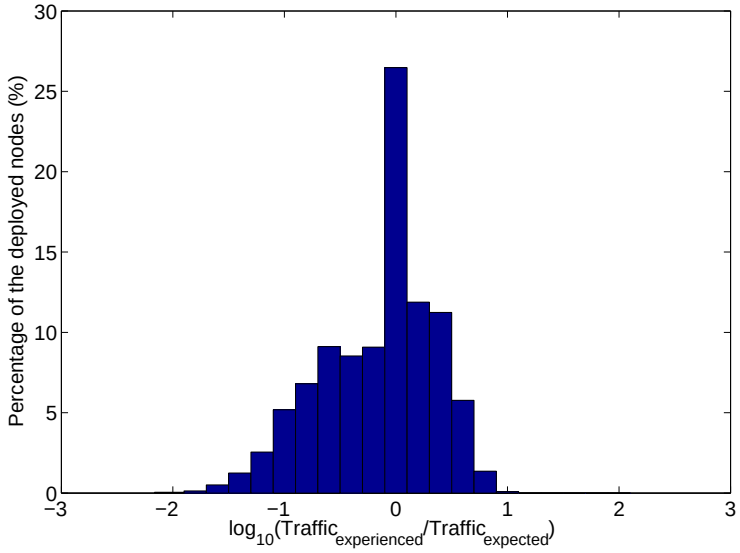


Figure 4.4: The deviation of a node's experienced traffic load from its expected traffic load.

the approximation of the hop length when calculating the expected traffic load on individual nodes and when allocating energy for those edge nodes using RIFES. Moreover, all the nodes have the same packet generation rate of $\bar{o} = 1$ packet per unit time. We assume that each packet transmission causes a fixed energy consumption E_{pkt} at the forwarding node and the packet transmission is the only source that consumes energy. The total available energy in the network or the energy budget is $10^6 E_{pkt}$.

Obviously, the energy allocation scheme would be optimal if the amount of energy allocated to each node is strictly proportional to the traffic load experienced by that node. Given that RIFES utilizes the knowledge of the expected traffic load distribution over the nodes, the error arises when a node's experienced traffic load deviates from its expected traffic load. Let $Traffic_{experienced}$ be a node's experienced traffic load, and let $Traffic_{expected}$ be the same node's expected traffic load. Fig. 4.4 demonstrates the distribution of the logarithm of the ratio $\frac{Traffic_{experienced}}{Traffic_{expected}}$ from one sample simulation. It can be seen that the deviation is actually centered somewhere close to zero, which proves the sound basis of RIFES. However, Fig. 4.4 also shows that the difference between a node's expected and experienced traffic loads can be up to a factor of ten. This inconsistency will degrade the performance of RIFES.

In the following, the performance of RIFES is evaluated by investigating the ∂ network lifetime achieved by using it. The ∂ network lifetime is considered as the operational time up to the point at which $\partial \times 100$ percent of the total deployed nodes run out of energy or are partitioned from the sink due to the destruction of their routing paths. We firstly present the results when there is no rerouting, namely a node will be permenantly disconnected from the sink once its routing path to the

sink is broken. We then consider a rerouting strategy (Algorithm 1), namely that a new routing path can be built if the old routing path is broken. The rerouting strategy will lead to a more efficient use of the allocated energy. We discover that although neighboring nodes each may experience a traffic load which greatly deviates from that which was expected, their average experienced traffic load coincides with their expected traffic load (for neighboring nodes, their expected traffic loads can be viewed as the same because they have similar distances to the sink). Thus, by shifting the traffic load from the heavy-load node to its low-load neighboring nodes, a rerouting strategy assists in providing a balanced traffic distribution. Because neighboring nodes together experience a total traffic load which coincides with their total expected traffic load when there is a rerouting stragegy, the optimal performance targeted by RIFES may be approached. In both scenarios, the ∂ network lifetimes achieved by using RIFES are compared to those achieved by using a straight-forward even energy allocation scheme, and in addition to those achieved through the use of an ideal energy allocation scheme where energy allocation is directly based on the experienced traffic loads.

Algorithm 1: REROUTING

Input: The list of dead nodes, which is initialized with those nodes known to be dead soon; the list of live nodes; network topology; current routing paths.
Output: Updated list of live nodes; updated routing paths.
1) **repeat**
2) Get the first item from the list of dead nodes, and denote it as i
3) Remove i from the list of live nodes
4) **foreach** (immediate child node j of node i)
5) {
6) **if** There is live neighboring nodes on j's way to the destination **then**
7) Rebuild j's routing path according to the shortest-path routing algorithm
8) **else**
9) Put j into the list of dead nodes
10) **end if**
11) }
12) Remove i from the list of dead nodes
13) **until** The list of dead nodes becomes empty

Since the goal is to provide an optimal energy allcoation scheme and thus the operational lifetime during which all nodes are alive is maximized, we are only interested in investigating the ∂ network lifetime acquired when the death of up to a small percentage of the deployed nodes is reached. Fig. 4.5 shows the average results of 100 simulations when there is no rerouting strategy adopted. For RIFES and the even energy allocation scheme, their corresponding ∂ network lifetimes are respectively acquired by allocating energies to the deployed nodes individually according to RIFES and by distributing the available energy evenly among the deployed nodes. As to the ideal energy allocation scheme, all the deployed nodes have

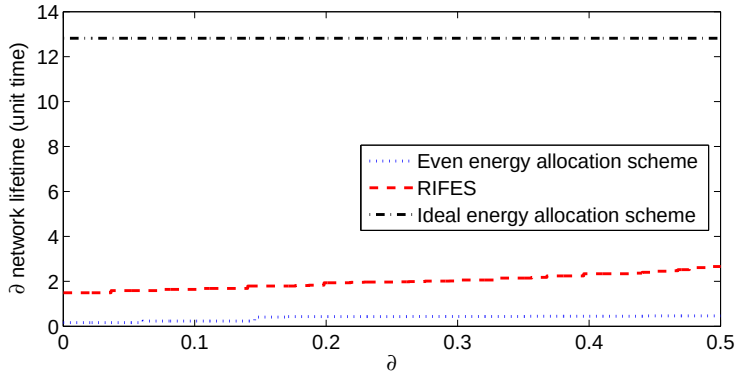


Figure 4.5: ∂ network lifetimes achieved when there is no rerouting strategy.

the same exhaustion time. Thus, the first node dies simultaneously with the last node, and all ∂ network lifetimes have the same value. In our presented results, the "single" ∂ network lifetime achieved by using the ideal energy allocation scheme is calculated by dividing the energy budget by the total energy consuming rate, where the latter equals the multiplication of the total traffic load and E_{pkt}. It can be seen that the ∂ network lifetimes achieved by using RIFES are much longer than those achieved by using the even energy allocation scheme. This is particularly the case, when $\partial = 0$ (i.e. the network operates till the first node dies) using RIFES achieves a ∂ network lifetime which is about 9 times that acheived by using the even energy allocation scheme; when $\partial = 0.5$ the ∂ network lifetime achieved by using RIFES is about 6 times that achieved by using the even energy allocation scheme. However, the achieved ∂ network lifetime by using RIFES is only 12% for $\partial = 0$ and 21% for $\partial = 0.5$ of that achieved by using the ideal energy allocation scheme. The reason behind this is the deviation between the experienced traffic load distribution and the expected traffic load distribution, where the latter is the basis of RIFES. In the following, it will be shown that a rerouting strategy can significantly improve the ∂ network lifetimes achieved by using RIFES.

In the above, RIFES has been shown to have an overwelming advantage compared to the straight-forward even energy allocation scheme in terms of the ∂ network lifetime achieved. However, it also shows that there is a performance gap between RIFES and the ideal energy allocation scheme. In this case it is shown that this performance gap can be mitigated if a rerouting strategy is adopted. We use the rerouting strategy described in Algorithm 1. At each occasion when there is a node running out of energy or being unable to find a path to the sink (i.e. there is a new dead node), the rerouting procedure will be launched to rebuild those routing paths affected by the death of this node. Algorithm 1 can be implemented in a distributed manner in reality if each deployed node remains aware of the updated live/dead states of all its neighboring nodes.

Fig. 4.6 shows the average results for the same 100 simulations when the rerouting strategy is adopted. It can be seen that both RIFES and the even energy alloca-

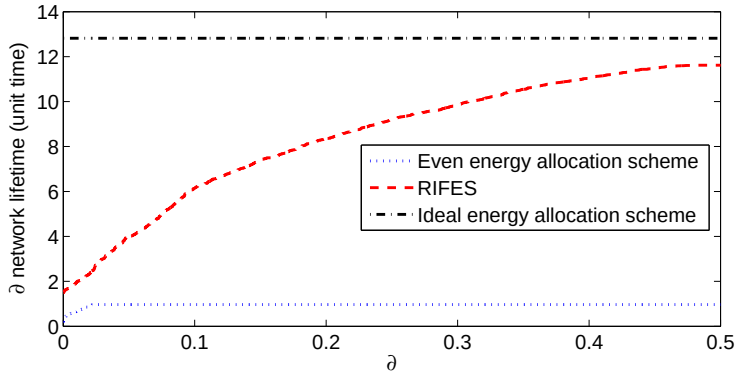


Figure 4.6: ∂ network lifetimes achieved when there is a rerouting strategy.

tion scheme exhibit performance enhancements by reconnecting those disconnected nodes to the sink. When $\partial = 0$, there is no rerouting and thus there is no performance enhancements for the simulated energy allocation schemes. However, the performances of the simulated energy allocation schemes are improved quickly as the value of ∂ increases. When $\partial = 0.5$, using RIFES leads to a ∂ network lifetime which is 91% of that achieved by using the ideal scheme, while using the even energy allocation scheme leads to a ∂ network lifetime which is 7.5% of that achieved by using the ideal scheme.

To sum up, RIFES performs about 10 times better than the straight-forward even energy allocation scheme. By using an appropriate rerouting strategy, it is also possible for RIFES to arrive at a network lifetime which is not significantly worse than that achieved by using the ideal scheme if the deaths of a small percentage of the deployed nodes are acceptable.

As stated at the end of Section 4.3.2, the total energy allocated by using RIFES may have a small discrepancy to the energy budget. The total energies finally allocated by using RIFES in the above 100 simulations are compared to the energy budget in order to investigate the extent of the possible discrepancy. Let $Energy_{final}$ be the final allocated energy when RIFES is used, and let $Energy_{budget}$ be the energy budget. Fig. 4.7 shows the distribution of the difference between $Energy_{final}$ and $Energy_{budget}$. It can be seen that there is no obvious difference between $Energy_{final}$ and $Energy_{budget}$. Thus, RIFES is very good at maintaining the energy cost within the budget.

4.3.4 Summary of Optimal Energy Allocation in Dense Sensor Networks

An efficient optimal energy allocation scheme for dense wireless sensor networks is proposed based on the traffic load distribution results presented in Section 3.4. The energy allocation scheme provided, matches the energy distribution to the traf-

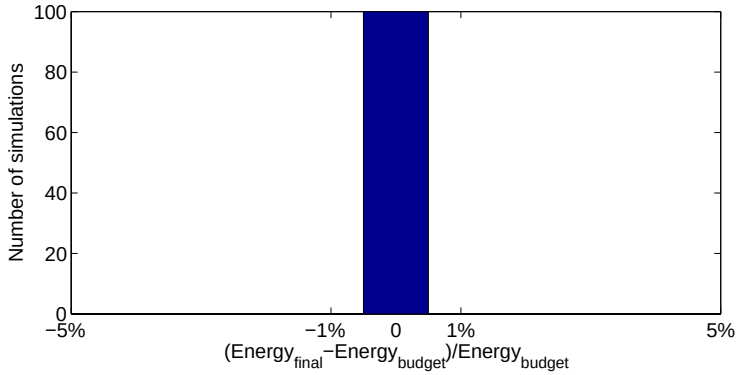


Figure 4.7: The distribution of the difference between the final allocated energy and the energy budget.

fic load distribution over the nodes. In addition, the presented energy allocation scheme can be considered to be routing independent in the sense that the energy allocation does not rely on the knowledge of the routing hop length for the majority of the deployed nodes. Thus it is of practical significance. Experiments show that the presented energy allocation scheme exhibits significant performance improvements in comparison to the straight-forward even energy allocation scheme in terms of the network lifetime achieved. When a rerouting strategy is adopted, using the presented energy allocation scheme can result in a ∂ network lifetime which is not significantly different to that achieved by using an ideal scheme.

4.4 Chapter Summary

In this chapter, the author presents his work on the optimiztion of WSNs. This has benefited from previous research results and an understanding of the traffic characteristics in sensor networks. By modeling the traffic flows and their related energy consumption, a linear programming problem has been formulated to solve the network optimization problem. The special many-to-one communication pattern actually forms a bottleneck zone around the sink. By investigating the traffic loads and their related energy consumption within a defined bottleneck zone, the performance effect of this bottleneck zone is formularized. Work relating to optimal resource allocation is also presented. By utilizing the knowledge of the traffic load distribution in dense WSNs, an optimal energy allocation scheme called RIFES is presented. RIFES maximizes the network lifetime during which all sensor nodes are operational.

Chapter 5

Packet Traffic: A Good Source for Detecting Sensor Network Anomalies?

In the previous chapters, the sensor network traffic characteristics and their applications on network optimization have been studied. In this chapter, the potential use of the learned traffic knowledge in detecting network anomalies, based on which efficient responses can be made to enhance network security and reliability, will be deliberated.

It has been mentioned in Chapter 3 that the traffic patterns of WSNs are much simpler and less dynamic than those of the more traditional networks (e.g. Internet). This makes it possible to build high-precision traffic profiles for individual sensor nodes and for the whole WSN. Actually, the traffic profiles evolve quite slowly for most WSNs where there is only limited mobility. If a sudden change occurs on the observed traffic profile, it usually means there is something unexpected happening.

In the following, two methods of detecting anomaly events in WSNs based on traffic profile modeling and matching are discussed. One method uses the knowledge that the communications in WSNs are mainly local (besides those communications joined by the sink nodes) and limited in types. Thus, it is proposed that the sequence relationships of the packets arriving at individual nodes can be learned and used to build the node traffic profiles. By matching the runtime packet arriving sequence with a database of the learned normal packet arriving sequences, anomalies caused by malicious attacks can be identified. Another method uses the ON/OFF model to capture the traffic bursts caused by intermittent target observations in a target-tracking sensor network. Each ON period indicates an event that the neighborhood of a considered sensor node has been visited by the target, while the length of an ON period provides information relating to the duration of a visit. An OFF period corresponds to the idle time period when there is no target observation. It is found that both the ON and OFF period distributions are steady (i.e. not changing

with time) if the target follows a certain random mobility model. Thus, anomalies can be detected by matching the runtime target observation statistics with the historical ON/OFF period distributions.

5.1 Anomaly Detection on Sequence of Arriving Packets

It has been presented in Section 3.2 that the observed packet arriving sequences at a node can be used to build the traffic profile for that node. Due to the simple communication scenario and limited mobility in WSNs, a traffic profile built in this manner evolves quite slowly over time and can be used as the baseline profile to detect anomaly traffic caused by malicious attacks.

5.1.1 Basic Idea and Assumptions

WSNs are usually designed for specific applications, such as information collection, and usually have low mobility. Each sensor node involved has its own role assignment and only performs the necessary and specified operations. Exhibited on the communication, obvious and differentiable patterns exist on the sequence of packets (both incoming and outgoing) arriving at each sensor node. The method of extracting patterns for packet arriving sequences have been presented in Section 3.2. In the following, this pattern extraction process is proposed to be conducted automatically by means of online training. Once a stable set of the learned patterns has been acquired, it can be used as the normal traffic profile for the node of interest. As malicious attacks usually appear with unreasonable, even stochastic packet sequence, such as the case in which a Routing Reply is generated without a pre-received Routing Request or a HELLO message received from a non-neighbor node, then it is to be hoped that they can be detected by matching the runtime observed packet arriving sequence with the normal traffic profile prebuilt.

However, there are prerequisites for the functioning of such a type of anomaly detection proposal. The prerequisites include:

1. The network is not too dynamic, so that traffic profiles can be learned within a reasonable time span and do not require frequent updating.

2. The detection system is still able to function in a satisfactory manner at the point at which anomalies appear. Thus the detection system is required to be fault tolerant as well as resistant to malicious attacks.

These prerequisites serve as assumptions in the author's proposals.

5.1.2 System Architecture

The profile-based anomaly detection consists of two stages: the Profile Learning stage and the Anomaly Detecting stage. Additionally, the Profile Learning stage consists of five modules: *Packet Capturing Module*, *Feature Extraction Module*, *Packet Classification Module*, *Event Translation Module* and *Pattern Extraction Module*. The Anomaly Detecting stage consists of the same types of modules apart from the final one, where *Pattern Matching Module* is used instead of *Pattern Extraction Module*. A conceptual view of our approach is shown in Fig. 5.1. Each time a new packet arrives at the node of interest, it is firstly captured by the *Packet Capturing Module*. Then the captured packet is sent to the *Feature Extraction Module* where selected features in each packet are extracted. The *Packet Classification Module* further classifies each arriving packet according to the predefined classification set. Packets classified into the same category are treated as the same event and are further translated into an internal format in the *Event Translation Module*. Finally, based on the arriving order of these translated events, the *Pattern Extraction Module* on the Profile Learning stage extracts patterns and builds a node profile, or the *Pattern Matching Module* on the Anomaly Detecting stage matches the arriving event sequence with the learned patterns and detects anomalous node behaviors.

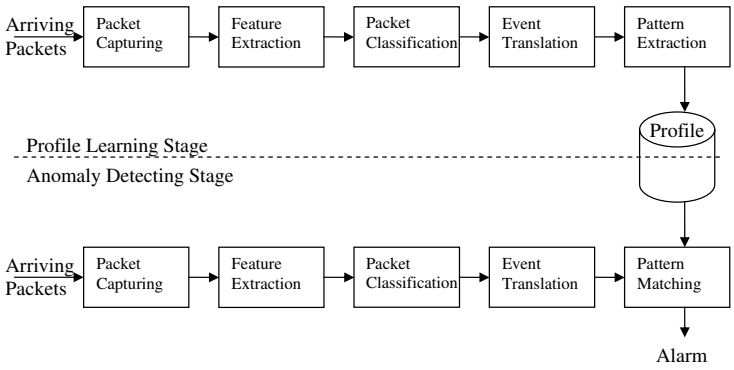


Figure 5.1: Architecture for anomalous sensor node behavior detection

The *Packet Capturing Module* can be realized by inserting a packet monitoring layer above the MAC layer in the network stack such as what it is done in NS2-MIUN [11]. All interested packets can be filtered out and made copies for later profile building and anomaly detection use. The rest modules then receive and process the output data of the *Packet Capturing Module*. For the implementation of *Feature Extraction Module*, *Packet Classification Module*, *Event Translation Module* and *Pattern Extraction Module*, we can follow the instructions presented in Chapter 3. Namely, selected packet features (i.e. source, destination, and packet type) are firstly extracted from an interested arriving packet. Then, the source and destination addresses are generalized and translated to a value among the set of {me; neighbor; local; unlocal; sink/cluster head} from the point of view of the node of interest. Afterwards, those packets with the same generalized features are classified into the same category. For

ease of internal processing, the set of packet categories are mapped to a set of characters. Finally, the sequence of packets arriving at a node of interest can be viewed as a large (or asymptotically infinite) string of characters, and the pattern extraction is performed by scanning all given, length k, unique subsequences. As long as there is no new observed unique subsequence, a traffic profile for the node of interest is built. During the anomaly detecting stage, a *Pattern Matching Module* also scans for the given length unique subsequences. Instead of putting a new observed unique subsequence into the pattern database representing the traffic profile, an alarm may be triggered for an unknown subsequence. This will be described in the following sections.

5.1.3 Pattern Matching and Alarm

The pattern matching is similar to the pattern extraction. A buffer window of length k is maintained across the sequence of arriving packets during runtime monitoring. Each time a new interested packet arrives, the buffer window is moved forward by one position and checked for a *match*, i.e., whether there is a pattern that matches the subsequence in the buffer window. If no matching pattern exists, then this is called a *mismatch*.

Note that the method for raising an alarm must not depend on the sequence length of packet arriving events. Arriving packets have to be processed in real time, and waiting until all packets have arrived before a check for possible anomalies is not an option. The following texts explain the idea of finding a *mistmatch* and launching an alarm in real time.

Let a and b be two sequences of length k. The expression a_i designates the character at position i. The difference $d(a, b)$ between a and b is defined as

$$d(a,b) = \sum_{i=1}^{k} f_i(a,b),$$

where $f_i(a,b) = \begin{cases} 0 & \text{if } a_i = b_i \\ 1 & \text{otherwise} \end{cases}$.

During pattern matching, we determine for each subsequence u of the arriving packet sequence the *minimum* distance $d_{min}(u)$ between u and the entries in the pattern database:

$$d_{min}(u) = \min\{\, d(u,p) \quad \forall \text{ patterns } p \,\}.$$

To detect an anomaly event, at least one of the observed subsequences affected by this event must be classified as anomalous. In terms of the above measure, there is at least one subsequence u for which

$$d_{min}(u) > 0.$$

In the ideal case, any $d_{min}(u)$ value that is greater than 0 can be considered a sign of an anomalous event. However, a complete match cannot always be achieved,

expecially for a network with mobility and a dynamic routing strategy. Therefore, a threshold is defined such that only subsequences whose $d_{min}(u)$ value is above this threshold are considered suspicious.

Once a subsequence of packet events has been detected as suspicious, an alarm is launched.

5.1.4 Exemplifying the Detection of Malicious Attacks

In the following, the system's ability to detect some example attacks against WSNs is analyzed.

ID Spoofing and Sybil Attacks

In this attack, an attacker presents one (ID spoofing) or more (Sybil attack) spoofed identities to the network [98, 99, 100]. Those identities could either be newly fabricated identities or identities stolen from legitimate nodes.

Anomaly Detection Analysis: This attack will violate the normal profile of the malicious node which launches this attack. Most likely a spoofed identity will be unknown to the anomaly detection system, and in this case the attack will no doubt trigger an alarm. Even a spoofed identity can be previously known to the anomaly detection system, the attack is also prone to disturb the normal packet arriving sequences by introducing an outgoing packet without observing its incoming counterpart or by introducing a certain kind of packet at an inappropriate time. Further, all the neighboring nodes of this malicious node will suprisingly discover several extra new neighbors.

Sinkhole Attack

In the sinkhole attack [98, 99], a malicious node manages to attract routes from many nodes to go through it thus acting as a "sinkhole". This attack typically works by making the malicious node appear to be especially attractive for the surrounding nodes, for example, by claiming a short or a fast route to the destination. If the attacker succeeds, then data traffic attacks can be launched and these can thus prevent the discovery of other legitimate routes.

Anomaly Detection Analysis: This attack will violate the normal profile of the malicious node. Since the malicious node attempts to attract routes which could never be passed through in the normal situation, many unlocal packets (with previously unknown addresses as the source or destination) will be observed at the "sinkhole" during the attack. Thus, the normal traffic profile is violated and alarms can be launched accordingly.

Wormhole Attack

In the wormhole attack [98, 99], a malicious node captures packets from one location in the network, and "tunnels" them to another malicious node at a distant point, which replays them locally. The tunnel can be established in many different ways, such as through an out-of-band hidden channel, packet encapsulation, or high powered transmission. The tunnel creates the illusion that the two end points are very close to each other, by making tunneled packets arrive either sooner or by a lesser number of hops compared to the packets sent over normal routes. This allows an attacker to subvert the correct operation of the routing protocol, by controlling numerous routes in the network. At a later stage it is possible for this to be used to perform traffic analysis or selectively drop data traffic.

Anomaly Detection Analysis: On one hand, the two malicious nodes that cooperate in "tunneling" packets are not very likely to be neighbors, otherwise the attack is not too meaningful. If these two malicious nodes are far away from each other, the attack will generate anomalous communication between "me" (the local malicious node) and an "unlocal" node (another malicious node) at the places of both malicious nodes. On the other hand, many communications will be attracted by the efficiency of the wormhole channel and thus change their routes to pass through the wormhole channel. This will change the traffic profiles of the malicious nodes participating in the wormhole communication.

5.1.5 Summary of Anomaly Detection on Sequence of Arriving Packets

The communications within most of the proposed WSN scenarios are non-dynamic. This makes it possible to build precise traffic profiles for the deployed nodes. In Chapter 3, the author has presented the means to model the packet arriving sequences at individual nodes and has used the learned sequence patterns to build traffic profiles. In this section, the author argues that the learned traffic profiles can actually be used as baseline profiles to detect network and node anomalies. The author thus proposes the architecture and the details in implementing such an anomaly detection system. Finally, the detection of example attacks are discussed to validate the proposal.

5.2 Anomaly Detection in WSNs for Target Tracking

In a sensor network for target tracking, bursty source traffic can originate from any corner of the sensing field if there is a continuous target observation. The sensor nodes can also remain silent if they have not observed the target for a while. In Section 3.3, it has been shown that the precise distributions of the active and silent periods can be modelled for any sensor node of interest. The modeling results concerning the source traffic burst and its period contributes to the understanding of the

traffic dynamics in event-driven WSNs. In the following, we show that the modeling results are also useful when dealing with anomaly detection.

5.2.1 Reiteration of Modeling Results Presented in Section 3.3

An ON/OFF model is found to be good at modeling the bursty source traffic in a sensor network for target tracking purposes. An ON period corresponds to a period of continuous target observation by the node of interest. An OFF period is a silent period between two continuous ON periods when there is no target observation. It is found that the distributions for both the ON and OFF periods follow the generalized Pareto distribution.

5.2.2 Analysis of ON/OFF Period Distribution

Table 3.2 and Table 3.3 in Section 3.3 show the parameter values of the generalized Pareto distributions when they are used to fit the statistical ON/OFF period distributions acquired through simulations.

In the generalized Pareto distribution, parameter $k < 0$ means the tail of the distribution is finite. Table 3.2 and 3.3 show that nine out of a total of twelve fitted distributions have a negative k, and the other three have small k values which are less than 0.1. This means that the observed ON/OFF periods' lengths in all analyzed cases either have theoretical upper bounds, or their distributions have tails which decay very rapidly. This discovery has been called the short-tail property of the investigated ON/OFF period distributions.

It may also be of interest to guarantee with a 0.99 probability (or 99% confidence) that the length of an ON/OFF period is less than an upper length limit. This probabilistic upper length limit $x|F(x) = 0.99$ is calculated for each fitted distribution and the results are shown in Table 5.1 and 5.2. A comparison is made between this probabilistic upper length limit and the mean period length for each fitted distribution, and the results show that the probabilistic upper length limit is always less than 5 multiples of the mean period length. This enhances our allegation that all the investigated distributions exhibit a short-tail property.

The short-tail property exhibited by the ON/OFF period distributions is very useful for anomaly detection. In anomaly detection, an unusually long ON/OFF period is more worthy of investigation than are others. For example, an unusually long ON period could be due to a compromised sensor node or an energy exhuastion attack, and an unusually long OFF period may indicate a node failure because of energy exhuastion. Because of the short-tail property, it is possible to quickly identify the few unusually long ON/OFF period instances.

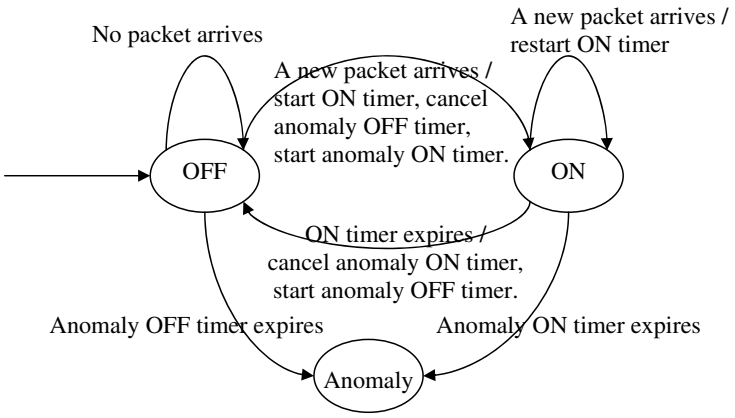


Figure 5.2: The state transition diagram for ON/OFF model with the function of anomaly detection

Table 5.1: Parameters used when fitting the generalized Pareto distribution to statistical ON period distributions

	node-edge			*node-center*		
	1.5s ON timer	5s ON timer	20s ON timer	1.5s ON timer	5s ON timer	20s ON timer
$x\|F(x)$ =0.99	16.7s	22.3s	45.5s	51s	84s	654s
$Mean$	5.703s	9.291s	24.958s	14.389s	23.525s	144.46s
$\frac{x}{Mean}$	2.93	2.40	1.82	3.54	3.57	4.53

5.2.3 Detecting Anomaly ON/OFF Periods

The goal is to detect those unusually long ON/OFF periods such that they can be identified for further analysis. Given that the ON/OFF period distributions can be modelled, a probabilistic upper length limit (e.g. $x|F(x) = 0.99$, where x is the length) for the ON/OFF periods can be easily acquired for any node of interest. Our strategy is: an anomalous ON/OFF period is detected for further analysis, whenever there is an unusually long ON/OFF period which has a length more than the specified upper length limit. We thus describe the new ON/OFF state transition diagram (an old diagram is available at Fig. 3.2 in Section 3.3.2) for anomaly detection in Fig. 5.2, where the length of "anomaly ON/OFF timer" is set to be a probabilistic upper length limit. Because the ON/OFF period distributions exhibit short-tail properties in the former analysis, an anomaly regarding a long ON/OFF period can be quickly detected within a reasonable time span in our simulation scenario.

Table 5.2: Parameters used when fitting the generalized Pareto distribution to statistical OFF period distributions

	node-edge			*node-center*		
	1.5s ON timer	5s ON timer	20s ON timer	1.5s ON timer	5s ON timer	20s ON timer
$x \vert F(x)$ =0.99	1356s	1355s	1353s	46.8s	47.2s	45.2s
$Mean$	306.98s	307.40s	302.24s	10.549s	10.117s	10.769s
$\frac{x}{Mean}$	4.42	4.41	4.48	4.44	4.67	4.20

5.2.4 Summary of Anomaly Detection in WSNs for Target Tracking

In this section, an anomaly detection strategy is proposed to detect anomalous sensor node activities in a sensor network for target tracking. Because the target observation by a sensor node is intermittent, the source traffic triggered by the target observation will be bursty. An ON/OFF model has been used to model the bursty source traffic in target-tracking sensor networks in Section 3.3. In the proposed ON/OFF model, an ON period corresponds to a period of continuous target observation and an OFF period is the silent period between two successive ON periods. Because an unusually long ON period could mean that the node of interest has been compromised and an usually long OFF period could mean that the node of interest is dead, it is especially important to quickly identify those abnormally long ON/OFF periods to mitigate their negative influences. The proposed anomaly detection strategy watches the current ON/OFF period and triggers an alarm if the current ON/OFF period has been longer than a certain probabilistic upper length limit. Because both the ON and OFF period distributions have been shown to exhibit a short-tail property (i.e. the probability of observing a certain length of ON/OFF period decreases near-exponentially as the length increases) in the simulated target-tracking WSN scenario, an abnormal ON/OFF period could be detected quickly with high confidence and thus the anomaly detection becomes very efficient in the simulated target-tracking WSN scenario.

5.3 Chapter Summary

The traffic patterns of WSNs are much simpler and less dynamic compared to those of the more traditional networks (e.g. Internet). This makes it possible to build precise traffic profiles for individual sensor nodes as well as for the whole WSN. Due to the fact that normal network operations would not violate the built traffic profiles, any observed traffic profile violation is a signal of an anomaly. In this chapter, two methods of detecting anomaly events in WSNs have been discussed based on traffic profile modeling and matching. One method matches the runtime packet arriving sequence with the prelearned normal packet arriving sequences at the place

of any node of interest. Those prelearned normal packet arriving sequences serve as the normal traffic profile. If a "match" cannot be found between the runtime packet arriving sequence and any prelearned normal packet arriving sequence, the traffic profile has been violated and an alarm can be launched. Another method uses an ON/OFF model to capture the traffic bursts caused by intermittent target observations in a target-tracking sensor network. Each ON period indicates an event that the neighborhood of the considered sensor node has been visited by the target, while the length of an ON period provides information relating to the duration of a visit. An OFF period corresponds to the idle time period when there is no target observation. It has been found that both the ON and OFF period distributions are steady (i.e. not changing with time) if the target follows a certain random mobility model. Further, both the ON and OFF period distributions have tails which decrease near-exponentially as the period lengths increase. That means, an unusual long ON/OFF period can only be observed with an extremely low probability in the normal situation, and its runtime observation should trigger an anomaly alarm to receive speical attention.

Chapter 6

Conclusions and Future Work

6.1 Overview

Wireless sensor network (WSN) has emerged as a promising technology because of the recent advances in electronics, networking, and information processing. The WSN research was initially driven by military applications such as battlefield surveillance and enemy tracking. Now, many civil applications of WSN have also been proposed, which include habitat monitoring, environmental observation and forecasting systems, health monitoring, etc. In these applications, many low power and inexpensive sensor nodes are deployed in a vast space to cooperate as a network.

Although WSN is a promising technology which can be used in many applications, there are still a few obstacles to overcome before it finally becomes a mature technology. For example, traffic dynamics in WSNs are application dependent. For many WSN application scenarios, the traffic dynamics are still very obscure. If accurate and analytically tractable models for sensor network traffic are offered, WSNs could be improved to become more efficient and optimized. Because the energy constraint has little hope of being removed in the near future, optimizing the design of WSNs to produce the minimum energy consumption continues to be of prime importance. As more and more WSNs become available for practical deployment, problems relating to sensor failures and malicious attacks will attract more and more attention. Developing a technique which can instantly detect those anomalies caused by sensor failures and malicious attacks will be very useful.

In this dissertation, the author presents his research results and contributions within the fields of traffic analysis & modeling, network optimization and anomaly detection for WSNs. In the field of traffic analysis & modeling for WSNs, the author contributes by presenting several means of modeling different aspects of WSN traffic. To learn the sequence relations among arriving packets observed at the places of individual sensor nodes, the author maps the packet arriving sequence to an infinite character string and uses a window-based scanning process to extract those unique string patterns. To model the bursty source traffic in a target tracking sensor net-

work scenario, the author views the source traffic arrving as a process consisting of intermittent active/silent periods, and uses an ON/OFF model to fit this intermittent traffic arriving process. The special communication pattern in WSNs also lead to an unbalanced traffic load distribution over the deployed nodes. In order to have a better understanding in relation to this traffic imbalance, the author has researched the traffic load distribution in a symmetric dense WSN and has discovered that the traffic load distribution over the nodes is a function of their distances from the sink.

Further research shows that network optimization for WSNs can actually greatly benefit from the understanding of traffic dynamics. By modeling the interrelationships among communication traffic, energy consumption and WSN performances, the author has studied the manner in which the energy is consumed inside the network and has investigated several performance bounds constrained by the available energy resources. The research results provide some insights for future routing design and the proper deployment of WSNs. By utilizing the knowledge concerning the traffic load distribution in a symmetric dense WSN, the author has also proposed an optimal energy allocation scheme which maximizes the network lifetime and minimizes the energy waste.

The research works related to traffic analysis & modeling also show that the traffic patterns in WSNs are much simpler and less dynamic than those in more traditional networks such as the Internet. This makes it possible to build precise traffic profiles for individual sensor nodes as well as for the whole network. Because normal operations in WSNs generate traffic which obeys the normal traffic profiles, any violation of the normal traffic profiles signals an anomaly. In this dissertation, the author uses two examples to demonstrate the feasibility and the goodness of detecting sensor network anomalies through the analysis of network traffic.

6.2 Future Work

In the future, traffic analysis & modeling for WSNs should focus on those event-driven WSN scenarios because traffic dynamics in event-driven WSNs are much more uncertain than those in periodic data generation WSNs. Further, as node mobility has been utilized in a few WSN applications such as healthcare monitoring, it will be useful to investigate the traffic dynamics in WSNs when there is node mobility. In-network processing is viewed as an essential method to reduce and balance the energy consumption within WSNs. Because in-network processing elimates the need to transmit raw data to a central point, it also changes those familar traffic patterns in WSNs. Investigating traffic dynamics in WSNs, when different in-network processing strategies are applied, will be very necessary.

In the future, network optimization for WSNs will continue to be of prime importance given the inherent nature of limited resources. For those WSNs with node mobility and in-network processing, the fundamental performance bounds are still not clear. More network optimization models could be built to investigate the fundamental performance bounds of such WSNs, and the provision of accurate traffic models will be a pre-condition for this option. For those well-investigated WSNs

without mobility and in-network processing, the optimal performances which are achievable by centralized coordination algorithms are already known. The research focus should shift to the development of distributed coordination algorithms which are more practical in a real implementation. As to the optimal resource allocation for WSNs, this dissertation has determined an optimal energy allocation scheme for symmetric dense WSNs. The development of more resource allocation schemes for more general WSN scenarios should be very useful.

In the future, anomaly detection for WSNs will become more and more important as more and more WSNs become available for real deployment. This dissertation argues that packet traffic is a good source for anomaly detection in WSNs. This argument requires more support in the future. As malicious attacks will be low probability events in many WSNs, high false alarm rates are not tolerable in these WSNs. Designing an anomaly detection system with an extremely low false alarm rate will be a challenge. After a network anomaly is detected, either a person is required to be sent to the identified problem region or the network must take some measures to automatically recover from the possible damage. The development of such accompanying emergency response strategies will be necessary for future anomaly detection in WSNs.

Bibliography

[1] H. Karl and A. Willig, "A Short Survey of Wireless Sensor Networks," in *TKN Technical Report TKN-03-018, Technical University Berlin*, October 2003.

[2] L. Doherty, B. A. Warneke, B. E. Boser, and K. S. J. Pister, "Energy and performance considerations for smart dust," *International Journal of Parallel and Distributed Systems and Networks*, vol. 4, no. 3, pp. 121–133, 2001.

[3] G. J. Pottie and W. J. Kaiser, "Wireless integrated network sensors," *Communications of the ACM*, vol. 43, no. 5, pp. 51–58, 2000.

[4] Q. Wang and T. Zhang, "A survey on security in wireless sensor networks," in *Security in RFID and Sensor Networks*, Y. Zhang and P. Kitsos, Eds. CRC Press, Taylor & Francis Group, 2009, ch. 14, pp. 293–320.

[5] V. Paxson and S. Floyd, "Wide-area traffic: The failure of poisson modeling," *IEEE/ACM Transactions on Networking*, vol. 3, pp. 226–244, 1995.

[6] H. Øverby and N. Stol, "Effects of bursty traffic in service differentiated optical packet switched networks," *Optics Express*, vol. 12, no. 3, pp. 410–415, 2004.

[7] J.-H. Chang and L. Tassiulas, "Maximum lifetime routing in wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 12, pp. 609–619, August 2004.

[8] J.-H. Chang and L. Tassiulas, "Routing for maximum system lifetime in wireless ad-hoc networks," in *Proc. of the 37th Annual Allerton Conference on Communication, Control, and Computing*, September 1999.

[9] F. Ordonez and B. Krishnamachari, "Optimal information extraction in energy-limited wireless sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 22, pp. 1121–1129, August 2004.

[10] "The network simulator - ns-2," http://www.isi.edu/nsnam/ns/.

[11] Q. Wang, "Ns2-miun," 2009, http://apachepersonal.miun.se/∼qinwan/resources.htm.

[12] "Ns2 learning guide," http://hpds.ee.ncku.edu.tw/∼smallko/ns2/ns2.htm.

[13] C.-Y. Chong and S. P. Kumar, "Sensor networks: Evolution, opportunities, and challenges," *Proceedings of the IEEE*, vol. 91, no. 8, pp. 1247–1256, August 2003.

[14] *Proceedings of the Distributed Sensor Nets Workshop*, Pittsburgh, USA, 1978, Department of Computer Science, Carnegie Mellon University.

[15] R. Rashid and G. Robertson, "Accent: A communication oriented network operating system kernel," in *Proc. of the 8th Symposium on Operating System Principles*, 1981, pp. 64–75.

[16] C. Myers, A. Oppenheim, R. Davis, and W. Dove, "Knowledge-based speech analysis and enhancement," in *Proc. of the International Conference on Acoustics, Speech and Signal Processing*, 1984.

[17] S. Kumar and D. Shepherd, "Sensit: Sensor information technology for the warfighter," in *Proc. of the 4th International Conference on Information Fusion (FUSION'01)*, August 2001, pp. 3–9 (TuC1).

[18] "IEEE 802.15 wpan task group 4," http://www.ieee802.org/15/pub/TG4.html.

[19] "Zigbee alliance," http://www.zigbee.org.

[20] "21 ideas for the 21st century," *Business Week*, pp. 78–167, August 1999.

[21] L. M. Ni, "China's national research project on wireless sensor networks," in *Proc. of the 2008 IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC'08)*, June 2008, p. 19.

[22] "Crossbow technology," http://www.xbow.com.

[23] "Dust networks, Inc." http://www.dustnetworks.com.

[24] J. L. Hill, "System Architecture for Wireless Sensor Networks," Ph.D. dissertation, Doctor of Philosophy in Computer Science, University of California at Berkeley, USA, 2003.

[25] S. Sudevalayam and P. Kulkarni, "Energy Harvesting Sensor Nodes: Survey and Implications," Department of Computer Science and Engineering, Indian Institute of Technology Bombay, Technical Report TR-CSE-2008-19, 2008.

[26] "Tinyos community forum," http://www.tinyos.net.

[27] "Contiki," http://www.sics.se/contiki.

[28] "Ipso alliance - promoting the use of ip for smart objects," http://www.ipso-alliance.org.

[29] "Sos embedded operating system," https://projects.nesl.ucla.edu/public/sos-2x/doc/.

[30] "Liteos," http://www.liteos.net.

[31] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, August 2002.

[32] I. Howitt, W. W. Manges, P. T. Kuruganti, G. Allgood, J. A. Gutierrez, and J. M. Conrad, "Wireless industrial sensor networks: Framework for qos assessment and qos management," *ISA Transactions*, vol. 45, no. 3, pp. 347–359, July 2006.

[33] Q. Wang and T. Zhang, "Sec-snmp: Policy-based security management for sensor networks," in *Proc. of the International Conference on Security and Cryptography (SECRYPT'08), in conjunction with ICETE 2008*, July 2008.

[34] K. S. J. Pister, "Military applications of sensor networks," in *of Institute for Defense Analyses Paper P-3531, Defense Science Study Group*, 2000.

[35] D. Steere, A. Baptista, D. McNamee, C. Pu, and J. Walpole, "Research challenges in environmental observation and forecasting systems," in *Proc. of 6th International Conference on Mobile Computing and Networking (MOBICOMM'00)*, 2000, pp. 292–299.

[36] K. Martinez, P. Padhy, A. Riddoch, H. L. R. Ong, and J. K. Hart, "Glacial environment monitoring using sensor networks," in *Proc. of Workshop on Real-World Wireless Sensor Networks (REALWSN'05)*, June 2005.

[37] G. Barrenetxea, F. Ingelrest, G. Schaefer, and M. Vetterli, "Wireless sensor networks for environmental monitoring: The sensorscope experience," in *Proc. of 20th IEEE International Zurich Seminar on Communications (IZS'08)*, March 2008.

[38] "Bsn research in imperial college london," http://ubimon.doc.ic.ac.uk/bsn/m621.html.

[39] "Streetline, Inc." http://www.streetlinenetworks.com.

[40] "Rohrback cosasco systems," http://www.cosasco.com.

[41] M. Connolly and F. O'Reilly, "Sensor networks and the food industry," in *Proc. of Workshop on Real-World Wireless Sensor Networks (REALWSN'05)*, June 2005.

[42] "Expo 2010 Shanghai China," http://www.expo2010.cn.

[43] I. Demirkol, F. Alagoz, H. Delic, and C. Ersoy, "Wireless sensor networks for intrusion detection: Packet traffic modeling," *IEEE Communications Letters*, vol. 10, no. 1, pp. 22–24, January 2006.

[44] S. Cui, R. Madan, A. J. Goldsmith, and S. Lall, "Joint routing, mac, and link layer optimization in sensor networks with energy constraints," in *Proc. of IEEE International Conference on Communications ICC'05*, May 2005, pp. 725–729.

[45] Y. Ma and J. H. Aylor, "System lifetime optimization for heterogeneous sensor networks with a hub-spoke topology," *IEEE Transactions on Mobile Computing*, vol. 3, no. 3, pp. 286–294, July-September 2004.

[46] S. Tang, "An analytical traffic flow model for cluster-based wireless sensor networks," in *Proc. of 1st International Symposium on Wireless Pervasive Computing*, 2006.

[47] Q. Wang and T. Zhang, "Source traffic modeling in wireless sensor networks for target tracking," in *Proc. of the 5th ACM International Symposium on Performance Evaluation of Wireless Ad-Hoc, Sensor, and Ubiquitous Networks (PE-WASUN'08)*, October 2008, pp. 96–100.

[48] P. Wang and I. F. Akyildiz, "Spatial correlation and mobility aware traffic modeling for wireless sensor networks," in *Proc. of IEEE Global Communications Conference (GLOBECOM'09)*, December 2009.

[49] C. Tseng, P. Balasubramanyam, C. Ko, R. Limprasittiporn, J. Rowe, and K. Levitt, "A specification-based intrusion detection system for aodv," in *Proc. of the 1st ACM Workshop on Security of Ad Hoc and Sensor Networks*, 2003.

[50] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing," in *Proc. of the 2nd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'99)*, Feb. 1999, pp. 90–100.

[51] P. Yi, Y. Jiang, Y. Zhong, and S. Zhang, "Distributed intrusion detection for mobile ad hoc networks," in *Proc. of the 2005 Symposium on Applications and the Internet Workshops (SAINT-W'05)*, 2005.

[52] D. B. Johnson, D. A. Maltz, and J. Broch, "Dsr: The dynamic source routing protocol for multi-hop wireless ad hoc networks," in *Ad Hoc Networking*, C. E. Perkins, Ed. Addison-Wesley, 2001, ch. 5, pp. 139–172.

[53] Q. Wang and T. Zhang, "Detecting anomaly node behavior in wireless sensor networks," in *Proc. of the 21st International Conference on Advanced Information Networking and Applications Workshops (AINAW'07)*, May 2007, pp. 451–456.

[54] M. Noori and M. Ardakani, "Characterizing the traffic distribution in linear wireless sensor networks," *IEEE Communications Letters*, vol. 12, no. 8, pp. 554–556, August 2008.

[55] R. Subramanian and F. Fekri, "Sleep scheduling and lifetime maximization in sensor networks: Fundamental limits and optimal solutions," in *Proc. of the 5th International Conference on Information Processing in Sensor Networks (IPSN'06)*, April 2006, pp. 218–225.

[56] Q. Wang and T. Zhang, "Characterizing the traffic load distribution in dense sensor networks," in *Proc. of the 2nd International Workshop on Wireless Sensor Networks: theory and practice (WSN'09)*, December 2009.

[57] Q. Wang and T. Zhang, "Traffic load distribution in large-scale and dense wireless sensor networks," in *Proc. of the 5th Annual International Wireless Internet Conference (WICON'10)*, March 2010, to appear.

[58] Q. Wang and T. Zhang, "Fair energy allocation in large-scale and dense sensor networks," in *Proc. of IEEE Global Communications Conference (GLOBECOM'10)*, 2010, submitted.

[59] H. Ren, M. Q.-H. Meng, and X. Chen, "Investigating network optimization approaches in wireless sensor networks," in *Proc. of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, October 2006, pp. 2015–2021.

[60] C. K. Toh, "Maximum battery life routing to support ubiquitous mobile computing in wireless ad hoc networks," *IEEE Communications Magazine*, vol. 39, pp. 138–147, June 2001.

[61] S. Singh, M. Woo, and C. S. Raghavendra, "Power-aware routing in mobile ad hoc networks," in *Proc. of the 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'98)*, October 1998, pp. 181–190.

[62] Y. Zhang, M. Ramkumar, and N. Memon, "Information flow based routing algorithms for wireless sensor networks," in *Proc. of IEEE Global Telecommunications Conference (Globecom'04)*, November 2004, pp. 742–747.

[63] J.-H. Chang and L. Tassiulas, "Energy conserving routing in wireless ad-hoc networks," in *Proc. of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'00)*, March 2000, pp. 22–31.

[64] Q. Wang, T. Zhang, and S. Pettersson, "Bounding the information collection performance of wireless sensor network routing," in *Proc. of the 5th Annual Conf. on Communication Networks and Services Research (CNSR'07)*, May 2007, pp. 55–62.

[65] Q. Wang, T. Zhang, and S. Pettersson, "An effort to understand the optimal routing performance in wireless sensor network," in *Proc. of the IEEE 22nd Int. Conf. on Advanced Information Networking and Applications (AINA'08)*, March 2008, pp. 279–286.

[66] E. Shih, S. Cho, N. Ickes, R. Min, A. Sinha, A. Wang, and A. Chandrakasan, "Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks," in *Proc. of the 7th Annual ACM International Conference on Mobile Computing and Networking (MobiCom'01)*, July 2001, pp. 272–286.

[67] T. van Dam and K. Langendoen, "An adaptive energy-efficient mac protocol for wireless sensor networks," in *Proc. of the International Conference on Embedded Networked Sensor Systems (SenSys'03)*, November 2003, pp. 171–180.

[68] S. Singh and C. Raghavendra, "Pamas: Power aware multi-access protocol with signalling for ad hoc networks," *ACM SIGCOMM Computer Communication Review*, vol. 28, no. 3, pp. 5–26, July 1998.

[69] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient mac protocol for wireless sensor networks," in *Proc. of the 21st International Conference of the IEEE Computer and Communications Societies (INFOCOM'02)*, June 2002, pp. 1567–1576.

[70] M. Rabbat and R. Nowak, "Distributed optimization in sensor networks," in *Proc. of the 3rd International Symposium on Information Processing in Sensor Networks*, 2004, pp. 20–27.

[71] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proc. of the 33rd Hawaii International Conference on System Sciences*, January 2000.

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

[73] S.-C. Huang and R.-H. Jan, "Energy-aware, load balanced routing schemes for sensor networks," in *Proc. of the 10th International Conference on Parallel and Distributed Systems (ICPADS'04)*, July 2004, pp. 419–425.

[74] I. Teixeira, J. F. de Rezende, and A. de Castro P. Pedroza, "Wireless sensor network: Improving the network energy consumption," in *Proc. of the XXI Simposio Brasileiro de Telecommunicações (SBT'04)*, 2004.

[75] E. Hyytiä and J. Virtamo, "On traffic load distribution and load balancing in dense wireless multihop networks," *EURASIP Journal on Wireless Communications and Networking*, 2007, article ID 16932.

[76] L. Popa, A. Rostamizadeh, R. M. Karp, C. Papadimitriou, and I. Stoica, "Balancing traffic load in wireless networks with curveball routing," in *Proc. of the 8th ACM International Symposium on Mobile Ad Hoc Networking and Computing (Mobihoc'07)*, September 2007, pp. 170–179.

[77] C. Efthymiou, S. Nikoletseas, and R. Jose, "Energy balanced data propagation in wireless sensor networks," *Wireless Networks*, vol. 12, no. 6, pp. 691–707, December 2006.

[78] S. Tang and W. Li, "Qos supporting and optimal energy allocation for a cluster based wireless sensor network," *Elsevier Computer Communications*, no. 29, pp. 2569–2577, March 2006.

[79] Q. Wang and T. Zhang, "Bottleneck zone analysis in energy-constrained wireless sensor networks," *IEEE Communications Letters*, vol. 13, no. 6, June 2009.

[80] Q. Gao, K. J. Blow, D. J. Holding, I. W. Marshall, and X. H. Peng, "Radio range adjustment for energy efficient wireless sensor networks," *Ad Hoc Networks Journal*, vol. 4, no. 1, pp. 75–82, January 2006.

[81] C. Song, M. Liu, J. Cao, Y. Zheng, H. Gong, and G. Chen, "Maximizing network lifetime based on transmission range adjustment in wireless sensor networks," *Elsevier Computer Communications*, February 2009.

[82] C.-Y. Chang, K.-P. Shih, H.-R. Chang, and H.-J. Liu, "Energy-balanced deployment and topology control for wireless sensor networks," in *Proc. of IEEE Global Telecommunications Conference (Globecom'06)*, November 2006, pp. 1–5.

[83] A. P. da Silva, M. Martins, B. Rocha, A. Loureiro, L. Ruiz, and H. Wong, "Decentralized intrusion detection in wireless sensor networks," in *Proc. of the 1st ACM International Workshop on Quality of Service & Security in Wireless and Mobile Networks (Q2SWinet'05)*, October 2005, pp. 16–23.

[84] I. Onat and A. Miri, "A real-time node-based traffic anomaly detection algorithm for wireless sensor networks," in *Proc. of Systems Communications*, August 2005, pp. 422–427.

[85] Y. Huang, W. Fan, W. Lee, and P. S. Yu, "Cross-feature analysis for detecting ad-hoc routing anomalies," in *Proc. of the 23rd International Conference on Distributed Computing Systems*, 2003.

[86] Y. Huang and W. Lee, "A cooperative intrusion detection system for ad hoc networks," in *Proc. of the 1st ACM Workshop on Security of Ad Hoc and Sensor Networks*, 2003, pp. 135–147.

[87] S. Forrest, S. Hofmeyr, A. Somayaji, and T. Longstaff, "A sense of self for unix processes," in *Proc. of the 1996 IEEE Symposium on Security and Privacy*, May 1996, pp. 120–128.

[88] S. Hofmeyr, S. Forrest, and A. Somayaji, "Intrusion detection using sequences of system calls," *Journal of Computer Security*, vol. 6, pp. 151–180, 1998.

[89] X. Zhao, D. Massey, M. Lad, and L. Zhang, "ON/OFF Model: A New Tool to Understand BGP Update Burst," in *Technical Report 04-819, USC-CSD*, August 2004.

[90] I. Downard, "Simulating sensor networks in ns-2," 2004, http://cs.itd.nrl.navy.mil/pubs/docs/nrlsensorsim04.pdf.

[91] A. Busson, G. Chelius, and E. Fleury, "From euclidian to hop distance in multihop radio networks: A discrete approach," *Technical Report No. 5505 - version 2, INRIA*, September 2005.

[92] S. Lee, B. Bhattacharjee, and S. Banerjee, "Efficient geographic routing in multihop wireless networks," in *Proc. of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'05)*, May 2005, pp. 230–241.

[93] M. Bhardwaj, T. Garnett, and A. Chandrakasan, "Upper bounds on the lifetime of sensor networks," in *Proc. of the 2001 IEEE Int. Conf. on Communications (ICC'01)*, 2001, pp. 785–790.

[94] C. H. Foh, J. W. Tantra, J. Cai, C. T. Lau, and C. P. Fu, "Modeling hop length distributions for reactive routing protocols in one dimensional manets," in *Proc. of IEEE International Conference on Communications (ICC'07)*, June 2007, pp. 3882–3886.

[95] "Zigbee specification version 1.0," June 2005, http://www.zigbee.org.

[96] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proc. of the 33rd Hawaii International Conference on System Sciences*, January 2000.

[97] M. Bhardwaj and A. Chandrakasan, "Bounding the lifetime of sensor networks via optimal role assignments," in *Proc. of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'02)*, June 2002, pp. 1587–1596.

[98] C. Karlof and D. Wagner, "Secure routing in wireless sesnor networks: Attacks and countermeasurements," *Ad Hoc Networks*, vol. 1, pp. 293–315, 2003.

[99] I. Khalil, S. Bagchi, and C. Nina-Rotaru, "Dicas: Detection, diagnosis and isolation of control attacks in sensor networks," in *Proc. of the 1st IEEE International Conference on Security and Privacy for Emerging Areas in Communication Networks*, September 2005, pp. 89–100.

[100] J. Newsome, E. Shi, D. Song, and A. Perrig, "The sybil attack in sensor networks: Analysis & defenses," in *Proc. of the 3rd International Conference on Information Processing in Sensor Networks (IPSN'04)*, April 2004, pp. 259–268.

Biography

Qinghua Wang was born on the 14^{th} of December 1980 in Dongtai, China. He received his Bachelor of Engineering in Automatic Control from Harbin Engineering University, China, in 2002. He was a PhD student in Systems Engineering in Xi'an Jiaotong University, China, from 2002 to 2005. From 2005 to present, he has been a PhD student in Computer Science in Mid Sweden University, Sweden. In 2007, he received the Best Paper Award at the 5th Annual Conference on Communication Networks and Services Research (CNSR). In 2009, he was a TPC member of the 6th Swedish National Computer Networking Workshop and 9th Scandinavian Workshop on Wireless Adhoc Networks. He has been a reviewer for Journal of Cluster Computing and Journal of Wireless Communications and Mobile Computing.

Qinghua was also a student of ARTES++ graduate school during 2006-2007. In November 2007, he was a visiting researcher at the University of Texas at Arlington, USA. He has been an IEEE student member since 2007.