Performance Analysis and Deployment Techniques for Wireless Sensor Networks

Huimin She

Stockholm 2012

Thesis submitted to the KTH Royal Institute of Technology in partial fulfillment of the requirements for the degree of Doctor of Technology

She, Huimin
Performance Analysis and Deployment Techniques for Wireless Sensor Networks

ISBN 978-91-7501-430-2
TRITA-ICT/ECS AVH 12:02
ISSN 1653-6363
ISRN KTH/ICT/ECS/AVH-12/02-SE

School of Information and Communication Technology
KTH Royal Institute of Technology
SE-164 40 Stockholm
SWEDEN

Abstract

Recently, wireless sensor network (WSN) has become a promising technology with a wide range of applications such as supply chain monitoring and environment surveillance. It is typically composed of multiple tiny devices equipped with limited sensing, computing and wireless communication capabilities. Design of such networks presents several technique challenges while dealing with various requirements and diverse constraints. Performance analysis and deployment techniques are required to provide insight on design parameters and system behaviors.

Based on network calculus, a deterministic analysis method is presented for evaluating the worst-case delay and buffer cost of sensor networks. To this end, traffic splitting and multiplexing models are proposed and their delay and buffer bounds are derived. These models can be used in combination to characterize complex traffic flowing scenarios. Furthermore, the method integrates a variable duty cycle to allow the sensor nodes to operate at low rates thus saving power. In an attempt to balance traffic load and improve resource utilization and performance, traffic splitting mechanisms are introduced for sensor networks with general topologies. To provide reliable data delivery in sensor networks, retransmission has been one of the most popular schemes. We propose an analytical method to evaluate the maximum data transmission delay and energy consumption of two types of retransmission schemes: hop-by-hop retransmission and end-to-end retransmission. In order to validate the tightness of the bounds obtained by the analysis method, the simulation results and analytical results are compared with various input traffic loads. The results show that the analytic bounds are correct and tight.

Stochastic network calculus has been developed as a useful tool for Quality of Service (QoS) analysis of wireless networks. We propose a stochastic service curve model for the Rayleigh fading channel and then provide formulas to derive the probabilistic delay and backlog bounds in the cases of deterministic and stochastic arrival curves. The simulation results verify that the tightness of the bounds are good. Moreover, a detailed mechanism for bandwidth estimation of random wireless channels is developed. The bandwidth is derived from the measurement of statistical backlogs based on probe packet trains. It is expressed by statistical service curves that are allowed to violate a service guarantee with a certain probability. The theoretic foundation and the detailed step-by-step procedure of the estimation method are presented.

One fundamental application of WSNs is event detection in a Field of Interest (FoI), where a set of sensors are deployed to monitor any ongoing events. To satisfy a certain level of detection quality in such applications, it is desirable that events in the region can be detected by a required number of sensors. Hence, an

important problem is how to conduct sensor deployment for achieving certain coverage requirements. In this thesis, a probabilistic event coverage analysis method is proposed for evaluating the coverage performance of heterogeneous sensor networks with randomly deployed sensors and stochastic event occurrences. Moreover, we present a framework for analyzing node deployment schemes in terms of three performance metrics: coverage, lifetime, and cost. The method can be used to evaluate the benefits and trade-offs of different deployment schemes and thus provide guidelines for network designers.

Keywords: wireless sensor network, performance analysis, network calculus, coverage analysis, deployment scheme

Acknowledgments

First I would like to express sincere gratitude to my supervisor Associate Prof. Zhonghai Lu. Prof. Lu helps me a lot on my research and study, as well as on personal life. He is not only a good supervisor, but also a good friend to me. Many thanks to him for all the constructive comments and fruitful discussions. I will never forget his encouragement when my papers were rejected. Special thanks go to Prof. Axel Jantsch. He has provided a very pleasant research environment for his students. He is always supportive for my ideas and personal potentials. I would also like to thank Prof. Li-Rong Zheng for introducing me to KTH. His support has provided a good basis for the present thesis.

I would like to thank my previous supervisors Prof. Dian Zhou and Prof. Xuan Zeng at Fudan university for introducing me to the world of scientific research. I would also thank Prof. Ahmed Hemani for reviewing my thesis. Special thanks go to Associate Prof. Ingo Sander and Dr. Qiang Chen for their help.

Many thanks are due to current and former colleagues at ES department and iPack center for their friendships and the pleasant atmosphere they have created. There are too numerous names to be mentioned here, thank you all. Also, special thanks should go to the administrative staff at ES and the IT support group at ICT for their assistance in all kinds of issues.

I am grateful to all my friends in Sweden and China for the fun and encouragement. Without you, the cold, dark winter of Stockholm could be much more unpleasant.

I am grateful to the Vinn Excellence centers program of Vinnova (The Swedish Governmental Agency for Innovation Systems). Without its financial support, this work could not have been possible.

Thanks to Prof. Yuming Jiang for being my opponent. I would also like to thank Prof. Björn Pehrson, Prof. Marko Hännikäinen, and Prof. Markus Fidler for being committee members.

I owe my loving thanks to my wife Liping for her understanding, continuous support and patience. Words are not powerful enough to express my gratefulness. In addition, I would like to thank my parents and brother for their unconditional support.

Huimin She

Stockholm, Sweden, May 2012

Contents

List of Tables

List of Figures

List of Abbreviations

ACK	ACKnowledgment
ARQ	Automatic Repeat reQuest
ATM	Asynchronous Transfer Mode
BER	Bit Error Rate
CSI	Channel State Indicator
CSMA	Carrier Sense Multiple Access
DNC	Deterministic Network Calculus
ETS	Even Traffic Splitting mechanism
FBS	Flow Based Splitting
FIFO	First-In-First-Out
FoI	Field of Interest
MAC	Medium Access Control
NACK	Negative ACKnowledgment
NOS	General routing with No Splitting
PAN	Personal Area Network
PSFQ	Pump Slowly and Fetch Quickly
PTS	Probabilistic Traffic Splitting mechanism
QoS	Quality of Service
RFID	Radio-Frequency IDentification
RN	Relay Node
Rx	Receiving
SN	Sensor Node
SNC	Stochastic Network Calculus
SNR	Signal-to-Noise-Ratio
TDMA	Time Division Multiple Access
Tx	Transmission
WFP	Wait for First Packet
WSN	Wireless Sensor Network
WTS	Weighted Traffic Splitting mechanism

List of Publications

Articles

- **Huimin She**, Zhonghai Lu, Axel Jantsch, Li-Rong Zheng, "Probabilistic Event Coverage in Heterogeneous Sensor Networks", submitted to ACM Transactions on Sensor Networks, 2012.
- **Huimin She**, Zhonghai Lu, Axel Jantsch, Dian Zhou, Li-Rong Zheng, "Performance Analysis of Flow Based Traffic Splitting Strategy on Cluster-Mesh Sensor Networks", *International Journal of Distributed Sensor Networks*, Volume 2012, no-232937, 2012.
- **Huimin She**, Zhonghai Lu, Axel Jantsch, Li-Rong Zheng, Dian Zhou, "Analysis of Traffic Splitting Mechanisms for 2D Mesh Sensor Networks", *International Journal of Software Engineering and Its Applications (IJSEIA)*, Vol.2 No.3, July 2008.

In Proceedings

- **Huimin She**, Zhonghai Lu, Axel Jantsch, "System-Level Evaluation of Sensor Networks Deployment Strategies: Coverage Lifetime and Cost", *the 8th International Wireless Communications and Mobile Computing Conference (IWCMC'2012)*, Cyprus, August 2012.
- **Huimin She**, Zhonghai Lu, Axel Jantsch, Li-Rong Zheng, "Estimation of Statistical Bandwidth through Backlog Measurement", *Workshop on Network Calculus (WoNeCa'2012), in conjunction with MMB&DFT 2012*, Germany, March 2012.

- **Huimin She**, Zhonghai Lu, Axel Jantsch, Dian Zhou, Li-Rong Zheng, "Stochastic Coverage in Event-Driven Sensor Networks", *the 22nd IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'2011)*, Canada, September 2011.

- **Huimin She**, Zhonghai Lu, Axel Jantsch, Dian Zhou, Li-Rong Zheng, "Modeling and Analysis of Rayleigh Fading Channels using Stochastic Network Calculus", *the IEEE Wireless Communication and Networking Conference (WCNC'2011)*, Mexico, April 2011.

- **Huimin She**, Zhonghai Lu, Axel Jantsch, Li-Rong Zheng, Dian Zhou, "Analytical Evaluation of Retransmission Schemes in Wireless Sensor Networks", *IEEE 69th Vehicular Technology Conference (VTC2009-Spring)*, Spain, April 2009.

- **Huimin She**, Zhonghai Lu, Axel Jantsch, Li-Rong Zheng, Dian Zhou, "Deterministic Worst-case Performance Analysis for Wireless Sensor Networks", *IEEE International conference on Wireless Communications and Mobile Computing (IWCMC'08)*, Greece, August 2008

- **Huimin She**, Zhonghai Lu, Axel Jantsch, Li-Rong Zheng and Dian Zhou, "Traffic Splitting with Network Calculus for Mesh Sensor Networks," *in Proc. of IEEE conference on Future Generation Communication and Networking (FGCN'07)*, volume 2, South Korea, December 2007.

Others

- **Huimin She**, Zhonghai Lu, Axel Jantsch, "WSN-Dim: Methodology for Dimensioning and Planning of Wireless Sensor Networks", *Poster in ICES 4th Annual Conference: New Businesses based on Embedded Systems - How to Succeed!*, Sweden, September 2011

- **Huimin She**, Zhonghai Lu, Axel Jantsch, "Stochastic Coverage Analysis of Event-Driven Sensor Networks", *in Proceedings of The Second Nordic Workshop on System and Network Optimization for Wireless (SNOW'2011)*, Sweden, March 2011

- **Huimin She**, Zhonghai Lu, Axel Jantsch, Li-Rong Zheng, Dian Zhou, "A network-based system architecture for remote medical applications", *the 24th Asia-Pacific Advanced Network (APAN'07)*, China, August 2007.

Chapter 1

Introduction

This chapter provides a brief background and outlines the design challenges in wireless sensor networks. It also gives an overview of the research presented in the thesis and highlight the author's contributions.

1.1 Background

With the development of wireless communications and micro-electronics, *wireless sensor network* (WSN) has become a promising technology and received significant research attention in recent years [73, 2, 65, 21, 50, 27, 105, 68].

A typical sensor network consists of a large number of sensor nodes deployed either inside the phenomenon of interest or close to it. These sensor nodes are devices equipped with sensing, computation, and wireless communication capabilities. They take measurements and forward their observation values via the wireless interfaces to single or multiple fusion centers, which can also be called a sink node. Typical sensing tasks for sensors could be temperature, light, humidity, vibration, sound, etc. There are a variety of WSNs applications, which typically involve monitoring, tracking and controlling. Specific applications include environment surveillance, next-generation health-care systems, structural monitoring, supply chain management, disaster area monitoring, and military assistance [34, 3, 27]. Figure 1.1 shows an application of sensor networks for volcanic monitoring [104]. When an earthquake or an eruption occurs, sensor nodes detect the seismic event and send data to a base station via a multi-hop network. Collecting and analyzing data from multiple base stations can produce precise mappings of the volcano.

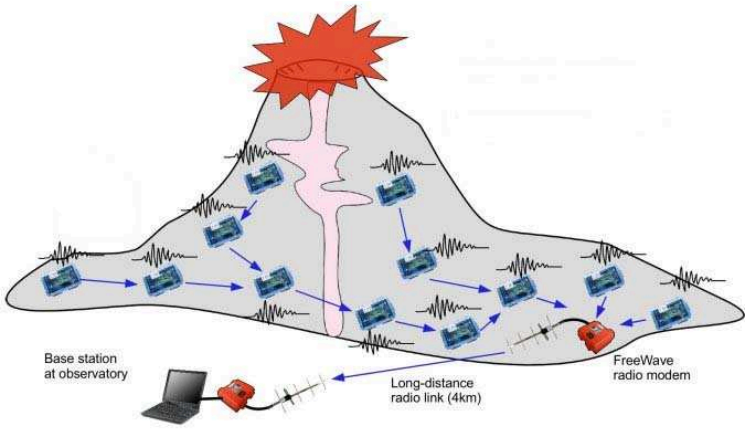


Figure 1.1. A wireless sensor network for volcanic monitoring

Although WSNs share many commonalities with existing ad hoc networks, there are still a number of unique features and application requirements [3, 50, 27]. These features, which are stated below, make the design of WSNs challenging.

- *Application specific* [10]: Sensor networks can be deployed in a variety of application scenarios. It is unlikely that there will be one all-purpose solution for all the potential possibilities. Therefore, it is essential to design a sensor network that can meet the requirements and constraints of specific domains.

- *Constrained resource* [27]: Unlike traditional networks, sensor networks are limited in power, computational capability and communication bandwidth. New solutions are needed both to meet the requirements of a specific application and to balance the trade-offs between performance and cost. On the other hand, in order to allocate the limited resource, evaluating the performance of WSNs is therefore a crucial task.

- *Limited memory size* [78]: Too much memory space would increase the cost and size of sensors, while too little memory space can not meet the requirements of applications. Hence, it is essential to estimate the required memory spaces before the deployment of sensor networks.

- *Low power requirements* [75, 16]: Due to size and cost constraints, the tiny sensor nodes can only be equipped with a limited power source. Moreover, in some application scenarios, recharge of power source is impossible. Hence,

efficient power management takes on great importance in sensor networks. One of the challenges is to design power efficient protocols.

- *Unpredictability* [3, 2]: A sensor network is subject to a number of uncertainties from several factors. First, the sensor network may be deployed in a hostile environment dealing with uncontrollable factors. Second, the wireless communication is unreliable due to interference, attenuation and fading effects. Third, some nodes may die over time and some new nodes may join later. As a consequence, the network topology and routing structures change dynamically.

1.2 Motivation

In many application scenarios of sensor networks, sensor data must be delivered to the base station within time constraints so that appropriate actions can be made. Hence, it is crucial to evaluate the performance limits, such as maximum delay, of traffic flows under all conditions [42]. Moreover, sensor networks present several technical challenges in terms of extremely low cost, low energy requirements and limited communication capabilities, while dealing with various workloads and diverse constraints [35, 76]. Addressing all these problems inevitably requires performance analysis techniques to provide insight on the design parameters and system behavior.

Since WSNs are in their early stage of development, most of the solutions are built, tested and evaluated either by simulations or testbeds. In [106], Woon *et al.* present a preliminary performance investigation, including throughput, packet delivery ratio, and average delay, of IEEE 802.15.4 standard focusing on multiple sources and multi-hop wireless sensor networks. In [31], the authors propose a mechanism to improve network coverage, increase throughput and reduce delay of a static sensor network by deploying several mobile sensors. In [111], Zhang *et al.* investigate the performance of an experimental WSN. The dynamic relaying and fixed relaying algorithms are tested and compared in terms of throughput, packet loss rate and data efficiency.

Simulations are useful in evaluating the average performance for specific cases. However, they are usually time consuming. If one configuration parameter is changed, the simulation needs to be relaunched. Moreover, simulations are very difficult or even impossible to cover worst cases. Therefore, a more efficient and faster approach is required to design and analyze a sensor network before it is deployed. The analysis needs to provide evidences that the network can meet its requirements and to indicate the performance of the network. In order to design a

WSN with predictable delay, backlog and energy consumptions, formal methods are desired to dimension sensor networks in an analytical way rather than case-by-case simulations. Starting with the seminal work by Cruz [25, 26], *network calculus* has been developed as a powerful tool for the performance analysis of networked system [11, 44, 46] . In contrast to queueing theory, network calculus deals with performance bounds, such as worst-case delay and backlog bounds, rather than average values. In general, network calculus has been developed along two tracks: *deterministic network calculus (DNC)* and *stochastic network calculus (SNC)*. The DNC generally considers the worst-case performance analysis through deterministic arrival curve and service curve. Recently, it has been extended and applied for worst-case performance analysis of sensor networks by several researchers [81, 51, 82, 49]. To incorporate non-deterministic service provisioning, the performance bounds have to be complemented with certain violation probabilities. SNC is such a tool that can be employed in the design of wireless networks to provide stochastic service guarantees. It has been applied to QoS analysis for wireless networks by many researchers recently [95, 45, 22, 60, 47].

WSNs generally have two fundamental application scenarios: tracking and monitoring. In both applications, it is essential to ensure that information of the target or the environment can be discovered and collected by sensors. To achieve good coverage, sensors are usually densely deployed. One of the most popular metrics to quantify the coverage performance is the *k-coverage* [43, 113]. An FoI (Field of Interest) is said to be k-covered if every point in it is covered by at least k sensors. The coverage problem can be classified in different ways depending on the way of sensor deployment and the features of applications. Moreover, the lifetime of WSN is determined by the energy budgets of sensors. To obtain longer network lifetime, more energy budgets should be assigned to sensors. Since sensors are usually equipped with batteries which are limited and expensive resources, deploying spare sensor nodes would cause high installation and maintenance cost. Therefore, in order to deploy minimum necessary sensors that can achieve the requirements, it is important to evaluate these performance metrics of a WSN before its deployment.

Performance analysis such as coverage, connectivity, energy consumption, cost for sensor networks has been studied by many researchers. In [20], the authors proposed a general framework for the analysis of the network lifetime and costs for several network deployment strategies in sensor networks. They investigated deployment strategies to maximize the network lifetime by mitigating the hot-spot traffic problem around the data sink. The coverage problem in sensor networks has been extensively investigated [59, 112, 100, 52, 6, 8, 72, 115, 108]. Zhang

and Hou [112] studied the problem of deriving the node density for maintaining k-coverage for a given network area in both random and deterministic deployment strategies. However, most attentions focused on analyzing the minimum coverage. In [88], the authors proposed an analytical method to model the event coverage problem in sensor networks. In [72], the authors investigated the coverage, energy consumption and message transfer delay of large-scale WSNs. They considered the square grid based deployment scheme which shows very good coverage performance and the Tri-Hexagon Tiling (THT) deployment strategy, which outperforms other schemes for energy consumption and worst-case delay.

Network calculus is a mathematical tool dealing with performance guarantees in packet switching networks [25, 26, 17, 11, 44]. With the abstraction of *arrival curve* for traffic flows and *service curve* for network elements, it has been widely applied in communication networks for performance analysis. In general, network calculus has been developed in two tracks: deterministic network calculus (DNC) and stochastic network calculus (SNC). The DNC generally considers the worst-case performance analysis through deterministic arrival curve and service curve. Recently, it has been extended and used for worst-case performance analysis of sensor networks by several researchers [81, 51]. Since data communication in wireless networks is unstable and irregular, it is very difficult or impossible to find the deterministic performance bounds. To incorporate nondeterministic service provisioning, the performance bounds have to be complemented with certain violation probabilities. SNC is such a tool which can be employed in the design of wireless networks to provide stochastic service guarantees.

In this thesis, the research has been orienting in two tracks: performance analysis for WSNs using network calculus, and deployment mechanisms especially coverage analysis in WSNs.

1.3 Contributions and Outline

The remainder of this thesis is structured as follows:

Chapter 2

In this chapter, we present a deterministic analysis method integrating traffic splitting and traffic multiplexing for worst-case performance analysis in WSNs. Moreover, two retransmission schemes are analytically evaluated: hop-by-hop retransmission and end-to-end retransmission. Most of the materials were published in:

- Huimin She, Zhonghai Lu, Axel Jantsch, Dian Zhou, Li-Rong Zheng, "Performance Analysis of Flow Based Traffic Splitting Strategy on Cluster-Mesh Sensor Networks", *International Journal of Distributed Sensor Networks*, Volume 2012, no-232937, 2012.

  Author's contributions: The author developed the algorithm, implemented the experiments and wrote the manuscript.

- Huimin She, Zhonghai Lu, Axel Jantsch, Li-Rong Zheng and Dian Zhou, "Deterministic Worst-case Performance Analysis for Wireless Sensor Networks," *in Proc. of the 2008 International conference on Wireless Communications and Mobile Computing Conference (IWCMC'08)*, Crete Island, Greece, August, 2008.

  Author's contributions: The author developed the algorithm, implemented the experiments and wrote the manuscript.

- Huimin She, Zhonghai Lu, Axel Jantsch, Li-Rong Zheng, Dian Zhou, "Analytical Evaluation of Retransmission Schemes in Wireless Sensor Networks", *IEEE 69th Vehicular Technology Conference (VTC2009-Spring)*, Spain, April 2009.

  Author's contributions: The author generated the ideas, developed the algorithm, implemented the experiments and wrote the manuscript.

Chapter 3

This chapter summarizes our research on applying stochastic network calculus for wireless channel modeling, and bandwidth estimation through backlog measurement. We propose a network calculus based approach for Quality of Service (QoS) analysis of wireless channels subject to Rayleigh fading. Furthermore, a

detailed mechanism for bandwidth estimation of random wireless channels based on stochastic network calculus is developed. The bandwidth is derived from the measurement of statistical backlogs based on probe packet trains. The theoretic foundation and the detailed step-by-step procedure of the estimation method are also presented.

Part of the results were published in:

- Huimin She, Zhonghai Lu, Axel Jantsch, Dian Zhou, Li-Rong Zheng, "Modeling and Analysis of Rayleigh Fading Channels using Stochastic Network Calculus", *the IEEE Wireless Communication and Networking Conference (WCNC'2011)*, Mexico, April 2011.

  Author's contributions: The author contributed with the idea, solution, experiments and wrote the manuscript.

- Huimin She, Zhonghai Lu, Axel Jantsch, Li-Rong Zheng, "Estimation of Statistical Bandwidth through Backlog Measurement", *Workshop on Network Calculus (WoNeCa2012), in conjunction with MMB&DFT 2012*, Germany, March 2012.

  Author's contributions: The author developed the algorithms, conducted the experiments and wrote the manuscript.

Chapter 4

This chapter contains the research on coverage analysis and deployment mechanisms in WSNs. A probabilistic event coverage analysis method is proposed for evaluating the coverage performance of heterogeneous sensor networks with randomly deployed sensors and stochastic event occurrences. Moreover, we introduce techniques for analyzing the coverage, network lifetime and cost both in random deployed and regular sensor networks.

Part of the results were published in:

- Huimin She, Zhonghai Lu, Axel Jantsch, Dian Zhou, Li-Rong Zheng, "Stochastic Coverage in Event-Driven Sensor Networks", *the 22nd IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'2011)*, Canada, September 2011.

  Author's contributions: The author contributed with the idea, solution, experiments and wrote the manuscript.

  An extended version of the conference paper is submitted to:

- Huimin She, Zhonghai, Axel Jantsch, Li-Rong Zheng, "Probabilistic Event Coverage in Heterogeneous Sensor Networks", submitted to ACM Transactions on Sensor Networks, 2012.

  Author's contributions: The author contributed with the idea, solution, experiments and wrote the manuscript.

- Huimin She, Zhonghai, Axel Jantsch, "System-Level Evaluation of Sensor Networks Deployment Strategies: Coverage Lifetime and Cost", *the 8th International Wireless Communications and Mobile Computing Conference (IWCMC'2012)*, Cyprus, August 2012.

  Author's contributions: The author contributed with the idea, solution, experiments and wrote the manuscript.

Chapter 5

This chapter summarizes the thesis and proposes several open topics for future investigation.

Chapter 2

Deterministic Performance Analysis

In this chapter, we present a deterministic analysis method integrating traffic splitting and traffic multiplexing for worst-case performance analysis in WSNs. Moreover, two retransmission schemes are analytically evaluated: hop-by-hop retransmission and end-to-end retransmission. The work in this chapter is mainly based on [83, 84, 85, 90, 86].

2.1 Introduction

A typical sensor network consists of a larger number of sensor nodes that are capable of sensing the environment and forwarding their observation values to a fusion center (sink) through multi-hop wireless links. Thus, the traffic pattern in sensor networks is usually in a many-to-one manner. The nodes near the sink may need to forward more data and thus consume more energy and buffer than the nodes far away. Consequently, the distributions of energy and buffer requirements in WSNs might be extremely uneven. Unfortunately, energy supplies and buffers are limited and expensive resources in typical WSNs, since sensor nodes are usually made as tiny devices with limited buffers and equipped with batteries that may not be convenient or economical for replacement. One way to address these challenges is applying traffic splitting strategies which have been adopted by many researchers for load balancing in communication networks [97, 66]. With traffic splitting, a main flow is divided into several subflows and forwarded to the destination through different routing paths. By distributing traffic over the network, the overall network load balance can be improved. It is shown in [109] that the spare capacity can

be reduced and thus the overall performance of the system can be improved by splitting traffic across multiple disjoint paths.

One popular application of WSNs is real-time monitoring and tracking, such as logistic chain tracking [114] and health-care application [40]. In such kind of applications, it is crucial to ensure sensor data delivered to the sink within time constraints so that appropriate actions can be made. In order to design a WSN with predictable delay, backlog and energy consumptions, formal performance analysis is desired for analyzing a sensor network before its actual deployment. While simulation running based methods can offer high accuracy, it can be very time-consuming and tedious to find the worst-case performance. Each simulation may take considerable time and evaluates only a single network configuration, traffic pattern, and load point. Hence, formal methods are desired to dimension sensor networks in an analytical way rather than case-by-case simulations. Starting with the seminal work by Cruz [25, 26], *network calculus* has been developed as a powerful tool for the performance analysis of networked system [11]. In contrast to queueing theory, network calculus deals with performance bounds, such as worst-case delay and backlog bounds, rather than average values. It has been applied to sensor networks by many researchers recently [81, 51, 82, 49].

Data transmission in WSNs is unreliable due to several factors such as fading, shadowing and multi-path effects of radio propagation. One of the most common approaches for enhancing transmission reliability is retransmission [71, 99, 69]. Park *et al.* [71] propose a scalable framework for reliable downstream data delivery using a *Wait-for-First-Packet (WFP)* pulse. In [99], Wan *et al.* propose a reliable transport protocol called PSFQ *(Pump Slowly and Fetch Quickly)*. These two protocols are typical examples that make use of hop-by-hop retransmissions. In [69], Pai *et al.* present an adaptive retransmission mechanism which allows a fusion center to select the sensors to retransmit their local information according to the reliability of the received information. This protocol belongs to end-to-end retransmission.

In this chapter, an analytical method for the performance analysis of WSNs based on deterministic network calculus is presented. Section 2.4 contains the analysis of traffic splitting strategies for mesh sensor networks. In Section 2.5, a method to analyze the retransmission schemes is presented.

2.2 Related Work

In general packet switching networks, network calculus provides methods to deterministically reason about timing properties and resource requirements. Based

on the powerful abstraction of *arrival curve* for traffic flows and *service curve* for network elements, it allows computing the worst-case delay and backlog bounds. Systematic introduction of network calculus can be found in books [11, 17].

Network calculus has been extremely successful for ensuring performance bounds when it is applied to Asynchronous Transfer Mode (ATM), Internet, and other networks. It has been recently extended and applied for performance analysis and resource dimensioning of WSNs by several researchers. In [81], Schmitt *et al.* firstly applied network calculus to sensor network and proposed a generic framework for performance analysis of WSNs with various traffic patterns. They further extended the general framework to incorporate computational resources besides the communication aspects of WSNs [82]. In [51], Anis *et al.* proposed a methodology for the modeling and worst-case dimensioning of cluster-tree sensor networks. They derived plug-and-play expressions for the end-to-end delay bounds, buffering and bandwidth requirements as a function of the WSN cluster-tree and traffic characteristics. Lenzini *et al.* [55] proposed a method for deriving tight end-to-end *least upper delay bounds* in sink-tree networks. The least upper delay bound is defined as the minimum value of the upper delay bound. In [49], the authors presented a method for computing the worst-case delays, buffering and bandwidth requirements while assuming that the sink node can be mobile.

Traffic splitting strategies have several common features with multi-path routing protocols. There have been plenty of research works on multi-path routing and traffic splitting for sensor networks [36, 62, 61, 116, 56, 67]. The authors in [36] proposed a multi-path routing scheme that finds several disjoint paths. In this scheme, the source node or an intermediate node chooses one path from the available paths to deliver the data to sink based on the performance requirements such as delay and throughput. An energy efficient multi-path routing protocol for WSNs with multiple sinks is presented in [62]. The path construction is implemented by the source node sending route messages to its neighbors. Traffic is distributed over the multiple paths according to a load balancing algorithm. The results show that the proposed scheme results in a higher energy efficiency. In [61], authors proposed an N-to-1 multi-path routing protocol, in which nodes are arranged in a spanning tree. Multi-paths are constructed by traversing the tree. The multi-path scheme is a combination of end-to-end multi-path traffic dispersion and per-hop alternate path salvaging. Zou *et al.* [116] studied the interplay between data aggregation and flow splitting in WSNs and proposed a flow-based scheme. The flows are preserved until the aggregation point. The aggregated data is splitted into multiple flows on the rest of the path to the destination. The results show that the scheme can balance energy consumption and therefore prolong the lifetime of WSNs. In [56], the authors

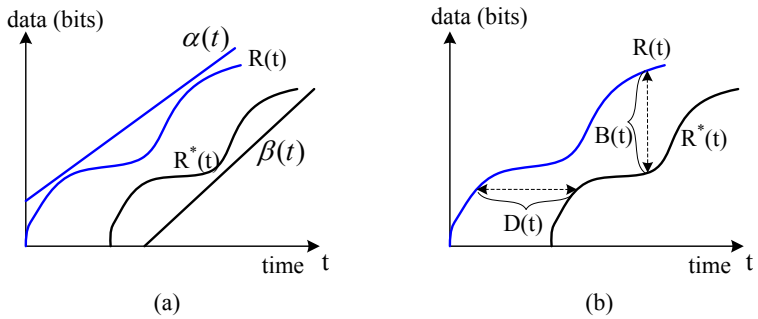


Figure 2.1. (a): arrive curve and service curve; (b): delay bound and backlog bound

investigated a joint coding/routing optimization of network costs and capacity in WSNs. By combining the link rate allocation and network coding-based multi-path routing, the total energy consumption of encoding power, transmission power and reception power can be reduced. A back-pressure collection protocol (BCP) for sensor networks is presented in [67]. In this protocol, routing and forwarding decisions are made based on a per-packet basis. By using ETX optimization and floating LIFO queues, BCP is capable of improving throughput and delivery performance under static and dynamic settings, respectively.

2.3 Basics of Deterministic Network Calculus

Network calculus contains two parts, which are deterministic network calculus [25, 26, 11, 1] and stochastic network calculus [44]. In this chapter, the analysis is based on deterministic network calculus. In general, network calculus is a min-plus system theory for deterministic queuing systems [11]. It has been developed for deterministic queuing, allowing to derive deterministic guarantees on throughput and delay, and bounds on buffer sizes. Specifically, network calculus analyzes the worst-case behavior of a network rather than the average behavior.

Next, some basic definitions and notations are provided. Detailed definitions and descriptions can be found in [11].

Definition 2.1 *Min-Plus Convolution*

Let $f(t)$ and $g(t)$ be *wide-sense increasing* functions [1] defined over real number $t \geq 0$, and $f(0) = g(0) = 0$. Then their convolution under min-plus algebra is

[1] A function *h* is wide-sense increasing if and only if $h(s) \leq h(t)$ for all $s \leq t$.

defined as:

$$(f \otimes g)(t) = \inf_{0 \le s \le t} \{f(t-s) + g(s)\} \tag{2.1}$$

An arrival process can be modeled by its cumulative traffic $R(t)$, which is defined as the number of bits that arrive in the time interval $[0, t]$. In particular, $R(0) = 0$ and $R(t)$ is wide-sense increasing. Similarly, the output function $R^*(t)$ of a system is the number of bits that depart from the system in time interval $[0, t]$. In particular, $R^*(0) = 0$ and $R^*(t)$ is wide-sense increasing.

Definition 2.2 *Arrival curve*

Given a flow with input function $R(t)$, a wide-sense increasing function $\alpha(t)$ is an arrival curve for $R(t)$ if and only if (Figure 2.1),

$$\forall t \ge 0, s \ge 0 \text{ and } s \le t : R(t) - R(t-s) \le \alpha(s) \tag{2.2}$$

One of the most commonly used arrival curve is the *affine arrival curve* [11], which is defined as:

$$\alpha(t) = \rho \cdot t + \sigma \tag{2.3}$$

where σ and ρ represent the burst tolerance (in units of data) and the rate (in units of data per unit time), respectively. The average data rate ρ gives an indication of the expected traffic volume in a given period of time. And burstiness σ describes the maximum traffic above the average rate during any time interval.

Definition 2.3 *Service Curve*

Assume that a server process is able to process $S(t)$ bits of input data until time t. Then, a wide-sense increasing function $\beta(t)$ is a service curve for $S(t)$ if and only if (Figure 2.1),

$$\forall t \ge 0, 0 \le s \le t : S(t) - S(t-s) \ge \beta(s) \tag{2.4}$$

One of the most commonly used service curve is the *rate-latency* service curve [11], which is defined as:

$$\beta(t) = R[t - T]^+ \tag{2.5}$$

where R represents the service data rate and T represents the maximum service delay. If $x \ge 0$, notation $[x]^+ = x$; otherwise it equals to 0.

With the arrival curve and service curve, the following theorems can be derived based on the network calculus theory. The detailed descriptions and proofs of these theorems can be found in [11].

Theorem 2.1 *Delay bound*

Assume a traffic flow $R(t)$, constrained by arrival curve $\alpha(t)$, traverses a system that provides a service curve $\beta(t)$. At any time t, the virtual delay $D(t)$ satisfies,

$$D(t) \leq \sup_{t \geq 0}\{\inf_{\tau \geq 0}\{\tau : \alpha(t) \leq \beta(t+\tau)\}\} \tag{2.6}$$

The delay bound defines the maximum delay that would be experienced by a bit arriving at time t. Graphically, the delay bound is the maximum horizontal deviation between $\alpha(t)$ and $\beta(t)$ (Figure 2.1).

Theorem 2.2 *Backlog Bound*

Assume a traffic flow $R(t)$, constrained by arrival curve $\alpha(t)$, traverses a system that provides a service curve $\beta(t)$. The backlog $B(t)$ for all t satisfies,

$$B(t) \leq \sup_{t \geq 0}\{\alpha(t) - \beta(t)\} \tag{2.7}$$

The backlog is the amount of bits that are held inside the sensor node. The required buffer size of a sensor node is determined by the maximum backlog. Graphically, the backlog bound is the maximum vertical deviation between $\alpha(t)$ and $\beta(t)$ (Figure 2.1).

Theorem 2.3 *Output bound*

Assume a traffic flow $R(t)$, constrained by arrival curve $\alpha(t)$, traverses a system that provides a service curve $\beta(t)$. The output flow is constrained by the following arrival curve:

$$\alpha^*(t) = \sup_{s \geq 0}\{\alpha(t+s) - \beta(s)\} \tag{2.8}$$

Theorem 2.4 *Concatenation*

Assume a flow sequentially traverses two systems which offer a service curve of β_1 and β_2, respectively. Then the concatenation of the two systems offers the flow the service curve $\beta(t)$, which is defined by,

$$\beta(t) = (\beta_1 \otimes \beta_2)(t) = \inf_{0 \le s \le t} \{\beta_1(t-s) + \beta_2(s)\} \tag{2.9}$$

where $\otimes$ represents *min-plus convolution* (Definition 2.1). If β_1 and β_2 are rate-latency service curves, i.e. $\beta_1(t) = R_1[t - T_1]^+$ and $\beta_2(t) = R_2[t - T_2]^+$, then $\beta_1 \otimes \beta_2 = R^*[t - T^*]^+$, where $R^* = min(R_1, R_2)$ and $T^* = T_1 + T_2$.

Theorem 2.5 *Aggregate Multiplexing*

Consider a node multiplexing two flows 1 and 2 in *First-In-First-Out* (FIFO) order. Assume that the node provides a guaranteed service curve $\beta(t)$ to the aggregate of the two flows and flow 2 is constrained by an arrival curve $\alpha_2(t)$. Then, for any $\theta \ge 0$ and $t \ge \theta$, flow 1 is guaranteed with the following service curve,

$$\beta_1^\theta(t) = [\beta(t) - \alpha_2(t - \theta)]^+ \tag{2.10}$$

This is also called leftover service, since it is the leftover capability of the node after serving flow 2.

2.4 Analysis of Traffic Splitting Scheme

In this section, based on network calculus, we propose a flow based traffic splitting strategy and an analytical method for worst-case performance analysis on cluster-mesh sensor networks. The traffic splitting strategy is useful in balancing network load and power consumption. Aiming to evaluate the worst-case performance in terms of end-to-end least upper delay bound, least upper backlog bound and power consumption, a splitting model is built for a single node analysis and an analytical method is proposed for the network analysis. Through an example, it is shown that the performance analysis method is able to derive closed-form formulas of these bounds. The numerical results indicate that the backlog and power consumption can be balanced by applying the traffic splitting strategy. In addition, simulations are performed to validate the performance bounds of our analytical method. The results show that their tightnesses are satisfactory.

2.4.1 System Model

This section presents system models, including the cluster-mesh network topology and power consumption model.

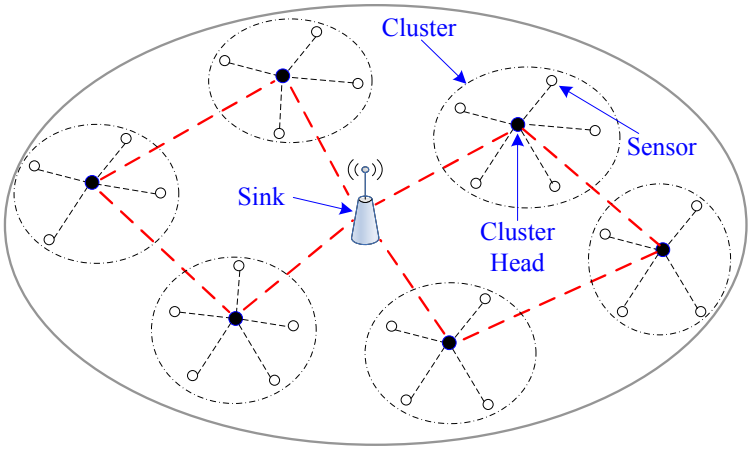


Figure 2.2. A cluster-mesh sensor network

A WSN may consist of a large number of sensors that are densely deployed either inside the phenomenon of interest or close to it. These sensors can be organized in various topologies, such as mesh and cluster based topologies. The mesh networking has advantages like supporting path diversity which enables better balance on traffic load and energy consumption [9]. The cluster based topologies are also quite suitable for WSNs with demanding requirements in terms of Quality of Service (QoS) support and real-time communications [51]. Considering these aspects, we adopt the *cluster-mesh* topology that merges advantages of mesh and cluster [54, 70]. It is a two-layered architecture with the mesh defining a backbone that consists of a set of *cluster heads* (CH). A cluster is formed by grouping a number of sensors within a geographic neighborhood. We define the network composed by cluster heads and the sink as the layer-1 network, and the network inside a cluster as the layer-2 network.

In summary, the cluster-mesh network contains three types of nodes: *sink*, *cluster head* and sensor. Like in most sensor networks, the sink is responsible for controlling the network and collecting data from all the other nodes. A cluster head and multiple sensors form a cluster. In order to reduce the cost and complexity, sensors do not communicate with each other and data generated by them is collected by their cluster head and delivered to the sink through neighbor cluster heads. For simplicity and conciseness, we consider cluster heads static and they do not sense the environment nor generate input data. However, this assumption can be easily relaxed and the subsequent analysis is straightforward. In the mesh network composed by cluster heads and the sink, links are considered bidirectional.

Fig. 2.2 shows an example of the cluster-mesh topology. A cluster-mesh network is a mesh network where each cluster head and its connected sensors form it own logical cluster. The layer-1 network can be modeled as a direct graph G($\mathcal{N}$,$\mathcal{L}$), where $\mathcal{N}$ is the set of all sensor nodes and the sink and $\mathcal{L}$ is the set of all direct links in the network. In this paper, our work concentrates on analyzing the layer-1 network.

In most types of sensor nodes, the energy consumption is mainly contributed by the transmitter, receiver and computation module [75]. The application scenarios of sensor networks for fresh food monitoring in warehouses are considered. In this scenario, sensors may perform tasks and send packets periodically. Consequently, the power consumption of the computation module can be considered as nearly constant denoted by p^c. Let ϵ^r denote the energy consumption of the receiver electronics for receiving one bit data. In practical applications, the power consumption of the receiving is usually stable [75]. So ϵ^r can be considered as a constant. According to the results in [74], the energy required to transmit a given amount of data is a convex and monotonically increasing function of the transmission rate, i.e., the energy per bit can be expressed as (Fig. 2.3),

$$\epsilon^t = \frac{N_0}{RG}\left(2^{\frac{R}{\eta W}} - 1\right) \tag{2.11}$$

where R is the transmission rate, W is the channel bandwidth, G is the channel gain, N_0 is the noise power, and $\eta \in (0,1)$ is the probability that the information can be reliably transmitted at a given transmission rate [2]. A simplified model for channel gain is adopted, i.e., $G = G_0(d/d_0)$ (Section 2.6 of [37]), where θ is the path loss exponent $(2 \leq \theta \leq 4)$, d is the distance between the transmitter and receiver, d_0 is a reference distance where the reference channel gain G_0 is measured.

Therefore, the total power consumption P_i of a node i can be expressed as,

$$\begin{aligned} P_i &= \epsilon^r \sum_{k\in N_{in}(i)} \rho_{ki} + \sum_{j\in N_{out}(i)} \epsilon^t_{ij}\rho_{ij} + p^c \\ &= \epsilon^r \sum_{k\in N_{in}(i)} \rho_{ki} + \sum_{j\in N_{out}(i)} \frac{\rho_{ij}N_0}{R_{ij}G_{ij}}\left(2^{\frac{R_{ij}}{\eta W}} - 1\right) + p^c \end{aligned} \tag{2.12}$$

where ρ_{ki} and ρ_{ij} denote the data rates on link ki and ij, respectively, R_{ij} denotes the transmission rate of node i sending data to node j, $G_{ij} = G_0(d_{ij}/d_0)^{-\theta}$ denotes the channel gain between node i and j, $N_{in}(i)$ denotes the set of nodes which

[2] In an optimal channel coding scheme, a transmission rate $R = \eta C$ can be guaranteed for any $0 < \eta < 1$, where $C = W\log_2(1 + GP/N_0)$ is the Shannon capacity [74].

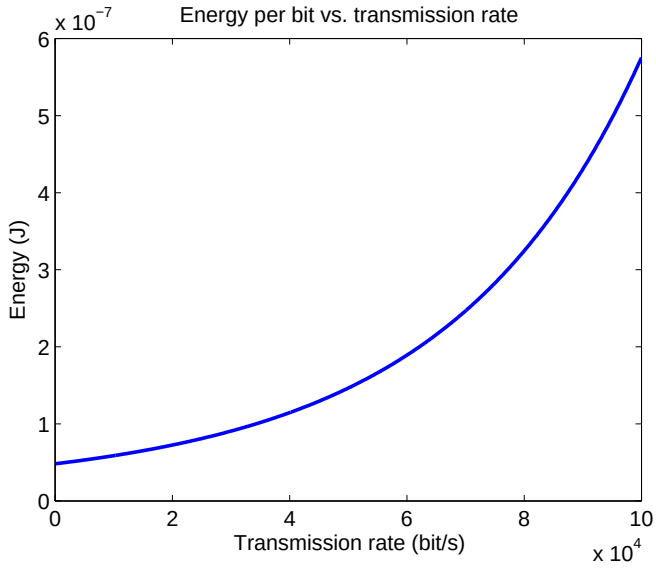


Figure 2.3. Energy per bit vs. transmission rate [74]: $W = 20\,kHz$, $N_0 = -100\,dB$, $G_0 = -50\,dB$, $d = 5\,m$, $\theta = 3$.

are the direct sources of incoming data flows of node i; $N_{out}(i)$ denotes the set of nodes which are the direct destinations of output data flows of node i.

As stated in the previous section, sensor nodes inside a cluster generate input data and then send them to their cluster head. A traffic *flow* is defined as an infinite stream of data from a source to a destination. Following network calculus, the input flow at a cluster head is modeled using its cumulative traffic $R(t)$, defined as the number of bits coming from the flow in time interval $[0, t]$. Furthermore, we use a wide-sense increasing function $\alpha(t)$ to constrain this cumulative traffic flow $R(t)$, defined as

$$R(t) - R(s) \leq \alpha(t - s); \quad \forall t \geq 0, t \geq s \tag{2.13}$$

where $\alpha(t)$ is called the arrival curve of the input flow $R(t)$ [11]. Affine arrival curve is one of the most commonly used arrival curves, which has been adopted in many works [81, 51, 82, 49]. The application scenario of this work is real-time monitoring, the sensor nodes sense the environment and send packets periodically to the cluster heads. Therefore, the affine arrival curve model can be used to abstract the input traffic of cluster heads, defined as $\alpha(t) = \gamma_{\sigma,\rho} = \rho \cdot t + \sigma$, where σ and ρ represent the burst tolerance (in bits) and the average data rate (in $bits/s$),

respectively. Fig. 2.4-(a) shows examples of a periodic cumulative flow $R(t)$ and an affine arrival curve $\alpha(t)$.

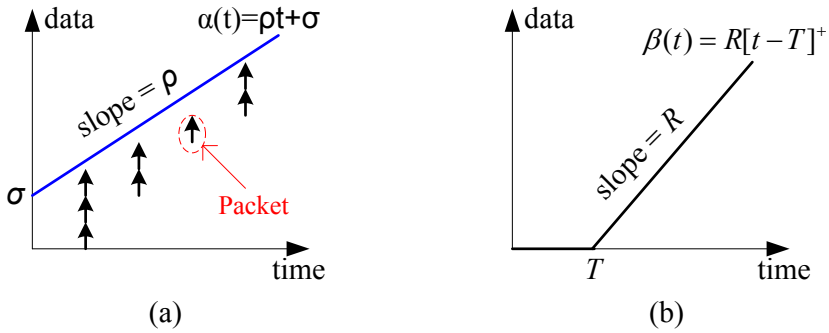


Figure 2.4. (a) An affine arrival curve: the arrows show the packet generation process; (b) A rate-latency service curve.

Service curve is an abstraction to model the processing capability of a node, depending on link layer characteristics, such as transmission rate, channel characteristics, and packet scheduling. The node and the channel together are modeled as a network element which provides a service curve β^s to the input flows. If the node forwards packets with the rate R ($bits/s$) and delays packets for T (s) at maximum due to scheduling and queuing, it can be modeled by a *rate-latency* service curve [11] that consists of two components: a *rate service curve* and a *latency service curve*. The rate-latency service curve can be formally defined as $\beta(t) = \chi_{R,T} = R[t-T]^+$, where notation $[x]^+$ denotes $\max\{0, x\}$.

In wireless networks, data transmission over wireless channels is usually unreliable due to their inherent uncertainties. The actual transmission rate and success probability are influenced by the transmission power, path loss, noise power and interference. In spite of these uncertainties, deterministic network calculus can still be useful in modeling wireless networks by making reasonable assumptions and abstractions. First, the uncertainties in some applications of WSNs are low. An example scenario for which our framework suits well is the process monitoring and tracking in logistics systems [114]. Second, the link unreliability and data loss rate can be mitigated by applying high transmission powers, especially for the cases with small distances between a transmitter and a receiver. Third, the interference between adjacent nodes can be alleviated by using appropriate MAC layer protocols. There are plenty of research works on designing TDMA-based link protocols which can create collision-free slot schedules [33, 79].

Based on these assumptions about link reliability and interference, the service

capability of a node can be abstracted and approximated by a deterministic service curve with the idea of effective transmission rate. From information theory, the Shannon capacity of a wireless channel can be expressed as, $C = W\log_2(1 + GP/N_0)$, where W, G, N_0 are the same as those in Eq. (2.11). The service rate is defined as the rate that the information can be reliably transmitted. With an optimal or suboptimal channel coding scheme, a service rate of $R = \eta \cdot C$ can be guaranteed for any $0 < \eta < 1$ [74]. R defines a lower bound on the transmission rate. Therefore, the rate-latency service curve $\beta(t) = \chi_{R,T} = R[t - T]^+$ can be applied to model the service capability of a wireless channel, where T denotes the maximum possible processing/queueing delay.

Given the arrival curve and service curve of a node, the least upper delay bound, least upper backlog bound and output bounds can be derived according to network calculus [11]. The least upper backlog bound is defined as the minimum value of the upper backlog bound. Consider a node i provides a service curve β_i^s to the input flow which is constrained by an arrival curve α_i. According to network calculus , the least upper delay bound of the flow can be computed by

$$D_i = h(\alpha_i, \beta_i^s) = \sup_{t\geq 0}\left\{\inf_{\tau\geq 0}\{\tau : \alpha_i(t) \leq \beta_i^s(t+\tau)\}\right\}. \tag{2.14}$$

Moreover, the least upper backlog bound of node i can be calculated by

$$B_i = v(\alpha_i, \beta_i^s) = \sup_{t\geq 0}\{\alpha_i(t) - \beta_i^s(t)\}. \tag{2.15}$$

Additionally, the arrival curve of the departure flow can be derived by

$$\tilde{\alpha}_i = \alpha_i \oslash \beta_i^s = \sup_{\tau\geq 0}\{\alpha_i(t+\tau) - \beta_i^s(\tau)\}. \tag{2.16}$$

2.4.2 Analysis

In this section, the splitting and multiplexing models are introduced. Then, the formal performance analysis procedure is presented. After that, the scope and assumptions of the analysis approach are discussed.

The Splitting Model

To analyze the splitting strategy, we build a splitting model that identifies the relations of input, output, delay and backlog for a single node. Without losing generality, a main flow is split into two subflows. The node that f_1 traverses is abstracted as the combination of a buffer plus a *splitter* depicted in Fig. 2.5-(a).

Consider that the node performs a weighted proportional splitting scheme, in which the main flow is split according to the configured weights, ϕ_i for subflow i. In each round, the splitter will try to forward ϕ_i packets to output link i before moving to the next one. The values of ϕ_i can be set either according to a pre-defined rule or randomly. Increasing the value of ϕ_i can result in increased packets to output link i. By adjusting ϕ_i, the amount of traffic over each link can be controlled. If the service rate is R bits/s, the maximum length of a round is consequently equal to $\sum_i \phi_i l/R$ seconds and the time for packets of subflow i to be forwarded within a round is bounded by $\phi_i l/R$ seconds, where l is the packet length. In the weighted proportional splitting scheme, the worst case appears when the packets of a subflow just misses its turn in the current round. Consequently, it will have to wait for its turn at the next round. In the worst case, packets of the subflow i have to wait up to $\sum_{i \neq j} \phi_j l/R$ seconds to be served.

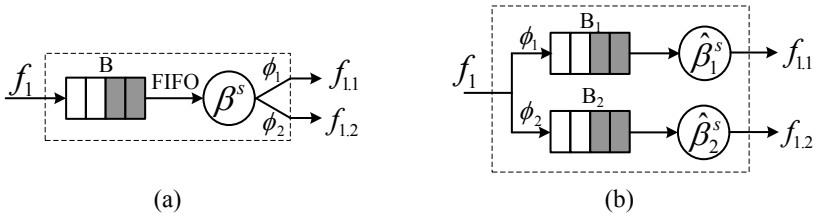


Figure 2.5. (a) The main flow f_1 is split into two subflows $f_{1.1}$ and $f_{1.2}$; (b) The equivalent model.

Consider a main flow f_1 that is upper constrained by arrival curve $\alpha_1 = \gamma_{\sigma_1,\rho_1}$, be split into two subflows $f_{1.1}$ and $f_{1.2}$ according to weights ϕ_1 and ϕ_2, where σ_1 and ρ_1 denote the burstiness and average data rate of f_1, respectively. Burstiness is defined as the amount of data inputted/outputted to/from a system or a node at one time. Consequently, it should be equal or bigger than the packet size l. Let $\alpha_{1.1} = \gamma_{\sigma_{1.1},\rho_{1.1}}$ and $\alpha_{1.2} = \gamma_{\sigma_{1.2},\rho_{1.2}}$ denote the arrival curves of $f_{1.1}$ and $f_{1.2}$, respectively. Then,

$$\rho_{1.1} = \frac{\phi_1}{\phi_1 + \phi_2}\rho_1, \qquad \sigma_{1.1} = \max\left(\left\lceil \frac{\phi_1\sigma_1}{(\phi_1+\phi_2)l} \right\rceil \cdot l, l\right) \tag{2.17}$$

$$\rho_{1.2} = \frac{\phi_2}{\phi_1 + \phi_2}\rho_1, \qquad \sigma_{1.2} = \max\left(\left\lceil \frac{\phi_2\sigma_1}{(\phi_1+\phi_2)l} \right\rceil \cdot l, l\right) \tag{2.18}$$

where $\lceil \cdot \rceil$ denotes the minimum integer equal to or bigger than the number inside.

Assume that the splitter provides a service curve $\beta^s = \chi_{R,T}$. Since the splitter serves one subflow at one time, the service rate for each subflow also equals R. Therefore, the equivalent service curve for subflow $f_{1.1}$ can be derived by

$$\widehat{\beta}_1^s = \beta^s \otimes \delta_{\frac{\phi_2 l}{R}} = \chi_{R,T+\frac{\phi_2 l}{R}}. \tag{2.19}$$

Analogously, the equivalent service curve for $f_{2.2}$ is

$$\widehat{\beta}_2^s = \beta^s \otimes \delta_{\frac{\phi_1 l}{R}} = \chi_{R,T+\frac{\phi_1 l}{R}}. \tag{2.20}$$

Furthermore, the equivalent bounds on backlogs can be calculated by

$$B_1 = \sigma_{1.1} + \frac{\phi_1\rho_1}{\phi_1+\phi_2}\left(T_1 + \frac{\phi_2 l}{R}\right), \quad B_2 = \sigma_{1.2} + \frac{\phi_2\rho_1}{\phi_1+\phi_2}\left(T_1 + \frac{\phi_1 l}{R}\right) \tag{2.21}$$

Therefore, the least upper bound of the total backlog is computed by

$$B = B_1 + B_2 = \sigma_1 + \rho_1\left[T + \frac{2\phi_1\phi_2 l}{R(\phi_1+\phi_2)}\right]. \tag{2.22}$$

The least upper delay bounds consist of three parts: the processing time, the time to serve input burstiness, and the scheduling delay. Let $D_{1.1}$ and $D_{1.2}$ denote the delay bounds of subflow $f_{1.1}$ and $f_{1.2}$, respectively. They can be computed by

$$D_{1.1} = T + \frac{\sigma_{1.1}}{R} + \frac{\phi_2 l}{R} \quad D_{1.2} = T + \frac{\sigma_{1.2}}{R} + \frac{\phi_1 l}{R}. \tag{2.23}$$

Furthermore, the departure arrival curves of $f_{1.1}$ and $f_{1.2}$ can be derived by

$$\begin{aligned}\tilde{\alpha}_{1.1} &= \frac{\phi_1\rho_1}{\phi_1+\phi_2}t + \sigma_{1.1} + \frac{\phi_1}{\phi_1+\phi_2}(\rho_1 T + \frac{\rho_1\phi_2 l}{R}) \\ \tilde{\alpha}_{1.2} &= \frac{\phi_2\rho_1}{\phi_1+\phi_2}t + \sigma_{1.2} + \frac{\phi_2}{\phi_1+\phi_2}(\rho_1 T + \frac{\rho_1\phi_1 l}{R}).\end{aligned} \tag{2.24}$$

The Multiplexing Model

In order to analyze resource sharing when multiple input flows share the bandwidth of a link at a node, a multiplexing model is proposed. This model should be used for analyzing a network with various traffic flowing scenarios.

Without loss of generality, let us consider a node serve two flows f_1 and f_2 in the FIFO order as shown by Fig. 2.6-(a). And its equivalent model is drawn in Fig. 2.6-(b). Let the node provide a service curve β^s to the aggregating flows, and f_1

and f_2 have α_1 and α_2 as arrive curves, respectively. We define $\widehat{\beta}_1^s = \kappa(\beta^s, \alpha_2)$ as the *equivalent service curve* [55] provided to flow f_1, where $\kappa(\cdot,\cdot)$ is an operator to compute the equivalent service and τ is an intermediate argument. Thus, the departure arrival curve of f_1 can be derived by $\tilde{\alpha}_1 = \alpha_1 \oslash \kappa(\beta^s, \alpha_2)$, and its least upper delay bound is computed by $h(\alpha_1, \kappa(\beta^s, \alpha_2))$, and the least upper backlog bound of the node is $v(\alpha_{\{1,2\}}, \beta^s)$, where $\alpha_{\{1,2\}}$ denotes the arrival curve of the aggregating flow $f_{\{1,2\}}$. Similarly, the equivalent service curve provides to flow f_2 can be derived by $\widehat{\beta}_2^s = \kappa(\beta^s, \alpha_1)$, and its delay and backlog bounds, and departure arrival curve can be derived accordingly.

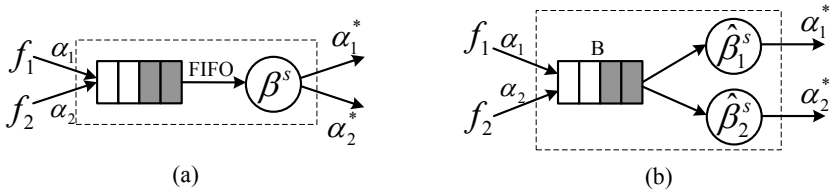


Figure 2.6. (a) A node serves two input flows; (b) The equivalent model.

An example is given to show how to compute $\kappa(\cdot,\cdot)$. Let $\beta^s(t) = \chi_{R,T} = R[t-T]^+$, and $\alpha_2(t) = \gamma_{\sigma_2,\rho_2}(t) = \rho_2 t + \sigma_2$, then applying Corollary 4.5 in [55], the equivalent service curve for f_1 can be calculated by

$$\widehat{\beta}_1^s = \kappa(\beta^s, \alpha_2) = \gamma_{R\cdot\tau, R-\rho_2} \otimes \delta_{T+\frac{\sigma_2}{R}+\tau} \quad (\tau \geq 0) \tag{2.25}$$

where τ is an intermediate argument for calculating the least upper delay bound, and $\delta_T(t) = +\infty$ for $t > T$, and 0 otherwise.

The least upper delay bound of f_1 is calculated by

$$D_1 = h(\alpha_1, \widehat{\beta}_1^s) = \inf_{\tau \geq 0} \left\{ T + \frac{\sigma_2}{R} + \frac{\sigma_1 - R\tau}{R - \rho_2} + \tau \right\}. \tag{2.26}$$

Furthermore, the least upper backlog bound of the node can be derived by

$$B = \sigma_1 + \sigma_2 + (\rho_1 + \rho_2)T. \tag{2.27}$$

Additionally, the arrival curve of the departure flow of f_1 is computed by

$$\tilde{\alpha}_1 = \alpha_1 \oslash \widehat{\beta}_1^s = \rho_1 t + \sigma_1 + \rho_1 \left(T + \frac{\sigma_2}{R} + \tau \right), \tag{2.28}$$

where τ is the same as the value obtained in equation (2.26).

The Splitting Based Performance Analysis Procedure

There have been several research works on the traffic splitting strategy in packet networks [66, 109], due to its efficiency in load balancing. In the flow based splitting strategy, a traffic flow is split into multiple subflows at its source node, and these subflows are forwarded to the sink through different routing paths. The source node decides how the subflows are split. Given the traffic patterns, service models, the routing protocols and the splitting strategy, the general performance analysis procedure is shown as follows:

Step 1: Based on the traffic pattern, routing protocols, and the traffic splitting strategy, construct a performance analysis model that converts the original network into an equivalent network.

Step 2: Derive the input and departure arrival curves of all nodes in the network based on network calculus.

Step 3: Derive the end-to-end equivalent service curves for the subflows; and then compute the end-to-end least upper delay bound using $D = h(\alpha, \hat{\beta})$, where α denotes the input arrival curve and $\hat{\beta}$ denotes the end-to-end equivalent service curve.

Step 4: Using the results in Step 2, compute the least upper backlog bound of each node $B_s = v(\sum_{i \in \mathcal{I}(s)} \alpha_i^s, \beta^s)$, where $\mathcal{I}(s)$ represents the set of input flows of node s.

Step 5: Compute the power consumption of a node s by $P_s = \epsilon^r \sum_{i \in N_{in}(s)} \rho_{is} + \sum_{j \in N_{out}(s)} \epsilon_{sj}^t \rho_{sj} + p_s^c$, where $N_{in}(i)$ and $N_{out}(i)$ denote the set of nodes which are the direct sources of incoming data flows and the direct destinations of output data flows of node i, respectively.

2.4.3 Discussions

Flow Based Splitting vs. Multipath Routing

Basically, a traffic splitting process consists of two stages: establishing multiple routing paths and allocating traffic on each path according to the splitting strategy. Multi-path routing is a technique exploiting routing diversity by using multiple source-destination pairs. It has been receiving plenty of research attentions [36, 29, 30, 18]. This work focuses on analyzing the performance of the splitting strategy rather than finding multi-path routes. So it is assumed that multiple paths have already been established between source nodes and the sink.

There are several common features between multi-path routing and flow based traffic splitting: first, both of them use multipaths to explore routing diversities. Second, both of them aim for balancing the load performance. Apart from these common features, there exist significant differences between them. In multi-path routing, routing decisions are made on a per-packet basis, i.e., each packet chooses its routing path and is forwarded to the destination individually. Multi-path routing is mainly used for improving network performance in terms of reliability and robustness [18, 30]. While in flow-based splitting strategy, the routing and forwarding is made on a per-flow basis. So it is capable of realizing a controlled splitting and providing quality of service. For example, if there are two paths between a source and a destination, a flow may be split half to one path and half to the other. So the delay guarantees can be reasoned about.

Flow Based Splitting vs. Node Based Splitting

According to the way that a traffic flow split, the traffic splitting strategy can be classified into two categories: flow based splitting and node based splitting.

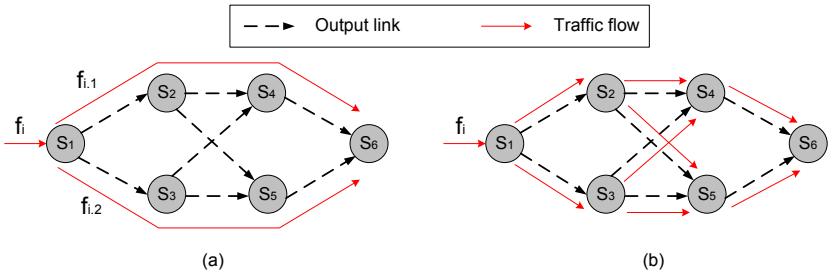


Figure 2.7. An example of splitting strategies. s_1 is the source node of the traffic flow and s_6 is the sink. (a) Flow based traffic splitting strategy; (b) Node based traffic splitting strategy.

In flow based splitting, the source node decides how the subflows are scheduled and split. The subflows can be identified after splitting. As shown in Fig. 2.7-(a), the traffic flow f_i is split into two subflows ($f_{i.1}$ and $f_{i.2}$) only at its source node s_1 and forwarded to the sink (s_6) through two routing paths: $\{s_1, s_2, s_4\}$ and $\{s_1, s_3, s_5\}$. In the node based traffic splitting strategy (Fig. 2.7-(b)), the traffic flow can be split at the intermediate nodes that have multiple output links (such as nodes s_1 s_2 s_3 in Fig. 2.7-(b)), and these nodes decide how their input flows are allocated to their output links. An example of node based routing strategy is the back-pressure-based routing protocol (BCP) [67]. Implementing a flow-based traffic splitting strategy is more complex than a node-based traffic splitting strategy in practice, but a flow-based traffic splitting strategy also has its own advantages. In the flow-based splitting strategy, the routing and forwarding decisions are made on a per-flow basis. So it is capable of realizing a controlled splitting and satisfying quality of service requirements.

How to Set Splitting Parameters?

This work mainly provides a framework for quality of service analysis of the flow based splitting strategy. Another research issue is on splitting parameter exploration, i.e., how to set splitting factors. One way is to utilize static network state information, such as link capacity and buffer length, to set splitting parameters. For example, in Fig. 2.7-(a), source node S_1 can select appropriate splitting factors based on the link capacity information of its downstream links. A larger amount of traffic can be allocated to the links with higher capacity. In [19], authors presented an explicit rate-based flow control scheme, in which each route ran a proportional max-min fair bandwidth sharing algorithm to divide the measure bandwidth among the passed flows. Alternatively, the splitting decision can be made based on dynamic network state information. Each node records its current or historic buffer lengths and the information is sent to the source node, so that the source node can select appropriate splitting parameters. For example, authors in [48] proposed a congestion-aware routing scheme which could redirect a certain amount of traffic to other paths under heavy traffic load. In this scheme, the congestion status information at each route is detected depending on the average MAC layer utilization and queue length. If congestion happens, traffic is split into other paths according to its services type.

2.4.4 An Analysis Example

In this section, we exemplify the general performance analysis and derive close-form formulas of delay bounds, backlog bounds and power consumptions

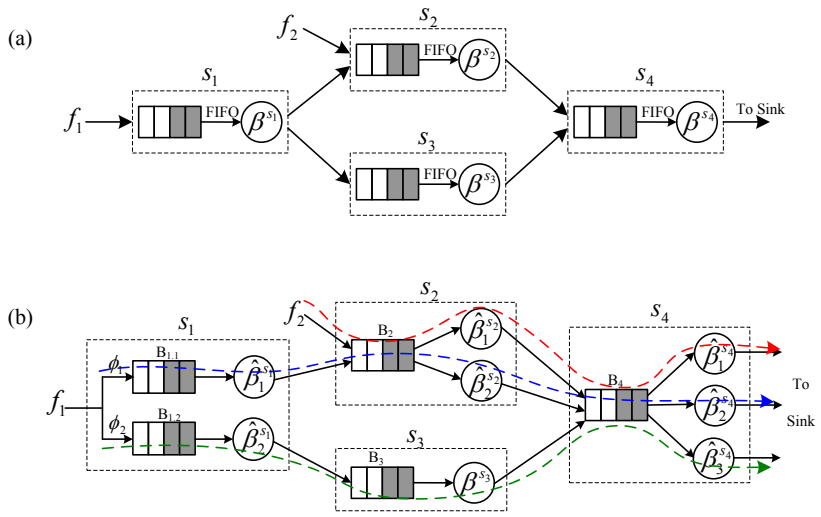


Figure 2.8. (a) A network analysis example: the main flow f_1 is split into two subflows $f_{1.1}$ and $f_{1.2}$. f_2 is the contention flow; (b) The equivalent analysis network: the blue, green, and red dashed lines show the routing path of flow $f_{1.1}$, $f_{1.2}$ and f_2 respectively.

under the conditions of affine arrival curve and rate-latency service curve. Consider that a network consists of four nodes as shown in Fig. 2.8. We define the *tagged main flow* as the main flow for which we shall derive the end-to-end delay bound. In this example, f_1 is chosen as the tagged main flow. Let f_1 be constrained by an arrival curve $\alpha_1 = \gamma_{\sigma_1,\rho_1}$, and split into two subflows $f_{1.1}$ and $f_{1.2}$, and respectively traverse two different routing paths $\mathcal{R}(f_{1.1}) = \{s_1, s_2, s_4\}$ and $\mathcal{R}(f_{1.2}) = \{s_1, s_3, s_4\}$ from the source node s_1 to the sink. ϕ_1 and ϕ_2 are the splitting weights of $f_{1.1}$ and $f_{1.2}$, respectively. f_2 is the contention flow that is modeled by an arrival curve $\alpha_2 = \gamma_{\sigma_2,\rho_2}$. Let $\alpha_j^{s_i}$ and $\tilde{\alpha}_j^{s_i}$ denote the input and departure arrival curves of flow j at node s_i, respectively. Next, we need to derive the end-to-end least upper delay bound for flow f_1 and the least upper backlog bound and power consumption for each node.

In order to compute the end-to-end least upper delay and the least upper backlog bounds, the input and departure arrival curves of each node need to be derived firstly.

Arrival Curves of Input and Output

According to the results of the splitting model, the arrival curves of departure flows at node s_1 can be derived as

$$\begin{aligned}\tilde{\alpha}_{1.1}^{s_1} &= \frac{\phi_1\rho_1}{\phi_1+\phi_2}t + \sigma_{1.1} + \frac{\phi_1}{\phi_1+\phi_2}\left(\rho_1 T_1 + \frac{\rho_1\phi_2 l}{R_1}\right)\\ \tilde{\alpha}_{1.2}^{s_1} &= \frac{\phi_2\rho_1}{\phi_1+\phi_2}t + \sigma_{1.2} + \frac{\phi_2}{\phi_2}\left(\sigma_1 + \rho_1 T_1 + \frac{\rho_1\phi_1 l}{R_1}\right).\end{aligned} \tag{2.29}$$

Node s_2 has two input flows, with arrival curves α_2 and $\alpha_{1.1}^{s_2} = \tilde{\alpha}_{1.1}^{s_1}$. According to the multiplexing analysis results (Section 2.4.2), the arrival curves of the departure flows of node s_2 is derived as follows,

$$\begin{aligned}\tilde{\alpha}_2^{s_2} &= \rho_2 t + \sigma_2 + \rho_2\left(T_2 + \frac{\sigma_{1.1}^{s_2}}{R_2} + \tau_1\right)\\ \tilde{\alpha}_{1.1}^{s_2} &= \tilde{\alpha}_{1.1}^{s_1} + \frac{\phi_1\rho_1}{\phi_1+\phi_2}\left(T_2 + \frac{\sigma_2}{R_2} + \tau_2\right).\end{aligned} \tag{2.30}$$

where $\sigma_{1.1}^{s_2} = \sigma_{1.1} + \frac{\phi_1}{\phi_1+\phi_2}(\rho_1 T_1 + \frac{\rho_1\phi_2 l}{R_1})$, and τ_1 and τ_2 are defined by

$$\begin{aligned}\arg\min_{\tau_1}\zeta_1(x) &= \{\tau_1 \geq 0 : \frac{\sigma_2 - R_2\tau_1}{R_2 - \phi_1\rho_1/(\phi_1+\phi_2)} + \tau_1\}\\ \arg\min_{\tau_2}\zeta_2(x) &= \{\tau_2 \geq 0 : \frac{\sigma_{1.1}^{s_2} - R_2\tau_2}{R_2 - \rho_2} + \tau_2\}.\end{aligned} \tag{2.31}$$

For node s_3, the arrival curve of its input flow is $\alpha_{1.2}^{s_3} = \tilde{\alpha}_{1.2}^{s_1}$, and it provides a service curve $\beta^{s_3} = \chi_{R_3,T_3}$. Consequently, the arrival curve of its departure flow is,

$$\tilde{\alpha}_{1.2}^{s_3} = \tilde{\alpha}_{1.2}^{s_1} + \frac{\phi_2\rho_1}{\phi_1+\phi_2}T_3. \tag{2.32}$$

According to the connection relations, we can get the arrival curves of three input flows at node s_4, which are $\alpha_2^{s_4} = \tilde{\alpha}_2^{s_2}$, $\alpha_{1.1}^{s_4} = \tilde{\alpha}_{1.1}^{s_2}$ and $\alpha_{1.2}^{s_4} = \tilde{\alpha}_{1.2}^{s_3}$. Since it is not necessary to compute the arrival curves of the departure flows at node s_4, the derivation is skipped.

The End-to-End Delay Bound

In order to compute the end-to-end delay bound, we first need to derive the service curve provided by individual nodes. Let $\hat{\beta}_k^{s_i}$ represent the equivalent service

curve provided by node s_i to its kth input flow. As shown in Fig. 2.8, node s_2 serves two flow $f_{1.1}$ and f_2. According to the multiplexing analysis in section 2.4.2, the equivalent service curve for $f_{1.1}$ at s_2 is $\hat{\beta}_2^{s_2} = \kappa(\beta^{s_2}, \alpha_2)$. Node s_4 serves three flows, and the equivalent service curve for $f_{1.1}$ is $\hat{\beta}_1^{s_4} = \kappa(\kappa(\beta^{s_4}, \alpha_2^{s_4}), \alpha_{1.2}^{s_4})$. Thus, the end-to-end equivalent service curve for $f_{1.1}$ can be derived by

$$\begin{aligned}
\beta_{1.1}^{e2e} &= \hat{\beta}_1^{s_1} \otimes \hat{\beta}_2^{s_2} \otimes \hat{\beta}_2^{s_4} \\
&= \beta^{s_1} \otimes \delta_{\frac{\phi_2 l}{R_1}} \otimes \kappa(\beta^{s_2}, \alpha_2) \otimes \kappa(\kappa(\beta^{s_4}, \alpha_2^{s_4}), \alpha_{1.2}^{s_4}) \\
&= \chi_{R_1,T_1} \otimes \delta_{\frac{\phi_2 l}{R_1}} \otimes \gamma_{R_2\tau_2,R_2-\rho_2} \otimes \delta_{T_2+\frac{\sigma_2}{R_2}+\tau_2} \otimes \gamma_{R_4\tau_3,R_4-\rho_4^{'}} \otimes \delta_{T_4+\frac{\sigma_4^{'}}{R_4}+\tau_3} \\
&= \chi_{R_1,T_1} \otimes \delta_{\frac{\phi_2 l}{R_1}+T_2+\frac{\sigma_2}{R_2}+\tau_2+T_4+\frac{\sigma_4^{'}}{R_4}+\tau_3} \otimes \gamma_{R_2\tau_2,R_2-\rho_2} \otimes \gamma_{R_4\tau_3,R_4-\rho_4^{'}}
\end{aligned} \tag{2.33}$$

where τ_1 and τ_2 are the same as those in Eq. (2.30), τ_3 is calcuated by $\arg\min_{\tau_3}\{\tau_3 \geq 0 : \frac{\sigma_{1.1}^{s_4} - R_4\tau_3}{R_4-\rho_4^{'}} + \tau_3\}$, $\rho_4^{'} = \rho_2 + \phi_2\rho_1/(\phi_1+\phi_2)$ and

$$\sigma_4^{'} = \rho_2(T_2 + \frac{\sigma_{1.1}^{s_2}}{R_2} + \tau_1) + \sigma_{1.2} + \frac{\phi_2[\rho_1(T_1+T_3) + \frac{\rho_1\phi_1 l}{R_1}]}{\phi_1+\phi_2} + \sigma_2. \tag{2.34}$$

Analogously, the end-to-end equivalent service curve for $f_{1.2}$ can be derived by

$$\begin{aligned}
\beta_{1.2}^{e2e} &= \hat{\beta}_2^{s_1} \otimes \beta^{s_3} \otimes \hat{\beta}_3^{s_4} \\
&= \beta^{s_1} \otimes \delta_{\frac{\phi_1 l}{R_1}} \otimes \beta^{s_3} \otimes \kappa(\kappa(\beta^{s_4}, \alpha_2^{s_4}), \alpha_{1.1}^{s_4}) \\
&= \chi_{R_1,T_1} \otimes \delta_{\frac{\phi_1 l}{R_1}} \otimes \chi_{R_3,T_3} \otimes \gamma_{R_4\tau_4,R_4-\rho_4^{''}} \otimes \delta_{T_4+\frac{\sigma_4^{''}}{R_4}+\tau_4}
\end{aligned} \tag{2.35}$$

where τ_4 is defined by $\arg\min_{\tau_4}\{\tau_4 \geq 0 : \frac{\sigma_{1.2}^{s_4} - R_4\tau_4}{R_4-\rho_4^{''}} + \tau_4\}$, $\rho_4^{''} = (\rho_2 + \phi_1\rho_1/(\phi_1 + \phi_2))$ and

$$\sigma_4^{''} = \frac{\phi_1\rho_1}{\phi_1+\phi_2}(T_1 + T_2 + \frac{\sigma_2}{R_2} + \tau_2) + \frac{\rho_1\phi_1\phi_2 l}{R_1(\phi_1+\phi_2)} + \rho_2(T_2 + \frac{\sigma_{1.1}^{s_2}}{R_2} + \tau_1) + \sigma_{1.1} + \sigma_2 \tag{2.36}$$

where τ_1 and τ_2 are the same as those in Eq. (2.30).

After getting the end-to-end service curves, the least upper delay bounds of $f_{1.1}$ and $f_{1.2}$ can be respectively computed by $h(\alpha_{1.1}, \beta_{1.1}^{e2e})$ and $h(\alpha_{1.2}, \beta_{1.2}^{e2e})$,

$$\begin{aligned} h(\alpha_{1.1}, \beta_{1.1}^{e2e}) &= T_1 + T_2 + T_4 + \frac{\phi_2 l}{R_1} + \frac{\sigma_2}{R_2} + \frac{\sigma_4^{'}}{R_4} + \inf_{\substack{\tau_2 \geq 0 \\ \tau_3 \geq 0}} \left\{ \tau_2 + \tau_3 \right. \\ &\quad \left. + \left[\frac{\sigma_{1.1}}{R_1} \vee \frac{\sigma_{1.1} - R_2\tau_2}{R_2 - \rho_2} \vee \frac{\sigma_{1.1} - R_4\tau_3}{R_4 - \rho_4^{'}} \right] \right\} \\ h(\alpha_{1.2}, \beta_{1.2}^{e2e}) &= T_1 + T_3 + T_4 + \frac{\phi_1 l}{R_1} + \frac{\sigma_4^{''}}{R_4} \\ &\quad + \inf_{\tau_4 \geq 0} \left\{ \tau_4 + \left[\frac{\sigma_{1.2}}{R_1} \vee \frac{\sigma_{1.2}}{R_3} \vee \frac{\sigma_{1.2} - R_4\tau_4}{R_4 - \rho_4^{''}} \right] \right\} \end{aligned} \tag{2.37}$$

Hence, the end-to-end least upper delay bound for the flow f_1 equals the maximum of the delays of two subflows, namely,

$$D_{f_1} = \max \left\{ h(\alpha_{1.1}, \beta_{1.1}^{e2e}), h(\alpha_{1.2}, \beta_{1.2}^{e2e}) \right\}. \tag{2.38}$$

The Backlog Bound

Let B_{s_i} denote the backlog bound of node s_i $(i = 1, \cdots, 4)$. As we have already derived the arrival curves of input and output flows at each node, its least upper backlog bound can be calculated very easily. According to the result in Eq. (2.22), it shows

$$B_{s_1} = B_{1.1} + B_{1.2} = \sigma_1 + \rho_1 \left[T_1 + \frac{2\phi_1\phi_2 l}{R_1(\phi_1 + \phi_2)} \right]. \tag{2.39}$$

Node s_2 has two input flows α_2 and $\alpha_{1.1}^{s_2}$, so its least upper backlog bound is computed by

$$B_{s_2} = \sigma_{1.1} + \sigma_2 + \frac{\phi_1(\rho_1 T_1 + \frac{\rho_1\phi_2 l}{R_1})}{\phi_1 + \phi_2} + \left(\frac{\phi_1\rho_1}{\phi_1 + \phi_2} + \rho_2 \right) T_2. \tag{2.40}$$

Analogously, the least upper backlog bounds of node s_3 and s_4 can be derived by,

$$\begin{aligned} B_{s_3} &= \sigma_{1.2} + \frac{\phi_2}{\phi_1 + \phi_2} \left[\rho_1(T_1 + T_3) + \frac{\rho_1\phi_1 l}{R_1} \right] \\ B_{s_4} &= \rho_1(T_1 + T_4) + \sigma_1 + \rho_2 \left(T_2 + T_4 + \frac{\sigma_{1.1}^{s_2}}{R_2} + \tau_1 \right) + \sigma_2 \\ &\quad + \frac{2\rho_1 l\phi_1\phi_2}{R_1(\phi_1 + \phi_2)} + \frac{\phi_2\rho_1 T_3}{\phi_1 + \phi_2} + \frac{\phi_1\rho_1}{\phi_1 + \phi_2} \left(T_2 + \frac{\sigma_2}{R_2} + \tau_2 \right). \end{aligned} \tag{2.41}$$

Power Consumption

According to the power model, the total power consumption of a node is contributed by the radio transmitter, radio receiver, and computation electronics. Thus, the power consumptions of all the nodes can be computed by

$$
\begin{aligned}
P_{s_1} &= \epsilon^r \rho_1 + \epsilon^t_{s_1 s_2} \rho_{1.1} + \epsilon^t_{s_1 s_3} \rho_{1.2} + p^c \\
&= \rho_1 \left[\epsilon^r + \frac{\phi_1 N_0 (2^{\frac{R_{s_1 s_2}}{\eta W}} - 1)}{R_{s_1 s_2} G_{s_1 s_2} (\phi_1 + \phi_2)} + \frac{\phi_2 N_0 (2^{\frac{R_{s_1 s_3}}{\eta W}} - 1)}{R_{s_1 s_3} G_{s_1 s_3} (\phi_1 + \phi_2)} \right] + p^c \\
P_{s_2} &= \epsilon^r (\rho_{1.1} + \rho_2) + \epsilon^t_{s_2 s_4} (\rho^{s_4}_{1.1} + \rho^{s_4}_2) + p^c \\
&= \left[\epsilon^r + \frac{N_0 (2^{\frac{R_{s_2 s_4}}{\eta W}} - 1)}{R_{s_2 s_4} G_{s_2 s_4}} \right] \left(\frac{\phi_1 \rho_1}{\phi_1 + \phi_2} + \rho_2 \right) + p^c \\
P_{s_3} &= \left[\epsilon^r + \frac{N_0 (2^{\frac{R_{s_3 s_4}}{\eta W}} - 1)}{R_{s_3 s_4} G_{s_3 s_4}} \right] \frac{\phi_2 \rho_1}{\phi_1 + \phi_2} + p^c \\
P_{s_4} &= \left[\epsilon^r + \frac{N_0 (2^{\frac{R_{s_4 s_0}}{\eta W}} - 1)}{R_{s_4 s_0} G_{s_4 s_0}} \right] (\rho_1 + \rho_2) + p^c
\end{aligned}
\tag{2.42}
$$

where $d_{s_4 s_0}$ denotes the distance between node s_4 and the sink node s_0. Here, we assume that the link capacity can meet the requirements of the traffic bandwidth, i.e., the service rate is bigger than the sum of input data rates.

2.4.5 Performance Evaluation

To show benefits of the traffic splitting strategy and to validate the network calculus based performance analysis method, numerical results and simulations are presented under the scenario of a fresh food monitoring application. In the numerical results, the end-to-end least upper delay bounds, the least upper backlog bounds and power consumptions are compared under two scenarios: *general routing with no traffic splitting (NOS)* and *flow based splitting strategy (FBS)*. In the simulations, the results obtained by the analytical method are compared with the simulation results also under these two scenarios.

The numerical results are based on an application example of a real-time fresh food monitoring system deployed in a warehouse [114] [70]. As shown in Fig. 2.9, one sink and 9 cluster heads are uniformly distributed in a 20m$\times$10m warehouse. Each cluster head connects with 5 sensor nodes. The coordinates of cluster heads

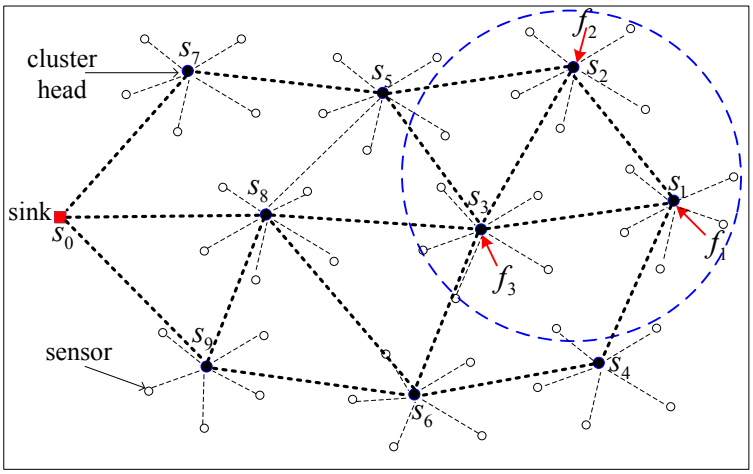


Figure 2.9. A cluster-mesh sensor network: s_0 is the sink. An event happens in the blue circle and three traffic flows f_1, f_2, f_3 are generated.

and sink (s_0) are: $s_0(0,0)$, $s_1(17.2,1.7)$, $s_2(14.1,5.5)$, $s_3(11,0.5)$, $s_4(14.8,-3.6)$, $s_5(8.3,4)$, $s_6(9.1,-4.4)$, $s_7(2.5,4.7)$, $s_8(4.2,0.8)$, $s_9(3.3,-3.6)$. We consider the application scenario of real-time monitoring, where sensor nodes periodically generate packets when there is a request.

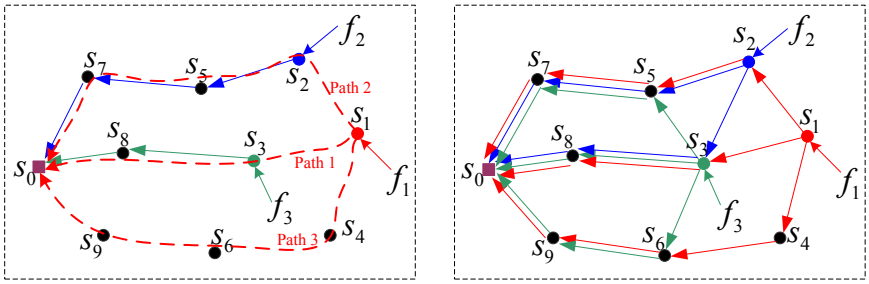
Numerical Results

Assume that sensor nodes in cluster $\mathcal{C}_1, \mathcal{C}_2$ and $\mathcal{C}_3$ are requested to send packets to the sink via cluster head s_1, s_2 and s_3, respectively. Consequently, there are three traffic flows in the network: f_1, f_2 and f_3. They are characterized by arrival curves $\alpha_1 = \sum_{n_j \in \mathcal{C}_1} \alpha_1^j$, $\alpha_2 = \sum_{n_j \in \mathcal{C}_2} \alpha_2^j$, $\alpha_3 = \sum_{n_j \in \mathcal{C}_3} \alpha_3^j$, where α_i^j $(i = 1,2,3)$ denotes the arrival curve of traffic generate by sensor node j in cluster $\mathcal{C}_i$. For periodic packets generation applications, the arrival process can be characterized by the affine arrival curve [81], i.e., $\alpha_1 = \gamma_{\sigma_1,\rho_1}$, $\alpha_2 = \gamma_{\sigma_2,\rho_2}$ and $\alpha_3 = \gamma_{\sigma_3,\rho_3}$. Moreover, assume cluster head s_i provides a rate-latency service curve $\beta^{s_i} = \chi_{R_i,T_i}$. f_1 is the tagged main flow, and f_2, f_3 are the contention flows. Other parameters are listed in Table 2.1. We shall derive the end-to-end least upper delay bound for f_1, the least upper backlog bounds and power consumptions of all nodes in two scenarios: NOS and FBS (Fig. 2.10).

From the power model in Eq. (2.11) and Fig. 2.3, the energy per bit is a monotonically increasing function of the transmission rate [74] if other parameters are

Table 2.1. System Parameters

System Parameter	Notation	Value	Unit
Packet length	l	400	bits
Computation power	p^c	10	uJ/s
Path loss factor	θ	3	-
Service delay	$T_i(i = 1 \cdots 9)$	0.1	s
Data rate of f_2	ρ_2	1.2	kbps
Data rate of f_3	ρ_3	1.6	kbps
Burstiness	$\sigma_1, \sigma_2, \sigma_3$	400	bits
Channel bandwidth	W	20	kHz



(a) (b)

Figure 2.10. (a) General routing with no splitting (NOS): the tagged main flow f_1 chooses one of path 1, 2 or 3, and the routing paths of f_2 and f_3 are shown by the blue and green line respectively; (b) Flow based splitting (FBS): all three flows are split as shown by the red, blue and green lines respectively.

fixed. Thus, it is better to use low transmission power for the sake of energy efficiency. On the other hand, the delay bound is a monotonically decreasing function of the service rate [3]. Therefore, there is a trade-off between energy consumption and delay. In order to study this trade-off, numerical experiments are implemented

[3]For example, given an arrival curve $\alpha(t) = \rho t + \sigma$, and service curve $\beta(t) = R[t - T]^+$, the delay bound is $T + \sigma/R$ when $R \geq \rho$.

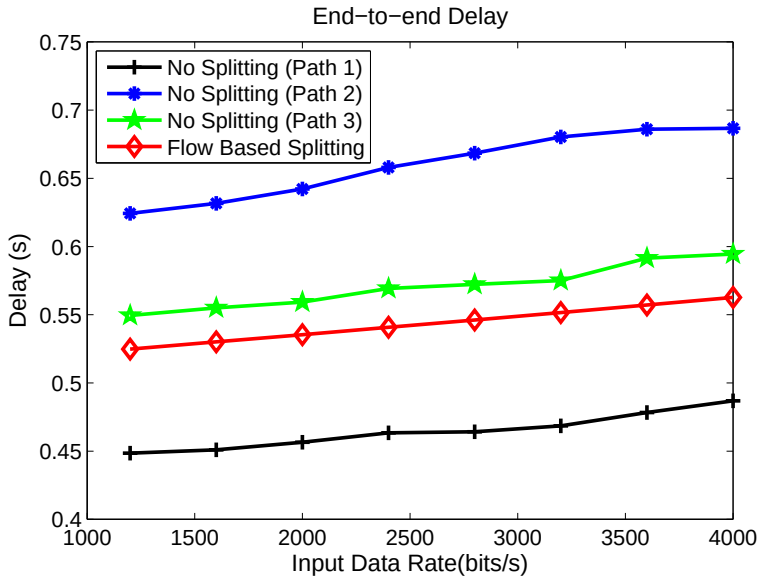


Figure 2.11. End-to-end least upper delay bound

in two scenarios: 1) Uniform service rate: the service rate of all cluster heads are the same and fixed; 2) Heterogeneous service rate: In order to guarantee a limited delay and backlog, the service rate should be equal to or bigger than the data arrival rate. So the service rate R is set to be equal to the arrival rate ρ.

The end-to-end least upper delay of f_1, the backlog bounds and power consumptions of all nodes are compared in two scenarios: NOS and FBS. In NOS, f_1 chooses one of the three paths: path 1-$\{s_1, s_3, s_8\}$, path 2-$\{s_1, s_4, s_6, s_9\}$, and path 3-$\{s_1, s_2, s_5, s_7\}$. In FBS, f_1 is evenly split into three subflows, and they are allocated on these three paths.

Uniform Service Rate

In the first numerical example, a fixed service rate is chosen $R_i = 9.6\ kbps$ $(i = 1, \cdots 9)$.

Fig. 2.11 shows the comparison of the end-to-end least upper delay bounds of the tagged main flow f_1 in NOS and FBS scenarios. In the NOS scenario, the end-to-end delays of f_1 going through three different routing paths are compared (Fig. 2.10-(a)). The input data rate of f_1 varies from $1.2\ kbps$ to $4\ kbps$. From this figure, we can see that the end-to-end delays in all scenarios increase with the input data rates. Furthermore, on average, the delays in FBS are 23.6% and 9.4%

less than those of path 2 and path 3 in NOS, respectively. And the delay in FBS is 12.4% bigger than those of path 1 in NOS. This is because path 1 is shorter than other two paths.

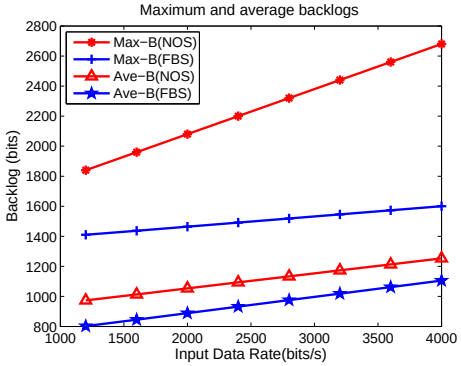


Figure 2.12. Least upper backlog bounds (In NOS: flow 1 chooses path 3).

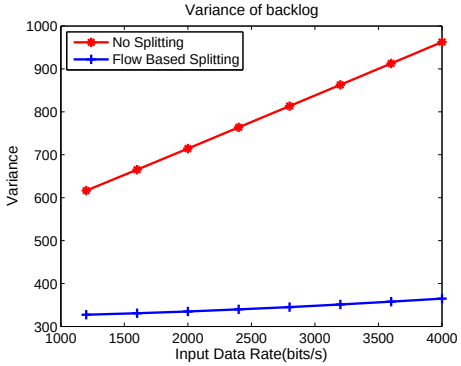


Figure 2.13. Variance of least upper backlog bounds (In NOS: flow 1 chooses path 3).

Fig. 2.12 shows the least upper and average backlog bounds in the FBS and NOS scenarios with input data rates vary, where 'Max-B(NOS)' means the maximum backlog in NOS which denotes the maximum value of backlogs among all nodes, and 'Ave-B(FBS)' means the average backlog in FBS which denotes the average value of backlogs over all nodes. From this figure, we can find that both the maximum and average backlogs in FBS are less than those in NOS. It indicates that the traffic splitting strategy can reduce backlogs. Moreover, the differences between the maximum backlogs of the two scenarios are much bigger than those of average backlogs. The average backlogs in FBS is 14.5% less than those in NOS on the average. While the maximum backlog in FBS is 23.4% less than that in NOS when the input data rate is 1.2 $kbps$, and the value increases to 40% when the input data rate is 4 $kbps$. We can also observe similar reduction in the variance of maximum backlogs (as shown in Fig. 2.13), where the variance of backlogs in NOS is much bigger than that in FBS. It means that in NOS some nodes have very small backlogs, but some nodes have very large backlogs. Since the buffer size of a sensor node is basically determined by the value of maximum backlog, larger backlog would brings higher hardware cost and bigger buffer. Therefore, applying the flow based splitting strategy can bring better load balance and thus reduce overall cost.

The maximum and average power consumptions of all nodes in NOS and FBS

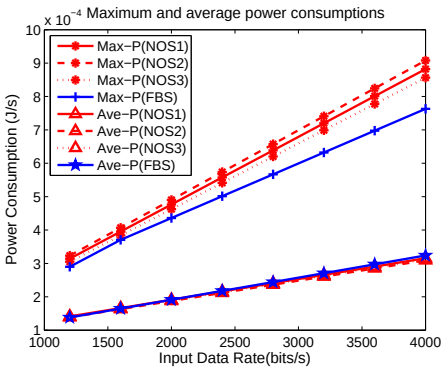


Figure 2.14. Power consumption (In NOSi: flow 1 chooses path i, where $i = 1, 2, 3$).

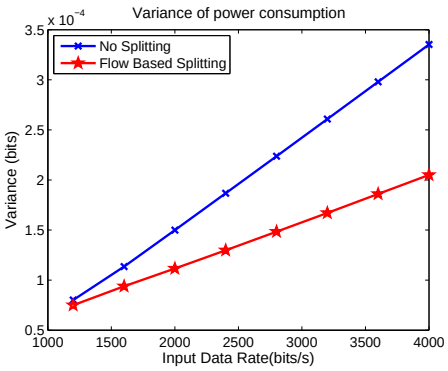


Figure 2.15. Variance of power consumption.

are shown in Fig. 2.14, where 'Max-P(NOS)' and 'Ave-B(NOS)' respectively denotes the maximum and average power consumptions in the NOS scenario. First, we see from the figure that all the power consumptions increase with the input data rates. Furthermore, when the data rate increases, the average power consumptions in the NOS and FBS are almost the same. However, the maximum power consumption in NOS increases much faster than that in FBS, with the maximum differences between FBS and NOS increasing from 0.8% to 12%. It indicates that the power consumptions of nodes are uneven in NOS. We can also see this from Fig. 2.15 showing the variance of power consumption of all nodes. From this figure, we can find the variance in NOS increases much faster than that in FBS. Usually, the lifetime of a WSN is determined by the first node exhausting its energy. Hence, the flow based splitting scheme can be used for balancing power consumption and consequently increasing the lifetime of the network.

Heterogeneous Service Rate

The data rate of f_1 varies from 1.2 to 4 $kbps$. The data rate of f_2 and f_3 are given in Table 2.1. The service rate of each node is equal to its arrival rate.

Being different from Fig. 2.11, the end-to-end delays of FBS are basically bigger than those of NOS in this case (Fig. 2.16). Moreover, when the input data rate increases, the end-to-end delay decreases. The reason is that the service rate increases with the input data rate. While the input burstiness is the same, so the delay would decrease.

Fig. 2.17 shows the comparison of backlog bounds in the FBS and NOS scenarios. From this figure, we can find that the average backlogs in FBS and NOS

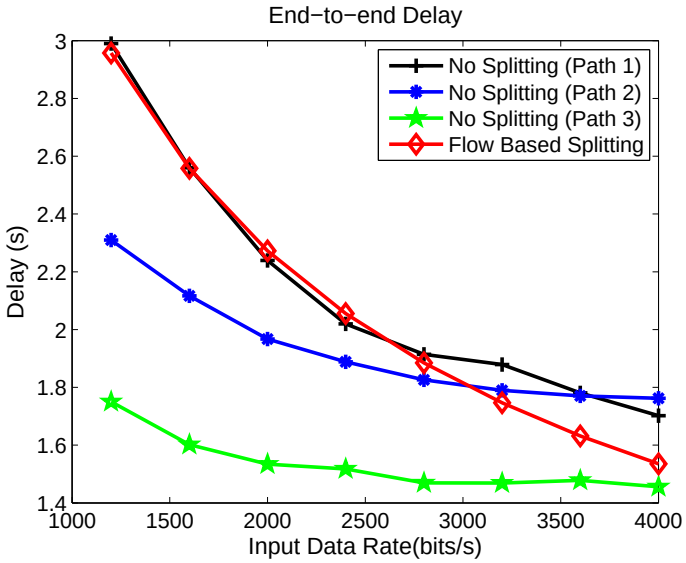


Figure 2.16. End-to-end delay.

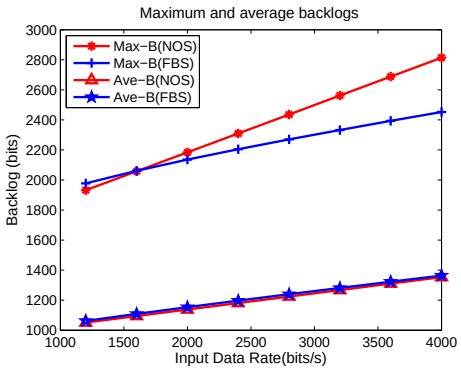


Figure 2.17. Least upper backlog bounds (In NOS: flow 1 chooses path 3).

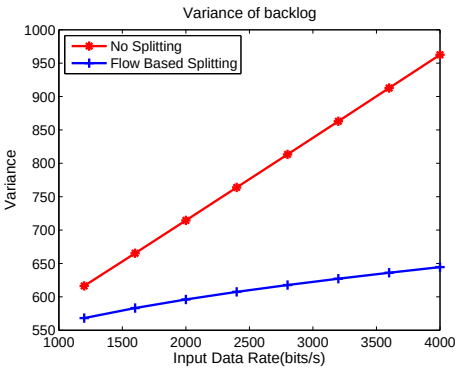


Figure 2.18. Variance of least upper backlog bounds (In NOS: flow 1 chooses path 3).

are almost the same. The maximum backlog in FBS is 2.4% bigger than that in NOS when the input data rate is $1.2\ kbps$. When the input data rate is increased, the maximu backlog in FBS becomes less than that in NOS. Similar to Fig. 2.13, from Fig. 2.18, we can see that the variance of backlogs in NOS is bigger than that in FBS. It also means that in NOS some nodes have very small backlogs, but some

nodes have very large backlogs.

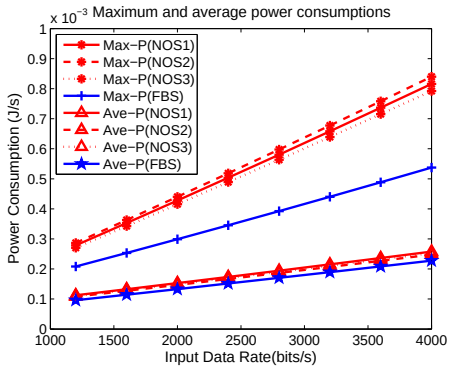


Figure 2.19. Power consumption (In NOSi: flow 1 chooses path i, where $i = 1, 2, 3$).

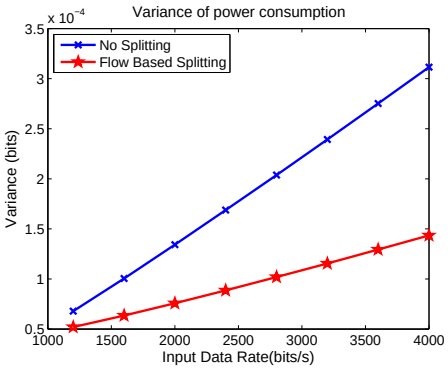


Figure 2.20. Variance of power consumption.

In the case of heterogeneous service rates, we see from the figure (Fig. 2.19) that all the power consumptions increase with the input data rates. Furthermore, the average power consumptions in the NOS are approximately 11.4% bigger than those in FBS. However, the maximum power consumption in NOS increases much faster than that in FBS, with the maximum differences between FBS and NOS increasing from 23.9% to 33%. The differences of maximum power consumption in this case are much bigger than those in uniform service rate case Fig. 2.14. We can also see this from Fig. 2.20 showing the variance of power consumption. From this figure, we can find the variance in NOS increases much faster than that in FBS.

From all those results and comparison, we can have the following conclusions: First, applying FBS strategy can balance traffic load and power consumption, so as to reduce overall system cost and increase the network lifetime. Second, there is a trade-off between power consumption and system performance. Under uniform service rate, the end-to-end delays of FBS are less than those of NOS in most cases, and the power consumptions of FBS are slightly less than those of NOS. While under heterogeneous service rates, the end-to-end delays of FBS are generally bigger than those of NOS, but the power consumptions of FBS are much less than those of NOS. It means that the decreasing of power consumption is obtained at the cost of increasing delay.

Comparison of End-to-end and Hop-by-hop Methods

As stated in [85], there are two ways to compute the end-to-end delay bound. The first method is to sum up the per-hop delay together. The second method is to

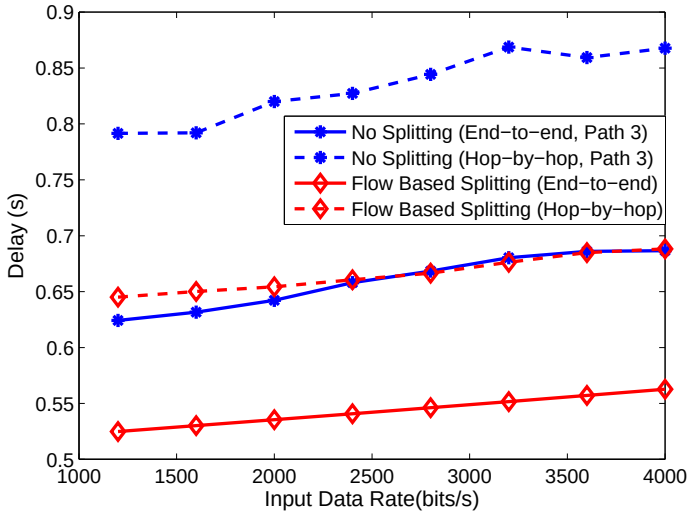


Figure 2.21. Compare the end-to-end delay computed by two methods.

derive an equivalent service curve for a given traffic flow. And then the end-to-end delay bound is calculated using the equivalent service curve. In [85], we use the first method (hop-by-hop). While in this thesis, we adopt the second method (end-to-end). Fig. 2.21 illustrates the comparison of these two methods in the scenario of FBS and NOS. On average, the hop-by-hop delay in NOS is 26.4% bigger than the end-to-end delay. And in FBS, the hop-by-hop delay is 22.5% bigger than the end-to-end delay. Therefore, the end-to-end method can achieve tighter bound than the hop-by-hop method.

Simulation Results

Since a simulation environment allows us to create a realistic sensor network behavior while still controllable, experiments are conducted in a simulation environment based on OMNeT++ 3.3 rather than in a field trial. The *tightness* is defined as the ratio of maximum simulation value divided by the analytical value.

In the simulation, a most common log-normal path loss model [77] is adopted. This model can provide more accurate multi-path channel models than Rayleigh and Nakagami models for indoor environments [110]. The simulation is also based on the application scenario shown in Fig. 2.9. Parameters used in simulations are the same as those in Table 2.1. Other parameters used are: $d_0 = 1\ m$, reference

channel gain $G_0 = 10^{-5}$ $(-50\ dB)$, noise power $N_0 = 10^{-10}$ $(-100\ dB)$, and the channel noise is subject to a Gaussian random variable with deviation 4. 50 simulation runs are conducted. In each run, the total simulation period is 25000 cycles and the source generates one packet every cycle. The packet generation rate is based on the pre-defined data rate. In order to bypass the initial non-stationary stage, the data of first 5000 cycles are omitted. In each run, the delay of every packet is recorded, and the value of backlog is recorded in every cycle. In order to compare the analytical results with the worst-case simulation results, the results of runs leading to maximum delay and maximum backlog are selected as the simulation results.

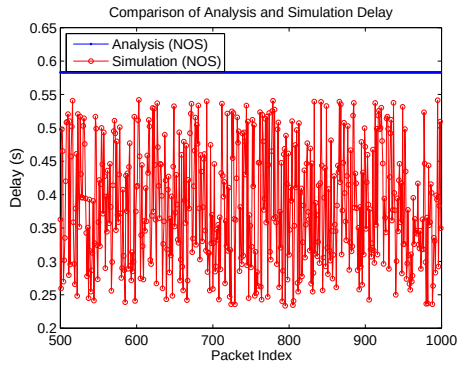


Figure 2.22. End-to-end delays in NOS.

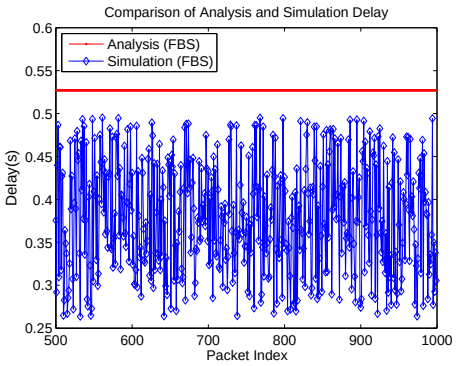


Figure 2.23. End-to-end delays in FBS.

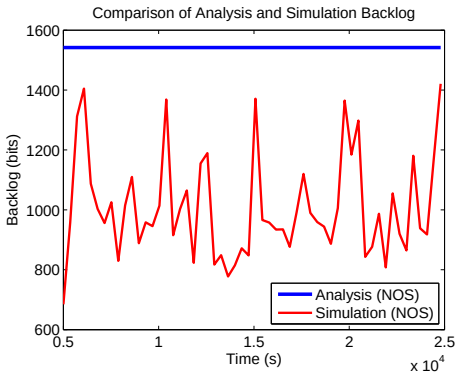


Figure 2.24. Nodes' backlogs in NOS.

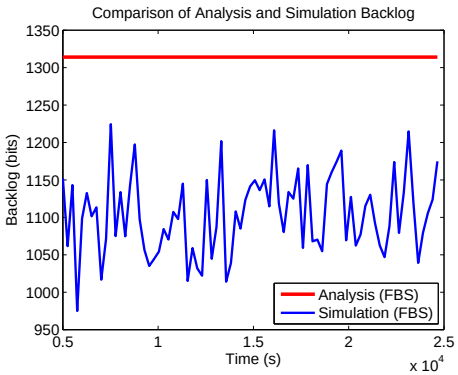


Figure 2.25. Nodes' backlogs in FBS.

Fig. 2.22 and Fig. 2.23 show the comparison of simulation results and analytical results of end-to-end delays of flow f_1 in the NOS and FBS scenarios, respectively. In NOS, the arrival rate of flow 1 is $2.8\ kbps$, and it selects path 3 as its routing path. In order to make the figure easy to read, we do not illustrate the delays of all packets, but extract 500 values from them. From these two figures, we observe that all the simulation values are bounded by the analytical results. And the tightness in NOS and FBS are 91% and 93.2%, respectively. This indicates our analysis performs well on bounding the end-to-end data delivery delay.

For the backlog analysis, node s_8 is chosen as the observation node. Its backlogs in different time points are recorded and compared as shown in Fig. 2.24 and Fig.2.25. In NOS, the arrival rate of flow 1 is $2.8\ kbps$, and it selects path 3 as its routing path. We can see that all the simulation values of backlogs are within the scopes of the analytical values. Moreover, the backlogs in the FBS scenario are less than those in NOS, which also indicates that the flow based splitting scheme can reduce the maximum backlogs by balancing traffic load over the network. Additionally, the tightness of the analytical results in NOS and FBS are 93.5% and 92.1%, respectively. In summary, the proposed analysis method is correct on deriving the backlog bound and the tightness is satisfactory.

2.5 Analysis of Retransmission Schemes

Many previous works have been done on reliable transport issues in experimental ways. However, there still lack analytical techniques to evaluate different reliable transport solutions. In [58], Liu *et al.* analyze the roles of packet retransmission and erasure coding in the reliable transport of WSNs by establishing the probability models. In this section, we propose analytical techniques to evaluate retransmission schemes in WSNs. We first introduce the traffic model, service model and energy model. Based on these models and network calculus, we analytically evaluate the maximum packet transfer delay and energy efficiency of two basic retransmission schemes, which are hop-by-hop retransmission and end-to-end retransmission. From the experiment results, the maximum delay and energy consumption of these two schemes are compared in several scenarios. Moreover, the analytical maximum delay is compared with the simulation results. With our method, appropriate retransmission scheme can be chosen based on different requirements and constraints.

2.5.1 System Model

To characterize the traffic generated by the sensor nodes, the arrival flow at a node is modeled using its cumulative traffic $R(t)$. An affine arrival curve $\alpha(t)$ is used to constrain the cumulative traffic flow, namely, $\alpha(t) = \rho \cdot t + \sigma$ (Definition 2.2 in Section 2.3). To model the processing capability of a node, the following service curve is used,

$$\beta(t) = C \cdot \frac{S}{T_s} \cdot [t - (T_s - S)]^+ \tag{2.43}$$

Following the energy model presented in [75], the energy consumption of a packet transmission between two nodes is abstracted in a similar way,

$$E = 2E_{start} + \frac{L}{R}(P_{tx} + P_{rx} + 2P_{cir}) \tag{2.44}$$

where E_{start} represents the energy for startup the radio; P_{tx} and P_{rx} represent the power consumption of the radio in transmission mode and receive mode, respectively; P_{cir} represents the power consumption of the electronic circuitry; L denotes the packet length in bits; and R denotes the transmission data rate. The energy consumption in the sleeping mode is ignored since it is much smaller than that for packet transmission or reception [75]. However, it is straightforward to extend our model to include the energy consumption in the sleep mode.

2.5.2 Analysis

There have been a lot of papers on designing retransmission schemes in WSNs [71] [69] [96]. These retransmission schemes can be classified into two basic categories, namely hop-by-hop retransmission and end-to-end retransmission (Figure 2.26).

Assume that there is a multi-hop path with n hops between a source node S and a destination node D. And there is an *automatic repeat request* (ARQ) mechanism running until a packet successfully arrives at the receiver. A packet is accepted only if every bit of the packet is received without error (for non-coded systems). Furthermore, assume an ideal MAC protocol where there is no interference and collision, so packet delivery failures are only due to channel errors. The packet error rate p_e can be computed by $p_e = 1 - (1 - p_b)^L$, where p_b is bit error rate (BER).

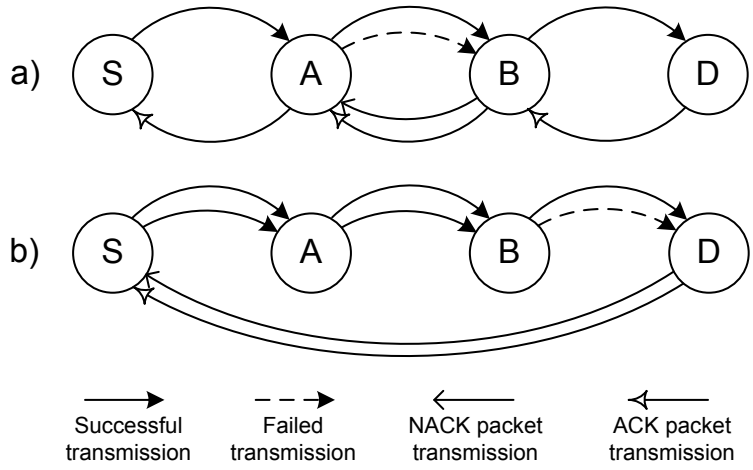


Figure 2.26. (a): Hop-by-hop retransmission; (b): End-to-end retransmission.

Hop-by-hop Retransmission

In hop-by-hop retransmission scheme, at every hop, the receiver checks the correctness of the packet and requests for a retransmission with an *negative-acknowledgment* (NACK) packet until a correct packet arrives. After that, an *acknowledgment* (ACK) packet is sent to the transmitter indicating a successful transmission. An example is shown in Figure 2.26 . If the first packet transmission fails between A and B, then B sends an NACK packet to A asking for a retransmission. After that, A retransmits the packet for a second time. B sends an ACK packet after successfully receiving the packet.

Let m_i denote the number of transmission trials at hop i, and p_i denote the packet error rate at hop i. Then, we shall derive the transmission delay and energy consumption.

Let L_a denote the length of an ACK and NACK packet. At the source node S, the arrival curve is expressed by $\alpha_1(t) = \rho_1 \cdot t + \sigma_1$. According to (2.43), the service curve at hop i $(1 \leq i \leq n)$ is expressed as follows,

$$\beta_i(t) = C \cdot \frac{S_i}{T_s} \cdot [t - (T_s - S_i)]^+ \tag{2.45}$$

where S_i denotes length of the slot assigned to link i. Since the input of current hop equals the output of previous hop, i.e. $\alpha_i(t) = \alpha_{i-1}^*(t)$ $(2 \leq i \leq n)$, the arrival curve of the traffic at the i^{th} $(1 \leq i \leq n)$ hop can be recursively derived based on

Theorem 2.3 (Section 2.3, Equation (2.8)),

$$\alpha_i^*(t) = \sup_{s \geq 0}\{\alpha_i(t+s) - \beta_i(s)\} = \alpha_i(t) + \rho_i \cdot (T_s - S_i) \tag{2.46}$$

Based on the theorems of determinsitic network calculus, the maximum delay at hop i can be derived as follows,

$$D_i = \sup_{t \geq 0}\{\inf_{\tau \geq 0}\{\tau : \alpha_i(t) \leq \beta_i(t+\tau)\}\} = \frac{\sigma_i T_s}{CS_i} + (T_s - S_i) \tag{2.47}$$

At each hop, the expected number of transmissions can be evaluated by $1/(1-p_i)$. Therefore, the expected maximum delay (D_{hbh}) of sending a packet from S to D can be calculated by summing up the delays at each hop,

$$D_{hbh} = \sum_{i=1}^{n} \frac{1}{1-p_i} D_i \tag{2.48}$$

The energy consumption is caused by data packets and ACK (NACK) packets. For simplicity, the energy consumption for decoding is ignored although it is straightforward to include it. According to the energy model, the energy consumption at the i^{th} hop can be calculated by,

$$E_i = 2E_{start}^i + \frac{L + L_a}{R}(P_{tx}^i + P_{rx}^i + 2P_{cir}^i) \tag{2.49}$$

Therefore, the total expected energy consumption E_{hbh} of transmitting a packet from S to D can be computed by,

$$E_{hbh} = \sum_{i=1}^{n} \frac{1}{1-p_i} E_i \tag{2.50}$$

End-to-end Retransmission

In end-to-end retransmission scheme, the intermediate nodes simply forward received packets to the next hop and do not check the correctness of the packets. When a packet arrives at the destination D, D checks the packet, and asks for a retransmission with an NACK packet directly to S if the packet is incorrect. Otherwise, it sends an ACK packet to S indicating a successful packet transmission. In this scheme, ACKs/NACKs can be sent to source nodes directly. This is reasonable since the sink node (i.e. the destination) is usually very powerful and it can reach every node in the network. See example in Figure 2.26.

Let p_i denote the packet error rate at hop i, and m denote the number of transmission trials. In this scheme, the retransmission is performed in an end-to-end manner, so an equivalent service curve can be derived for the whole link based on Theorem 2.4 (Section 2.3, Equation (2.9)) and Equation (2.45),

$$\beta_{e2e} = \beta_1 \otimes \beta_2 \otimes \cdots \otimes \beta_n = R_{e2e} \cdot (t - T_{e2e}) \tag{2.51}$$

where R_{e2e} and T_{e2e} can be calculated by,

$$R_{e2e} = \min_{1 \le i \le n} (C \cdot \frac{S_i}{T_s}), \quad T_{e2e} = \sum_{i=1}^{n} (T_s - S_i) \tag{2.52}$$

According to the traffic model, the arrival curve of the input flow at S is defined as: $\alpha_{in}(t) = \rho_{in} \cdot t + \sigma_{in}$. Based on Theorem 2.1 (Section 2.3, Equation (2.6)), the maximum delay D_{st} for one single transmission from S to D can be calculated by,

$$D_{st} = \sup_{t \ge 0} \{ \inf_{\tau \ge 0} \{ \tau : \alpha_{in}(t) \le \beta_{e2e}(t + \tau) \} \} = \frac{\sigma_{in}}{R_{e2e}} + T_{e2e} \tag{2.53}$$

In end-to-end retransmission, the total expected number of transmissions can be evaluated by $1/p_{st}$, where $p_{st} = \prod_{i=1}^{n}(1 - p_i)$. Then, the expected maximum delay D_{e2e} can be calculated by,

$$D_{e2e} = \frac{1}{p_{st}} D_{st} \tag{2.54}$$

In the end-to-end retransmission scheme, only the sink node needs to send ACK and NACK packets, other intermediate nodes simply forward data packets. According to the energy model, the energy consumption at the i^{th} hop can be calculated by,

$$E_i = 2E^i_{start} + \frac{L}{R}(P^i_{tx} + P^i_{rx} + 2P^i_{cir}) \tag{2.55}$$

Therefore, the total expected energy consumption E_{hbh} of transmitting a packet from S to D can be computed by,

$$E_{e2e} = \frac{1}{p_{st}} \left[\sum_{i=1}^{n} E_i + \frac{L_a}{R}(P^i_{tx} + P^i_{rx} + 2P^i_{cir}) \right] \tag{2.56}$$

In Equation (2.56), the first item computes the energy consumption for transmitting data packets, while the second item computes the energy for ACK and NACK packets transmissions.

$$E_{e2e} = \frac{1}{\prod_{i=1}^{n}(1 - p_i)} \left[\sum_{i=1}^{n} E_i + \frac{L_a}{R}(P^i_{tx} + P^i_{rx} + 2P^i_{cir}) \right]$$

2.5.3 Experiment Results

Experiment Setup

Table 2.2. Experimental Parameters

Parameter	Notation	Value	Unit
Tx power	P_{tx}	19.1	mW
Rx power	P_{rx}	14.6	mW
Circuit power	P_{cir}	12	mW
Start energy	E_{start}	1.0	uJ
Data pkt length	L	240	bits
ACK(NACK) pkt length	L_a	80	bits
Link capacity	C	19.2	kbps

In this section, the maximum transmission delay and energy consumption of hop-by-hop and end-to-end retransmission schemes are compared. The parameters used in experiments are shown in Table 2.2, which follow those used in [75] [24]. The link distance is randomly selected between $5\ m$ and $10\ m$, which is typical for most applications. Set the frame length T_s and slot length S to $0.2\ s$ and $0.01\ s$, respectively. The input data rate of end-to-end retransmission scheme $\rho_{in} = 30\ bps$, which corresponds to one packet in every eight seconds. The burstiness is set to $240\ bits$, which equals to the size of one packet. For hop-by-hop scheme, the number of ACK (NACK) packets are the same as data packets, so the data rate at the first hop $\rho_1 = (1 + L/L_a)\rho_{in}$.

Comparisons of Two Schemes

Figure 2.27 shows the comparisons of the maximum delay with required target success probability varying. From this figure, we can see that the maximum delays of both schemes increase as the target success probability increases. Further more, when the BER is low, i.e., $(1e-4)$, the maximum delay of end-to-end scheme is less than that of hop-by-hop scheme. But when the BER is high, i.e., $(1e-3)$, the hop-by-hop scheme has less delay. This indicates that when the BER is high, more trials of retransmissions are required by end-to-end scheme to achieve the same target success probability.

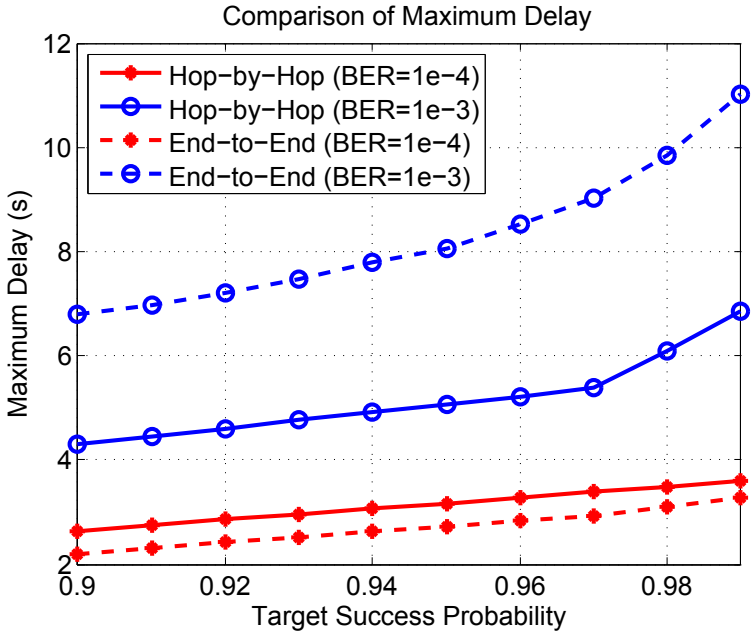


Figure 2.27. Compare the maximum transmission delay

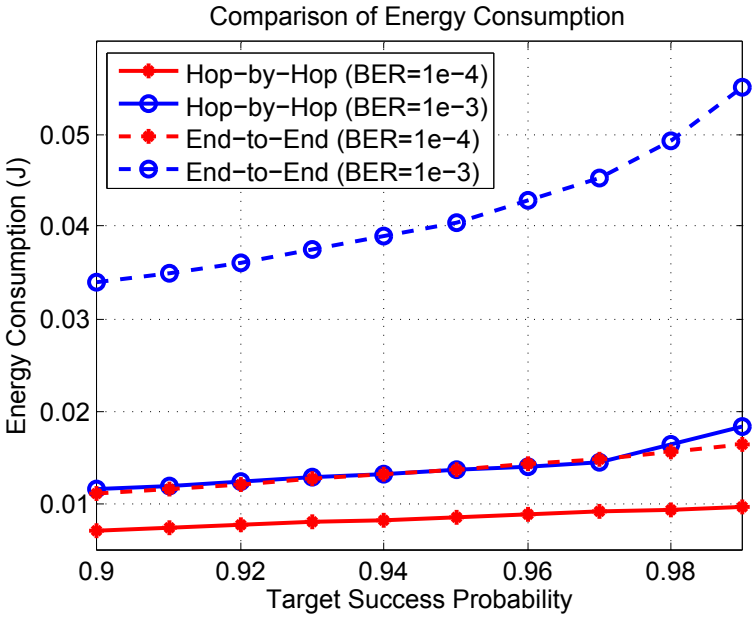


Figure 2.28. Compare the energy consumption

Figure 2.28 plots the energy consumption varies with the required target success probability. In this figure, we observe the end-to-end scheme consumes more energy than the hop-by-hop scheme both for low and high BERs. Especially, when the BER is $(1e-3)$, the energy consumption of the end-to-end scheme is much higher than the other one (BER is $1e-4$). The reason is that, in the end-to-end scheme, erroneous packets are not dropped until their destination is reached. And the energy for delivering these packets is wasted. When BER is higher, there are more erroneous packets which would lead to higher energy consumption.

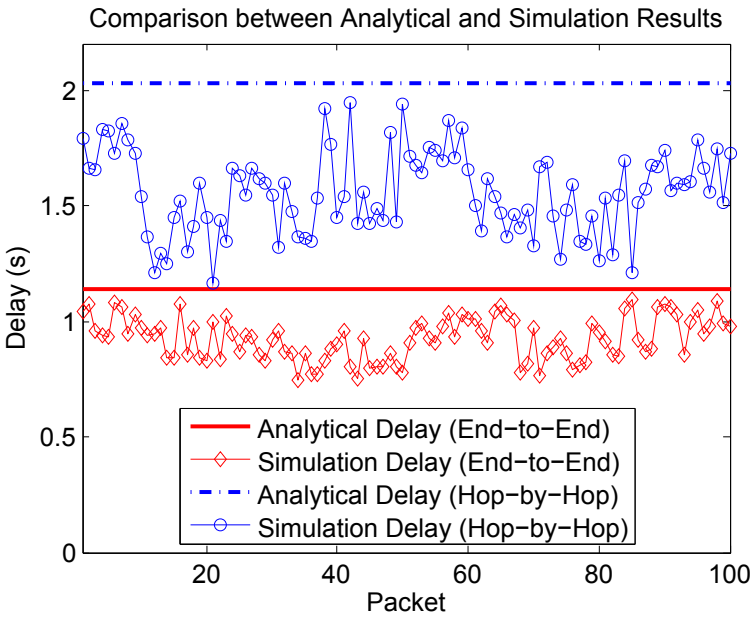


Figure 2.29. Compare analytical maximum transmission delay with the simulation results

To validate results of delay bound, the analytical results are compared with the simulation results in a chain scenario. The simulations are performed using Omnet++ 3.3. The path length is 4 hops and BER is $5e-4$. Other parameters are shown in table 2.2. From Figure 2.29, we observe that all the simulation values are within the scopes of the analytical results. This indicates network calculus performs well on bounding the packet transfer delay. For end-to-end and hop-by-hop retransmission scheme, the analytical delays are 4.3% and 5.8% bigger than simulated maximum delays, respectively.

2.6 Summary

Dimensioning timing-critical sensor networks requires formal methods to ensure performance and cost in any conditions. In Section 2.4, we present a network-calculus-based analysis method to compute the worst-case end-to-end delay bounds for individual flows, backlog bounds and power consumptions for individual nodes. Based on network calculus and the flow splitting model, it is able to compute per-flow equivalent service curve provided by the tandem of visited nodes and the input and departure arrival curves of each node. Consequently, the performance bounds for the network which applies the flow based traffic splitting strategy can be derived. Under the assumptions of affine arrival curve and rate-latency service curves, closed-form formulas of these bounds are computed. The numerical results for the example scenario show that by applying the splitting strategy the end-to-end delay can be reduced in most cases, the maximum backlog can be reduced up to 40%, and the power consumption can be reduced up to 15%. Furthermore, the simulation results verify that the theoretical bounds of our analysis are valid and fairly tight.

Due to the unreliable wireless links and limited energy budget, providing reliable data transmission has turned out to be a non-trivial problem in WSN. Retransmission has been adopted as one of the most prevalent schemes for addressing this issue. In Section 2.5, we presented analytical techniques to evaluate the maximum transmission delay and energy consumption of two categories of retransmission schemes: hop-by-hop retransmission and end-to-end retransmission. With the experiment results, the maximum packet transfer delay and energy efficiency of two types of retransmission schemes are studied and compared. Moreover, the analytical method for deriving delay bound is validated through simulations.

Chapter 3

Channel Modeling and Bandwidth Estimation

This chapter summarizes our research on applying stochastic network calculus for wireless channel modeling, and bandwidth estimation through backlog measurement. The work in this chapter is mainly based on [87, 89].

3.1 Introduction

As it has been discussed in the previous chapter, network calculus is a theory dealing with performance guarantees in packet switching networks [25, 26, 17, 11, 44]. With the abstraction of arrival curve for traffic flows and service curve for network elements, it has been widely applied in communication networks for performance analysis. The DNC generally considers the worst-case performance analysis through deterministic arrival curve and service curve. Recently, it has been extended and applied for worst-case performance analysis of sensor networks by several researchers [81, 51]. However, since data communication in wireless networks is unstable and irregular, it is very difficult even impossible to find the deterministic performance bounds. To incorporate nondeterministic service provisioning, the performance bounds have to be complemented with certain violation probabilities. SNC is such a tool which can be employed in the design of wireless networks to provide stochastic service guarantees.

Bandwidth estimation is essential for Quality-of-Service (QoS) provision in wireless networks. Accurate bandwidth estimation can decrease the packet loss rate and thus improve the network performance. Due to randomness of wireless

channels, it is a very challenging task to accurately estimate the bandwidth in wireless networks [80]. There have been a lot of research efforts in developing measurement techniques for bandwidth estimation. The most popular ones are to infer bandwidth from traffic measurements based on probe packets [41, 57, 64].

In Section 3.4, we propose a network calculus based approach for Quality of Service (QoS) analysis of wireless channels subject to Rayleigh fading. A stochastic service curve model for the Rayleigh fading channel is developed. Based on this model, we provide formulas to derive the probabilistic delay and backlog bounds in the cases of deterministic and stochastic arrival curves. The simulation results verify that the bounds are tight. Furthermore, through numerical experiments, it is shown that the analysis method is capable of deriving performance bounds with corresponding violation probabilities.

In Section 3.5, we develop a detailed mechanism for bandwidth estimation of random wireless channels based on stochastic network calculus. The bandwidth is derived from the measurement of statistical backlogs based on probe packet trains. It is expressed by statistical service curves that are allowed to violate a service guarantee with a certain probability [14]. The theoretic foundation and the detailed step-by-step procedure of the estimation method are presented. For case study, simulations are conducted to validate the correctness and accuracy of the proposed method over Rayleigh channels, although our method can be applied to any channels with various characteristics.

The rest of this chapter is organized as follows. Section 3.2 includes related work. Section 3.3 introduces basics of stochastic network calculus. Section 3.4 contains modeling of a Rayleigh fading channel. A bandwidth estimation method is presented in Section 3.5. Finally, conclusions are given in Section 3.6.

3.2 Related Work

In general packet switching networks, network calculus provides methods to deterministically reason about timing properties and resource requirements. Systematic descriptions of network calculus can be found in books [17, 11, 46].

DNC is recently extended and applied for performance analysis and resource dimensioning of WSNs by several researchers [81, 51, 82, 15, 49, 86]. In [81], Schmitt *et al.* first applied network calculus to sensor network and proposed a generic framework for performance analysis of WSNs with various traffic patterns. They further extended the general framework to incorporate computational resources besides the communication aspects of WSNs [82]. In [51], Anis *et al.*

proposed a methodology for the modeling and worst-case dimensioning of cluster-tree sensor networks. They derived plug-and-play expressions for the end-to-end delay bounds, buffering and bandwidth requirements as a function of the WSN cluster-tree and traffic characteristics. In [49], the authors presented a method for computing the worst-case delays, buffering and bandwidth requirements while assuming that the sink node can be mobile.

Research on SNC has the potential of providing insights into stochastic service guarantees of packet networks [22, 13, 47, 98]. In [13], Burchard *et al.* introduced the concept of statistical service curves as a probabilistic bound on the service received by an aggregation of flows or a single flow. Ciucu *et al.* [22] extended the stochastic network calculus by providing a network service curve formulation which is capable of calculating stochastic end-to-end delay and backlog bounds for a number of arrival and service distributions. In [45], Jiang and Emstad proposed a server model to facilitate stochastic service guarantee analysis and address the challenges of delay guarantee, backlog guarantee, output characterization and concatenation property. There are a lot of works providing theoretic fundamentals of stochastic network calculus, but few of them study the problem of mapping the theory to a specific application. In [98], the authors presented a method for analyzing wireless channels by modeling channels as Markov chains. And they evaluated the delay tail distribution. However, they did not give closed-form service curves for the channel.

Recently, system-theoretic approaches [57, 64] have been proposed for estimating available bandwidth based on the network calculus. In [57], Liebeherr *et al.* proposed a traffic measurement based approach for available service estimation of deterministic systems through the measurement of deterministic backlogs. However, this method is not suitable for bandwidth estimation in wireless networks due to the stochastic behavior of wireless links. In [64], the authors extended the work in [57], and proposed to estimate the available bandwidth in networks with random traffic load or link capacities through the measurement of time stamps of probing packet trains.

3.3 Basics of Stochastic Network Calculus

The deterministic network calculus provides hard worst-case performance bounds which might be overly pessimistic in some applications, such as wireless networks. Stochastic extensions of the network calculus have been of significant interest to overcome the limitations of the deterministic network calculus. The arrival curves, service guarantees, and performance bounds are typically statistical bounds with a

certain violation probability. A systematic description of stochastic network calculus is provided in book [46].

There are several ways to model the stochastic arrival traffic and service guarantees. In this thesis, we adopt one of the definitions in [46], where other types of arrival curves and service curves can be found.

Definition 3.1 A flow $R(t)$ is said to have a stochastic service curve α with bounding function f, denoted by $R(t) \ < f, \alpha >$, if for all $t \geq 0$ and all $x \geq 0$ there holds

$$Pr\left\{\sup_{0\leq s\leq t}[R(t)-R(s)-\alpha(t-s)]>x\right\}\leq f(x). \tag{3.1}$$

Definition 3.2 A system is said to provide a stochastic service curve $\beta(t)$ with bounding function $g(x)$ if during any period $(s,t]$, the system is capable of providing the amount of service $S(t)-S(s)$,

$$Pr\left\{S(t)-S(s)<\beta(t-s)-x\right\}\leq g(x) \tag{3.2}$$

for any $x \geq 0$.

After giving the definition of arrival curve and service curve, the performance bounds can be derived accordingly. Details are introduced in the next Section.

3.4 Modeling of Rayleigh Fading Channel

3.4.1 Channel Model

Fig. 3.1 shows the system model of a discrete-time flat-fading Rayleigh channel [77], which can be expressed by,

$$Y = |h_t|e^{j\varphi}X + Z \tag{3.3}$$

where X and Y are the channel input and output, respectively; Z is the independent and identically distributed (i.i.d.) Gaussian noise; $|h_t|e^{j\varphi}$ is a complex channel gain with amplitude $|h_t|$ which is a random variable with a Rayleigh distribution, and phase φ is uniformly distributed in $[0, 2\pi)$.

Some assumptions about this model are: 1) Channel distribution information (CDI) is available both at the transmitter and receiver; 2) Channel state information (CSI) is only available at the receiver; 3) it is a slow-fading channel, i.e., the channel state does not change during the transmission of a packet.

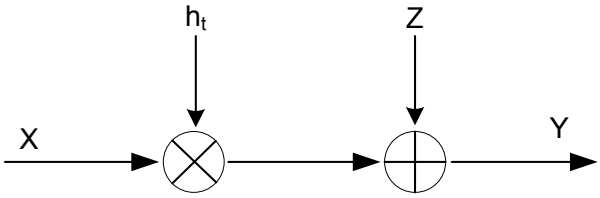


Figure 3.1. System model of a fading channel.

Let P_{tx}, W, and N_0 denote the average transmission power, channel bandwidth, and power spectral density of the noise, respectively. The channel capacity can be expressed as,

$$C = W \log_2(1 + 10^{\gamma_t/10}) = W \log_2 \left(1 + \frac{P_{tx}|h_t|^2}{WN_0}\right). \tag{3.4}$$

Since the transmitter does not know the instantaneous signal-to-noise ratio (SNR) γ_t (in dB), it can not adjust its transmission power according to the channel condition. Hence, the transmission data rate can be considered as a constant regardless of the SNR of the received signal and there is probability of outage. For a transmission data rate R, the outage probability of a Rayleigh fading channel can be expressed by,

$$\begin{aligned} p_{out}(R) &= Pr\{C < R\} \\ &= Pr\left\{W \log_2 \left(1 + \frac{P_{tx}|h_t|^2}{N_0 W}\right) < R\right\} \\ &= Pr\left\{|h_t|^2 < (2^{R/W} - 1) \cdot \frac{N_0 W}{P_{tx}}\right\}, \end{aligned} \tag{3.5}$$

where the channel gain $|h_t|$ has a Rayleigh distribution with PDF

$$f(x) = x \cdot \exp(-\frac{x^2}{2}).$$

Applying the transformation theorem, $|h_t|^2$ has an exponential distribution as PDF,

$$g(x) = \frac{1}{2} \exp(-\frac{x}{2}).$$

Therefore, the outage probability is derived as,

$$p_{out}(R) = 1 - \exp\left(\frac{1 - 2^{R/W}}{2 \cdot 10^{SNR/10}}\right), \tag{3.6}$$

where SNR $= 10\log_{10}[P_{tx}/(N_0 W)]$ denotes the signal-to-noise-ratio in dB.

3.4.2 Stochastic Service Curve

Since the channel capacity C is random, deterministic service curve is not suitable for capturing its characteristics. Thus, we adopt the stochastic service curve to characterize the service capability of the channel, which is described by two parameters: the data transmission rate R, and the error function ϵ. According to the previous analysis of the Rayleigh channel, it can be modeled by the stochastic service curve $\langle \beta(t), \epsilon \rangle$, where

$$\beta(t) = R \cdot t \quad \text{and} \quad \epsilon(R) = 1 - \exp\left(\frac{1 - 2^{R/W}}{2 \cdot \text{SNR}}\right).$$

The violation probability function ϵ defines the outage probability of the channel. It also means that the probability that the channel can not provide the transmission rate R is less than $\epsilon(R)$. ϵ is mainly impacted by the transmission data rate R and the SNR. R is determined by the modulation and coding schemes of the transmitter. SNR is determined by the transmission power and channel condition.

3.4.3 Performance Bounds

This section presents the results of performance bounds with traffic sources transmitting data over the Rayleigh channel. Two cases are considered: 1) the source periodically transmits data, which can be modeled by a deterministic arrival curve; 2) The traffic source transmits data randomly, which can be modeled by a stochastic arrival curve. Part of these derivations and proofs are based on the results in [46].

Let $R(t)$ and $R^*(t)$ denote the arrival process and departure process[1], respectively.

Deterministic Arrival Traffic

Lemma 3.1 Consider a traffic arrival process $R(t)$ bounded by a deterministic arrival curve $\alpha(t)$, and $R(t)$ receives a stochastic service curve $\langle \beta(t), \epsilon \rangle$, the performance bounds can be derived as follows [46],

[1]The input and output of the channel are called as arrival process and departure process, respectively.

1) *Backlog bound.* The stochastic backlog bound $B(t)$ is expressed as:

$$Pr\{B(t) \geq \alpha \oslash \beta(0)\} \leq \epsilon, \tag{3.7}$$

where $\alpha \oslash \beta(0) = \sup_{t\geq 0}\{\alpha(t) - \beta(t)\}$.

2) *Delay bound.* A stochastic upper bound for delay $D(t)$ is given by:

$$Pr\{D(t) \geq h(\alpha, \beta)\} \leq \epsilon, \tag{3.8}$$

where $h(\alpha, \beta) = \sup_{t\geq 0}\{\inf[\tau \geq 0 : \alpha(t) \leq \beta(t+\tau)]\}$ denotes the maximum horizontal difference between the arrival curve and service curve.

Proof [46]: Let $t \geq s \geq 0$. From known conditions, we have $B(t) = R(t) - R^*(t)$, $\alpha \oslash \beta(0) = \sup_{t\geq 0}\{\alpha(t) - \beta(t)\}$, $R(t) - R(s) \leq \alpha(t-s)$, and $Pr\{R^*(t) \leq R \otimes \beta(t)\} \leq \epsilon$. Consequently,

$$\begin{aligned}
&Pr\left\{B(t) \geq \alpha \oslash \beta(0)\right\} \\
=&Pr\left\{R(t) - R^*(t) \geq \sup_{\tau\geq 0}[\alpha(\tau) - \beta(\tau)]\right\} \\
\leq&Pr\left\{R(t) - R^*(t) \geq \sup_{0\leq\tau\leq t}[R(t) - R(t-\tau) - \beta(\tau)]\right\} \\
=&Pr\left\{R(t) - R^*(t) \geq R(t) - \inf_{0\leq\tau\leq t}[R(t-\tau) + \beta(\tau)]\right\} \\
=&Pr\left\{R^*(t) \leq R \otimes \beta(t)\right\} \leq \epsilon.
\end{aligned} \tag{3.9}$$

Let $\tau_0 = h(\alpha, \beta)$. Since the delay is defined as: $D(t) = \inf\{\tau \geq 0 : R(t) \leq R^*(t+\tau)\}$, we get $Pr\{D(t) \geq \tau_0\} \leq Pr\{R(t) \geq R^*(t+\tau_0)\}$. The problem becomes how to prove the bound of $Pr\{R(t) \geq R^*(t+\tau_0)\}$.

$$\begin{aligned}
&R(t) - R^*(t+\tau_0) \\
=&R(t) - R \otimes \beta(t+\tau_0) + R \otimes \beta(t+\tau_0) - R^*(t+\tau_0) \\
=&R(t) - \inf_{0\leq s\leq t+\tau_0}[R(s) + \beta(t+\tau_0 - s)] \\
&+ [R \otimes \beta(t+\tau_0) - R^*(t+\tau_0)].
\end{aligned} \tag{3.10}$$

Let $V = [R \otimes \beta(t+\tau_0) - R^*(t+\tau_0)]$, we have

$$\begin{aligned}
&R(t) - R^*(t+\tau_0) \\
\leq& \sup_{0\leq s\leq t+\tau_0}[R(t) - R(s) - \alpha(t-s)] \\
&+ \sup_{0\leq s\leq t+\tau_0}[\alpha(t-s) - \beta(t+\tau_0 - s)] + V \\
\leq& \sup_{0\leq s\leq t+\tau_0}[\alpha(t-s) - \beta(t+\tau_0 - s)] + V.
\end{aligned} \tag{3.11}$$

Since τ_0 is the maximum horizontal difference between $\alpha(t)$ and $\beta(t)$, we get $\alpha(t-s) \leq \beta(t-s+\tau_0)$. Therefore,

$$Pr\{D(t) \geq \tau_0\} \leq Pr\{R^*(t) \leq R \otimes \beta(t)\} \leq \epsilon. \tag{3.12}$$

Stochastic Arrival Traffic

Apart from deterministic arrival curve, an arrival process can be stochastically bounded, such as the exponentially bounded burstiness (EBB) model [107]. We consider a stochastic arrival curve model with parameters (ρ, a_1, a_2) as follows [13]:

$$Pr\left\{\sup_{0\leq s\leq t}[R(t)-R(s)-\alpha(t-s)] \geq 0\right\} \leq a_1 e^{-a_2\sigma}, \tag{3.13}$$

where $0 \leq s \leq t, \sigma > 0$, and $\alpha(t) = \rho \cdot t + \sigma$.

Lemma 3.2 Consider a traffic arrival process constrained by a stochastic arrival curve, i.e., $R(t) \sim \langle \alpha(t), f(\sigma)\rangle$, where $\alpha(t) = \rho \cdot t + \sigma$ and $f(\sigma) = a_1 e^{-a_2\sigma}$. $R(t)$ receives a stochastic service curve $\langle \beta(t), \epsilon\rangle$. The performance bounds can be expressed as follows [46],

1) *Backlog bound.* A stochastic backlog bound $B(t)$ can be derived by,

$$Pr\{B(t) \geq \alpha \oslash \beta(0)\} \leq \epsilon + f(\sigma). \tag{3.14}$$

2) *Delay bound.* A stochastic upper bound for delay $D(t)$ is given by:

$$Pr\{D(t) \geq h(\alpha, \beta)\} \leq \epsilon + f(\sigma). \tag{3.15}$$

Since the probability can not be bigger than one, we define $\epsilon + f(\sigma) = \min(\epsilon + f(\sigma), 1)$.

Proof [46]: The backlog is defined as the amount of data stored in the system. So we have,

$$\begin{aligned}
B(t) &= R(t) - R^*(t) \\
&= R(t) - R\otimes\beta(t) + R\otimes\beta(t) - R^*(t) \\
&= R(t) - \inf_{0\leq s\leq t}[R(s)+\beta(t-s)] + [R\otimes\beta(t) - R^*(t)] \\
&\leq \sup_{0\leq s\leq t}[R(t)-R(s)-\beta(t-s)-\alpha(t-s)+\alpha(t-s)] \\
&\quad + [R\otimes\beta(t) - R^*(t)] \\
&\leq \sup_{0\leq s\leq t}[R(t)-R(s)-\alpha(t-s)] + \sup_{t\geq 0}[\alpha(t)-\beta(t)] \\
&\quad + [R\otimes\beta(t) - R^*(t)].
\end{aligned} \tag{3.16}$$

The prove of delay bound is very similar to the case with deterministic arrival curve (Eq. (3.11)). The difference is in the last step of Eq. (3.11): for deterministic arrival curve, $Pr\{\sup_{0\leq s\leq t}[R(t)-R(s)-\alpha(t-s)]\geq 0\}=0$; while for stochastic arrival curve, $Pr\{\sup_{0\leq s\leq t}[R(t)-R(s)-\alpha(t-s)]\geq 0\}\leq f(\sigma)$. Therefore, the delay bound is given by $Pr\{D(t)\geq h(\alpha,\beta)\}\leq \epsilon+f(\sigma)$.

Although our work focuses on the EBB arrival model, it is straightforward to extend the method for deriving performance bounds for other types stochastic arrival curves as long as the deterministic arrival curve and violation probability are known.

3.4.4 Performance Evaluation

This section presents both simulation and numerical results. The relations between performance bounds and channel characteristics are studied through numerical results, which also provide hints on designing transmission strategies in wireless networks.

Simulation Results

In order to validate the correctness and tightness of the network calculus based modeling and QoS analysis method, we conduct simulations and compare analytical results with simulation results. The parameters used in the simulation are: channel bandwidth $W=30\,kHz$, $SNR=0\,dB$, packet size is $1\,kbit$. Each simulation is performed 50 runs with different seeds. In each run, the simulation period is 20000 cycles and the source generates one packet every cycle. The delay of every packet is recorded, and the backlog is recorded in every cycle. Since we can not obtain the violation probability from the simulation directly, we need to map the simulation results to delay/backlog bounds with corresponding violation probabilities. Let $m=50$ denote the total number of runs, and $n=20000$ denote the total number of packets (cycles) in one simulation run. Let $d(i,j)$ $(i=1,2,\cdots,m; j=1,2,\cdots,n)$ denote the delay of packet j in simulation run i. We sort $d(i,:)$ in descending order, where $d(i,:)$ represents the set of values of delay in simulation run i. Let $\epsilon_0(l)=(l-1)\times\kappa$ $(l=1,2,\cdots,\lceil 1/\kappa\rceil)$ denote the set of violation probabilities, where κ is a scaler (In the figures, we set $\kappa=0.05$). If the violation probability is $\epsilon_0(l)$, the corresponding delay bound is computed by $d_0(l)=\max_{1\leq i\leq m}[d(i,\lceil n\times\epsilon_0(l)\rceil)]$. For example, the delay bound with violation probability $\epsilon_0(3)=0.1$ is computed as $d_0(3)=\max_{1\leq i\leq m} d(i,2000)$.

In the first simulation, the traffic source sends packets periodically with data rate $r=10\,kbps$ over the Rayleigh fading channel. The arrival process can be

modeled by an affine arrival curve $\alpha(t) = rt + b$, with $b = 1\ kbit$. The analytical and simulation results of packet delays are studied and compared under different violation probabilities.

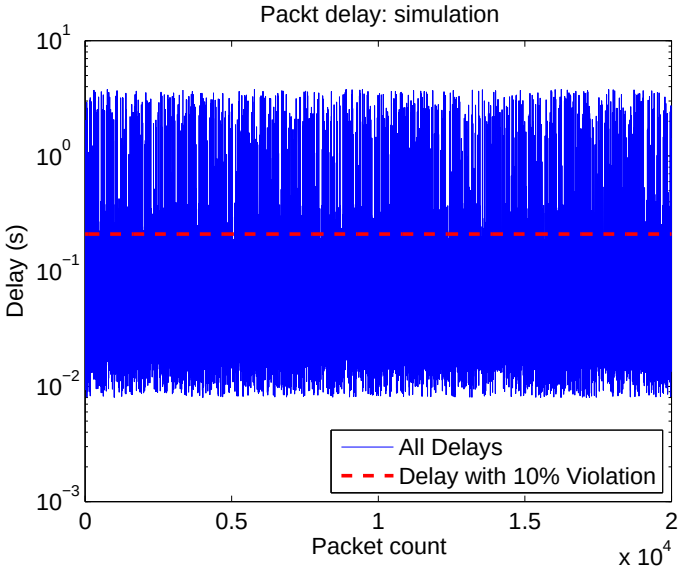


Figure 3.2. Simulation results: delay.

Fig. 3.2 illustrates the packet delays in one simulation run, where 20000 packets are sent by the source. From this figure, we can see that the packet delays are very heterogeneous, varying from $0.01\ s$ to $5\ s$. For the simulation data, we find that there are very few packets experiencing high delays. So the deterministic bound may be very loose and it is more meaningful to derive the stochastic bound. From these simulation results, we can derive the delay bounds with corresponding violation probabilities using the mapping method proposed in the previous discussion. For example, we first sort the delay $d(i)$ ($i = 1, 2, \cdots, 20000$) in the descending order. If the violation probability is 10%, the delay bound is $d(2000)$ (since $2000/20000 = 0.1$), which is $0.21\ s$ (shown by the dashed line in Fig. 3.2). The delay bounds with other violation probabilties (as shown by $'+'$ marker in Fig. 3.3) can be computed similarly.

The analytical results can be computed by Eq. (3.8). Fig. 3.3 shows the comparison of analytical results and simulation results. In this figure, the maximum, average, and minimum differences between the simulation and analytical results are 5.19%, 9.88% and 13.5%, respectively. From this figure, we can see that all the simulation results are within the bound of the analytical results. Moreover, the

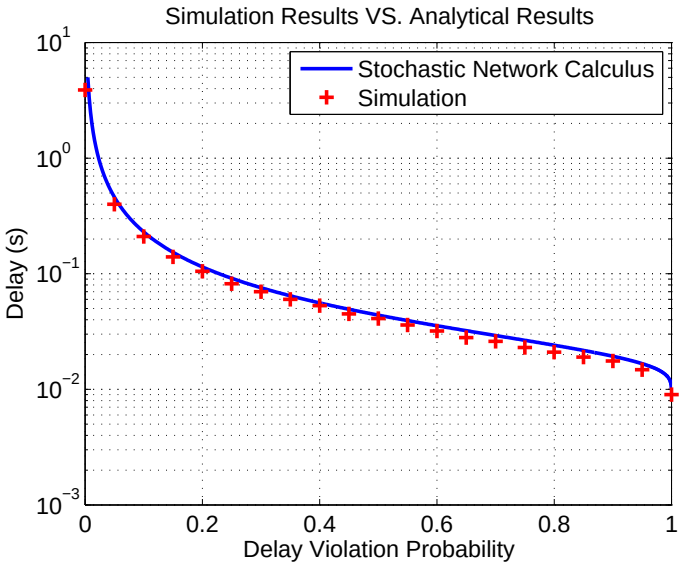


Figure 3.3. Compare simulation and analytical results.

comparison indicates that the performance bounds derived by network calculus is reasonably tight.

Since the arrival process is constrained by a deterministic arrival curve $\alpha(t) = rt+b$, and the channel provides a stochastic rate service curve with zero processing delay, the backlog bound (Eq. (3.7)) is only determined by the burstiness of the input. Hence, the analytical backlog bound is constant and independent of the violation probability. In this case, we need to use empirical method to derive an empirical backlog bound by combining the analytical and simulation results.

Simulation with Stochastic Arrival Traffic: In the second simulation, the traffic arrival process is a stationary two-state Markov process $R(t)$, $t = (i-1)t_0, i = 1, 2, 3, \cdots$ (t_0 is the inter-arrival time) with transition probabilities $p_{12} = 10^{-6}$ and $p_{21} = 10^{-2}$. When in state 1, the source is generating n_0 packets with probability $1/8$ and no packet with probability $7/8$. When in state 2, the source is generating n_0 packets with probability $9/20$ and no packet with probability $11/20$. The packet size is $1\ kbit$. This process can be characterized by the EBB arrival model as follows, $Pr\{\sup[R(t) - R(s)] \geq \rho(t-s) + \sigma\} \leq \exp(-a_2\sigma)$, where $a_2 = 2.73e - 3$ and $0 \leq s \leq t$ [95].

Fig. 3.4 shows the comparison of analytical results (by network calculus) and simulation results. For the analytical results, we set input burstiness $10\ kbit$. Two simulations are implemented with input data rate $\rho = 10\ kbps$, and burstiness σ

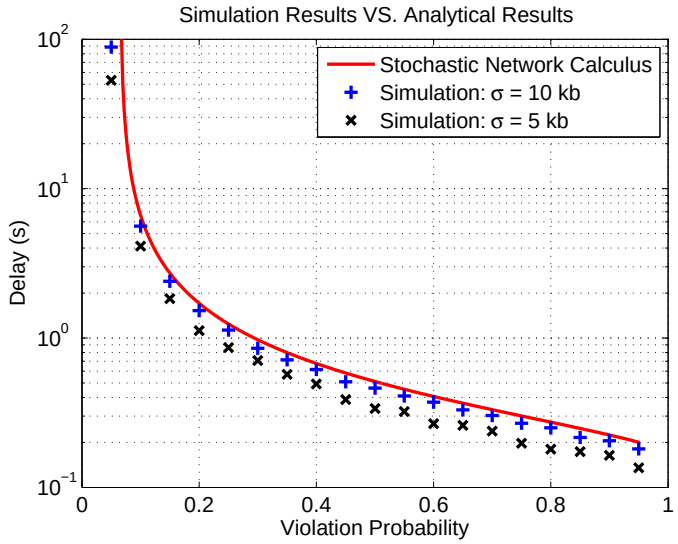


Figure 3.4. Compare simulation results with analytical results: Markov arrivals.

equals to $5\ kbit$ and $10\ kbit$, respectively. The input burstiness denotes the maximum number of packets that the source generates at a time. For example, $\sigma = 5$ corresponds to $n_0 = 5$ and $t_0 = 0.5\ s$, and $\sigma = 10$ corresponds to $n_0 = 10$ and $t_0 = 1\ s$. From the comparison, we can see that both simulation results are within the bounds of analytical results, which validate the correctness of the analytical method. Furthermore, with the same setting of input burstiness, i.e., $\sigma = 10\ kbit$, the simulation and analysis results are very close to each other. The maximum, average, and minimum differences between them are 8.25%, 10.5% and 15.8%, respectively. The comparison indicates that the tightness of delay bound derived by network calculus is acceptable.

Numerical Results

In the first experiment, the performance bounds are studied under the case of a periodic traffic source sending packets over the Rayleigh fading channel. The traffic input process is modeled an affine arrival curve [11], which is defined as $\alpha(t) = rt + b$, where b and r represent the burst tolerance (in $bits$) and the average data rate (in bit/s), respectively. In this experiment, we set $b = 1\ kbit$ and $r = 10\ kbps$.

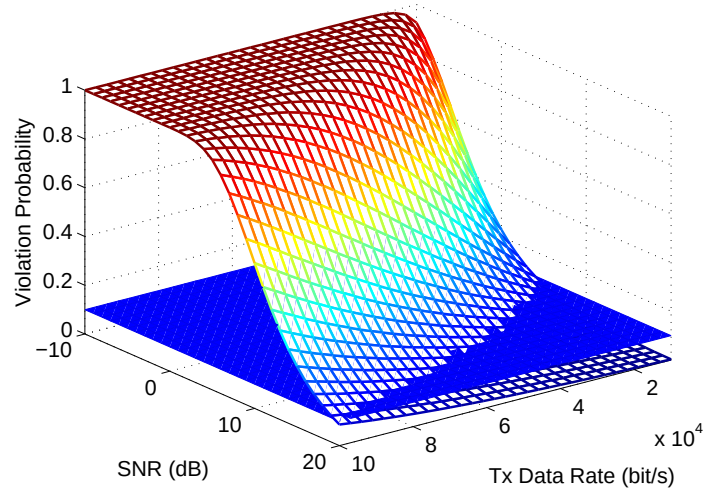


Figure 3.5. Delay bound under different SNR and transmission data rates.

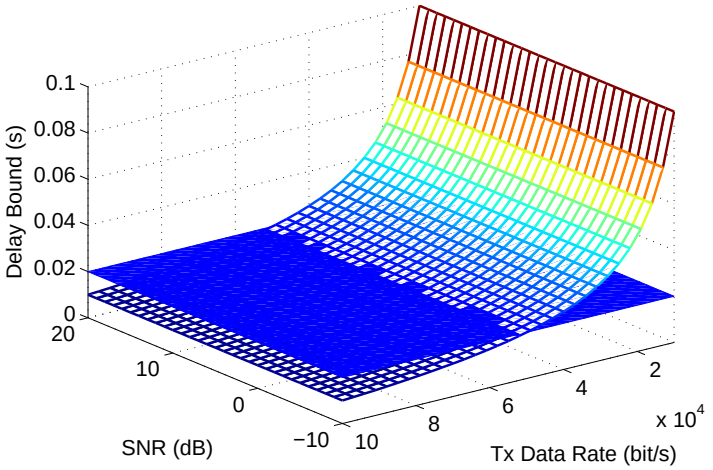


Figure 3.6. Violation probability of the delay and backlog bounds.

If $R \geq r$ [2], the delay bound can be calculated by $h(\alpha, \beta) = b/R$, and the

[2] If $R < r$, both the delay and backlog would increase with time. It is impossible and meaningless to find the finite delay and backlog bounds. Therefore, we consider the case that $R \geq r$.

backlog bound is $\alpha \oslash \beta(0) = b$. Fig. 3.5 shows the values of delay bound varying with $R \in [10\ kbps, 100\ kbps]$ and $SNR \in [-10\ dB, 20\ dB]$. The delay bound d is determined by the input burstiness b and the transmission data rate R. From Fig. 3.5, we can see that with a fixed b, the delay bound increases as transmission rate increases, and it is independent of the SNR. The backlog bound is only determined by the burstiness of the input traffic. However, the corresponding violation probabilities of these two bounds are impacted both by the transmission data rate and SNR (as shown in Fig. 3.6). The violation probability increases as transmission data rate increases and/or SNR decreases. There is a trade-off between the delay bound and its violation probability. If we want to reduce the transmission delay, a high transmission rate should be chosen. But high transmission rate brings high violation probability.

The analysis method not only can be used to derive performance bounds and their corresponding violation probabilities, but also can provide guidelines for designing modulation and coding schemes, i.e., the designer can choose appropriate transmission strategies according to performance requirements. For example, if the performance requirement is given as: $d < 0.02$ (as shown by the plane in Fig. 3.5) with violation probability less than 10% (as shown by the plane in Fig. 3.6), we can get the following system configurations: $R > 50\ kbps$ and $SNR > 14dB$, which are the values constraining delay and violation probability within the threshold. Hence, the designer can choose corresponding transmission strategy (transmission power, modulation and coding schemes) according to R and SNR.

Numerical Results with Stochastic Arrival Traffic: In the following experiments, we study the performance bounds under the case of a random traffic source sending packets over the Rayleigh fading channel. The traffic input process is modeled by the EBB arrival curve (Eq. (3.13)). The values of parameters are: $a_1 = 1$, $a_2 = 1e - 3$, and $\rho = 10\ kbps$. Similar to the previous experiment, it is assumed $\rho \leq R$.

In this experiment, the transmission data rate is set to $R = 30\ kbps$, and the values of input burstiness (σ) and SNR are changed. Both the delay ($d = \sigma/R$) and backlog ($B = \sigma$) bounds are linearly increasing with input burstiness. Their corresponding violation probabilities are illustrated in Fig. 3.7, where we can see that the violation probabilities decrease as SNR and/or σ increase. Hence, there are trade-offs between performance bounds and their violation probabilities. High σ would cause low violation probability, but bring high delay and backlog. Therefore, appropriate values of σ and SNR should be chosen according to the performance requirements.

In this experiment, we fix the $SNR = 10\ dB$ and change the values of input

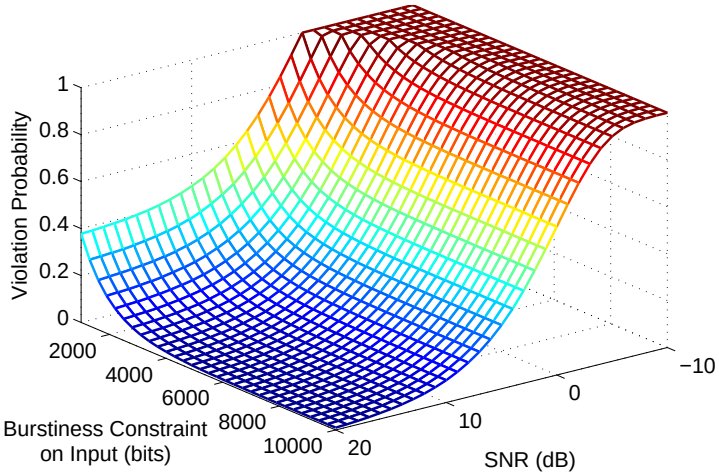


Figure 3.7. Violation probability of the delay and backlog bounds.

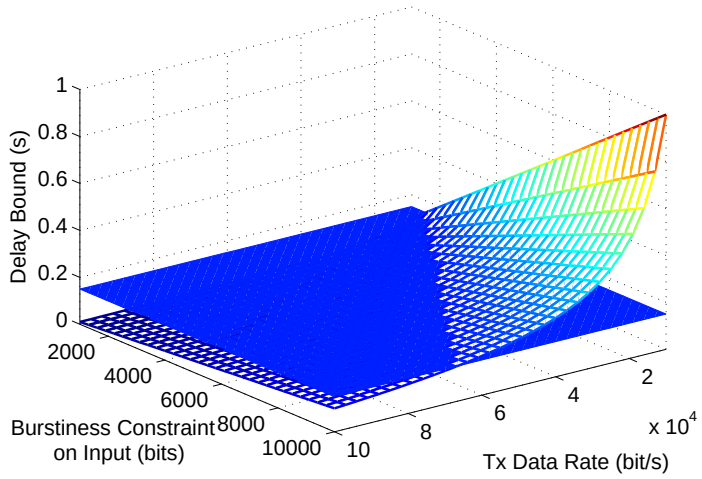


Figure 3.8. The delay bound.

burstiness (σ) and transmission data rate R. Similarly, the backlog bound is only determined by σ. The delay bound is shown in Fig. 3.8. The violation probabilities of these two bounds are illustrated in Fig. 3.9.

From Fig. 3.8, we can see that the delay bound increases as input burstiness

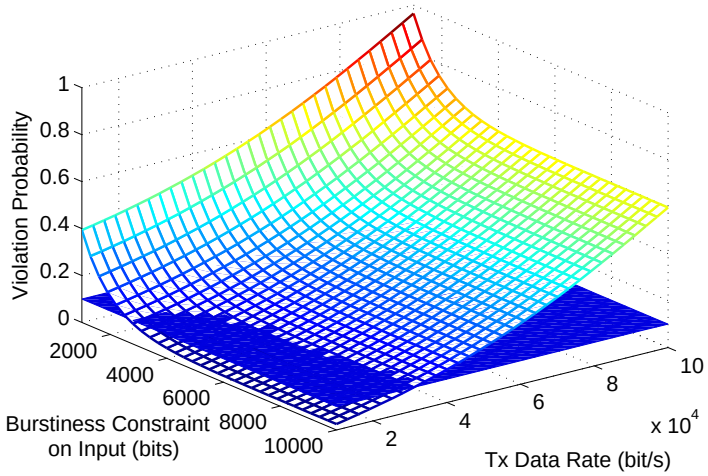


Figure 3.9. Violation probability of the delay and backlog bounds.

increases, and it decreases as transmission data rate increases. While the violation probability of the bounds has an opposite behavior, i.e., it decreases as the input burstiness increases and it increases as the transmission data rate increases. Moreover, from these performance results we can derive the system configurations which can meet the requirements. For example, if the performance requirements are defined as: $d < 0.15\ s$, $B < 5\ kbit$ with violation probability less than 10%, then we should select the values of SNR and σ below the planes in Fig. 3.8 and Fig. 3.9.

3.5 Bandwidth Estimation through Backlog Measurement

3.5.1 Preliminaries

Based on the min-plus algebra and max-plus algebra, network calculus can transform complex network systems into analytically tractable systems [11, 46]. Let $R(t)$, $S(t)$, and $R^*(t)$ denote the arrival process, service process, and departure process of a system, respectively. Their relations can be defined as follows,

$$R^*(t) = R \otimes S(t) = \inf_{0 \le \tau \le t} \{R(\tau) + S(t-\tau)\} \tag{3.17}$$

where $\otimes$ denotes the min-plus convolution. In the min-plus algebra, the service capability of a system is expressed in terms of service curves. For wireless networks,

it is very difficult, if not impossible, to find deterministic service guarantees due to the stochastic nature of wireless channels. A popular way to model the stochastic service process is using *statistical service curve* [14], which provides probabilistic abstraction of the service guarantees.

A non-decreasing function $\tilde{S}(t)$ is a statistical service curve of the system, if

$$Pr\left\{R^*(t) \geq R \otimes \tilde{S}(t)\right\} > 1 - \xi \tag{3.18}$$

where ξ denotes the violation probability ($0 \leq \xi \leq 1$). The statistical service curve provides a probabilistic abstraction of the service at a system, which is expressed by a deterministic service curve ($\tilde{S}(t)$) and a corresponding violation function.

The objective of our bandwidth estimation framework is to model the service capability of wireless channels. Since it is very difficult to derive the exact service process $S(t)$ due to the randomness of the channels, we try to derive its corresponding statistical service curve $\tilde{S}(t)$.

Lemma 3.3: For a system with service process $S(t)$, any function $\tilde{S}(t)$ that satisfies,

$$Pr\left\{S(t) \geq \tilde{S}(t)\right\} > 1 - \xi \tag{3.19}$$

for $0 \leq t \leq T$, is a statistical service curve of the system, where T is a limit on the maximal time scale on which the service curve relates service guarantees to arrivals [14]. The detailed proof of this lemma is presented in [14] [64].

The bandwidth utilization (real service) of a wireless channel is determined by its channel characteristics, behaviors of the sender and the receiver. It can be reflected by the amount of backlogged traffic at the sender. Therefore, the channel bandwidth estimation can be implemented by measuring the backlog at the sender through sending probe packets. We adopt the *rate scanning* probe scheme proposed in [57], where the packets are organized in packet trains. The probe packet trains are injected and the backlog bounds are measured. Based on the Legendre transform [57], the statistical service curve and bandwidth are derived through the measured backlog bounds.

According to the traffic pattern of probe packets, the arrival process at the sender can be expressed by $R(t) = rt$, where r is the transmission rate of the probe train. Given the arrival and departure functions $R(t) = rt$ and $R^*(t)$, the backlog bound is derived by [46],

$$B(r) = \sup_t\{R(t) - R^*(t)\} = \sup_t\{rt - R^*(t)\} \tag{3.20}$$

Due to the stochastic behavior of the wireless channel, the backlog bound is also stochastic. We can model the stochastic backlog bound by a statistical backlog bound, which is formally defined by the following equation,

$$Pr\{B(r) \leq B^{\epsilon}(r)\} > 1 - \epsilon \tag{3.21}$$

where $B^{\epsilon}(r)$ denotes the statistical backlog bound with a violation probability ϵ when the probing rate is r. The following proposition formalizes the bandwidth estimation method.

Proposition 1: Consider a system with probing packet trains constrained by the arrival curve $R(t) = rt$. Based on the measurement of the statistical backlog bound $B^{\epsilon}(r)$, the statistical service curve of the system can be derived by[3],

$$\tilde{S}(t) = \sup_{r} \{rt - B^{\epsilon}(r)\} \tag{3.22}$$

where the violation probability of the statistical service curve is calculated by $\xi = \min(\sum_r \epsilon, 1)$.

Proof: According to Eq. (3.20), the backlog bound of a system can be calculated by $B(r) = \sup_t \{rt - R^*(t)\}$. Based on Eq. (3.17) and the Legendre transform [57] [64], the relations of backlog and service curve can be expressed as,

$$B(r) = \sup_{t} \{rt - S(t)\} \Rightarrow S(t) = \sup_{r} \{rt - B(r)\} \tag{3.23}$$

From Eq. (3.21), we have

$$\begin{aligned} &Pr\{B(r) \leq B^{\epsilon}(r)\} > 1 - \epsilon \\ \Rightarrow &Pr\{rt - B(r) \geq rt - B^{\epsilon}(r)\} > 1 - \epsilon \end{aligned} \tag{3.24}$$

By the application of union bound (Eq. (3.17) in [46]), it follows that,

$$Pr\left\{S(t) \geq \sup_{r} \{rt - B^{\epsilon}(r)\}\right\} > 1 - \sum_{r} \epsilon \tag{3.25}$$

If we define $\tilde{S}(t) = \sup_r \{rt - B^{\epsilon}(r)\}$, it completes the proof. $\tilde{S}(t)$ is a statistical service curve of the system with the violation probability calculated by $\xi = \sum_r \epsilon$. Since the probability is non-negative, we set $\xi = \min(\sum_r \epsilon, 1)$.

Proposition 1 relates the statistical backlog bound with the statistical service curve based on the Legendre transform. Service curve for random wireless channels can be estimated by using probe packet trains transmitted at different rates. Since the service curve is expressed as the integration of instant service rate over time, the bandwidth (service rate) is equal to the slope of statistical service curve.

[3]We have independently given and proved the proposition. Later we found that this proposition matches an online publication [63].

3.5.2 Backlog-Based Bandwidth Estimation

After establishing the theoretic foundation of the bandwidth estimation method in Section 3.5.1, a detailed estimation procedure is given as follows.

Let a probe packet train be composed of N_0 packets with length l. The transmission time interval between two adjacent packets is T_0. The transmission rate (denoted by r) of a probe train is calculated by $r = N_0 \cdot l/[T_0(N_0 - 1)]$. Probe packet trains with the same rate are repeatedly injected into the system and the backlogs are measured according to pre-defined sampling rates. Subsequent probe packet trains are injected with increasing rates until the system is saturated. The bandwidth estimation procedure can be divided into three iterative steps. Fig. 3.10 shows the flow chart of the estimation algorithm.

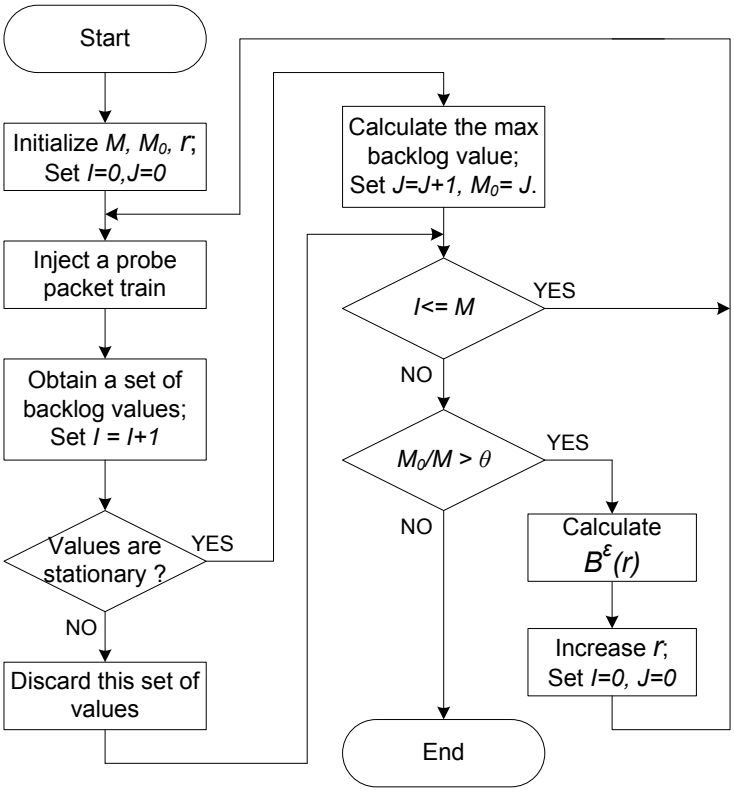


Figure 3.10. Procedure of the bandwidth estimation method

• *Inject packet train and obtain stationary backlog*

From injecting a packet train with probe rate r, we can obtain a set of backlog values at different time points from the measurements. If the values are stationary (i.e. steady-state), we keep them and calculate the max value $B_{max}(r)$. If the values are not stationary, they are discarded. The stationarity of the backlog series can be detected by using the square root test from Elliot, Rothenberg and Stock (RES) [64] [32]. The ERS test is based on an autoregressive moving average model which returns a single value called ERS statistic. If the ERS statistic falls below a threshold value, the data series is considered as stationary.

• *Calculate statistical backlog bound*

In the experiment, each packet train (with rate r) is sent for M times and M_0 ($M_0 <= M$) steady-state values of maximum backlog $B_{max}(r)$ can be obtained. If $M_0/M > \theta$ [4], it means that most of the measurements of backlog values are stationary. Then, the probe rate is increased and a new probe packet train is sent. Otherwise, it indicates that the probe rate is too high, leading to system saturation. In this case, the probe rate increase is stopped and the program ends.

In order to obtain the statistical backlog bound from a large amount of measurements, we apply the statistical method called *quantile* (Chapter 2 in [12]). A $p\%$-quantile leaves $p\%$ of the measurement values below and $(100-p)\%$ above with certain confidence levels. It is defined as follows. Assume there are n data points $x_1, ..., x_n$ from the measurements. Sort the points in an increasing order and obtain $x_{(1)} \leq ... \leq x_{(n)}$. The $p\%$-quantile is defined as $(x_{k'} + x_{k''})/2$ with $k' = \lfloor p(n-1)/100 + 1 \rfloor$ and $k'' = \lceil p(n-1)/100 + 1 \rceil$. $\lfloor u \rfloor$ is the largest integer not bigger than u and $\lceil u \rceil$ is the smallest integer not less than u. The quantile reveals information about the dispersion of the observations of random variables. It consists of an upper bound and a lower bound. Based on the quantile principle, these M_0 measurements of backlog bound are quantified and we can get the upper and lower bounds of $p\%$-quantile of the backlog bound (denoted by $B^{\epsilon}(r)$) with certain confidence levels (the typical value of confidence level is 0.95).

• *Bandwidth estimation*

Each probe rate r reveals one point $B^{\epsilon}(r)$ of the service curve in the Legendre domain, which composes a piecewise linear segment of the service curve $\tilde{S}(t) = rt - B^{\epsilon}(r)$. The probe rate of packet train is increased and the probe packet train is sent repeatedly until the proportion of stationary measured backlog values is below the threshold (i.e. $M_0/M \leq \theta$). According to Proposition 1, we can re-build the estimated statistical service curve, which is composed of piecewise linear segments. Since the service curve is expressed as the integration of instant

[4] θ denotes the threshold of the proportion of valid measurements that can produce stationary backlog values.

service rate over time, the bandwidth can be derived by calculating the slope of the statistical service curve.

3.5.3 Experimental Results

Experiment Purpose and Setup

Simulations are conducted to validate the estimation method. The sender transmits probe packet trains to the receiver through a time-variant Rayleigh fading channel. Consider a slow fading channel so that the capacity does not change during the transmission of one packet. A packet train contains 1000 packets, and the arrival interval between two adjacent packets is $9\ ms$. The length of a packet changes from $100\ bits$ to $3000\ bits$ with an increment of $100\ bits$ in each step. Hence, the corresponding probe data rate varies from $11\ kbps$ to $330\ kbps$ with an increment of $11\ kbps$. For each probe rate, the simulation runs 1000 times. The values of backlog are recorded every millisecond until the last packet has been sent. Since the results are not sensitive to the value of threshold θ, θ is set to 0.8 as an example.

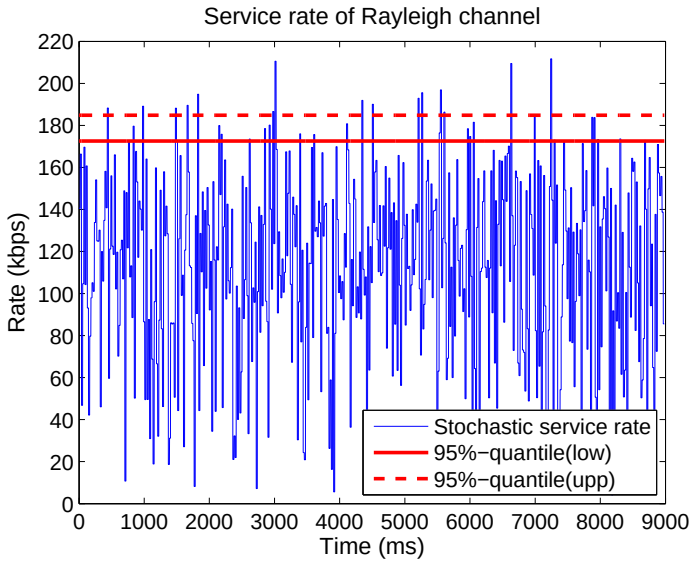


Figure 3.11. Service rate of Rayleigh channel

The channel capacity of the Rayleigh channel is given by $C = BW \cdot log_2[1 + S/(BW \cdot \delta_0)]$, where BW denotes the channel bandwidth, δ_0 is the noise power,

and S represents the received signal power. We set $BW = 30\ kHz$, $\delta_0 = -55\ dBm$ ($3.3 \times 10^{-9}\ W$), and S is an exponential distributed random variable with a mean of 2×10^{-3}. Fig. 3.11 illustrates a realization of the service rate which is defined as the capacity of the channel. The 95%-quantile of the service rate is obtained by using the method described in Section 3.5.2.

Estimation Results

In each simulation run, a backlog bound is recorded if the set of backlog results are stationary (Section 3.5.2). These backlog bounds are quantified to obtain the statistical backlog bound. Fig. 3.12 shows the 99%-quantile of the upper and lower bounds of maximum backlog with confidence level of 95%.

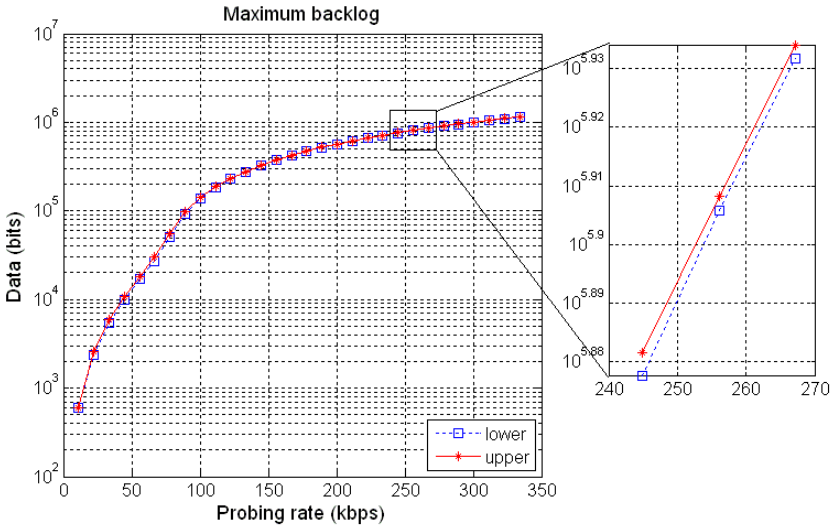


Figure 3.12. Maximum backlog for different probing rates.

Both the statistical service curves of the reference and estimation are expressed by deterministic service curves and corresponding violation probabilities. Fig. 3.13 shows the comparison of *reference service curve* (Ref SC) and the *estimated service curve* (Est SC). The confidence level of them are 95%. The reference service curve is described by a rate service curve with rate in the range $[173, 185]\ kbps$ and the violation probability of 5% (as shown by the solid and dashed lines). In the simulation, the estimated service curve is contributed by piecewise linear segments with 99%-quantile backlog bounds from 5 different probe rates. So the violation

probability of the estimated service curve is 5% (Eq. (3.25)). The estimated service curve is divided into two regions: starvation region and stable region. In the starvation region, the service capability of the channel is higher than the input traffic rate. So the estimated bandwidth is smaller than the actual service rate. In the stable region, the estimated bandwidth is $178\ kbps$, which is within the range of the reference service rate. The results in the stable region show that our method can accurately estimate the bandwidth.

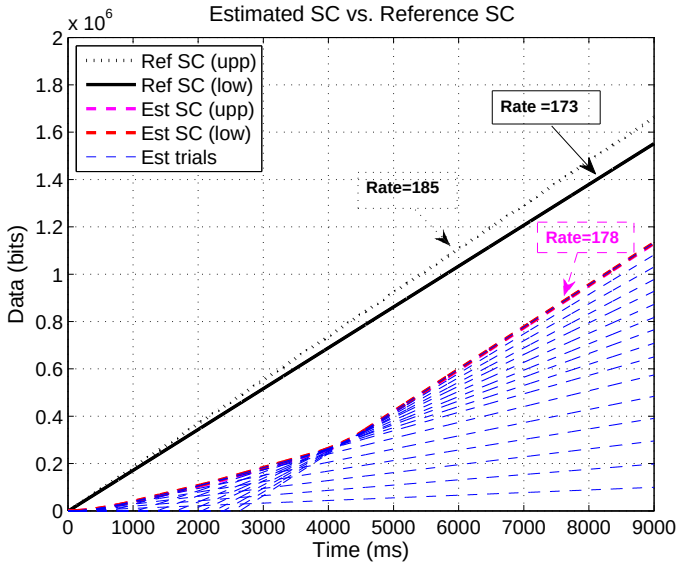


Figure 3.13. Comparison of reference service curve and estimated service curve: in simulation trials, the probe packet trains are sent with various rates and the backlog bounds are measured. The thin dashed blue lines represent the results of bandwidth estimation from all the trials.

In Fig. 3.14, the estimated bandwidth is compared with the reference service rate (i.e. bandwidth) under different channel parameters. Since the difference between the upper and lower bounds of the estimated bandwidth is very small, we only show the mean value of the estimated bandwidth in the figure. From Fig. 3.14, we can see that the estimated bandwidths are within the range of the upper and the lower bounds of the reference bandwidth for the second and fourth bar groups. When the channel bandwidth is $20\ kHz$ and the mean of the received signal power is 0.001, the estimated bandwidth is 5% bigger than that of the upper reference bandwidth. For the third bar group, the estimated bandwidth is 8% larger

than the upper reference bandwidth. The deviation is due to imprecise sampling of backlog measurement.

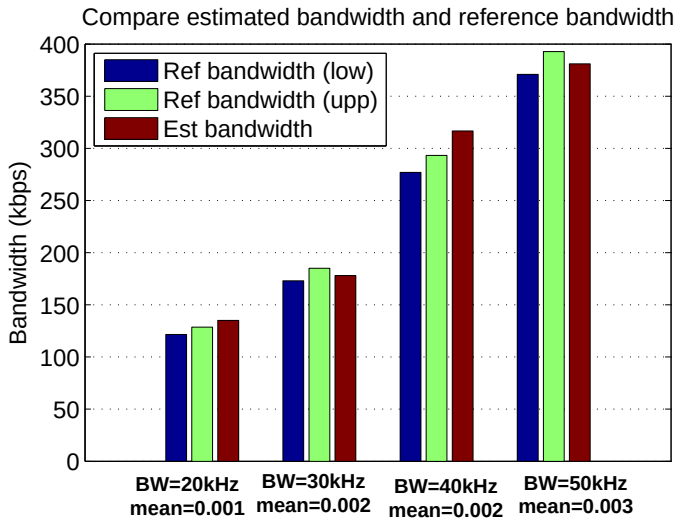


Figure 3.14. Comparison of reference service curve and estimated service curve with various channel parameters: 1) channel bandwidth; 2) mean of the received signal power (in Watt).

We design the following experiment to study the impact of inter-packet interval and packet size. In each simulation, the packet length is fixed, and we change the probe data rate by varying the inter-packet interval from $2\ ms$ to $20\ ms$ with an increment of $1\ ms$. The bandwidth is $BW = 40\ kHz$, and the mean is 0.002. Other parameters remain the same. We run 15 simulations with packet lengths increasing from $600\ bits$ to $2000\ bits$. Fig. 3.15 illustrates the comparison of estimated bandwidth and reference bandwidth. As we can see, most of the estimated bandwidth are within the lower and upper bounds of the reference bandwidth. Only a few are slightly deviated from the reference because of sampling effect. In summary, from Fig. 3.14 and Fig. 3.15, we can see that the accuracy of the method is satisfactory under different channel characteristics, packet sizes, and inter-packet intervals.

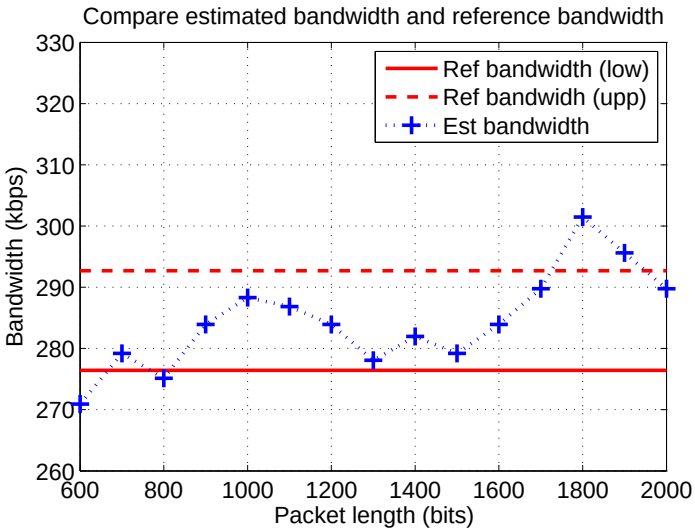


Figure 3.15. Comparison of reference service curve and estimated service curve with various inter-packet interval and packet length.

3.6 Summary

In this chapter, we propose a method for modeling and QoS analysis of wireless channels subject to Rayleigh fading. The key challenge in analyzing wireless systems is the inherently temporal uncertainties in fading channels. To this end this work applies stochastic network calculus to model Rayleigh fading channels and derive stochastic delay and backlog bounds. The analysis method is validated through simulations.

In Section 3.5, we introduce a stochastic network calculus based method for statistical bandwidth estimation of random service networks. The statistical bandwidth is estimated from the measurement of statistical backlog bounds through probe packet trains. We propose a step-by-step procedure on how to estimate the bandwidth from backlog measurement. The method can be applied to any wireless networks with various channel characteristics. To validate the method, simulations are performed to compare the estimated bandwidth with the reference bandwidth over a Rayleigh fading channel. The estimation results show good accuracy of our bandwidth estimation method.

Chapter 4

Coverage and Deployment

This chapter presents our research on coverage analysis and deployment techniques in wireless sensor networks. The work in this chapter is mainly based on [88, 91, 92].

4.1 Introduction

One fundamental application of WSNs is event detection in a Field of Interest (FoI), where a set of sensors are deployed to monitor any ongoing events. For example, sensor networks can be deployed to detect the occurrences of fires, earthquakes, volcanic eruptions, and motions of wild animals [3, 27, 102]. To satisfy a certain level of detection quality in such applications, it is desirable that events in the region can be detected by a required number of sensors. Hence, an important problem is how to deploy sensor nodes for achieving certain coverage requirements.

Coverage analysis in WSNs has attracted plenty of research interests [59, 100, 52, 6, 8, 115, 108]. One of the most popular metrics to quantify the coverage performance is *k*-coverage [43, 113]. An FoI is said to be k-covered if every point in it is covered by at least k sensors. The coverage problem can be classified in different ways depending on the way of sensor deployment and the features of applications. Based on the characteristics of applications, the k-coverage problem can be categorized into *area coverage* and *event coverage*, where the former requires each location within the sensing field must be covered by at least k sensors. While the objective of event coverage analysis is to study how well the events in the sensing field are detected. Fig. 4.1 shows an example of area coverage and event coverage (As an example, the FoI in this figure is a rectangle. However, in our analysis the

FoI can be irregular). In the left figure, the purpose is to ensure the whole area of the FoI is covered by sensors. In the right figure, the purpose is to ensure the events in the FoI are covered by sensors.

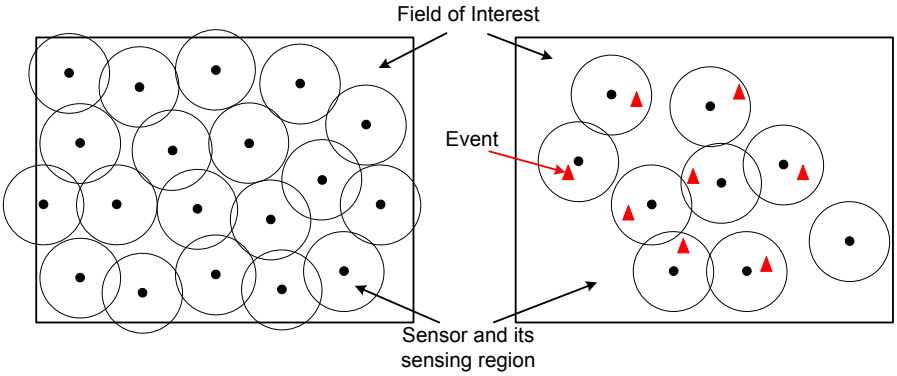


Figure 4.1. Left: area coverage; right: event coverage

In many applications, the tasks of sensor networks are monitoring or detecting events, such as the behaviors of animals in a forest [93]. For those scenarios, covering of locations does not mean covering of events. For example, in a WSN application, we need to deploy video sensors in a forest to monitor the behaviors of wild animals, and a sensor can not monitor two animals simultaneously. Assume five video sensors can cover the whole area of the forest from the analysis of area coverage. However, if more than five animals enter into the monitoring region, five sensors are not able to track the activities of all the animals. To study the performance of coverage in those cases, we need event coverage analysis method. However, there is very few research on event coverage analysis considering both area coverage analysis and event occurrence.

In this chapter, a probabilistic event coverage analysis method is proposed for evaluating the coverage performance of heterogeneous sensor networks with randomly deployed sensors and stochastic event occurrences. Different from traditional area coverage analysis, the event coverage is to quantify the probability that events can be covered by sensors, given the density of events and the density of sensor deployment. We provide formulas to calculate event coverage probability under different settings of sensor and event densities with the considerations of boundary conditions. Moreover, the event coverage analysis is extended to heterogeneous sensor networks. In this chapter, we define heterogeneous sensor network as the network where the sensor distribution in the FoI is heterogeneous. As shown in Fig. 4.2, in the homogeneous network, the sensor density in the whole FoI is uniform.

In the heterogeneous network, the FoI is logically divided into sub-FoIs according to the event density. In different sub-FoIs, the sensors are deployed with different densities. The probability of event coverage in heterogeneous sensor networks is derived. The analytical results show how the probabilities of event coverage change with the sensor and event densities. Moreover, the results also indicate that for sensor networks with irregular event occurrence, heterogeneous sensor deployment can achieve better coverage performance than homogeneous deployment. In addition, the comparisons of simulation and analytic results validate our proposed event coverage analysis method. This method can provide guidelines in determining how many sensors are required to guarantee a certain level of coverage for given users' requirements and characteristics of applications.

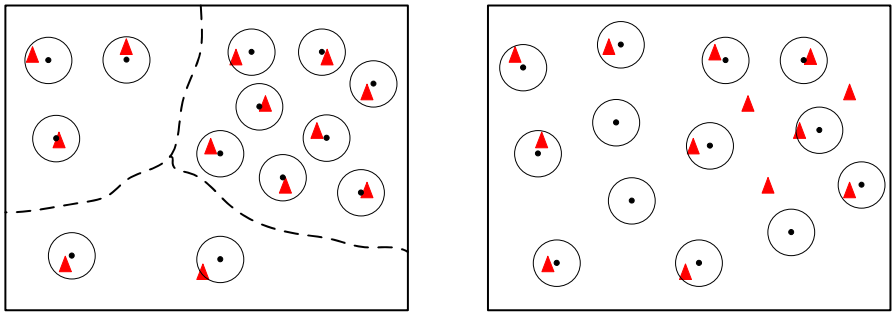


Figure 4.2. Left: heterogeneous sensor networks; right: homogeneous sensor networks

4.2 Related Work

In recent years, a host of research activities have been performed on coverage analysis in sensor networks regarding to different objectives and metrics [38, 52, 59, 100, 101, 7]. These work can be divided into two classes of coverage analysis: area coverage and event coverage [94].

Liu and Towsley in [59] defined three coverage measures to characterize area coverage, node coverage and detectability in large-scale sensor networks. They studied the coverage problem from a theoretical perspective and explored the fundamental coverage limits. In [52], the authors studied the problem of stochastic coverage in heterogeneous sensor networks. By mapping the coverage problem to the set intersection problem, they provided formulas for k-coverage in the cases where sensors' sensing regions can have any arbitrary shapes. Gupat *et al.* [38]

investigated the problem of choosing the minimum number of sensors from a set of sensors that are uniformly deployed such that the FoI is covered. In [100], the authors have studied the asymptotic node density requirement for achieving k-coverage in uniformly random deployment and Poisson process deployment. Furthermore, Zhang and Hou [112] proposed a different method to derive the sufficient and necessary node densities for maintaining k-coverage for a given sensor network under both random and deterministic deployment strategies.

Clouqueur *et al.* [23] formulated the target detection problem as an unauthorized traversal problem and proposed deployment strategies which can achieve the required target detection probability and minimize the network cost. In the deployment strategy, only part of the sensors are randomly deployed if those sensors can ensure the coverage performance. In [103], the authors considered a hybrid sensor network consisting of static and mobile sensor. Static sensors are fixed to detect events and mobile sensors move closer to such events when they are detected. In [53], the authors mapped the target detection problem in sensor networks to a line-set intersection problem. Based on that, the probability of detecting targets is analytically evaluated for both deterministic and stochastic deployment. In [94], Snyder *et. al* proposed an efficient sensor movement strategy to detect the dynamic events in sparse mobile sensor networks. The locations and durations of events in the network are dynamic. Bai *et al.* [7] studied the problem of achieving optimal multiple-coverage with minimum number of sensor nodes. Their proposed results are helpful in constructing two-coverage deployment patterns with certain optimization.

4.3 Probabilistic Event Coverage

In this section, we derive the probability of event coverage when the sensors and events are distributed according to the spatial Poisson process. In order to do that, it firstly needs to derive the probability of area coverage.

4.3.1 System Model

Spatial Poisson Process

A spatial point process is a model for a random pattern of points in *d*-dimension ($d = 2$ in this paper) space. It can be used as statistical models in the analysis of observed patterns of locations of objects, such as animals in forests and sensors in a building. One of the most popular spatial point processes is the *spatial Poisson process* [5], which is formally defined as follows.

Definition 4.1 Spatial Poisson Process: For an FoI A_0 with area F_0, the spatial Poisson process, with a uniform intensity $\lambda > 0$, is a point process in $\mathbb{R}^2$ such that

1) The number of events $M(A_0)$ in the FoI A_0 has a Poisson distribution with mean λF_0;

2) If $A_0^1, \cdots, A_0^m$ are disjoint regions in A_0, then $M(A_0^1), \cdots, M(A_0^m)$ are independent.

The intensity λ is defined as the expected number of targets per unit area.

Network Model

Consider a two-dimension *Field of Interest* (FoI) $A \in \mathbb{R}^2$. Sensors are placed inside the FoI randomly according to a spatial Poisson process of intensity $\lambda > 0$. Thus for any region $A_i \subseteq A$, the number of sensors in A_i is given by a Poisson variable of mean $\lambda |A \cap A_i|$, where $|A \cap A_i|$ is the area of the region A_i. Assume all sensors have homogeneous sensing regions which can be irregular shapes. The event occurrence in the FoI is also modeled by the spatial Poisson point process, which is a random collection of points representing the location of events. A target event is said to be covered if and only its distance to the nearest sensor is equal to or less than the sensing range. Equivalently, given a target, the probability it is not covered by any sensors is equal to the probability that it is not within the range of any sensor or enough number of sensors. In event-driven sensor networks, the purpose is to detect event occurrences. Sensors start to work and collect information only when there are events within their sensing ranges.

In some applications of sensor networks, the probability of event occurrence in different regions of the FoI are different. In order to provide good coverage in these networks, we adopt the heterogeneous sensor deployment method. Namely, the density of sensor deployment in different regions of the FoI are different depending on the event occurrence probability. Details are given in Section 4.3.5.

4.3.2 Area Coverage

Basically, the probabilistic area coverage can be formulated as follows: A number of sensors are randomly deployed in an FoI. The areas and perimeters of the FoI and the sensing region of each sensor are given. We need to calculate the probability that a random point in the FoI is monitored by at least k sensors, namely, it is k-covered. In [52], the authors mapped this stochastic coverage problem into a set intersection problem and proposed an analytic method to derive the probability of k-coverage based on integrate geometry. The detailed descriptions are defined as follows:

Lemma 4.1 Let A_0 be an FoI with area F_0 and perimeter L_0 on a plane. N $(N \geq 1)$ sensors are uniformly randomly deployed in the FoI A_0. The area and perimeter of the sensing region of each sensor are denoted by F_s and L_s, respectively. Let $P_1(N,k)$ and $P_2(N,k)$ represent the probability that a randomly selected point in A_0 is covered by exact and at least k sensors, respectively. For $1 \leq k \leq N$, we have

$$P_1(N,k) = \frac{\binom{N}{k}(2\pi F_s)^k (2\pi F_0 + L_0 L_s)^{N-k}}{[2\pi(F_0+F_s)+L_0 L_s]^N}, \tag{4.1}$$

$$P_2(N,k) = 1 - \sum_{i=0}^{k-1} P_1(N,i) \tag{4.2}$$

The boundary condition is $P_1(N,0) = P_2(N,0) = 1$. For simplicity, we assume the sensing regions of all sensors are homogeneous. However, it is proposed in [52], the derivations can be easily extended to the cases with heterogeneous sensing regions as long as the area and perimeter of each sensor are given.

In our network model, sensors are deployed in the sensing field according to a spatial Poisson process. The probability of k-coverage can be derived as follows.

Lemma 4.2 Let A_0 be an FoI with area F_0 and perimeter L_0 on a plane. Sensors are deployed in the FoI A_0 according to a spatial Poisson process with intensity λ. The area and perimeter of the sensing region of each sensor are denoted by F_s and L_s, respectively. Let $P_{ex}(\lambda,k)$ and $P_{le}(\lambda,k)$ denote the probability that a randomly selected point in A_0 is covered by exact and at least k sensors, respectively. We have $P_c^{ex}(\lambda,0) = P_c^{le}(\lambda,0) = 1$, and for $k \geq 1$,

$$P_{ex}(\lambda,k) = \sum_{j=k}^{\infty} \frac{\binom{j}{k} e^{-\lambda F_0} (\lambda F_0)^j (2\pi F_s)^k (2\pi F_0 + L_0 L_s)^{j-k}}{j! \times [2\pi(F_0+F_s)+L_0 L_s]^j} \tag{4.3}$$

$$P_{le}(\lambda,k) = 1 - \sum_{i=0}^{k-1} P_{ex}(\lambda,i) \tag{4.4}$$

Proof: If the sensors are deployed according to a spatial Poisson process with intensity λ, the probability that there are j sensors in the FoI A_0 is given by,

$$\mathbb{P}\{N(A_0) = j\} = \frac{(\lambda F_0)^j}{j!} e^{-\lambda F_0} \tag{4.5}$$

Given there are j sensors, the probability of exact k-coverage is given by Eq. (4.1) . By summing up all the possibilities, we can get

$$\begin{aligned} & P_{ex}(\lambda, k) \\ &= \sum_{j=k}^{\infty} \mathbb{P}\{N(A_0) = j\} P_1(j, k) \\ &= \sum_{j=k}^{\infty} \frac{\binom{j}{k} e^{-\lambda F_0} (\lambda F_0)^j (2\pi F_s)^k (2\pi F_0 + L_0 L_s)^{j-k}}{j! \times [2\pi(F_0 + F_s) + L_0 L_s]^j} \end{aligned} \tag{4.6}$$

Consequently, the probability of at least k-coverage $P_{le}(\lambda, k)$ equals to one minus the sum of the probabilities of less than exact k-coverage.

4.3.3 Event Coverage

The objective of this work is to study the probabilistic event coverage problem, which is different from traditional area coverage problem in WSNs. Generally, the event coverage is to quantify how well the events in an FoI is detected by sensors. The probabilistic event coverage problem is described as: Given an FoI, events may happen in any points of the FoI, and multiple sensors are deployed in the FoI to detect the events. Both sensors and events are distributed according to the spatial Poisson process. We need to compute the probability that a random event can be detected by at least k sensors. In this problem, the probability of event coverage depends on not only the intensity of events but also the intensity of sensor deployment. The event coverage is formally defined as follows:

Lemma 4.3 In an FoI A_0 with area F_0 and perimeter L_0, sensors are deployed according to a spatial Poisson process with intensity λ. The area and perimeter of the sensing region of each sensor are denoted by F_s and L_s, respectively. Events may happen in A_0 and they are modeled by a spatial Poisson process with intensity γ. Let $P_{e-cov}(k)$ denote the probability that every event is detected by at least k sensors. Then, we have

$$\begin{aligned} & P_{e-cov}(k) \\ &= \sum_{n=1}^{\infty} \{\mathbb{P}[M(A_0) = n] [P_{le}(\lambda, k)]^n\} \\ &= \left\{ \sum_{n=1}^{\infty} \frac{(\gamma F_0)^n}{n!} e^{-\gamma F_0} [1 - \sum_{i=0}^{k-1} P_{ex}(\lambda, i)]^n \right\} \end{aligned} \tag{4.7}$$

where q_0 denotes the probability that a given event is detected by less than k sensors. The value of k depends on the requirements of applications. For example, $k = 1$ can meet the requirements of monitoring applications, while in tracking or positioning applications, k might be 2 or 3.

4.3.4 Boundary Analysis

For sensors deployed near the boundary of the FoI, part of the coverage regions are outside the FoI. The coverage analysis should take this issue into account.

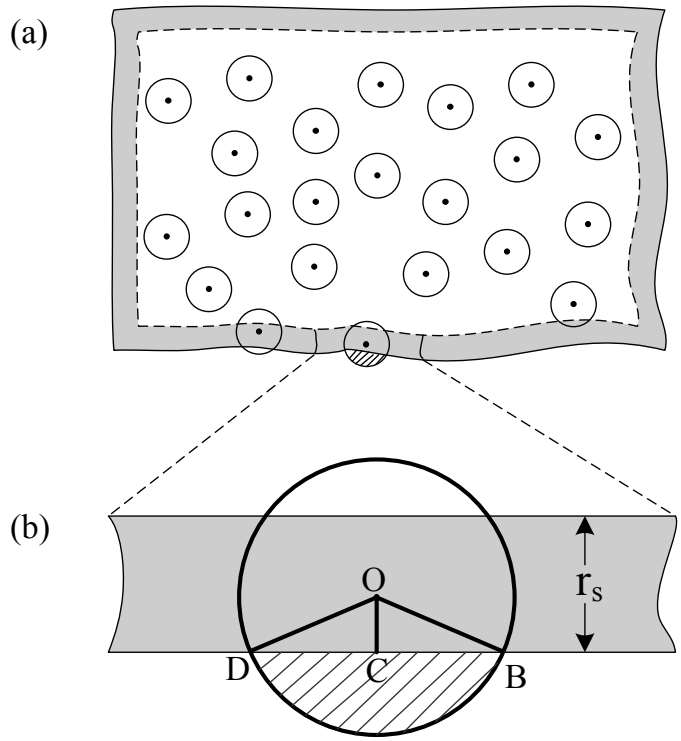


Figure 4.3. Boundary analysis

The *near boundary region* is defined as the the region with distance r_s to the boundary (Fig. 4.3-(a)), where r_s denotes the maximum sensing distance of the sensor s. As shown in Fig. 4.3, a sensor s is located at point O which is inside the near boundary region of the FoI. Let z $(0 < z < r_s)$ denote the distance between the sensor and the boundary of the FoI ($\overline{OC} = z$). From basic geometry knowledge, we know that the value of $\angle BOC$ equals to $\arccos(z/r_s)$.

Therefore, the area of the shadow region ($\frown DBC$) in Fig. 4.3-(b) is calculated by,

$$F_{shd} = r_s^2 \arccos \frac{z}{r_s} - z\sqrt{r_s^2 - z^2} \tag{4.8}$$

For the sensor s located in the near boundary region, the ratio of the coverage region inside the FoI is calculated by,

$$\eta_z = 1 - \frac{r_s^2 \arccos(z/r_s) - z\sqrt{r_s^2 - z^2}}{\pi r_s^2} \tag{4.9}$$

For a randomly deployed sensor in the near boundary region, the probability it is located in a strip (dz) with distance z to the boundary is given by,

$$q_z = \frac{(L_0 - \nu z)dz}{F_0 - F_0^{'}} \tag{4.10}$$

where $F_0^{'}$ denotes the area of the FoI excluding the near boundary region, ν is a constant depending on the shape of the FoI. If the FoI is a rectangle, the value of ν is 8. If the FoI is a circle, the value of ν is 2π. Hence, the mean coverage ratio η_0 is calculated by,

$$\eta_0 = \int_0^{r_s} \frac{L_0 - \nu z}{F_0 - F_0^{'}} \left[1 - \frac{r_s^2 \arccos \frac{z}{r_s} - z\sqrt{r_s^2 - z^2}}{\pi r_s^2}\right] dz \tag{4.11}$$

The detailed computation of Eq. (4.11) is described in the following.

Eq. (4.11) can be re-organized as follows,

$$\begin{aligned} \eta_0 = & \int_0^{r_s} \frac{L_0 - \nu z}{F_0 - F_0^{'}} dz \\ & + \int_0^{r_s} \frac{\nu r_s^2 z \arccos \frac{z}{r_s} - L_0 r_s^2 \arccos \frac{z}{r_s}}{(F_0 - F_0^{'})\pi r_s^2} dz \\ & + \int_0^{r_s} \frac{L_0 z\sqrt{r_s^2 - z^2} - \nu z^2\sqrt{r_s^2 - z^2}}{(F_0 - F_0^{'})\pi r_s^2} dz \end{aligned} \tag{4.12}$$

In the above equation, $r_s, F_0, F_0^{'}, L_0, \nu$ are non-integration variables. It is easy to compute the first part of Eq. (4.12) as follows,

$$\int_0^{r_s} \frac{L_0 - \nu z}{F_0 - F_0'} dz = \frac{2L_0 r_s - \nu r_s^2}{2(F_0 - F_0')} \tag{4.13}$$

According to the formula (306) in Chapter 7.4 of [28], we have

$$\begin{aligned} \int z \arccos(az) dz =& \frac{z^2}{2} \arccos(az) \\ & - \frac{1}{4a^2} \left[\arccos(az) + az\sqrt{1 - a^2 z^2} \right]. \end{aligned} \tag{4.14}$$

Based on the formula (304) in Chapter 7.4 of [28], we have

$$\int \arccos(az) dz = z \arccos(az) - \frac{1}{a}\sqrt{1 - a^2 z^2}. \tag{4.15}$$

Consequently, the second part of Eq. (4.12) can be calculated based on the above formulas.

Based on the formula (142) in Chapter 7.4 of [28], we can get,

$$\int z\sqrt{c^2 - a^2 z^2} dz = -\frac{1}{3a^2}(c^2 - a^2 z^2)^{3/2} \tag{4.16}$$

Moreover, from the formula (143) in Chapter 7.4 of [28], we have

$$\begin{aligned} \int z^2\sqrt{c^2 - a^2 z^2} dz =& -\frac{z}{4a^2}(c^2 - a^2 z^2)^{\frac{3}{2}} + \frac{c^2 z}{8a^2}\sqrt{c^2 - a^2 z^2} \\ & + \frac{c^4}{8a^3} \arcsin(az/c). \end{aligned} \tag{4.17}$$

Therefore, the third part of Eq. (4.12) can be computed accordingly.

Based on the above formulas, η_0 can be calculated.

So the amended event coverage probability with the consideration of boundary condition can be derived by the following equation,

$$P'_{e-cov}(k) = \frac{F_0' + (F_0 - F_0')\eta_0}{F_0} \cdot P_{e-cov}(k) \tag{4.18}$$

4.3.5 Probabilistic Event Coverage in Heterogeneous Networks

In traditional coverage analysis methods, the heterogeneity of events are usually not considered and the sensors are uniformly deployed with the same density

over the FoI. In many sensor network applications, there are surveillance regions with different types of events which require different numbers of sensors. Moreover, in some scenarios, the probabilities of event occurrence in different regions of the FoI are different. For example, the studies in [39] show that wildfire events occur in clusters. For those applications with heterogeneous event distributions, it is better to deploy various numbers of sensors in different regions of the FoI. In the following, we present the coverage analysis for sensor networks with homogeneous and heterogeneous sensor deployment.

Event Coverage in Heterogeneous Networks

In heterogeneous sensor networks, the probabilities of event occurrence in different parts varies. Considering the heterogeneity of event occurrence, the sensor deployment is also heterogeneous. Let the FoI be divided into multiple sub-FoIs $A_1, A_2, \cdots, A_H$. The area, perimeter and event density of the region A_h $(1 \leq h \leq H)$ are F_h, L_h, and γ_h, respectively. Let λ_h denote the sensor density in region A_h. If we consider the difference of event densities and deployment sensors with different densities, the probability of event coverage in the region A_h can be expressed as,

$$P_h^{cov}(k) = Q \cdot \left\{ \sum_{n=1}^{\infty} \frac{(\gamma_h F_h)^n}{n!} e^{-\gamma_h F_h} [1 - \sum_{i=0}^{k-1} P_{ex}(\lambda_h, i)]^n \right\} \tag{4.19}$$

where $P_{ex}(\lambda_h, i)$ is derived by Eq. (4.3). Q is the emendatory coefficient, which is computed by (refer to Section 4.3.4 for details),

$$Q = \frac{F_h^{'} + (F_h - F_h^{'})\eta_0}{F_h} \tag{4.20}$$

where $F_h^{'}$ denotes the area of A_h excluding the near boundary region.

In the homogeneous deployment case, the heterogeneity of event occurrence is not considered and the sensors are deployed with the same density. The computation of event coverage probability is similar except that the values of λ_h in Eq. (4.19) are equal. It is a special case of heterogeneous deployment.

The overall event coverage probability of the network can be calculated by the geometric mean of the event coverage probabilities of all the regions,

$$P_{net}^{cov}(k) = \sqrt[H]{\prod_{h=1}^{H} P_h^{cov}(k)} \tag{4.21}$$

.

The use of a geometric mean normalizes the ranges being averaged, so that no range dominates the weighting, and a given percentage change in any of the properties has the same effect on the geometric mean.

Overlap between Sub-FoIs

In Section 4.3.4, we derive the coverage analysis with the consideration of boundary condition. However, the boundary analysis can not be directly applied to the heterogeneous network since there are coverage overlaps between sub-FoIs. So we propose a method to analyze the overlap between sub-FoIs.

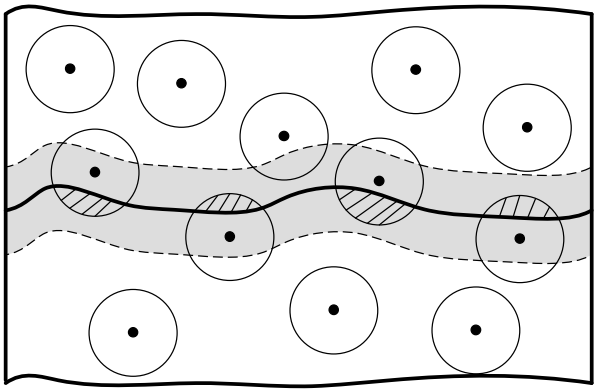


Figure 4.4. Coverage overlap

The two-sub-FoI case is taken as an example. As shown in Fig. 4.4, the FoI A_0 is divided into two sub-FoIs, A_1 and A_2. Let F_1, L_1, and γ_1 be the area, perimeter and event density of A_1, respectively. Let F_2, L_2, and γ_2 be the area, perimeter and event density of A_2, respectively. We consider the difference of event densities and deploy sensors with different densities (denoted by λ_1 and λ_2) in A_1 and A_2.

According to the analysis in Section 4.3.4, for a sensor located in the near boundary region of A_2 with distance z to the boundary, the ratio of its coverage region overlapping with A_1 can be calculated as,

$$\zeta_z = 1 - \eta_z = \frac{r_s^2 \arccos(z/r_s) - z\sqrt{r_s^2 - z^2}}{\pi r_s^2} \tag{4.22}$$

For a randomly deployed sensor in the overlap region, the probability with which it is located in a strip (dz) with distance z to the boundary is given by,

$$q_z^{'} = \frac{l_{1,2}}{F_2 - F_2^{'}} dz \tag{4.23}$$

where $F_2^{'}$ denotes the area of the A_2 excluding the overlap region, $l_{1,2}$ denotes the length of the border of A_1 and A_2. So the average coverage contribution from a sensor in A_2 is,

$$\zeta_{1,2} = \int_0^{r_s} \zeta_z \frac{l_{1,2}}{F_2 - F_2^{'}} dz \tag{4.24}$$

The computation of $\zeta_{1,2}$ can be found in the Appendix. Therefore, the event coverage probability is calculated by,

$$\widetilde{P}_1^{cov}(k) = P_1^{cov}(k) \cdot \frac{F_1^{'}\lambda_1 + (F_1 - F_1^{'})\lambda_2\zeta_{1,2}}{F_1\lambda_1} \tag{4.25}$$

where $P_1^{cov}(k)$ is calculated according to Eq. (4.19). And $P_2^{cov}(k)$ can be similarly computed.

The overall event coverage probability of the whole FoI A_0 is computed by,

$$P_{net}^{cov}(k) = \sqrt{\widetilde{P}_1^{cov}(k) \cdot \widetilde{P}_2^{cov}(k)} \tag{4.26}$$

We extend the analysis to a multi-sub-FoI network where a sub-FoI may have multiple neighbors (i.e., adjacent sub-FoIs). Let A_h denote the sub-FoI which has multiple neighbors (denoted by $NBG(A_h)$). For any of the neighbor A_i, the average coverage contribution from a sensor in A_i to A_h is derived by (according to Eq. (4.22), (4.23), (4.24)),

$$\zeta_{h,i} = \int_0^{r_s} \frac{r_s^2 \arccos(z/r_s) - z\sqrt{r_s^2 - z^2}}{\pi r_s^2} \cdot \frac{l_{h,i}}{F_h - F_h^{'}} dz \tag{4.27}$$

Consequently, the event coverage probability can be computed by,

$$\widetilde{P}_h^{cov}(k) = P_h^{cov}(k) \prod_{A_i \in NBG(A_h)} Q_{h,i} \tag{4.28}$$

where $Q_{h,i}$ is derived by

$$Q_{h,i} = \frac{F_h^{'}\lambda_h + (F_h - F_h^{'})\lambda_i\zeta_{h,i}}{F_h\lambda_h}. \tag{4.29}$$

After we compute the event coverage probability of each sub-FoI, the overall event coverage probability of the FoI can be calculated according to Eq. (4.21).

4.3.6 Experimental Results

To evaluate the coverage analysis method and to show the benefits and accuracy of the analysis method in heterogeneous sensor networks, numerical experiments and simulations are conducted. The analytic and simulation results of the probability of event coverage are presented and compared under different settings of sensor and event densities.

Analytical Results

In the following experiments, the probability of event coverage in homogeneous sensor networks is studied, namely, the sensor density and event density in different locations of the FoI are the same. The FoI is a rectangle with length and width of $20\ m$ and $16\ m$, respectively. The sensing region of each sensor is a disc with radius 3 m. So the area and perimeter of the sensing region are $28.3\ m^2$ and $18.8\ m^2$, respectively. The sensor density λ denotes the average number of sensor in an unit area. Similarly, the event density γ represents the average number of event in an unit area.

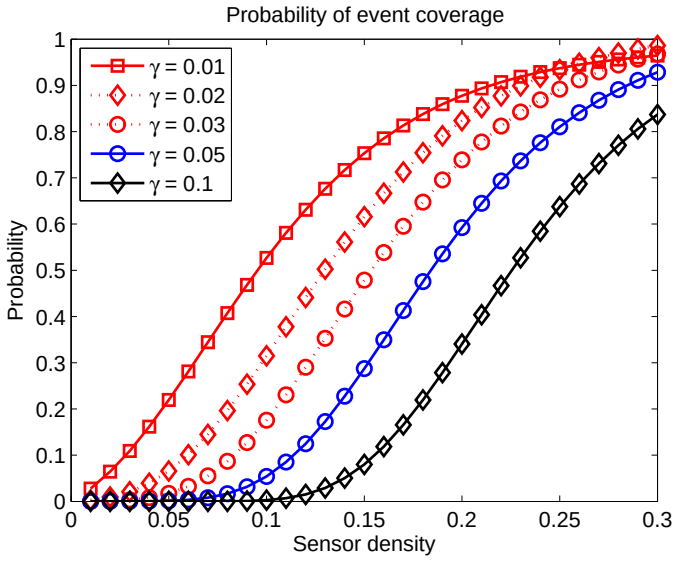


Figure 4.5. Probability of event coverage: 1-coverage

Fig. 4.5 and Fig. 4.6 illustrate the probabilities that every event in the FoI is covered by at least one sensor and two sensors, respectively. As we can see in

both figures, for the same event density, the probability of event coverage increases when the sensor density increases. For the same sensor density, when the event density increases, the probability of event coverage decreases. Given the event density in an application and the requirements on event coverage, these analytic results can be useful in determining the density of deployment and the total amount of sensors.

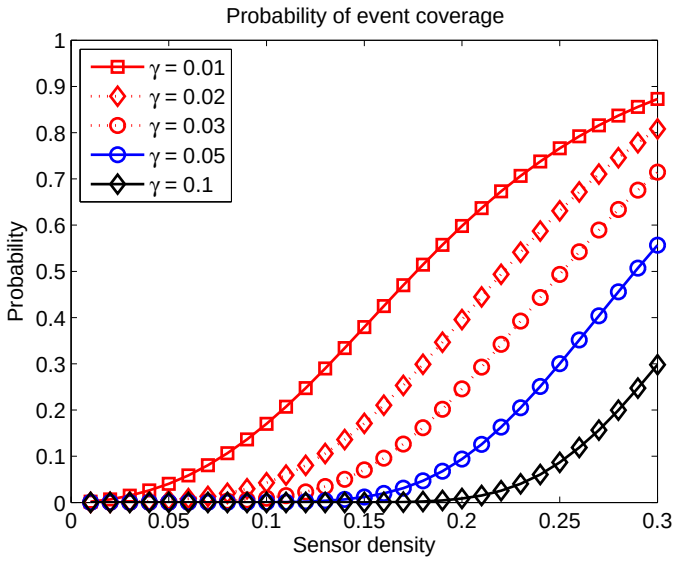


Figure 4.6. Probability of event coverage: 2-coverage

Fig. 4.7 shows the probabilities that every event in the FoI is at least k-covered ($k = 1, 2, 3, 4$) when the event density changes. The sensor density is set to 0.3. Similar with those in Fig. 4.5 and Fig. 4.6, we can find in this figure that the probability of event coverage decreases when the event density increases. Moreover, when the requirement on k-coverage is increased, the probability of event coverage significantly decreases. For example, when the event density is 0.15, the probability of 1-coverage and 2-coverage are 0.73 and 0.16, respectively. These results indicate that if the applications have higher coverage requirements, much more sensors need to be deployed. Consequently, the total deployment costs can be significantly increased.

The following experiments are performed to show the benefits of heterogeneous sensor networks. The coverage region of a sensor is the same as that in the previous section. In the network settings, the FoI is divided into sub-FoIs, and the event densities in different sub-FoIs are different. In homogeneous deployment, the

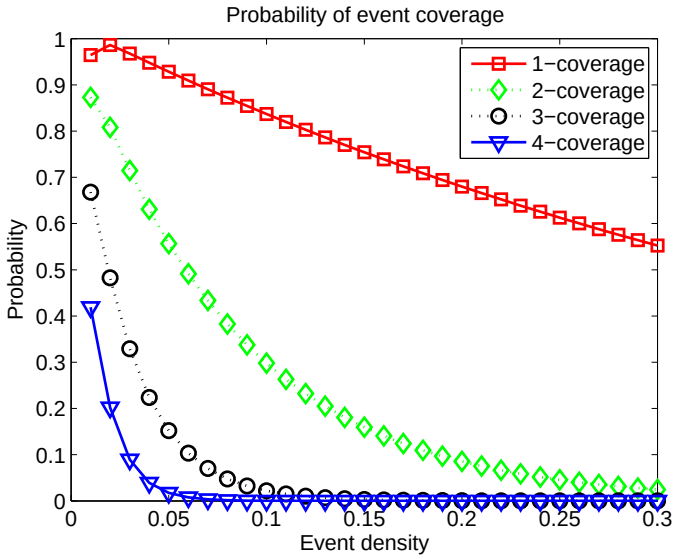


Figure 4.7. Probability of event coverage with different values of k ($\lambda = 0.3$)

sensor density in all sub-FoIs is the same. In heterogeneous deployment, the sensor density in different sub-FoIs are different. In order to implement fair comparisons, the average sensor density in the FoI of these two deployment scenarios are set to be equal. The overall event coverage probability of the network is calculated by the geometric mean of the probabilities of all the sub-FoIs (Eq. (4.21)).

In Fig. 4.8, we compare the probabilities of 1-coverage (1-cov) and 2-coverage (2-cov) under homogeneous and heterogeneous deployment in a two-sub-FoI network. We also consider the overlapping between adjacent sub-FoIs in the heterogeneous deployment. From this figure, we can see that the probabilities of both 1-coverage and 2-coverage of heterogeneous deployment are better than those of homogeneous deployment. It indicates that heterogeneous deployment can achieve better coverage performance. Moreover, considering overlapping between adjacent sub-FoIs can improve the event coverage probabilities since the sub-FoI can get coverage compensation contributed by the sensors in the near boundary region of its adjacent sub-FoI. In addition, the simulation results show that the results of overlapping analysis are accurate.

In this experiment, the FoI is divided into sub-FoI 1 and sub-FoI 2. The event density in sub-FoI 1 is $\gamma_1 = 0.03$, and that (γ_2) in sub-FoI 2 is changed from 0.04 to 0.16. In Fig. 4.9, we can see that the event coverage probabilities in heterogeneous deployment are larger than those in homogeneous deployment. Moreover, when

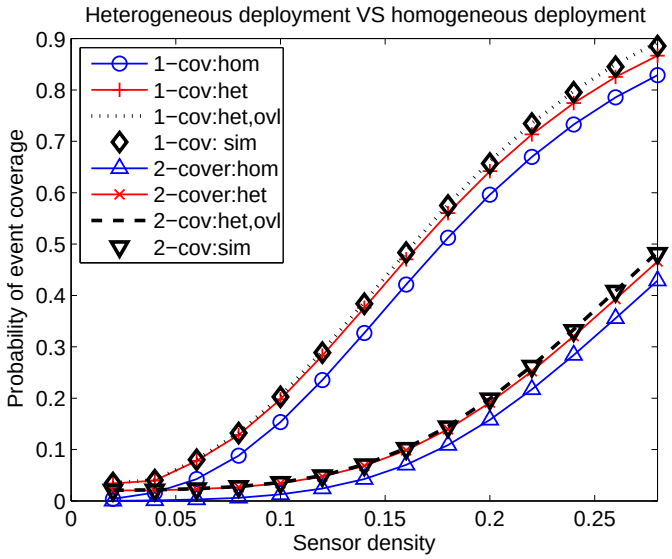


Figure 4.8. Event coverage probabilities in two-sub-FoI networks (event density $\gamma = 0.03, 0.06$). 'hom' denotes homogeneous deployment; 'het' denotes heterogeneou deployment; 'ovl' means 'overlapping' between adjacent sub-FoIs are considered.

$\gamma_1 = 0.03, \gamma_2 = 0.16$, the gap of the event coverage probability between the two deployment methods are bigger than those for $\gamma_1 = 0.03, \gamma_2 = 0.12(0.08/0.04)$. The comparisons indicate that the coverage performance of heterogeneous deployment is better than that of homogeneous deployment. And the improvement is more significant when the heterogeneity of event density is more obvious.

The following experiment is conducted to check the scaling property of the heterogeneous deployment method. The FoI is divided into 8 regions with the event densities in region i being $\gamma_i = i \times 0.02$ $(i = 1, \cdots, 8)$. In Fig. 4.10, $'Hom'$ denotes homogeneous deployment; $'Het - j'$ denotes heterogeneous deployment with j sub-FoIs $(j = 2, 4, 8)$. The average sensor densities in the above four cases are the same. As we can see from the figure, the coverage probability of $Het - 8$ is bigger than all the others cases. It means, for the network with heterogeneous event occurrence, heterogeneous sensor deployment leads to better coverage performance. The results of our analysis method can provide guidelines for constructing sensor deployment strategies in real applications.

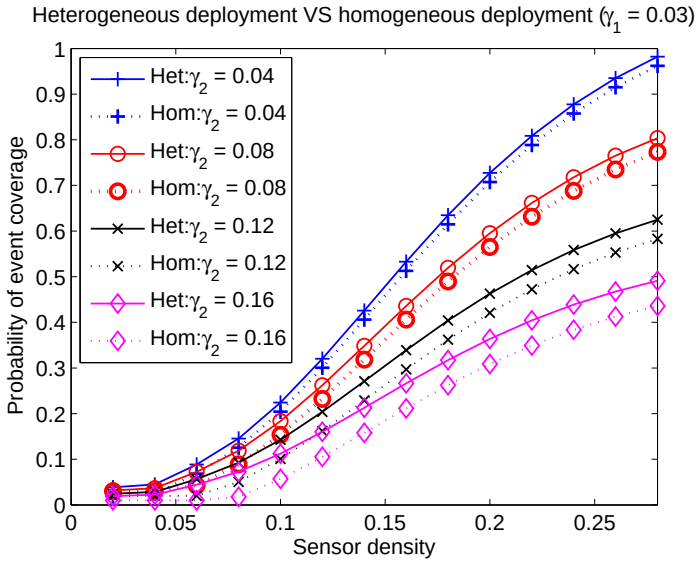


Figure 4.9. Comparison of the event coverage probabilities of homogeneous and heterogeneous sensor deployments in two-sub-FoI networks ($\gamma_1 = 0.03$, γ_2 are changed)

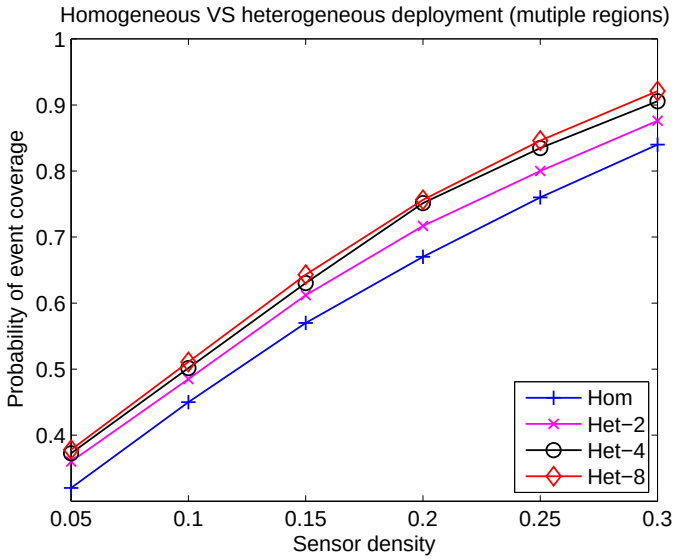


Figure 4.10. Comparisons of the event coverage probabilities in multi-sub-FoI networks

Validation through Simulations

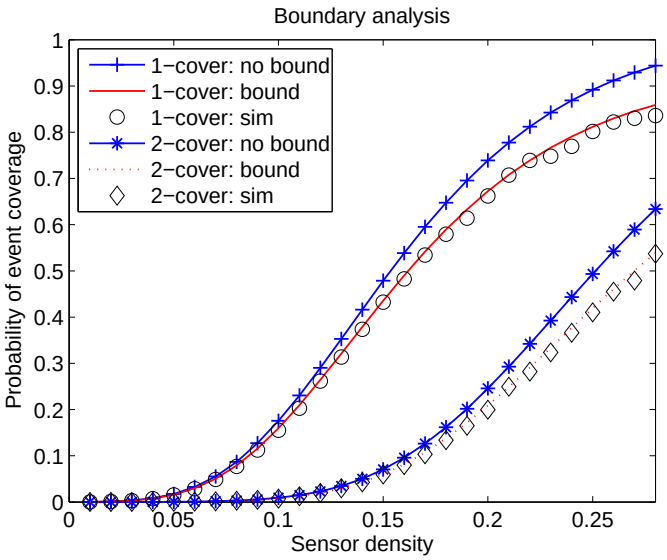


Figure 4.11. Boundary analysis ($\gamma = 0.03$)

In order to validate our proposed analytical method, numerical experiments and simulations are conducted with the same settings. The experiments are also used to study the impact and accuracy of boundary analysis (Section 4.3.4). Fig. 4.11 shows the comparison of event coverage probability in two cases: ignore the boundary of FoI, and consider the boundary conditions. The event density is set to 0.03. From the figure, we can see that the probability of both 1-coverage and 2-coverage are bigger if the boundary of FoI is ignored. Moreover, the difference between those two cases becomes larger and larger with the increase of sensor density. It means that the event coverage is over-estimated if the boundary condition is ignored, and the sensor density has impacts on the accuracy of the coverage probability. In addition, the simulation results are coincident with the numerical results of boundary analysis. It indicates good accuracy of the proposed event coverage analysis with boundary condition.

As shown in Fig. 4.11, there is over-estimation of the coverage probability if we do not consider the boundary of the FoI. In the following experiment, we investigate how the size of the FoI impacts the ratio of over-estimation (relative difference). The area of FoI changes from $200\ m^2$ to $400\ m^2$. The percentage (Y-axis) denotes the relative difference of the event coverage probability between the

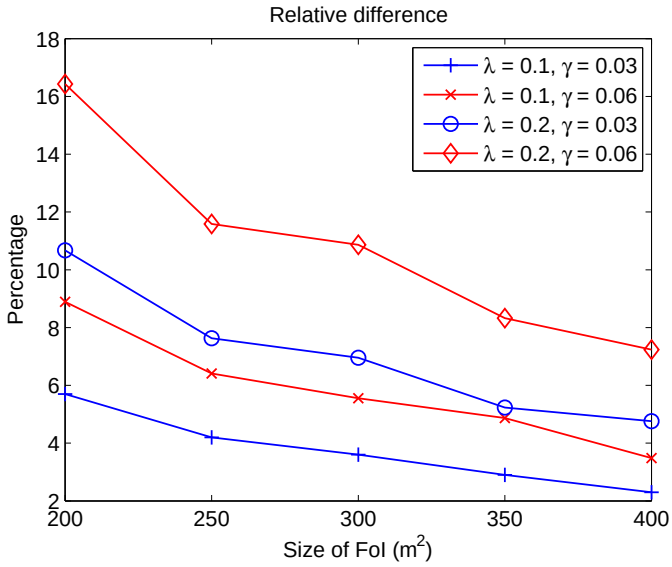


Figure 4.12. Relative error of boundary analysis($\gamma = 0.03$)

ignoring boundary case and the considering boundary case. From Fig. 4.12, we can see that the relative different decreases when the size of FoI increases. It means that for small-scale sensor networks, event coverage analysis with boundary conditions can give much more accurate results. Moreover, with the same sensor density, the relative difference between the two cases is bigger if the event density is larger. In summary, the comparisons between the numerical and simulation results validate the accuracy of our proposed event coverage analysis method.

4.4 Deployment Strategies

WSNs generally have two fundamental application scenarios: tracking and monitoring. In both applications, it is essential to ensure that information of the target or the environment can be discovered and collected by sensors. To achieve good coverage, sensors are usually densely deployed. Moreover, the lifetime of WSN is determined by the energy budgets of sensors. To obtain longer network lifetime, more energy budgets should be assigned to sensors. Since sensors are usually equipped with batteries which are limited and expensive, deploying spare sensor nodes would cause high installation and maintaining costs. Therefore, in

order to deploy minimum necessary sensors that can achieve the requirements, it is important to evaluate these performance metrics of a WSN before its deployment.

Performance analysis such as coverage, connectivity, energy consumption, cost for sensor networks has been studied by many researchers. In [20], the authors proposed a general framework for the analysis of the network lifetime and costs for several network deployment strategies in sensor networks. They investigated deployment strategies to maximize the network lifetime by mitigating the hot-spot traffic problem around the data sink. The coverage problem in sensor networks has been extensively investigated by researchers [112, 8, 72]. Zhang and Hou [112] studied the problem of deriving the node density for maintaining k-coverage for a given network area in both random and deterministic deployment strategies. However, most of the attentions focused on analyzing the minimum coverage. In [88], the authors proposed an analytical method to model the event coverage problem in sensor networks. In [72], the authors investigated the coverage, energy consumption and message transfer delay of large-scale WSNs. They considered the square grid based deployment scheme which shows very good coverage performance and the Tri-Hexagon Tiling (THT) deployment strategy. Their scheme outperforms other schemes for energy consumption and worst-case delay. However, the THT scheme shows poor performance for coverage and planning overhead. Moreover, since the topology of THT is more complex than the mesh-based deployment scheme, it is more difficult and expensive to configure and maintain the network.

In this section, we introduce techniques for analyzing the coverage, network lifetime and cost of random deployed and regular sensor networks. Different from most works which focused on the minimum k-coverage analysis, we put emphasis on analyzing the coverage represented by average k-coverage and the variance of exact k-covered points, which can describe the coverage performance more accurately than minimum k-coverage. Moreover, energy model and cost model are applied to investigate the network lifetime and installation cost of WSNs with different deployment strategies. Our results can be useful in the following aspects. First, our method is capable of evaluating the advantages and disadvantages of different deployment strategies. Second, our results can provide guidelines in designing deployment schemes which can satisfy the performance requirements.

4.4.1 Definitions and Models

This section contains descriptions of network topology, node deployment schemes, and sensing and coverage model.

Network Topology

In a large scale sensor network, the sensors can be organized in various topologies, such as mesh, star, and cluster-based topologies. Moreover, they can be organized as combinations of these topologies. One of the most popular candidates is the cluster-mesh topology [54], which merges the advantages of mesh and cluster networking. The cluster-mesh WSN contains three types of nodes: *sink*, *cluster head (CH)*, and sensor node *(SN)*. Like in most sensor networks, the sink can be located either inside or near the FoI. The sink is responsible for controlling the network and aggregating the sensing data from all the other nodes. The network is composed of multiple clusters. A cluster is constructed by grouping a cluster head and multiple sensor nodes within a geographic neighborhood. Inside a cluster, the information sensed by sensor nodes is firstly sent to the cluster head and then delivered to the sink through neighbor cluster heads.

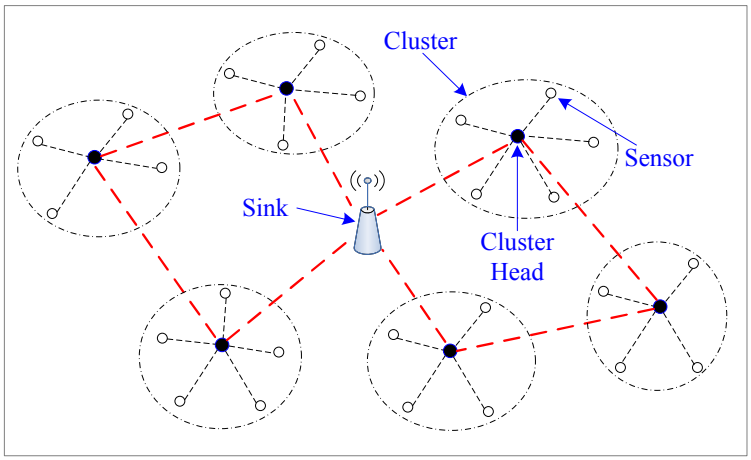


Figure 4.13. System Architecture.

Node Deployment Schemes

According to the features of the FoI, the sensor nodes can be deployed in either random or deterministic manner. For example, in the volcano and forest monitoring

application of WSNs [104, 4], sensor nodes may be thrown out from an airplane, which can lead to a random deployment. While in applications of WSNs for bridge or building monitoring [68], sensors can be deterministically deployed according to regular patterns.

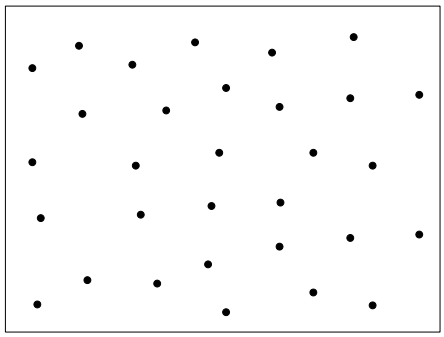


Figure 4.14. Uniform random

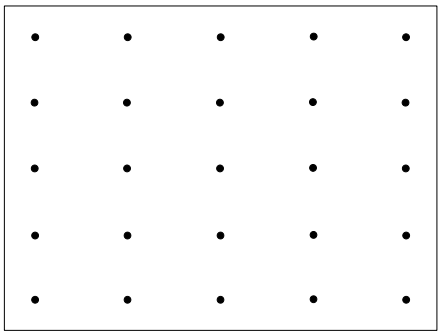


Figure 4.15. Rectangle mesh

In this work, we consider the uniformly random deployment scheme (Fig. 4.14) which is a typical random scheme, and the rectangle mesh scheme (Fig. 4.15) which belongs to the deterministic deployment category. In the uniformly random scheme, each SN has equal probability of being placed at any point. While in the rectangle mesh scheme, SNs are placed according to a regular mesh. In [72], the author considered the square grid based deployment scheme which shows very high coverage performance. Here, we extend the work by considering the rectangle mesh pattern, which can be more useful and applicable than the square grid scheme. In [72], the author introduced the Tri-Hexagon Tiling (THT) deployment strategy, which outperforms other scheme for energy consumption and worst-case delay. However, it shows poor performance for coverage and planning overhead. The planning overhead includes time and labor for deployment. Moreover, since the topology of THT is more complex than the mesh-based deployment scheme, it is more difficult and expensive to configure and maintain the network. That is why we choose the rectangle mesh as the representing deterministic deployment scheme.

4.4.2 Sensing and Coverage Model

The popular disc-based sensing model [6] is adopted, where each SN has a sensing range of r_s. Any point within the circle of radius r_s centered at a SN is covered by it. An FoI is said to have k-coverage if every point of it is covered by

at least k sensor nodes. Assume that the maximum communication range of a SN is bigger than its sensing range.

In the following, we introduce the analysis of three performance metrics: coverage, lifetime and cost. The variables are defined in Table 4.1.

4.4.3 Coverage Analysis

Since the minimum k-coverage is not accurate enough to represent the actual coverage performance, a method is proposed to investigate the average k-coverage and its variance. Our method can be more accurate in evaluating the coverage performance of WSNs than the minimum k-coverage method. For example, if the values of minimum k-coverage of two deployment schemes are the same, their coverage performance could be different if their average k-coverage values are different. Moreover, even if the average k-coverage of two WSNs are the same, their coverage performance may be different if their variances of k-coverage are different.

In the rectangle mesh scheme (Fig. 4.15), Fig. 4.16 shows the k-coverage map of all possible k-coverage values of a rectangle cell. It is assumed that the sensing range equals to the width of the rectangle. In order to guarantee full coverage, there should be some overlap among the sensing regions of sensors. The amount of overlapping can be determined by the coverage requirements.

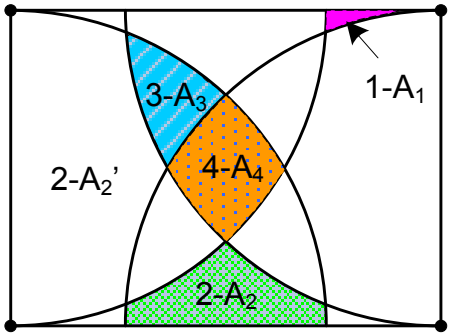


Figure 4.16. k-coverage map.

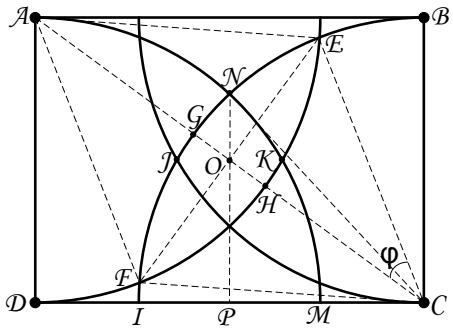


Figure 4.17. k-coverage map for calculation

In order to compute the average k-coverage value and its variance, we need to compute the area of each part, i.e., the values of A_1, A_2, A_2', A_3, A_4. The k-coverage problem can be modeled by using basic geometry (Fig. 4.17). Assume the width and length of the rectangle are l_1 and l_2, respectively. The sensing range $r_s = l_1$. In order to ensure full coverage, i.e., every point in the FoI is covered

Table 4.1. Variables and Descriptions

Variable	Description
N_0	Total number of sensor nodes
N_s	Number of sensor nodes in a cluster
N_c	Number of clusters
SN_i	Sensor node i
CH_i	Cluster head i
A_{net}	Total area of the network
d_{max}^{SN}	Maximum transmission distance of a SN
d_{max}^{CH}	Maximum transmission distance of a CH
r	Sensing range of a SN
d_{k0}	Distance between sensor k to its cluster head
D_i	Data generation rate of SN i
E_i^{SN}	Initial energy of SN i
E_i^{CH}	Initial energy of CH i
d_{ij}	Distance between cluster head i and j
E_{cir}	Energy of electronic circuit for transmit/receive one bit
ϵ_a	Transmitter amplifier coefficient
θ	Path loss exponent
θ_1	Path loss exponent (intra-cluster)
θ_2	Path loss exponent (inter-cluster)
C_i^{SN}	Cost of SN_i
C_i^{CH}	Cost of CH_i
C_{net}	Total cost of the network

at least by one sensor, l_1 and l_2 should satisfy $l_1 \leq l_2 \leq \sqrt{3}l_1$. Let φ denote the value of $\angle\mathcal{ECO}$. Let φ_1 denote the value of $\angle\mathcal{NCP}$. As shown in Fig. 4.17, let A_a, A_b, A_c, A_d, A_e denote the area of shape $\widehat{\mathcal{FGEHF}}$, $\widehat{\mathcal{EBCIFHE}}$, $\widehat{\mathcal{BJCB}}$, $\widehat{\mathcal{CJIC}}$, $\widehat{\mathcal{MNIM}}$, respectively. We can compute the area of each part (Fig. 4.16) using the following equations.

$$\begin{aligned} A_a &= 2\varphi l_1^2 - \frac{1}{2}\sqrt{(3l_1^2 - l_2^2)(l_1^2 + l_2^2)} \\ A_b &= \frac{\pi}{4}l_1^2 - A_a \\ A_c &= \left(\frac{\pi}{3} - \frac{\sqrt{3}}{4}\right) l_1^2 \\ A_d &= \frac{\pi}{4}l_1^2 - A_c \\ A_e &= \varphi_1 l_1^2 - \frac{1}{2}l_1 l_2 \sin(\varphi_1). \end{aligned} \tag{4.30}$$

where $\varphi = \arctan\left[\sqrt{(3l_1^2 - l_2^2)/(l_1^2 + l_2^2)}\right]$ and $\varphi_1 = \angle\mathcal{NCP} = \arccos\left[l_2/(2l_1)\right]$.

Hence, the areas of A_1, A_2', A_2, A_3, A_4 can be calculated as following:

$$\begin{aligned} A_1 &= \frac{1}{2}\left(l_1 l_2 - \frac{\pi}{4}l_1^2 - A_b\right) \\ A_2' &= l_1 l_2 - \frac{\pi}{4}l_1^2 - A_d - 2A_1 \\ A_3 &= \frac{\pi}{4}l_1^2 - A_e - A_1 - A_2' \\ A_2 &= \frac{\pi}{4}l_1^2 - A_a - A_1 - A_2' - A_3 \\ A_4 &= A_e - A_2 - 2A_3. \end{aligned} \tag{4.31}$$

The average k-coverage is defined as the total k-coverage divided by the total area.

$$\widetilde{k}_{rm} = \frac{1}{l_1 l_2}\left[4A_1 + 4(A_2 + A_2') + 12A_3 + 4A_4\right] \tag{4.32}$$

Accordingly, the variance of the k-coverage value can be calculated by,

$$\begin{aligned} var(\widetilde{k}_{rm}) = &\frac{1}{l_1 l_2}\Big[4A_1(1 - \widetilde{k}_{rm})^2 + 2(A_2 + A_2')(2 - \widetilde{k}_{rm})^2 \\ &+ 4A_3(3 - \widetilde{k}_{rm})^2 + A_4(4 - \widetilde{k}_{rm})^2\Big] \end{aligned} \tag{4.33}$$

The values of $\widetilde{k}$ and $var(\widetilde{k})$ above are calculated from the rectangle which is composed by four sensor nodes. For large scale sensor networks, the boundary

conditions can be ignored. Thus, the average k-coverage and its variance of the whole network can be accurately expressed by $\widetilde{k}$ and $var(\widetilde{k})$, respectively.

Different from the deterministic deployment strategy, in the uniform random deployment strategy, it is difficult or impossible to guarantee full coverage. From this aspect, the uniform random deployment is not as good as the deterministic deployment. The average k-coverage can be calculated by,

$$\widetilde{k}_{ur} = N_0 \pi r_s^2 / A_{net} \tag{4.34}$$

Since the sensors are randomly scattered in the network, it is very difficult or impossible to explicitly compute the variance of k-coverage, but it can be calculated through simulations.

4.4.4 Lifetime Analysis

In order to compute the network lifetime, we first need to introduce the energy model. We adopt the most popular two-order model [75], where the energy consumption of node i for transmitting one bit over distance d_{ij} is $e_i^{Tx} = E_{cir} + \epsilon_a (d_{ij})^{\theta}$, where θ denotes the path loss exponent. And the energy for receiving one bit is $e_i^{Rx} = E_{cir}$.

In the system architecture, a sensor directly sends data to its cluster head, and a cluster head directly sends data to the central server. Under this model, the power consumption of a SN (j) and a cluster head (i) can be expressed by,

$$\begin{aligned} P_j^{SN} &= \left[E_{cir} + \epsilon_a (d_{j0})^{\theta_1} \right] D_j \\ P_i^{CH} &= \sum_{j=1}^{N_S} D_j \left[2E_{cir} + \epsilon_a (d_{i0})^{\theta_2} \right] \end{aligned} \tag{4.35}$$

where d_{j0} denotes the distance between sensor j to its CH, and d_{i0} denotes the distance between CH i to the central server; D_j (bit/s) denotes the data generate rate of the sensor j. If the sensors can vary transmission power to accommodate the distance over which they transmit, the transmission power required to deliver data from the sensor j to its CH will be controlled in such a way that the transmission distance d_{j0} equals the physical distance. If the sensors use a fixed transmission power level, then the transmission distance can be considered as $d_{j0} = d_{max}^{SN}$.

In general, lifetime of a sensor network is defined as the time after which certain fraction of sensor nodes run out of their energies, resulting in a routing hole within the network. Moreover, death of a cluster head would also bring routing

problems. Hence, the lifetime is defined as the time after which a certain fraction of sensors run out of their energies or one cluster head runs out of its energy.

According to the power consumption model, the lifetime of $CH\ i$ and $SN\ j$ can be expressed as,

$$T_i^{CH} = \frac{E_i^{CH}}{P_i^{CH}}, \qquad T_j^{SN} = \frac{E_j^{SN}}{P_j^{SN}} \tag{4.36}$$

Let T_{th}^{SN} denote the time after which a certain fraction (denoted by q_0) of SNs run out their energy. Then, the network lifetime can be expressed as,

$$T_{net} = \min \left\{ \min_{1 \le i \le N_c} T_i^{CH}, \quad T_{th}^{SN} \right\}$$

T_{th}^{SN} can be calculated in this way: assume there are total $N_0 = 100$ SNs and $q_0 = 10\%$. We calculate the lifetimes of all SNs and sort them in the ascending order. Then, we take the 10^{th} $(\lceil N_0 * q_0 \rceil)$ value as T_{th}^{SN}.

4.4.5 Cost Analysis

The cost of the network is composed of the costs of SNs, CHs, and the deployment overhead. For the SN and CH, the cost consists of hardware cost (circuit) and energy cost (battery). For different types of sensor nodes and cluster heads, their costs are different. For a cluster head of type i, the cost of CH_i can be modeled as $C_i^{CH} = a_i^0 + b_i^0 E_i^{CH}$, where a_i^0 and b_i^0 denote the hardware cost and the proportionality constant for the battery cost, respectively. For a sensor of the type j, the cost of SN_j can be modeled as $C_j^{SN} = a_j^1 + b_j^1 E_j^{SN}$, where a_j^1 and b_j^1 denote the hardware cost and the proportionality constant for the battery cost, respectively. In order to consider the impact of deployment schemes, we define a parameter C_{over} to model the deployment overhead. For example, the deployment overhead a random scheme may be less than that of a deterministic scheme.

Depending on the applications, there are homogeneous and heterogeneous sensor networks. For the homogeneous sensor network, there are only one type of sensor node and one type of cluster head. The overall cost of a network can be expressed by,

$$C_{net} = N_c \cdot C^{CH} + N_c \cdot N_s \cdot C^{SN} + C_{over} \tag{4.37}$$

For the heterogeneous sensor networks, there are multiple types of sensor nodes and cluster heads. The overall cost of a network can be expressed by,

$$C_{net} = \sum_i N_c^i \cdot C_i^{CH} + \sum_j N_s^j \cdot C_j^{SN} + C_{over} \tag{4.38}$$

where N_c^i denotes the number of CHs of type i, and N_s^j denotes the number of SNs of type j.

In the cost model, we take into account every aspect of setting up sensor networks. It is a simple but effective method for comparing different deployment schemes on an equal basis [20].

4.4.6 Evaluation Results

To compare the three performance metrics of the random and regular node deployment scheme, performance evaluation is conducted in this section. In each experiment, a total number of $N_0 = 128$ sensor nodes are distributed in a rectangle FoI with the sink placed in the center. For simplicity, the average data generation rate for a sensor is set to unit rate, namely $D_j = 1\ bit/s$. It is straightforward to extend the analysis with other data rates. We conduct three types of experiments with the network divided into 2, 8, and 32 clusters, which means that a CH controls 64, 16 and 4 SNs, respectively. The Dijkstra's shortest routing algorithm is applied to deliver the data from a source CH to the sink. Other parameters used in the experiments are listed in Table 4.2 as following.

Table 4.2. Parameters

Variable	Value	Unit	Variable	Value	Unit
E_{cir}	50	nJ/bit	a^1	10	-
ϵ_a^1	100	$pJ/bit/m^3$	a^2	100	-
ϵ_a^2	1000	$pJ/bit/m^2$	E^{SN}	20	J
θ_1	2	-	E^{CH}	$2e4$	J
θ_2	3	-	b^1	0.2	-
$r_s = l_1$	5	m	b^2	0.01	-

Results of Coverage

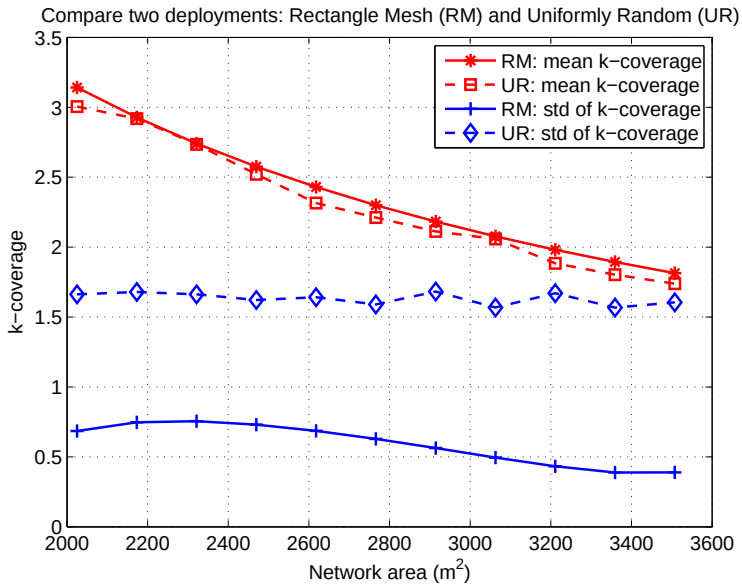


Figure 4.18. The average k-coverage and variance.

The comparisons of coverage performance of the network is shown in Fig. 4.18. In this figure, 'std' means standard deviation, which is equal to the square root of the variance expressed in Equ. (4.33). From the figure, we can see that the performance of average coverage is similar in two deployment schemes. However, the standard deviation of the rectangle mesh deployment scheme is smaller than that of the uniformly random scheme. It indicates that the rectangle mesh scheme has a better balanced coverage performance which is more useful in achieving the tracking or monitoring purpose of WSNs in practical applications. Moreover, the figure shows that the average k-coverage decreases when the network area increases. Since the total number of sensor nodes and their sensing range are fixed, the percentage of overlapped coverage area decreases when the total network area increases. Therefore, by adopting the rectangle mesh deployment scheme, it is able to adjust the coverage performance according to the requirements of the application even if the sensing range of the sensors can not be adjusted.

Results of Lifetime

In this experiment, two power schemes are adopted. In the uniform power scheme, the transmission power of sensors is fixed. While in the power control scheme, the transmission power of sensors are adjusted according to the transmission distances. The value of q_0 (refer to Section 4.4.4) is set to 5%. The lifetimes of CHs and SNs are shown in the following table.

Table 4.3. Lifetime

Lifetime of CHs and SNs (day)							
-		*Rectangle Mesh*			*Uniformly Random*		
-		*2CH*	*8CH*	*32CH*	*2CH*	*8CH*	*32CH*
Uniform Power	*CH*	155	267	756	147	256	711
	SN	182	842	3082	158	724	2661
Power Control	*CH:min*	155	267	756	143	254	708
	CH:max	155	516	2893	146	486	2087
	SN:min	182	842	3086	161	730	2603
	SN:max	3086	3086	3086	2758	2642	2715

In this table, 'CH:min' and 'CH:max' denote the minimum and maximum lifetime of CHs, respectively. Similarly, 'SN:min' and 'SN: max'denote the minimum and maximum lifetime of sensors, respectively. As we can see from the table, the minimum lifetime of SNs and CHs in the uniform power and power control schemes are almost the same. However, the maximum and minimum lifetime in the power control schemes are different which indicates the uneven power consumptions among SNs and CHs due to their positions.

Results of Cost

According to the cost model, the installation cost of the network is determined by the number of SNs and CHs and their initial energy assignments. Since the total number of SNs and their initial energy in three cases is the same, the network cost is 2392, 4192, and 11392 when the number of CHs is 2, 8 and 32, respectively. The values of a_1, b_1, a_2 and b_2 are based on realistic models of the slave sensor node and master sensor node [70] in the fresh food tracking project. These values may be different in other applications of WSNs. However, the point is that, it is able to estimate the deployment cost in an early stage based on our cost analysis method.

Discussions

There are trade-offs between the three performance metrics. If the users require better coverage performance, more sensors should be deployed which can lead to higher cost. Furthermore, from the results of lifetime and cost, we can find that deploying more CHs can increase the network lifetime but it can also bring higher cost. In order to design a cost-efficient network while fulfilling the users' requirements on coverage and network lifetime, we can design optimization algorithms and do extensive computations on the performance metrics and then choose an appropriate deployment scheme. That is where our performance analysis techniques can be used.

4.5 Summary

In most previous work, the coverage analysis usually focused on area coverage which is to model how well an area is covered by sensors. Those types of methods are not suitable for event-driven networks, where the objective is to quantify how well the events are detected by sensors. To this end, we propose a method to study the event coverage problem in heterogeneous sensor networks, where the sensor densities in different locations of FoI are different depending on the event densities. The sensors and events are distributed according to spatial Poisson processes. Formulas are derived to calculate the probability of event coverage by taking considerations of boundary conditions and overlapping. The numerical results illustrate the probabilities of *k*-coverage with different settings of event and sensor densities. Moreover, the results show that the heterogeneous deployment can achieve better coverage than homogeneous deployment if the event occurrence

in the FoI is heterogeneous. In addition, to validate the analytic method for probabilistic coverage analysis, simulations are conducted and the results are compared with the analytical results.

In addition, we present a method for analyzing the coverage, network lifetime and cost of sensor networks under random and deterministic node deployment strategies. We derive closed-form formulas for calculating the average coverage and variance of rectangle mesh WSNs. Furthermore, energy model and cost model are applied to analyze the network lifetime and installation cost of WSNs. Our method is capable of evaluating the advantages and disadvantages of different deployment strategies. Consequently, it can provide guidelines for network designers to implement an cost-efficient network while achieving requirements on network lifetime and quality of coverage.

Chapter 5

Summary and Future Work

This chapter concludes the thesis and outlines future directions.

5.1 Summary

Due to its non-deterministic and random characteristics, wireless sensor network design presents several challenges while dealing with various requirements and diverse constraints. To provide insight upon the design parameters effecting system behavior, such as data transmission delay and maximum required buffer, performance analysis techniques are required. In this thesis, we present a performance analysis method for WSNs based on network calculus. With this method, some performance metrics can be analytically evaluated rather than by case-by-case simulations. Moreover, we conduct investigation and analysis on event coverage and deployment schemes.

In chapter 2, we define traffic splitting and multiplexing models to characterize any traffic flowing scenarios in sensor networks. The worst-case delay bound and backlog bound of these models are separately analyzed using network calculus. Based on the results of these models, a deterministic worst-case performance analysis method is presented. Providing network parameters, our method can provide an effective way for a designer to estimate the worst-case performance and buffer cost of sensor networks. Moreover, we have presented an analytical technique to evaluate the maximum transmission delay and energy consumption of two categories of retransmission schemes: hop-by-hop retransmission and end-to-end retransmission. In order to validate the tightness of the two bounds obtained from theoretical calculation, several experiments are carried out using Omnet++. The

simulation results and analytical results are compared in the chain and mesh scenarios with various input traffic loads. From the results, it shows that the network calculus can be useful and accurate for performance analysis of WSNs.

In chapter 3, we propose a method for modeling and QoS analysis of wireless channels subject to Rayleigh fading. The key challenge in analyzing wireless systems is the temporal uncertainties inherent in fading channels. To this end, this work applies stochastic network calculus to model Rayleigh fading channels and derive stochastic delay and backlog bounds. The analysis method is validated through simulations. In Section 3.5, we introduce a stochastic network calculus based method for statistical bandwidth estimation of random service networks. The statistical bandwidth is estimated from the measurement of statistical backlog bounds through probe packet trains. We propose a step-by-step procedure on how to estimate the bandwidth from backlog measurement. The method can be applied to any wireless networks with various channel characteristics. The estimation results show good accuracy of our bandwidth estimation method.

In chapter 4, we propose a method to study the event coverage problem in heterogeneous sensor networks, where the sensor densities in different locations of FoI are different depending on the event densities. Formulas are derived to calculate the probability of event coverage by considering boundary conditions and overlapping. The numerical results illustrate the probabilities of k-coverage with different settings of event and sensor densities. Moreover, the results show that the heterogeneous deployment can achieve better coverage than homogeneous deployment if the event occurrence in the FoI is heterogeneous. In addition, to validate the analytic method for probabilistic coverage analysis, we conduct simulation and compare their results with analytic results. The comparisons show that these results match well validating the accuracy of the analytic method. In addition, we present a method for analyzing the coverage, network lifetime and cost of sensor networks under random and deterministic node deployment strategies. We derive closed-form formulas for calculating the average coverage and variance of rectangle mesh WSNs. Our method is capable of evaluating the advantages and disadvantages of different deployment strategies. Consequently, it can provide guidelines for network designers to implement an cost-efficient network while achieving requirements on network lifetime and quality of coverage.

5.2 Future Work

Wireless sensor network is a relatively new scientific and engineering field. Both technology and envisioned applications are changing at a rapid pace. The

work presented in this thesis has many possibilities for future work. Below we discuss some issues that is particularly interesting, some of which we have already started pursuing.

- **Experimental test platform**

  The validation of network-calculus-based performance analysis is implemented by simulations in this thesis. Simulation is more controllable and often simplified for modeling real system and application scenarios. But it may be less accurate than prototype-based filed tests. It would be interesting to design and implement WSN prototypes and validate further the performance analysis under realistic experimental conditions. The experimental platform will also help to validate the coverage and deployment analysis.

- **Applications of stochastic network calculus**

  There has been a lot of research on the theoretic part of stochastic network calculus. However, there is a large gap between the stochastic network calculus theory and the wireless networks. How to apply stochastic network calculus on real application scenarios is a very interesting and important research direction. In recent years, cognitive radio network is attracting a lot of research interests. It contains primary users who have priority to access the spectrum and secondary users who can use the spectrum when available. How to apply stochastic network calculus on such networks is worthy to investigate.

- **Fault tolerant sensor networks**

  Fault tolerance is a crucial issue in many applications of sensor network, since sensors will directly interact with the environment and might be subject to a variety of physical, chemical, and biological forces. Moreover, sensor networks often operate in an autonomous mode without human interactions. Therefore, research in fault tolerant sensor networks should receive a significant attention.

- **Trade-offs between power consumption and transmission reliability**

  Reliable transmission is essential for many applications of sensor networks, such as health care and intrusion detection. However, higher reliability requires more power consumption of sensors, and thus the lifetime of the network would be reduced. It is worthwhile to investigate the trade-offs between reducing power consumption and improving transmission reliability for these applications.

References

[1] R. Agrawal, R. L. Cruz, C. Okino, and R. Rajan. Performance bounds for flow control protocols. *IEEE Trans. on Networking*, 7(3):310–323, 1999.

[2] I. F. Akyildiz, W. Su, Sankarasubramanian, and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks*, 38(4):393–422, April 2002.

[3] I. F. Akyildiz, W. Su, Sankarasubramanian, and E. Cayirci. A survey on sensor networks. *IEEE Communications Magazine*, 40(8):102–114, Aug. 2002.

[4] Fadi M. Al-Turjman, Hossam S. Hassanein, and Mohamed A. Ibnkahla. Connectivity optimization for wireless sensor networks applied to forest monitoring. In *Proceeding of IEEE Int. Conference on Communications (ICC'09)*, 2009.

[5] Adrian Baddeley, Imre B., and Rolf Schneider. Spatial point processes and their applications. In *Stochastic Geometry*, volume 1892 of *Lecture Notes in Mathematics*, pages 1–75. Springer Berlin / Heidelberg, 2007.

[6] Xiaole Bai, Santosh Kumar, Dong Xuan, Ziqiu Yun, and Ten H. Lai. Deploying wireless sensors to achieve both coverage and connectivity. In *in Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing (MobiHoc '06)*, pages 131–142, 2006.

[7] Xiaole Bai, Ziqiu Yun, Dong Xuan, Biao Chen, and Wei Zhao. Optimal multiple-coverage of sensor networks. In *INFOCOM*, pages 2498–2506, 2011.

[8] Xiaole Bai, Ziqiu Yun, Dong Xuan, T.H. Lai, and Weijia Jia. Deploying four-connectivity and full-coverage wireless sensor networks. In *The IEEE 27th Conference on Computer Communications (INFOCOM'08)*, pages 296 –300, 2008.

[9] N. Baker. Real world wireless mesh sensor network solutions. In *the IEE Seminar on Industrial Networking and Wireless Communications for Control*, 2006.

[10] Edoardo S. Biagioni and K. W. Bridges. The application of remote sensor technology to assist the recovery of rare and endangered species. *International Journal of High Performance Computing Applications*, 16:112–121, 2002.

[11] J. Boudec and P. Thiran. Network calculus: A theory of deterministic queuing systems for the internet. *Springer, LNCS 2050*, 2004.

[12] Jean-Yves Le Boudec. *Performance Evaluation of Computer and Communication Systems*. EPFL Press, Switzerland, 2010.

[13] A. Burchard, J. Liebeherr, and S. D. Patek. A min-plus calculus for end-to-end statistical service guarantees. *IEEE Transactions on Information Theory*, 52(9):4105 –4114, Sept. 2006.

[14] Almut Burchard, Jörg Liebeherr, and Stephen D. Patek. A min-plus calculus for end-to-end statistical service guarantees. *IEEE Transactions on Information Theory*, 52(9):4105–4114, 2006.

[15] Y. Cao, Y. Xue, and Y. Cui. Network-calculus-based analysis of power management in video sensor networks. *the IEEE Global Communications Conference, Exhibition and Industry Forum (GLOBECOM'07)*, 2007.

[16] M. Cardei and D. Du. Improving wireless sensor network lifetime through power aware organization. *ACM Journal of Wireless Networks*, 11(3):333–340, May 2005.

[17] C. S. Chang. *Performance Guarantees in Communication Networks*. Springer-Verlag, 2000.

[18] Y. Charfi, N. Wakamiya, and M. Murata. Adaptive and reliable multipath transmission in wireless sensor networks using forward error correction and feedback. *IEEE Wireless Communications and Networking Conference(WCNC'07)*, 2007.

[19] K. Chen, K. Nahrstedt, and N. Vaidya. The utility of explicit rate-based flow control in mobile ad hoc networks. In *IEEE Wireless Communications and Networking Conference (WCNC'04)*, volume 3, pages 1921 – 1926, 2004.

[20] Zhao Cheng, Mark Perillo, and Wendi Heinzelman. General network lifetime and cost models for evaluating sensor network deployment strategies. *IEEE Trans. on Mobile Computing*, 7(4), 2008.

[21] Chee-Yee Chong and S. P. Kumar. Sensor networks: evolution, opportunities, and challenges. *Proceedings of the IEEE*, 91(8):1247–1256, Aug. 2003.

[22] Florin Ciucu, Almut Burchard, and Jörg Liebeherr. A network service curve approach for the stochastic analysis of networks. In *Proceedings of the 2005 ACM SIGMETRICS international conference on Measurement and modeling of computer systems (SIGMETRICS'05)*, pages 279–290, 2005.

[23] Thomas Clouqueur, Kewal K. Saluja, and Parameswaran Ramanathan. Fault tolerance in collaborative sensor networks for target detection. *IEEE Transaction on Computers*, 53(3):320–333, 2004.

[24] Crossbow. Mica2 date sheet. *http://www.xbow.com/ products/ Product_pdf_files /Wireless_pdf/MICA2_Datasheet.pdf.*

[25] R. Cruz. A calculus for network delay, part I: Network elements in isolation. *IEEE Transaction on Information Theory*, 37(1):114–121, 1991.

[26] R. Cruz. A calculus for network delay, part II: Network analysis. *IEEE Transaction on Information Theory*, 37(1):132–141, 1991.

[27] D. Culler, D. Estrin, and M. Srivastava. Overview of sensor networks. *IEEE Computer*, 37(8):41–49, 2004.

[28] Lennart Råde and Bertil Westergren. *Mathematics Handbook for Science and Engineering (Fifth Edition)*. 2010.

[29] S. De and C. Qiao. On throughput and load balancing of multipath routing in wireless networks. *IEEE Wireless Communications and Networking Conference(WCNC'04)*, 2004.

[30] P. Djukic and S. Valaee. Reliable packet transmissions in multipath routed wireless networks. *IEEE Transactions on Mobile Computing*, 5(5):548–559, May 2006.

[31] X. Du and F. Lin. Improving sensor network performance by deploying mobile sensors. *in Proc. of 24th IEEE International on Performance, Computing, and Communications(IPCCC'05)*, 2005.

[32] Graham Elliott, Thomas J. Rothenberg, and James H. Stock. Efficient tests for an autoregressive unit root. *Journal of Econometrica*, 64(4):813–36, July 1996.

[33] Sinem Coleri Ergen and Pravin Varaiya. Pedamacs: Power efficient and delay aware medium access protocol for sensor networks. *IEEE Transactions on Mobile Computing*, 2006.

[34] D. Estrin, L. Girod, G. Pottie, and M. Srivastava. Instrumenting the world with wireless sensor networks. *in Proc. of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2001.

[35] D. Ganesan, A. Cerpa, W. Ye, Y. Yu, J. Zhao, and D. Estrina. Networking issues in wireless sensor networks. *Journal of Parallel and Distributed Computing*, pages 799–814, 2004.

[36] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin. Highly-resilient, energy-efficient multipath routing in wireless sensor networks. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(4):11–25, 2001.

[37] Andrea Goldsmith and Anaïs Nin. *Wireless Communications*. Cambridge University Press, 2005.

[38] Himanshu Gupta, Zongheng Zhou, Samir R. Das, and Quinyi Gu. Connected sensor cover: self-organization of sensor networks for efficient query execution. In *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking and computing (MobiHoc'03)*, pages 189–200, 2003.

[39] Amanda Hering, Cynthia Bell, and Marc Genton. Modeling spatio-temporal wildfire ignition point patterns. *Environmental and Ecological Statistics*, 16:225–250, 2009.

[40] Fei Hu, Yang Xiao, and Qi Hao. Congestion-aware, loss-resilient biomonitoring sensor networking for mobile health applications. *IEEE Journal of Selected Areas in Communications*, 27(4):450–465, May 2009.

[41] Ningning Hu and P. Steenkiste. Evaluation and characterization of available bandwidth probing techniques. *IEEE Journal on Selected Areas in Communications*, 21(6):879 – 894, 2003.

[42] Zhihua Hu and Baochun Li. Fundamental performance limits of wireless sensor networks. *Journal of Ad Hoc and Sensor, Nova Science Publishers*, pages 81–101, 2004.

[43] Chi-Fu Huang and Yu-Chee Tseng. The coverage problem in a wireless sensor network. In *Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications (WSNA'03)*, pages 115–121, September 2003.

[44] Yuming Jiang. A basic stochastic network calculus. *in Proc. of ACM SIGCOMM*, 2006.

[45] Yuming Jiang and Peder J. Emstad. Analysis of stochastic service guarantees in communication networks: A server model. In *In Proceeding of the International Workshop on Quality of Service (IWQoS'05)*, pages 233–245, 2005.

[46] Yuming Jiang and Yong Liu. *Stochastic Network Calculus*. Springer Publishing Company, 2008.

[47] Yuming Jiang, Qinghe Yin, Yong Liu, and Shengming Jiang. Fundamental calculus on generalized stochastically bounded bursty traffic for communication networks. *Computer Networks*, 53(12):2011–2021, 2009.

[48] Hongzhi Jiao and F.Y. Li. A service-oriented routing scheme with load balancing in wireless mesh networks. In *IEEE International Symposium on Wireless Communication Systems (ISWCS'08)*, pages 658 –662, 2008.

[49] Petr Jurcik, Ricardo Severino, Anis Koubaa, Mario Alves, and Eduardo Tovar. Real-time communications over cluster-tree sensor networks with mobile sink behaviour. In *14th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA'08)*, 2008.

[50] H. Karl and A. Willig. A short survey of wireless sensor networks. *Technical Report TKN-03-018, Telecommunication Networks Group, Technical University Berlin*, 2003.

[51] Anis Koubaa, M. Alves, and E. Tovar. Modeling and worst-case dimensioning of cluster-tree wireless sensor networks. *the 27th IEEE International Real-time Systems Symposium (RTSS'06)*, 2006.

[52] Loukas Lazos and Radha Poovendran. Stochastic coverage in heterogeneous sensor networks. *ACM Transaction on Sensor Networks*, 2:325–358, August 2006.

[53] Loukas Lazos, Radha Poovendran, and James A. Ritcey. Analytic evaluation of target detection in heterogeneous wireless sensor networks. *ACM Transaction on Sensor Networks*, 5:18:1–18:38, April 2009.

[54] S. I. Lee and J. S. Lim. Hybrid cluster mesh scheme for energy efficient wireless sensor networks. *IEICE Transactions on Communications*, E91-B(08):2610–2617, 2008.

[55] L. Lenzini, L. Martorini, E. Mingozzi, and G. Stea. Tight end-to-end per-flow delay bounds in FIFO multiplexing sink-tree networks. *Performance Evaluation Journal (Elsevier)*, 63:956–987, 2006.

[56] Chenglin Li, Junni Zou, Hongkai Xiong, and Yongsheng Zhang. Joint coding/routing optimization for correlated sources in wireless visual sensor networks. In *IEEE Global Telecommunications Conference (GLOBECOM'09)*, Dec. 2009.

[57] Jörg Liebeherr, Markus Fidler, and Shahrokh Valaee. A system-theoretic approach to bandwidth estimation. *IEEE/ACM Trans. Netw.*, 18(4):1040–1053, 2010.

[58] B. Liu, F. Ren, C. Lin, and Y. Ouyang. Performance analysis of retransmission and redundancy schemes in sensor networks. In *IEEE International Conference on Communications (ICC'08)*, 2008.

[59] Benyuan Liu and Don Towsley. A study of the coverage of large-scale sensor networks. In *In the Proceedings of the IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS'04)*, pages 475–483, 2004.

[60] Yong Liu, Chen-Khong Tham, and Yuming Jiang. A calculus for stochastic qos analysis. *Performance Evaluation*, 64(6):547–572, 2007.

[61] Wenjing Lou. An efficient n-to-1 multipath routing protocol in wireless sensor networks. In *IEEE International Conference on Mobile Adhoc and Sensor Systems Conference (MASS'05)*, 7 2005.

[62] Ye Ming Lu and V.W.S. Wong. An energy-efficient multipath routing protocol for wireless sensor networks. In *IEEE 64th Vehicular Technology Conference(VTC-06-Fall)*, 2006.

[63] Ralf Lübben, Markus Fidler, and Jörg Liebeherr. A foundation for stochastic bandwidth estimation of networks with random service. *online, http://arxiv.org/abs/1008.0050*, 2010.

[64] Ralf Lübben, Markus Fidler, and Jörg Liebeherr. A foundation for stochastic bandwidth estimation of networks with random service. In *INFOCOM*, pages 1817–1825, 2011.

[65] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson. Wireless sensor networks for habitat monitoring. *in the Proceeding of the ACM International Workshop on Wireless Sensor Networks and Applications*, 2002.

[66] A. Meddeb. Benefits of multicast traffic split routing in packet switched networks. In *IEEE International Conference on Communications (ICC'04)*, June 2004.

[67] Scott Moeller, Avinash Sridharan, Bhaskar Krishnamachari, and Omprakash Gnawaliy. Routing without routes: The backpressure collection protocol. In *The 9th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN'10)*, 2010.

[68] Jeongyeup Paek, Krishna Chintalapudi, Ramesh Govindan, John Caffrey, and Sami Masri. A wireless sensor network for structural health monitoring: Performance and experience. *The Second IEEE Workshop on Embedded Networked Sensors (EmNetS'05)*, 2005.

[69] H. Pai, J. Sung, and Y. S. Han. Adaptive retransmission for distributed detection in wireless sensor networks. In *IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC'06)*, 2006.

[70] Zhibo Pang, Jun Chen, David Sarmiento, Zhi Zhang, Jie Gao, Qiang Chen, and Li-Rong Zheng. Mobile and wide area deployable sensor system for networked services. In *The 8th Annual IEEE Conference on Sensors*, October 2009.

[71] S. J. Park, R. Vedantham, R. Sivakumar, and I. F. Akyildiz. A scalable approach for reliable downstream data delivery in wireless sensor networks. *in Proc. of the 5th ACM MobiHoc*, pages 78–89, 2004.

[72] Wint Yi Poe and Jens B. Schmitt. Node deployment in large wireless sensor networks: coverage, energy consumption, and worst-case delay. In *Proceeding of ACM Asian Internet Engineering Conference (AINTEC'09)*, Thailand, 2009.

[73] G. J. Pottie and W. J. Kaiser. Wireless integrated network sensors. *Communications of the ACM*, 43(5), 2000.

[74] B. Prabhakar, E. Uysal Biyikoglu, and A. El Gamal. Energy-efficient transmission over a wireless link via lazy packet scheduling. In *The 20th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'01)*, pages 386 –394, 2001.

[75] V. Raghunathan, C. Schurgers, S. Park, and M. B. Srivastava. Energy-aware wireless microsensor networks. *IEEE Signal Processing Magazine*, 19:40–50, Mar. 2002.

[76] Nithya Ramanathan, Laura Balzano, Deborah Estrin, Mark Hansen, Thomas Harmon, Jenny Jay, William Kaiser, and Gaurav Sukhatme. Designing wireless sensor networks as a shared resource for sustainable development. *International Conference on Information and Communication Technologies and Development (ICTD'06)*, 2006.

[77] Theodore S. Rappaport. *Wireless Communications: Principles and Practice*. Prentice Hall, 2002.

[78] K. Romer and F. Mattern. The design space of wireless sensor networks. *IEEE Wireless Communications*, 11(6):54–61, Dec. 2004.

[79] Anthony Rowe and et al. Rt-link: A time-synchronized link protocol for energy-constrained multi-hop wireless networks. 2006.

[80] Cheikh Sarr, Claude Chaudet, Guillaume Chelius, and Isabelle Guérin Lassous. Bandwidth estimation for ieee 802.11-based ad hoc networks. *IEEE Trans. on Mobile Computing*, 7(10):1228–1241, 2008.

[81] Jens B. Schmitt and U. Roedig. Sensor network calculus - a framework for worst case analysis. *IEEE/ACM Int. conference on Distributed Computing in Sensor Systems (DCOSS'05)*, 2005.

[82] Jens B. Schmitt, F. A. Zdarsky, and L. Thiele. A comprehensive worst-case calculus for wireless sensor networks with in-network processing. In

Proceedings of the 28th IEEE Int. Real-time Systems Symposium (RTSS'07), 2007.

[83] Huimin She, Zhonghai Lu, Axel Jantsch, Li-Rong Zheng, and Dian Zhou. Traffic splitting with network calculus for mesh sensor networks. In *Proceedings of IEEE conference on Future Generation Communication and Networking (FGCN'07)*, 2007.

[84] Huimin She, Zhonghai Lu, Axel Jantsch, Li-Rong Zheng, and Dian Zhou. Analysis of traffic splitting mechanisms for 2D mesh sensor networks. *International Journal of Software Engineering and Its Applications (IJSEIA)*, 2(3):25–37, 2008.

[85] Huimin She, Zhonghai Lu, Axel Jantsch, Li-Rong Zheng, and Dian Zhou. Deterministic worst-case performance analysis for wireless sensor networks. In *Proc. of Int. Wireless Communications and Mobile Computing Conference (IWCMC'08)*, 2008.

[86] Huimin She, Zhonghai Lu, Axel Jantsch, Dian Zhou, and Li-Rong Zheng. Analytical evaluation of retransmission schemes in wireless sensor networks. In *Proc. of IEEE 69th Vehicular Technology Conference (VTC2009-Spring)*, 2009.

[87] Huimin She, Zhonghai Lu, Axel Jantsch, Dian Zhou, and Li-Rong Zheng. Modeling and analysis of rayleigh fading channels using stochastic network calculus. In *IEEE Wireless Communication and Networking Conference (WCNC'2011)*, 2011.

[88] Huimin She, Zhonghai Lu, Axel Jantsch, Dian Zhou, and Li-Rong Zheng. Stochastic coverage in event-driven sensor networks. In *Proceedings of the 22nd IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'2011)*, Canada, 2011.

[89] Huimin She, Zhonghai Lu, Axel Jantsch, Dian Zhou, and Li-Rong Zheng. Estimation of statistical bandwidth through backlog measurement. In *Workshop on Network Calculus (WoNeCa2012*, 2012.

[90] Huimin She, Zhonghai Lu, Axel Jantsch, Dian Zhou, and Li-Rong Zheng. Performance analysis of flow-based traffic splitting strategy on cluster-mesh sensor networks. *International Journal of Distributed Sensor Networks(IJDSN)*, (232937), 2012.

[91] Huimin She, Zhonghai Lu, Axel Jantsch, Dian Zhou, and Li-Rong Zheng. Probabilistic event coverage in heterogeneous sensor networks. In *submission to ACM Transactions on Sensor Networks*, 2012.

[92] Huimin She, Zhonghai Lu, Axel Jantsch, Dian Zhou, and Li-Rong Zheng. System-level evaluation of sensor networks deployment strategies: Coverage lifetime and cost. In *the 8th International Wireless Communications and Mobile Computing Conference (IWCMC'08)*, 2012.

[93] P. Sikka, P. Corke, P. Valencia, C. Crossman, D. Swain, and G. Bishop-Hurley. Wireless ad hoc sensor and actuator networks on the farm. In *The Fifth International Conference on Information Processing in Sensor Networks (IPSN' 2006)*, pages 492 –499, 2006.

[94] Mark Snyder and Sriram Chellappan. Event coverage in sparse mobile sensor networks. *International Conference on Network-Based Information Systems*.

[95] David Starobinski and Moshe Sidi. Stochastically bounded burstiness for communication networks. *IEEE Transaction on Information Theory*, 46(1):206–212, 2000.

[96] C. Taddia and G. Mazzini. On the retransmission method in wireless sensor networks. *in Proc. of VTC2004-Fall*, 2004.

[97] S. Tai, R. Benkoczi, H. Hassanein, and S. Akl. A performance study of splittable and unsplittable traffic allocation in wireless sensor networks. In *IEEE International Conference on Communications (ICC'06)*, 2006.

[98] G. Verticale and P. Giacomazzi. An analytical expression for service curves of fading channels. In *IEEE Global Telecommunications Conference (GLOBECOM'09)*, pages 1 –6, Nov. 2009.

[99] C. Y. Wan, A. T. Campbell, and L. Krishnamurthy. PSFQ: a reliable transport protocol for wireless sensor networks. *in Proc. of the 1st ACM international workshop on Wireless Sensor Networks and Applications (WSNA'02)*, 2002.

[100] Peng-Jun Wan and Chih-Wei Yi. Coverage by randomly deployed wireless sensor networks. *IEEE/ACM Transaction on Networking*, 14:2658–2669, June 2006.

[101] Demin Wang, Bin Xie, and Dharma P. Agrawal. Coverage and lifetime optimization of wireless sensor networks with gaussian distribution. *IEEE Transactions on Mobile Computing*, 7(12):1444–1458, 2008.

[102] Feng Wang and Jiangchuan Liu. Networked wireless sensor data collection: Issues, challenges, and approaches. *IEEE Communications Surveys Tutorials*, 2010.

[103] You-Chiun Wang, Wen-Chih Peng, Min-Hsien Chang, and Yu-Chee Tseng. Exploring load-balance to dispatch mobile sensors in wireless sensor networks. In *ICCCN*, pages 669–674, 2007.

[104] G. Werner-Allen, K. Lorincz, M. Ruiz, O. Marcillo, J. Johnson, J. Lees, and M. Welsh. Deploying a wireless sensor network on an active volcano. *IEEE Internet Computing*, 10(2):18–25, 2006.

[105] Daniel H. Whang, Ning Xu, Sumit Rangwala, Krishna Chintalapudi, Ramesh Govindan, and John W. Wallace. Development of an embedded networked sensing system for structural health monitoring. *in Proc. of the International Workshop on Smart Materials and Structures Technology*, 2004.

[106] Wilson T.H. Woon and T.C. Wan. Performance evaluation of ieee 802.15.4 ad hoc wireless sensor networks: Simulation approach. *in Proc. of IEEE International Conference on Systems, Man and Cybernetics (SMC'06)*, 2006.

[107] Opher Yaron and Moshe Sidi. Performance and stability of communication networks via robust exponential bounds. *IEEE/ACM Transaction on Networking*, 1(3):372–385, 1993.

[108] Ziqiu Yun, Xiaole Bai, Dong Xuan, T.H. Lai, and Weijia Jia. Optimal deployment patterns for full coverage and k-connectivity ($k \leq 6$) wireless sensor networks. *IEEE/ACM Transactions on Networking*, 18(3):934 –947, 2010.

[109] Andrew Zalesky, Hai Le Vu, and Moshe Zukerman. Reducing spare capacity through traffic splitting. *IEEE Communications Letters*, 8(9), September 2004.

[110] Marco Z. Zamalloa and Bhaskar Krishnamachari. An analysis of unreliability and asymmetry in low-power wireless links. *ACM Transactions on Sensor Networks*, 3(2), 2007.

[111] Gu-Chun Zhang, Xiao-Hong Peng, and Xuan-Ye Gu. Implementation and performance evaluation of an experimental wireless relay sensor network. In *in Proc. of 10th IEEE International Conference on High Performance Computing and Communications (HPCC'08)*, 2008.

[112] H. Zhang and J. C. Hou. Is deterministic deployment worse than random deployment for wireless sensor networks? In *The IEEE 25th Conference on Computer Communications (INFOCOM'06)*, pages 1–13, 2006.

[113] Honghai Zhang and Jennifer Hou. On deriving the upper bound of α-lifetime for large sensor networks. In *Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing (MobiHoc'04)*, pages 121–132, 2004.

[114] Zhi Zhang, Qiang Chen, Tobia Bergarp, Per Norman, Magnus Wikstrom, Xiaolang Yan, and Li-Rong Zheng. Wireless sensor networks for logistics and retail. In *The Sixth International Conference on Networked Sensing Systems (INSS'09)*, June 2009.

[115] Ming-Chen Zhao, Jiayin Lei, Min-You Wu, Yunhuai Liu, and Wei Shu. Surface coverage in wireless sensor networks. In *The IEEE 28th Conference on Computer Communications (INFOCOM'09)*, pages 109 – 117, 2009.

[116] Shoudong Zou, I. Nikolaidis, and J. Harms. Aggregation vs. load balancing in WSNs. In *IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'07)*, Sept. 2007.