Cross-Layer Design of Multi-Hop Wireless Networks

Chi Liu

A Dissertation Submitted in Fulfilment of Requirements for the Degree of
Doctor of Philosophy of Imperial College London and
Diploma of Imperial College

Communications and Signal Processing Research Group
Department of Electrical and Electronic Engineering
Imperial College London
July 2010

Abstract

MULTI -hop wireless networks are usually defined as a collection of nodes equipped with radio transmitters, which not only have the capability to communicate each other in a multi-hop fashion, but also to route each others' data packets. The distributed nature of such networks makes them suitable for a variety of applications where there are no assumed reliable central entities, or controllers, and may significantly improve the scalability issues of conventional single-hop wireless networks.

This Ph.D. dissertation mainly investigates two aspects of the research issues related to the efficient multi-hop wireless networks design, namely: (a) network protocols and (b) network management, both in cross-layer design paradigms to ensure the notion of service quality, such as quality of service (QoS) in wireless mesh networks (WMNs) for backhaul applications and quality of information (QoI) in wireless sensor networks (WSNs) for sensing tasks. Throughout the presentation of this Ph.D. dissertation, different network settings are used as illustrative examples, however the proposed algorithms, methodologies, protocols, and models are not restricted in the considered networks, but rather have wide applicability.

First, this dissertation proposes a cross-layer design framework integrating a distributed proportional-fair scheduler and a QoS routing algorithm, while using WMNs as an illustrative example. The proposed approach has significant performance gain compared with other network protocols. Second, this dissertation proposes a generic admission control methodology for any packet network, wired and wireless, by modeling the network as a black box, and using a generic mathematical

function and Taylor expansion to capture the admission impact. Third, this dissertation further enhances the previous designs by proposing a negotiation process, to bridge the applications' service quality demands and the resource management, while using WSNs as an illustrative example. This approach allows the negotiation among different service classes and WSN resource allocations to reach the optimal operational status. Finally, the guarantees of the service quality are extended to the environment of multiple, disconnected, mobile subnetworks, where the question of how to maintain communications using dynamically controlled, unmanned data ferries is investigated.

Acknowledgment

In most of mankind, gratitude is merely a secret hope for greater favors

- Duc de la Rochefoucauld, Maxims (1665)

Completing a Ph.D. is truly my greatest honor in my life so far, and I would not have been able to complete this journey without the generous aids and supports of countless people over the past four years.

I must first express my deepest gratitude towards my supervisor, Professor Kin K. Leung, Head of the Communications and Signal Processing Research Group, Imperial College. His support I will never forget in my entire life; his leadership, attention to detail, and hard work have set an example I hope to match some day.

Over the years, I have enjoyed the generous aids of several scholarships. From 2006 to 2009, I received the Electrical and Electronic Engineering Departmental scholarship from Imperial College. From 2006 to 2008, my research was generously supported by EU FP6 MEMBRANE project on wireless mesh networks; and for the last two years I have been supported by US Army-UK MoD co-funded ITA project on wireless sensor networks.

I would like to thank Dr. Chatschik Bisdikian, Dr. Joel Branch, Dr. Ting He, and Dr. Kang-won Lee at IBM's T. J. Watson Research Center in Hawthorne, USA, who provided me with golden opportunities to collaborate with world-class researchers, and a wonderful summer intern experience in 2009.

I would like to thank Dr. Athanasios Gkelias, for his guidance and help throughout my Ph.D. research and detailed descriptions for every single question I asked.

I would like to thank Dr. Erwu Liu for his generous help on building up the OPNET simulation platform for EU FP6 MEMBRANE project, and Dr. Yun Hou for her inspiring discussions on distributed scheduling algorithms, without which my ideas and thoughts could not be verified and the project could not be such a success. I look forward to a continuing collaboration with them all in the future.

I would like to thank Dr. Javier Barria and Dr. Vasilis Friderikos for their time serving as my Ph.D. examiners and for their helpful discussions to improve the presentation of my Ph.D. dissertation.

I am grateful to all faculty and staff members of the Department of Electrical and Electronic Engineering at Imperial College; and I also thank all fellow Ph.D. students of Communications and Signal Processing Research Group. They all helped make my time in the Ph.D. program more fun and interesting.

Finally, I really thank my parents for their understanding and love during the past four years. Their support and encouragement was in the end what made this dissertation possible. They both receive my deepest gratitude and love for their dedication and many years of support from a teenage boy to the completion of my Ph.D. degree; I own them a lot.

List of Publications

The following publications have been written during the course of the Ph.D. research:

- **C. H. Liu**, K. K. Leung, and A. Gkelias, "A Generic Admission Control Methodology for Packet Networks," in preparation for submission to *IEEE Trans. on Networking*.

- **C. H. Liu**, A. Gkelias, Y. Hou, and K. K. Leung, "Cross-Layer Design for QoS in Wireless Mesh Networks," in *Springer Wireless Personal Comm.*, Special Issue on "Cross-Layer Design for Future Generation Networks", Vol. 51, No. 3, pp. 593-613.

- **C. H. Liu**, C. Bisdikian, J. Branch, and K. K. Leung, "QoI-Aware Wireless Sensor Network Management for Dynamic Multi-Task Operations," in *IEEE SECON 2010*, Boston, USA (won IEEE Comm. Society Student Travel Grant); also in *IBM Res. Tech. Rep.* RC24933, January, 2010; and also will appear in *Annual Conf. of ITA 2010*, September, London, UK.

- **C. H. Liu**, T. He, K. W. Lee, K. K. Leung, and A. Swami, "Dynamic Control of Data Ferries under Partial Observations," in *IEEE WCNC 2010*, April, Sydney, Australia.

- **C. H. Liu**, K. K. Leung, and A. Gkelias, "Route Capacity Estimation Based Admission Control and QoS Routing for Mesh Networks," in *IEEE Globecom 2009*, November, Hawaii, USA, pp. 1-6.

- **C. H. Liu**, J. Branch, C. Bisdikian, and K. K. Leung. "A QoI-aware Middleware for Task-Oriented Applications in Wireless Sensor Networks," in *Annual Conf. of ITA 2009*, September, Maryland, USA.

- **C. H. Liu**, S. G. Colombo, A. Gkelias, E. Liu, and K. K. Leung. "An Efficient Cross-Layer Simulation Architecture for Wireless Mesh Networks," in *IEEE UKSim 2009*, March 25-27, Cambridge, UK, pp. 491-496.

- **C. H. Liu**, S. G. Colombo, E. Liu, A. Gkelias, and G. Paltenghi. "Efficient Cross-Layer Simulator for Performance Evaluation of Wireless Mesh Networks," in *ACM/ICST SimuTools 2009*, Rome, March 3-6, Italy.

- **C. H. Liu**, K. K. Leung, C. Bisdikian, and J. Branch, "A New Approach to Architecture of Sensor Networks for Mission-Oriented Applications," in *SPIE Defense, Security, and Sensing 2009*, April 13-17, Orlando, USA, vol. 7349; also in *IBM Res. Tech. Rep.* RC24765, April, 2009.

- **C. H. Liu**, A. Gkelias, Y. Hou and K. K. Leung. "A Distributed Scheduling Algorithm with QoS Provisions in Multi-Hop Wireless Mesh Networks," in *IEEE WiMob 2008*, October 12-14, Avignon, France, pp. 253-258.

- **C. H. Liu**, A. Gkelias, and K. K. Leung, "Connection Admission Control and Grade of Service for QoS Routing in Wireless Mesh Networks," in *IEEE PIMRC 2008*, September, France, pp. 1-5.

- A. Gkelias, F. Boccardi, **C. H. Liu**, and K. K. Leung, "MIMO Routing with QoS Provisioning," in *IEEE ISWPC 2008*, Greece, 2008, pp. 46-50.

- **C. H. Liu**, A. Gkelias, and K. K. Leung, "A Cross-Layer Framework of QoS Routing and Distributed Scheduling for Mesh Networks," in *IEEE VTC Spring 2008*, 11-14 May, Singapore, pp. 2193-2197.

- **C. H. Liu**, K. K. Leung, and A. Gkelias, "A Novel Cross-Layer QoS Routing Algorithm for Wireless Mesh Networks," in *IEEE ICOIN 2008*, January, Busan, Korea, pp. 1-5.

Contents

List of Figures

List of Tables

Statement of Originality

To the author's best of knowledge, the following aspects of this Ph.D. dissertation are believed to be original contributions:

1. A novel cross-layer design solution to support QoS requirements in wireless mesh networks is proposed, including:

   (a) A QoS routing algorithm, which overcomes the NP completeness of integrating multiple QoS performance metrics, including packet delay, packet error rate (PER), and throughput, in a unified utility function.

   (b) An integrated QoS routing and distributed scheduling algorithm to enforce the long-term QoS experience.

   (c) A multi-level QoS management scheme for grade of service (GoS), which allows connections to adapt their levels of QoS requirements to fit the network conditions.

2. A generic admission control (GAC) methodology is proposed for any packet networks, wired and/or wireless, completely transparent to the lower protocol layers, where:

   (a) A subnetwork between a pair of ingress and egress nodes is modeled as a "black box" for the amount of available network resources to share.

   (b) The potential connection admission is mathematically characterized as the input change to the black box, and the impact of this admission event on the overall network QoS satisfaction is analyzed in a closed-form.

(c) A GAC methodology is proposed to enforce the QoS control of the new connection without violating existing connections' QoS experience.

3. A QoI-aware network operation and management (O&M) framework for wireless sensor networks (WSNs) is proposed, where key design elements include:

   (a) The QoI satisfaction index of tasks, which quantifies the degree to which the required QoI is satisfied by the WSN;

   (b) The QoI network capacity, which expresses the ability of the WSN to host a new task (with specific QoI requirements) without sacrificing the QoI of other currently hosted tasks;

   (c) A negotiation process, which iteratively reconfigures and optimizes the usage of network resources and the degree of QoI acceptance of prioritized tasks.

4. The design of control policies for unmanned data ferries to maintain communications among multiple, disconnected, mobile subnetworks is proposed, including:

   (a) A Partially Observable Markov Decision Process (POMDP) mathematical model with a reward structure to maximize the total number of effective contacts with an exponential discount;

   (b) An efficient policy computation algorithm, based on the belief space quantization which limits the computation on the dimension of the simplex one smaller than the original simplex.

List of Notations

i	node indexed i
V_R	the set of WMRs with size n_r
V_G	the set of Internet gateways with size n_g
j	node i's one-hop neighbors, denoted as $\{j = 1, 2, ..., J_i\}$
(i, j)	the link between node i and node j
τ_{ij}	the aggregated routing demand for link (i, j)
U_{ij}	the scheduling utility for link (i, j)
ρ_{ij}	the instantaneous channel capacity for link (i, j)
ϕ_{ij}	the long-term throughput for link (i, j)
$\mathrm{N}_e(i, j)$	the average number of erroneous packets on link (i, j)
$\mathrm{N}_c(i, j)$	the average number of correct packets on link (i, j)
π_{sg}	the route set between source s and gateway g
π_{sg}^k	the k^{th} route between source s and gateway g
q	connection/task index
q'	newly arrived connection/task index
$\mathcal{Q}$	the ongoing connection set
$\mathcal{S}_q$	the set of participating sensors for task q
$l_q(t)$	the remaining lifetime of the task q at time t
λ	the parameter for Possion arrival process
μ	the parameter for exponential departure process

z	$z = \{D, T, E, \alpha\}$ denote ETE delay, throughput, PER, and the probability of detection, respectively
M_z	the dimension of the QoI requirements
z_q^r	connection/task's required QoS values for parameter z
z_q^a	connection/task's attained QoS values for parameter z
R_q^z	defined QoS outage ratios for parameter z
$\mathrm{N_{HQoS}}$	the number of ongoing connections with high QoS requirements
$\mathrm{N_{LQoS}}$	the number of ongoing connections with low QoS requirements
f	the mathematical function to represent the black box model
$\underline{x}(t)$	the input variables to the black box model
M_b	the dimension of the input variables of the black box model
$y(t)$	the single output from the black box model
$N(t)$	the number of ongoing connections in the network
$T(t)$	the total served throughput in the network
G_0	defined GoS level
$\theta_q(t)$	per-connection QoS performance index at time t, or per-task QoI satisfaction index at time t
$\mathrm{I}(t)$	network-wide QoS performance index at time t, or network-wide QoI satisfaction index at time t
u_q	the priority level of task q, where $u_q = 1, 2, \ldots, U$
$\underline{\xi}_q(t)$	the instantaneous resource occupancy for task q
$\underline{\mathcal{P}}(t)$	the instantaneous remaining resources in the network
M_r	the dimension of the instantaneous remaining resources
$\underline{\mathcal{C}}(t)$	the QoI network capacity
M_c	the dimension of the defined QoI network capacity
$\gamma_q(t)$	the power allocation for task q at time t

L	packet length
$\mathcal{F}$	the objective function for the negotiation process
p_{iq}	the achieved probability of detection from sensor i to task q when $\gamma_q(t) = 1$ is assumed
$\zeta_i(t)$	the remaining energy for sensor i at time t
n_s	the number of disconnected subnetworks
o_t	the observation for the data ferry at time t
a_t	the action taken by the data ferry at time t
1_t	the effective contact indicator of the data ferry at time t
s_t	the state of the data ferry at time t
$\underline{b}_t$	the belief of the data ferry at time t
$\underline{b}_0$	the limiting distribution of the belief of the data ferry
r_t	the reward earned by the data ferry at time t
$\mathcal{L}_i$	the set of state space of subnetwork i
$\mathcal{A}_f$	the "follow" action space of the data ferry
$\mathcal{A}_s$	the "switch" action space of the data ferry
$\mathcal{O}$	the observation space of the data ferry
$\mathcal{B}$	the set of selected belief points over the belief simplex
π_t	the policy mapping from previous preservations and actions to the new action at time t
H	the design horizon of the data ferry
P^i	the state transition matrix of the mobility pattern for the group of nodes in the subnetwork i
ω_1, ω_2	the functions for belief update within a time slot
κ	the discount factor for the accumulated reward within horizon H
$\mathbf{e}_a$	the unit vector with 1 at the a^{th} element and 0 elsewhere

List of Abbreviations

AMC: Adaptive Modulation and Coding Scheme

AP: Access point

CAC: Connection admission control

GAC: Generic admission control

GoS: Grade of service

HQoS: High quality of service requirement

IGW: Internet gateway

IQoSR: Integrated quality of service routing

LQoS: Low quality of service requirement

MAC: Medium access control

MCP: multi-constrained path

VI: Value interation

O&M: Operations and management

PER: Packet-error-rate

PHY Layer: Physical layer

POMDP: Partial observable markov decision process

SINR: Signal to noise plus interference ratio

SNR: Signal to noise ratio

TCP: Tranport control protocol

TDMA: Time Division Multiple Access

QoI: Quality of information

QoS: Quality of service

WMN: Wireless mesh network

WMR: Wireless mesh router

WSN: Wireless sensor network

Chapter 1

Introduction

THE past decade has seen enormous development in wireless technologies, which significantly boost the growth of diverse wireless networks, from single-hop wireless networks to multi-hop wireless networks. In the former, such as cellular networks and wireless local area networks (WLANs), every node is within one hop of a central controlled entity (*e.g.,* base stations, access points, etc.), and only communicates with the entity through single hop transmissions. Such networks require much infrastructure support, hence are expensive to deploy.

In comparison, multi-hop wireless networks are usually defined as a collection of nodes equipped with radio transmitters, which not only have the capability to communicate each other in a multi-hop fashion, but also be able to route data packets as a relay from the source to the destination. It is commonly popular in areas in which there is little or no communication infrastructure, or the existing infrastructure is expensive or inconvenient to use, where wireless users (or nodes) may still be able to communicate through the formation of a multi-hop wireless network. In other words, each node operates not only as a host but also as a router, forwarding packets for other nodes in the network (through discovering multi-hop routes) that may not be within direct wireless transmission range of each other. The idea of multi-hop wireless networking is sometimes also called infrastructure-

less networking, since nodes in the network dynamically establish routing among themselves to form their own network "on the fly."

The distributed nature of multi-hop wireless networks makes them suitable for a variety of applications where there are no assumed reliable central entities, or controllers, and their usage may significantly improve the scalability of conventional single-hop wireless networks. Some existing applications include but not limited to wireless backhaul networks supporting multimedia traffic, students using laptop computers to participate in an interactive lecture, business associates sharing information during a meeting, soldiers relaying information for situational awareness on the battlefield, and emergency disaster relief personnel coordinating efforts after a hurricane or earthquake, etc. Without loss of generality, multi-hop wireless networks can be further classified into three categories: mobile ad hoc networks (MANETs), wireless mesh networks (WMNs, [1]), and wireless sensor networks (WSNs, [2]).

1.1 Background Information

The initial research on multi-hop wireless networks started in the early 1970's when packet radio networks were studied. They received much wider attention since late 1990's, thanks to the IEEE standardization efforts and the commercial success in wireless networks. In this section, a brief background research is given on these networks.

1.1.1 Mobile Ad Hoc Networks (MANETs)

A mobile ad-hoc network consists of a collection of "peer" mobile nodes that are capable of communicating with each other without help of a fixed infrastructure [3, 4]. Each node is an end user as well as a router. The interconnections among nodes may change on a continual and arbitrary basis. Nodes within each other's

radio range communicate directly via wireless links, while those far apart use other nodes as relays in a multi-hop fashion. MANETs are suited for scenarios where an infrastructure does not exist, *e.g.,* in disaster recovery situations where existing communication networks are destroyed, and communications on battle fields where military units may move constantly and multi-hop connectivity may be desired. There has been extensive research on MANETs, especially the scheduling, routing and transport issues [5, 6].

1.1.2 Wireless Mesh Networks (WMNs)

The ongoing proliferation of wireless broadband data services is expected to lead to the increasing needs of wireless backhaul networks, where the typical upgrade of wired lines to high-speed fibre networks is not always an available and/or economically attractive solution [1, 7–11]. In these scenarios, WMNs, transporting data between the access network and the wired Internet, could potentially offer an appealing alternative. It could be widely adopted not only in hot-spots or fully wireless hot zones, but also in broader entire metropolitan area. WMNs must meet a number of technical requirements, namely, (a) the high capacity to forward the aggregated traffic from multiple access points, (b) the guarantees of a set of quality of service (QoS) requirements (*e.g.,* packet error rate (PER), throughput, and packet delay) of the end user applications, and (c) a large enough effective communication range. In order to satisfy these requirements, a range of novel techniques have to be exploited, including, but not limited to, multi-hopping, multiple antennas techniques, novel medium access control (MAC), routing, and admission control (AC) algorithms. However, it is also worth to note that WMNs are not restricted to wireless backhaul applications, but could be used for client access, scanning (required for high speed handover in mobile applications) applications, etc. as well.

1.1.3 Wireless Sensor Networks (WSNs)

A WSN consists of potentially large number of sensors, which are small, low-cost, low-power, and resource-constrained devices [2,12–14]. Similar to MANETs, the operations of WSNs do not require the infrastructure support, but sensors can propagate the sensed and partially-processed data over multiple hops. Furthermore, there are usually some sink nodes in WSNs, which are responsible for collecting the data, and may send the data to a processing unit via other wired or wireless links. WSNs are especially suited for environment monitoring in hazard or inaccessible places, where sensors are deployed densely and randomly, and the notion of quality of information (QoI) is required, such as the information accuracy, timeliness and completeness. Applications of WSNs include but are not limited to health care systems to monitor and assist patients, surveillance and targeting systems, smart home, etc. Many research efforts have been made on WSNs, especially energy efficiency, fault tolerance and scalability [2, 15] are among the active research topics due to resource constraints of sensors.

1.2 Thesis Motivations

Multi-hop wireless networks are becoming a new attractive network design paradigm owing to their low cost and ease of deployment, and have found more and more applications. However, to fully achieve the promising features of multi-hop wireless networks, many research problems still remain to be solved, from two general categories: the network protocols, and the network management.

1.2.1 Network Protocols

Medium Access Control (MAC) Protocols

MAC protocols, including scheduling algorithms, are responsible for coordinating the access from active nodes [16, 17]. They are responsible for providing efficient packet exchange between two or more nodes of the network. The challenges come from the error-prone nature of wireless channel, co-channel interference from adjacent concurrent transmissions, and the hidden/exposed-terminal problems. Since the MAC layer has a direct bearing on how reliably and efficiently packets can be transmitted among nodes along the routing path in the network, it does affect the QoS satisfaction of the network. Therefore, the design of a MAC protocol should address the unreliable time-varying channel properties in physical layer, scheduling conflicts, and applications' QoS issues together.

Routing Protocols

Routing is one of the core problems for packet exchange among nodes in multi-hop wireless networks [18, 19]. For QoS routing, it is not sufficient to only find a route from a source to one (or multiple) destination(s). This route also has to satisfy one (or more) QoS constraint(s). As the use of delay and bandwidth sensitive applications (*e.g.,* voice or video streams) increase, so does the need for QoS routing protocols in multi-hop wireless networks. The challenges thus arise. First, because of the nature of error-prone wireless links, resource reservations on adjacent links can influence each other in a 2-hop range, and thus it complicates the computation and the management of the bandwidth and delay restrictions. Second, even with successful reservations, the time-varying resource availability cannot always be guaranteed due to the dynamic aspects of the network.

Admission Control (AC) Schemes

The great deal of research attention for AC increases significantly recently [20–25], due to the growing popularity of multimedia applications (*e.g.,* voice, video, and broadband data services) and the central role AC scheme plays in QoS provisioning (in terms of the connection quality, blocking probabilities, packet delay, and throughput etc.) The challenges come from the inefficiency of lower layer protocols for the network resource management, where AC algorithms play major roles for QoS supports, not only for the individual user performance, but also for the overall network performance. Arriving new connections are granted, or denied, access to the network by the AC scheme based on predefined criteria, such as the network traffic conditions and resource availability. On the other hand, the heterogeneous nature of the protocols to use in any multi-hop wireless networks does require a degree of *transparency* of the AC algorithm, so that whatever advanced technologies to use, the AC scheme can always efficiently control the connection admission by estimating network resource availability.

1.2.2 Network Management

Building usually on top of the communication protocol layers, the core functionalities of the network management include network planning, deployment, configuration, operation, monitoring, tuning, repairing, and changing communication networks [26–31]. The difficulties of performing the network management come from the complexity of the network structure, stochastic traffic pattern, and heterogeneous operational contexts of communication protocols in use. Particularly, multi-hop wireless networks pose the new challenges as follows.

Large scale:

Multi-hop wireless networks such as WSNs and WMNs are expected to span a large scale, with respect to both the number of nodes and the size of the coverage area. WSNs may consist of tens of thousands of sensors, while WMNs may have tens to hundreds of nodes in a metropolitan area. To design efficient algorithms, methodologies, and network protocols to control such a large-scale network is a challenging issue.

The Notion of Service Quality:

Due to the nature for the majority of multi-hop wireless networks to support some notion of service quality, like QoS in WMNs and MANETs for delay and throughput sensitive applications, and QoI in WSNs to guarantee the sensing data quality, the new challenges for the network management thus come from the question of how to optimally allocate limited network resources serving multiple tasks with different data quality requirements at different time. Especially when these tasks dynamically come and go (which happens in most network scenarios), the resource availability also changes from time to time.

Inter-domain Communications:

There are not much research exposure on how to maintain inter-domain communications to support a notion of service quality, bridging wireless communications among multiple, disconnected, mobile subnetworks, which have not direct contact due to territory obstacles or extreme scenarios. How to perform the network management within such context to guarantee a notion of service quality is still an open issue.

1.3 Thesis Objectives

The overall aim of this Ph.D. dissertation is to propose several cross-layer design solutions to guarantee the notion of service quality in a variety of multi-hop wireless networks. The proposed design protocols, models, mythologies and policies belong to several layers of conventional communication protocols, *e.g.,* scheduling in MAC layer, routing and admission control in network layer, network management in application layer etc. Furthermore, this dissertation is also trying very hard to provide a few generic solutions wherever possible to a variety of underlined networks, wired and wireless. The main objective of this Ph.D. dissertation is to show that using some cross-layer design frameworks is a practical and effective way of improving the overall network performance and satisfying the applications' service quality requirements. To meet this objective, the dissertation addresses the following eight goals:

1. to propose a QoS routing algorithm to tackle the well-known NP-completeness problem of finding an optimal route in a wireless network with multiple QoS constraints;

2. to propose a cross-layer design solution integrating QoS routing and distributed scheduling algorithms, and to testify its efficiencies in a comprehensive OPNET simulation platform for WMNs;

3. to propose a generic admission control methodology to perform QoS control for any packet networks, wired and wireless, to show its applicability, scalability and feasibilities, and finally to testify the efficiencies of the proposed solution in a WMN scenario.

4. to propose a novel network management framework to fill the research gap between the external applications' service quality requirements and the inter-

nal network resource management, to derive the fundamental relationships for the trade-off among the service quality requirements, network lifetime, and task arrival and departure rate, to show the proposed solution's applicability, scalability and complexity, and finally to testify the its efficiencies in a WSN scenario.

5. to propose a POMDP control model to bridge communications among multiple, disconnected, and mobile subnetworks, to discuss the existence of the optimal policy and its computation complexity, to propose a heuristic algorithm for policy computation, and finally to show the efficiencies of the proposed algorithm compared with a few other schemes.

1.4 Thesis Structure

To address these research challenges, this Ph.D. dissertation proposes several cross-layer design solutions to ensure the notion of service quality, for instance the QoS supports in WMNs for backhaul applications and the QoI supports in WSNs for sensing tasks. An illustrative figure to show the thesis structure could be found in Figure 1.1; details of the motivations for each Chapter is introduced as follows.

Within the context of multi-hop wireless networks design, most of the current research on protocol designs are mainly based on a *layered* approach. By providing modularity and transparency in between, the layered approach has proven to be the robust and scalable in the Internet and become the *de facto* architecture for wireless systems. However, the spatial reuse of the spectral frequency, the broadcast, unstable, and error-prone nature of the wireless channel, and different operational time scales for protocol layers, all make the layered approach suboptimum for the overall network performance. For instance, bad resource scheduling in MAC layer can lead to huge amount of interference that affect the performance of the physical

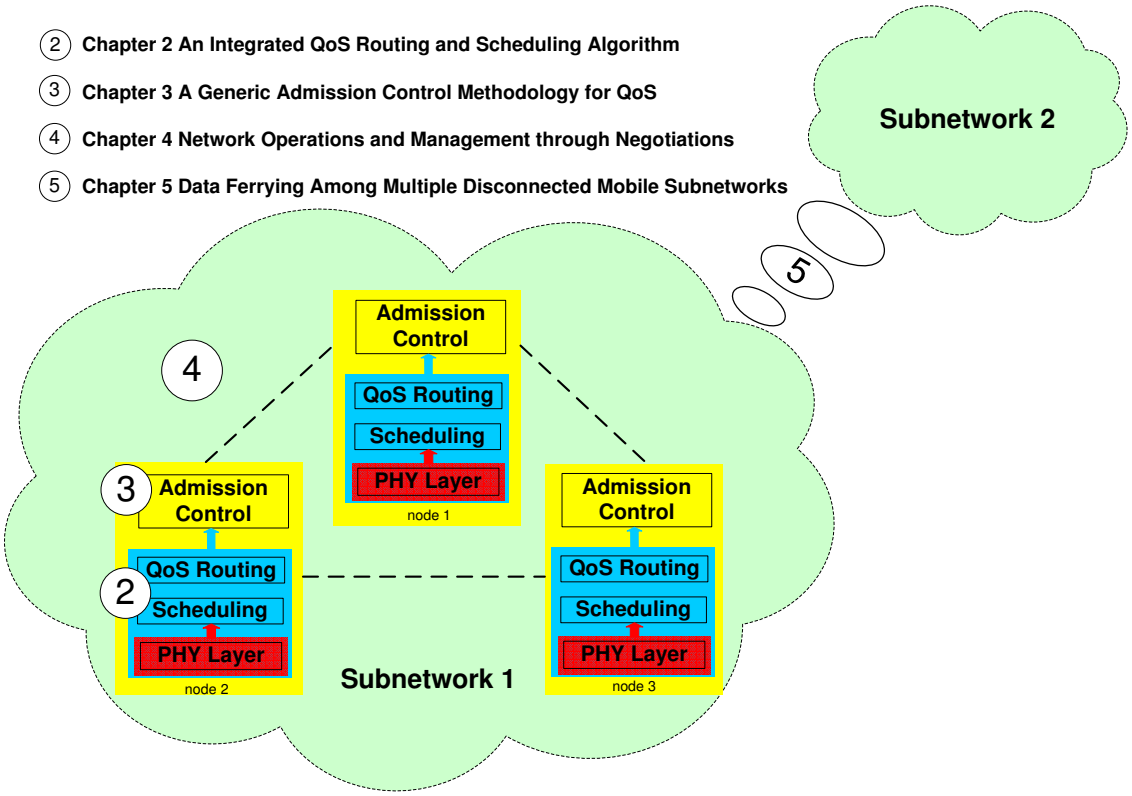

Figure 1.1: The structure of this Ph.D. dissertation in an illustrative figure.

(PHY) layer due to the reduced signal quality and ultimately deteriorate the overall network performance. Local capacity optimization with opportunistic scheduling techniques that exploit the multi-user diversity gain may increase the overall outgoing throughput of the nodes but they can also generate new bottlenecks in several routes in the network, etc.

Chapter 2 is thus motivated to adopt a *cross-layer* design approach, integrating a multi-constrained QoS routing algorithm and a distributed proportional-fair scheduling algorithm. The routing algorithm selects the satisfactory route to guarantee the end-to-end QoS, while the distributed scheduling algorithm enforces the routing demands in terms of aggregated link throughput to achieve in the long-run.

Chapter 3 is motivated by the research finding in Chapter 2 that one can-

not allow arbitrary large number of connections admitted into the network, and thus admission control needs to be enforced. Admission control is usually achieved by knowing the network capacity, however known to be difficult to estimate [32]. Especially for the multimedia traffic, connections with dynamic QoS requirements will consume different amount of network resources, making the network capacity highly dynamic and the estimation of the remaining resources extremely difficult. Furthermore, due to the co-channel interference and wireless channel fluctuations, the uncontrollable admission of the improper new connection can highly affect the resources of adjacent transmissions. Precise knowledge of the admission impact on overall network QoS will help the network perform optimal decision to the new connection without jeopardizing the proper operations of the existing connections.

Chapter 3 make three contributions. First, a subnetwork is generically modeled as a "black box", where the ingress node aggregates traffic as the input to the black box with parameters of the required QoS requirements, and an egress node serves as an output from the black box with a single defined parameter called *QoS performance index*. Second, the potential connection admission is characterized as the input change to the black box, and the change of output is approximated in a closed-form. Finally, admission control is performed by comparing the new output QoS performance index with its upper-bounded satisfactory value 1. Without loss of generality, we show that the proposed admission control methodology is generic, and has wide applicability to any packet networks, wired and/or wireless, and we discuss various feasibility issues of the proposed approach, *e.g.,* the impact of large connection throughput requirements, statistics feedback delay, and statistics collection time on the estimation accuracies.

Chapter 4 extends the contributions in previous chapters by identifying the research gap bridging applications' service quality demands (or, the "external" operations) and the network resource management (or, the "internal" operations). In

other words, how to optimally manage the network resources to satisfy all applications' service quality requirements (the bottom-up approach), and how to adapt the applications' service quality demands to fit in the network status (the top-down approach), should be addressed together in the research. In this chapter, the service quality is interpreted as QoI, like information accuracy, timeliness, and completeness, etc., from the applications' perspective for sensing tasks (or simply "tasks" used later) in WSNs, rather than the traditional QoS for backhaul applications.[1]

Different from existing approaches to always maximize a network utility [33, 34] known as *a priori*, first, we conduct *runtime* learning of the QoI benefit provided by the WSN to the tasks it supports by monitoring the level of QoI satisfaction (or, the *QoI satisfaction index* of a task) they attain in relation to the QoI they request. This relaxes the requirement for *a priori* knowledge of utility functions and facilitates the dynamic accommodation of tasks with heterogeneous requirements. Second, by proposing the concept of *QoI network capacity*, the ability of a WSN to host a new task (with specific QoI requirements) is expressed without sacrificing the QoI of existing tasks. Finally, an adaptive *negotiation* process is proposed to dynamically configure the usage of network resources to best accommodate all tasks' QoI requirements.

Chapter 5 is motivated by maintaining communications among multiple disconnected mobile subnetworks. Due to complex terrains (*e.g.,* obstacles or danger zone in between), the nodes operating in different network domains may not have direct contacts. Yet, to maintain the applications' service quality, communications are needed, for instance the emergency response scenarios in military coalition networks. We propose to use unmanned, sensor mounted data ferries (*i.e.,* the sensors are mounted on controllable mobile platforms such as UAVs [35]) to assist the

[1]It is interesting to see that QoI and QoS may have slightly different interpretations for their targeted applications, but they both focus on the broader aspects of service quality.

communications in a load-carry-and-deliver manner. In practice, complete sensing coverage of data ferries may not always be possible due to ground obstacles, vast network area, limitations of sensors, or simply because of the need of keeping the UAVs from being exposed to the adversary. In this chapter, we study in detail how to bridge communications in such challenged scenarios using dynamically controlled, unmanned data ferries.

Each data ferry is equipped with certain sensing, communications, and storage capabilities, and most importantly, with a *programmable* control logic which can control its navigation. Periodically, the data ferry senses the presence of nodes and uploads/downloads data upon contact, after which it will move to the next sensing point specified by the control logic and repeat the process. Meanwhile, the mobility of nodes make them move within their local network domain constantly. Although it is possible to infer statistically properties of their movements, it is often impractical to accurately predict how these nodes will move due to runtime randomness. The questions we investigate are: how should one control the data ferries to move intelligently based on the prior knowledge of node movements and the real-time partial observations? To the best of our knowledge, this is the first effort to address both runtime randomness and incomplete observations in the data ferry control.

To sum up, the notion of "cross-layer" design is carried through the presentation of this dissertation. We start from the integrated distributed scheduling and QoS routing algorithms, where the former exploits the multi-user diversity gain of the PHY layer and enforces the routing demands of the latter, while the latter requests QoS demands from the application layer to the former. Then, this dissertation further investigates the admission control aspect by estimating the admission impact, where it explicitly considers the QoS needs of application layer and exploits the PHY and MAC layer information to indicate the degree of QoS satisfaction. Next, this dissertation aims to further improve the design efficiency by proposing a negotiation-

based network management framework among applications' service quality demands and the network resource management in lower layers. Finally, the cross-layer design extends to maximize the link-level throughput (*i.e.,* the service quality) from the application's perspective among multiple disconnected subnetworks, where this dissertation proposes to use unmanned data ferries with programmable control logic. Extensive simulations have shown in each Chapter to verify the design efficiencies of the proposed models, protocols, methodologies and policies. Results have proved that the overall design framework achieves the best network performance compared with conventional network protocols or designs. Finally, conclusions are drawn and some future work is identified in Chapter 6.

Chapter 2

An Integrated QoS Routing and Scheduling Algorithm

THIS Chapter aims to tackle the afore-mentioned challenges for multi-hop wireless network design from *network protocol* perspectives, while using WMNs as illustrate examples. Cross-layer design for QoS in WMNs has attracted much research interest recently. Such networks are expected to support various types of applications with different and multiple QoS requirements. In order to achieve this, several key technologies spanning all layers, from physical up to network layer, have to be exploited and novel algorithms for harmonic and efficient layer interactions must be designed. Unfortunately, most of the existing works on cross-layer design so far focus on the interactions of up to two layers while different operational time scales for different protocol layers have been overlooked. In this chapter, we propose a unified framework that exploits the physical channel properties and multi-user diversity gain of WMNs, and by performing intelligent route selection and scheduling, we provide QoS satisfactions to a variety of underlying applications.

2.1 Introduction

WMNs are a relatively new and promising key technology for the next generation wireless networking that have recently attracted both the academic and industrial interests [36]. Such networks are expected gradually to partially substitute the wired network infrastructure functionality by being able to provide a cheap, quick and efficient solution for wireless data networking in urban, suburban and even rural environments. Their popularity comes from the fact that they are self-organized, self-configurable and easily adaptable to different traffic requirements and network changes. WMNs are composed of static wireless nodes/mesh routers (WMR) that have ample energy supply, as shown in Figure 2.1. Each node operates not only as an conventional access point (AP)/Internet gateway (IGW) to the internet but also as a wireless router able to relay packets from other nodes without direct access to their destinations [1]. The destination can be an IGW or a mobile user served by another AP in the same mesh network. Moreover, some nodes may only have the backhauling functionality, meaning that they do not serve any mobile user directly but their purpose is to forward other APs' packets.

WMNs must meet a number of technical requirements. First, they must meet the high capacity needs of the access nodes that have to forward the aggregated traffic of their underling users. Second, they have to cope with multiple, strict QoS requirements of the end user applications. Generally, QoS requirements can be divided into different groups according to their nature related to packet networks, *e.g.,* additive constraints, multiplicative constraints, and concave constraints. Let $d(n_i, n_j)$ be a metric for link (n_i, n_j) and $p = (n_1, n_2, \ldots, n_m)$ be a multi-hop route between the source node n_1 and the destination node n_m. Then the named

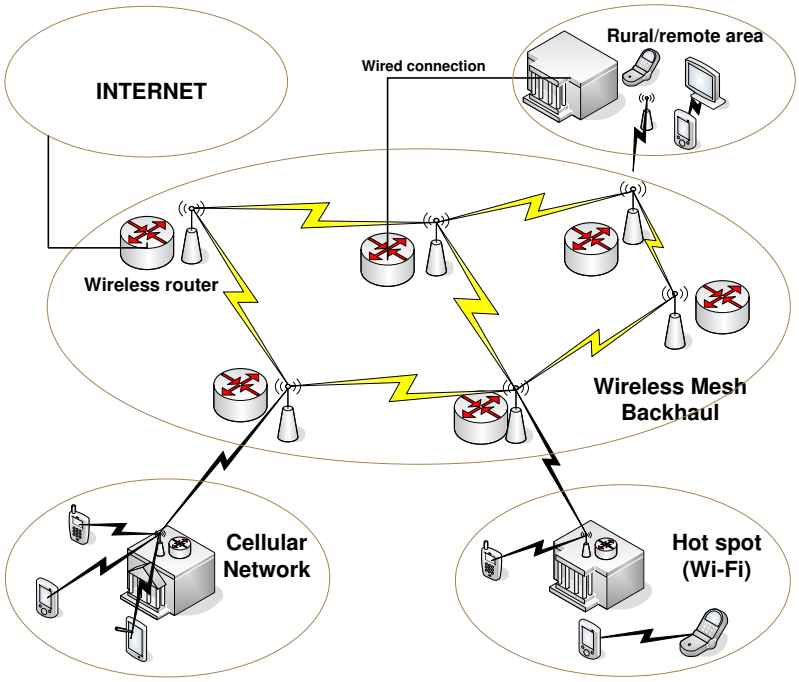

Figure 2.1: A typical wireless mesh network scenario.

constraints are defined as follows:

$$\begin{aligned} \text{Additive} \quad &: \quad d(p) = d(n_1, n_2) + d(n_2, n_3) + ... + d(n_{m-1}, n_m), \\ \text{Multiplicative} \quad &: \quad d(p) = d(n_1, n_2) \times d(n_2, n_3) \times \ldots \times d(n_{m-1}, n_m), \\ \text{Concave} \quad &: \quad d(p) = \min\left(d(n_1, n_2), d(n_2, n_3), \ldots, d(n_{m-1}, n_m)\right). \end{aligned} \tag{2.1}$$

The most commonly used constraints in WMNs are end-to-end (ETE) throughput, delay, and packet error rate (PER). Throughput (concave) denotes the amount of traffic along a certain route and is limited by the bottleneck link with the lowest throughput along this route. Delay (additive) indicates the time between sending out a packet from the source node and the reception of this packet at the destination node. ETE PER (multiplicative) refers to the probability of a packet to be erroneous on its way to the destination node, *e.g.,* because of collisions, topology changes or

weak radio signals etc.[1]

Finally, WMNs must provide a large enough effective communication range to ensure that no APs (or groups of APs) are isolated from the Internet gateways. In order to satisfy the above requirements, a range of novel techniques has to be exploited. Such technology enablers include but not limited to multi-hopping, various multiple antennas techniques, novel medium access control (MAC), and routing algorithms.

Furthermore, in traditional cellular network settings, the grade-of-service (GoS) has been a fundamental parameter to define the quality of voice services [37, 38], as a benchmark to define the desired performance of a particular trunked system by specifying a desired likelihood of a user obtaining channel access. However, in WMNs with different QoS requirements, the GoS can be defined as the probability that a specific QoS *level* will be guaranteed throughout the whole duration of the connection. Therefore, this GoS *threshold* can help control the number of connections that can be allowed at each level.

To provide both QoS and GoS provisions, unfortunately, most of the current work on WMNs protocol analysis and design is mainly based on a *layered* approach. This layered architecture by providing modularity and transparency between the layers, led to the robust scalable protocols in the Internet and it has become the *de facto* architecture for wireless systems. However, the spatial reuse of the spectral frequency, the broadcast, unstable and error prone nature of the channel and different operational time scales for protocol layers, make the layered approach *suboptimum* for the overall network performance. For instance, bad resource scheduling in MAC layer can lead to a significant amount of co-channel interference that affects the per-

[1]Besides these constraints, there are other interesting metrics for WMNs. The number of hops (additive) represents the number of links in a path. Energy (additive) takes the energy needed to send a packet from source to destination into account. Further QoS metrics include, *e.g.,* , signal strength (concave) and distance (additive).

formance of the PHY layer due to the reduced signal-to-interference-plus-noise-ratio (SINR) and ultimately deteriorates the overall network performance. Local capacity optimization with opportunistic scheduling techniques that exploit the multi-user diversity may increase the overall outgoing throughput of the transceivers but they can also generate new bottlenecks in several routes in the network, etc.

These are primarily why cross-layer designs for improving the network performance have been a focus of much recent work. In a cross-layer paradigm, the joint optimization of control over two or more layers can significantly yield performance improvement. Caution needs to be exercised though, since cross-layer design may potentially destroy the modularity and make the overall system fragile. Other importance challenges that have to be taken into account during the design of cross-layered solution for WMNs is the different operational time scales between coding, scheduling and routing algorithms; especially in the case that system performance estimations in different layers have to be performed. Moreover, since WMNs have to support a wide variety of applications and services, the multi-dimensionality of QoS requirements requires the joint satisfaction by the cross-layer approach, but proven to be NP-complete [39].

In this chapter, we propose a novel cross-layer design paradigm to support QoS in WMNs that includes a multi-constrained QoS routing algorithm in the network layer and a distributed opportunistic proportional fair (OPF) scheduler in MAC layer. Our contributions are summarized as follows:

1. We propose a multi-constrained QoS routing algorithm to overcome the NP completeness of integrating three QoS metrics, delay, throughput and PER, in a unified utility function.

2. We manage to successfully integrate the proposed routing algorithm with a novel opportunistic scheduling scheme to maximize the network throughput.

3. We propose a GoS management scheme supporting multi-level QoS requirements for connections, where network resources are organized and used in an optimal way.

The remainder of the Chapter is organized as follows. After summarizing the related work in Section 2.2, the system model is introduced in Section 2.3. The proposed QoS routing algorithm is discussed in Section 2.4. Section 2.5 describes the distributed opportunistic scheduler and its interaction with the routing algorithm. The GoS management for multi-level QoS is demonstrated in Section 2.6. Extensive simulation results follow on Section 2.7 while Section 2.8 summarizes this chapter.

2.2 Related Work

Cross-layer designs have been widely used to improve the network performance [40–44] that generally includes two aspects of design methods: theoretical mathematical modeling and practical protocol designs.

Layered protocol architecture is one of the most important factors that have made networking so successful. However, there has been a lack of a systematic approach to analyze whether layering of protocols is optimal or not. The layering as optimization decomposition [45] fills a gap between theoretical methods and practical aspects of protocol design. In this method, various protocol layers are integrated into one single coherent optimization function, in which asynchronous distributed computation over the network is applied to solve a global optimization problem in the form of generalized network utility maximization (NUM). The key idea of layering as optimization decomposition is to decompose the optimization problem into sub-problems, each corresponding to a protocol layer and functions of primal or Lagrange dual variables, coordinating these sub-problems correspond to the interfaces between layers. However, the above formulation is based on a deterministic fluid model that

cannot capture the packet-level details, microscopic queuing dynamics, and wireless link fluctuations.

On the other hand, cross-layer design through individual (or some) protocol layer(s) can significantly improve the network performance in two ways: loosely coupled and tightly coupled. In the loosely coupled cross-layer design, optimization is carried out without crossing layers but focusing on one protocol layer. Parameters in other protocol layers are taken into account by information exchange and deliveries from multiple layers to enforce the cross-layer design. With such information, the performance is improved because a better (*i.e.,* more accurate and reliable) parameter is used, but the algorithm itself does not need a modification. Nevertheless, in the tightly coupled cross-layer design, merely information sharing between layers is not enough, but algorithms in different layers are optimized altogether as one optimization problem. Our proposed cross-layer design architecture takes the advantage of loosely coupled design paradigm where scheduling and routing algorithms exchanges QoS information like delay, throughput, and PER, and they both obtain PHY layer information like SINR. Due to optimization execution across layers, we can expect that a better performance improvement can be achieved, and the advantage of adopting this design approach does not totally abandon the transparency between protocol layers.

Researchers, meanwhile, have been focusing on individual protocol layer design for QoS in wired/wireless networks. The problem of finding a path subject to two or more independent additive and/or multiplicative constraints in any possible combination, also known as the multi-constrained path (MCP) problem, is NP-Complete. QoS Routing is NP-Complete when the QoS metrics are independent, real numbers or unbounded integers. The proof of NP-Completeness relies heavily on the correlation of the link weight metrics. Garey and Johnson [46] were the first to list the MCP problem with there are only two metrics as being NP-complete, but

they did not provide a proof. Wang and Crowcroft have provided this proof for more than two metrics in [39,47], which basically consisted in reducing the MCP problem for a two metrics case to an instance of the partition problem, a well-known NP-complete problem [46]. [48,49] have addressed extensively on multi-constrained QoS routing algorithms in wired network based on network state [50,51] to overcome the NP-complete difficulties of providing optimal routes that guarantee multiple QoS constraints [39]. Meanwhile, QoS routing algorithms for wireless ad-hoc networks have been previously explored in [52–56]. However, they either overlook the multi-hop queuing delays since only the packet processing time was considered or simply calculate the available bandwidth in terms of slot and reserved for QoS connections that fails to exploit the opportunistic scheduling gain in fast-fading channels.

Scheduling for WMNs has drawn a lot of research attention recently that generally includes centralized [57,58] and distributed solutions. Centralized scheduling algorithms are based on graph theory assuming that a central controller has full knowledge on network. The method finds the optimal set of non-overlapping links with the highest total throughput of the graph, however proven NP-complete [59,60]. Distributed solutions like [61] is commonly used as the MAC protocol in wireless ad-hoc networks, and IEEE 802.16 standard specifies its own MAC layer in [62]. However, due to the completely random link selection, neither of the algorithms takes advantage of multi-user diversity in the wireless environments, nor providing QoS with routing algorithms.

2.3 System Model

Consider a WMN comprises a set of n_r number of WMRs, denoted as $V_R = \{v_r | r = 1, 2, \ldots, n_r\}$ and a set of n_g number of IGWs denoted as $V_G = \{v_g | g = 1, 2, \ldots, n_g\}$. If further consider an arbitrary node i, it may have J_i one-hop neighbors within fixed

transmission range, where these neighbors are denoted as $\{j = 1, 2, ..., J_i\}$. The MAC layer, or scheduling algorithm, of each node maintains a set of queues for each direction of transmission, i.e., the queue for direction $(i, j)\&(j, i), \forall j = 1, 2, ..., J_i$. In other words, newly arrived packets (the reception) will be placed into the corresponding queue according to the pre-determined route that they belong to, and the outgoing packets (the transmission) will be popped up from the corresponding queue as well.

We assume that the network runs under a time-division multiple access (TDMA) slotted framework while we also assume that all nodes are synchronized to the slot boundaries[2]. Each time frame consists of the control phase, comprises f_c fixed-size time slots for control messages, and the data transmission phase that consists of f_d fixed-size time slots for data. During the period of one time frame, we assume the block fading channel that remains relatively constant. Scheduling decisions are taken by all nodes in the network simultaneously at the beginning of each time frame at the control phase, and stay unchanged until the next frame. The PHY layer employs adaptive modulation and coding techniques (AMC), where there are a finite V transmission modes, each of which corresponds to a unique modulation and coding scheme and one particular interval of the received SINR. The transmission rate at each mode is proportional to its spectral efficiency, *i.e.,* transmission mode v can transmit maximum c_v packets in one time slot, or $H = f_d c_v$ packets in a time frame, where $v = 1, 2, ..., V$,. Furthermore, in order to reduce the interference to adjacent concurrent transmissions and increase the frequency reuse and channel capacity, the WMRs are equipped with directional antennas with steerable-beam. Power control is not considered in this phase, *i.e.,* all the nodes have the same fixed transmission power.

[2]The investigation of providing full synchronization among all WMRs is out of the scope of this Ph.D. dissertation, but could be found in [63–65]

Each WMR independently generates traffic, or connections, according to some stochastic traffic arrival and departure processes. Each connection q has to fulfill a set of QoS constraints that include ETE packet delay D_q^r, throughput T_q^r, and PER E_q^r, where superscript r denotes the *required* value. We denote this QoS requirement set as (D_q^r, T_q^r, E_q^r). We also consider the multi-level QoS case in the network where a typical application could be the transmission of the hierarchically encoded video where the video bit stream is composed of a set of hierarchical sub streams, each one of which enhances the quality through different level of required bit streams (*e.g.,* in MPEG video). Without loss of generality, we assume there are two levels of QoS requirements associated with the connection, high level (HQoS) and low level (LQoS). We further denote the the LQoS requirement set as: $(D_q^{r,l}, T_q^{r,l}, E_q^{r,l})$.

Let Ω_{sg} denote the possible route set from a source WMR s to a particular IGW g. A route Ω_{sg}^k from a source WMR with index s to a destination IGW indexed g within the route set Ω_{sg} is concatenated by a set of links $\{(v_i, v_j)\}, \forall v_i, v_j \in V_R \bigcup V_G$. Therefore, we could formally express the route from s to g as (2.2), where total m candidate routes exist. For the k^{th} route,

$$\Omega_{sg}^k = \left\{ \biguplus (v_i, v_j) \middle| \forall \{v_i, v_j\} \in V_R \cup V_G \right\}, \tag{2.2}$$

where $k = 1, 2, \ldots, m$, and notation $\biguplus$ denotes the concatenation of a set of links. In the following discussions, we use (v_i, v_j) and (i, j) for the link between node v_i and v_j interchangeably.

2.4 QoS Routing Algorithm

As it has been mentioned above that the problem of providing an optimal route that guarantees multiple QoS constraints has been proven to be NP-complete [39].

Therefore, in order to facilitate the information delivery and exchange among PHY, MAC and network layers, we define a generalized QoS utility that unifies multiple QoS constraints into one metric to uniquely denote the level of QoS satisfaction.

Given a connection q with three QoS requirements (D_q^r, T_q^r, E_q^r), ETE delay, throughput, and PER respectively, we introduce a concept of the *QoS outage ratio*, R, which is experienced by each QoS metric, defined as the ratio between the *attained* parameter measurement and the *requested* value. It is worth to note that the attained value here is the value attained by previous completed connections, which have traversed a particular route, but not the attained performance measurement of its own connection. More specifically, we define the ratio "R" for each of the QoS requirement as follows:

Delay Outage Ratio:

For connection q, ETE packet delay outage ratio $R_q^D(k)$ on route Ω_{sg}^k is defined as the ratio between the attained delay measurement $D^a(k)$ (including queuing and transmission delays), and the required delay value D_q^r, *i.e.,*

$$R_q^D(k) = \frac{D^a(k)}{D_q^r}. \tag{2.3}$$

Suppose that the receiving end of link (v_i, v_j) keeps track of the packet delay for each packet, and the mean value is denoted as $D^a(i,j)$, then the ETE packet delay on multi-hop route Ω_{sg}^k could be derived as:

$$D^a(k) = \sum_{(i,j)\in\Omega_{sg}^k} D^a(i,j). \tag{2.4}$$

PER Outage Ratio:

For connection q, PER outage ratio $R_q^E(k)$ on route Ω_{sg}^k is defined as the ratio between the attained PER $E^a(k)$, and the required PER value E_q^r. Therefore, we have:

$$R_q^E(k) = \frac{E^a(k)}{E_q^r}. \tag{2.5}$$

Suppose that the receiving end of link (v_i, v_j) keeps track of the number of erroneous packets $\mathrm{N}_e(i,j)$ received at v_j within an predetermined time interval, then the attained PER over the wireless link (v_i, v_j) could be approximated by the ratio of:

$$E^a(i,j) = \frac{\mathrm{N}_e(i,j)}{\mathrm{N}_e(i,j) + \mathrm{N}_c(i,j)}, \tag{2.6}$$

where $\mathrm{N}_c(i,j)$ denotes the number of correct packets. Next, without loss of generality we assume the errors occurred on each link is independent of the others, then the attained ETE PER over the multi-hop route Ω_{sg}^k could be derived as:

$$E^a(k) = 1 - \prod_{(i,j)\in\Omega_{sg}^k} \left[1 - E^a(i,j)\right], \tag{2.7}$$

as it is a multiplicative constrain.

Throughput Outage Ratio:

For connection q, throughput outage ratio $R_q^T(k)$ on route Ω_{sg}^k is formulated as the ratio between the throughput requirement T_q^r and attained *bottleneck* link throughput $T^a(k)$. Therefore, we have:

$$R_q^T(k) = \frac{T_q^r}{T^a(k)}. \tag{2.8}$$

Suppose that the receiving end of link (v_i, v_j) keeps track of the amount of packets received at v_j within an predetermined time interval t_{interval}, then the the throughput

could be computed within the sliding time window as:

$$T^a(i,j) = \frac{L(\mathrm{N}_e(i,j) + \mathrm{N}_c(i,j))}{t_{\text{interval}}}, \tag{2.9}$$

where L denotes the packet length, and $\mathrm{N}_e(i,j), \mathrm{N}_c(i,j)$ denote the number of erroneous and correct packets respectively. Then, the bottleneck throughput over the multi-hop route Ω_{sg}^k could be derived as:

$$T^a(k) = \min_{(i,j)\in\Omega_{sg}^k} T^a(i,j). \tag{2.10}$$

Averaging the Measurements:

The proposed QoS routing algorithm is a measurement-based approach, where the measurements collections and processing are central of the algorithm. Next we briefly introduce our approach. After each packet flow over the wireless link (i,j), it will receive the per-link delay, throughput, and PER information. These measurements are constantly averaged by exponential smoothing to obtain the average link quality measurement $D^a(i,j), T^a(i,j), E^a(i,j)$, and used for QoS routing.

2.4.1 QoS Routing Objective Function

It is worth noting that some constraints may not be critical in some applications (for instance, many broadband data services may not be delay sensitive). In order to efficiently cope with this issue we introduce the indication function 1_p, where $p = D, T, E$, expressed as:

$$1_p = \begin{cases} 1 & \text{if parameter } p \text{ is critical for connection } q, \\ 0 & \text{otherwise.} \end{cases} \tag{2.11}$$

An example of the resource reservation margin factors and indication func-

Table 2.1: The values of resource reservation factors β for different applications

	voice-over-IP	Interactive-video	Broadband Data
1_D	1	1	0
1_T	1	1	1
1_E	1	1	1

tions chosen for three types of traffic in the network, namely, (a) voice-over-IP, (b) interactive-video, and (c) broadband data services respectively, is demonstrated in Table 2.1.

Therefore, to fulfil the set of QoS requirements, a source-to-gateway route Ω_{sg}^{k} will be feasible if and only if all defined outage ratios are less than one, or mathematically to the maximum of the three ratios has to be less than one, *i.e.,*

$$\max\Big[1_D R_q^D(k), 1_T R_q^T(k), 1_E R_q^E(k)\Big] \leq 1, \forall \Omega_{sg}^{k} \in \Omega_{sg}. \tag{2.12}$$

2.4.2 Route Selection to Support QoS

The proposed route selection criterion is given by the minimum of the maximum QoS outage ratios for a set of route Ω_{sg}, as:

$$\Omega_{sg}^{k^*} \longleftarrow \min_{\forall \Omega_{sg}^{k} \in \Omega_{sg}} \max\Big[1_D R_q^D(k), 1_T R_q^T(k), 1_E R_q^E(k)\Big], \tag{2.13}$$

and route $\Omega_{sg}^{k^*}$ is finally chosen if and only if the value of the right hand side is smaller or equal than 1; otherwise there is no satisfactory route for connection q. In other words, we are choosing the route with the minimum overall QoS outage probability.

2.4.3 Routing Procedures

Routing discovery phase requires each receiving node on one side of edge (i, j) records one-hop delay, link throughput and PER information. Delay information is collected by stamping the packet when it arrives at the transmission node v_i and when it reaches the receiving node v_j; throughput information is collected by monitoring the amount of receiving traffic within an interval; PER information is collected by tracking the probability a packet to be erroneous over the wireless link (i, j).

It is worth noting that the proposed routing procedure is similar to probe based routing algorithm in [66–68], also the follow-up work in [69–71], and the QoS routing in cognitive packet networks [72]. For the former, probes are launched to detect different routes between a source and a given destination to collect routing information. QoS routing decisions are based on the success of probes, and this decision is usually not the optimal one but a satisfactory one. For the latter, smart or cognitive packets are used to discover routes for connections; they are routed using a reinforcement learning algorithm based on a QoS goal. However, the main contribution of our QoS routing algorithm is the unique way to integrate multiple QoS constraints into one single representative utility function, and used for route selection, but not he routing procedure itself. In other words, our defined QoS routing metric can also be implemented in any probe based routing algorithms and routing in cognitive networks.

Next, by introducing an example of routing discovery procedures in Figure 2.2, we show the mechanism of our proposed QoS routing algorithm. Routing discovery procedure is initialized when new connections are accepted by certain nodes. In Figure 2.2, suppose WMR 1 serves as the source, and in order to find the route to carry the connection, it generates a request packet *REQ* containing the required QoS constraints. Then, it sends the *REQ* to its one-hop neighbours

through the allocated time slot in the control frame, while ticking a clock with a certain duration of expiration. Before this clock expires, if WMR 1 does not receive any reply message *REP*, it will try to regenerate a request packet and broadcasts it to the whole network due to previous packet loss (but only limited REQ packets will be generated). In Figure 2.2(a), when WMR 2 receives the *REQ* from WMR 1, it computes the average of all previously received values for one-hop delay measurements $D^a(1,2)$, link throughput measurements $T^a(1,2)$, and PER measurements $E^a(1,2)$, which are then piggybacked in the original *REQ* packet. The next step is to send this packet to WMR 1, WMR 3 and WMR 6 through the allocated time slots in control frame. Nevertheless, WMR 1 will simply discard the packet as it is just bounced back from WMR 2 without reaching the destination. Correspondingly, when WMR 3 receives the REQ packet, they will average the delay measurement $D^a(2,3)$, throughput measurement $T^a(2,3)$, and PER measurement $E^a(2,3)$, over the wireless link (v_2, v_3), and piggyback into the original *REQ* packet. Same procedure applies to WMR 6. Up to now, the REQ packet has information over two routes, $1 \to 2 \to 3$ and $1 \to 2 \to 6$.

All other nodes in the network repeat the above procedures until the gateway node WMR 5 receives the *REQ* packet. Afterwards, the reply procedure is initialized as shown in Figure 2.2(b), where WMR 5 sends a reply packet *REP* back to WMR 1 through two different routes, *i.e.,* $5 \to 4 \to 3 \to 2 \to 1$ and $5 \to 4 \to 7 \to 6 \to 2 \to 1$. By calculating the value of right hand side in (2.13), WMR 1 chooses the "best" route obtained before the clock expires, where the "best" means the minimum QoS outage or the maximum QoS satisfaction. It is also worth to note that WMR 1 does not need to wait for all *REQ* packets to come back, not only because of the exponential number of possible routes even for reasonable network size, but also we only need to find a route to meet certain QoS requirements, but unnecessarily find the "optimal" one if generating too much control overhead. This is controlled by

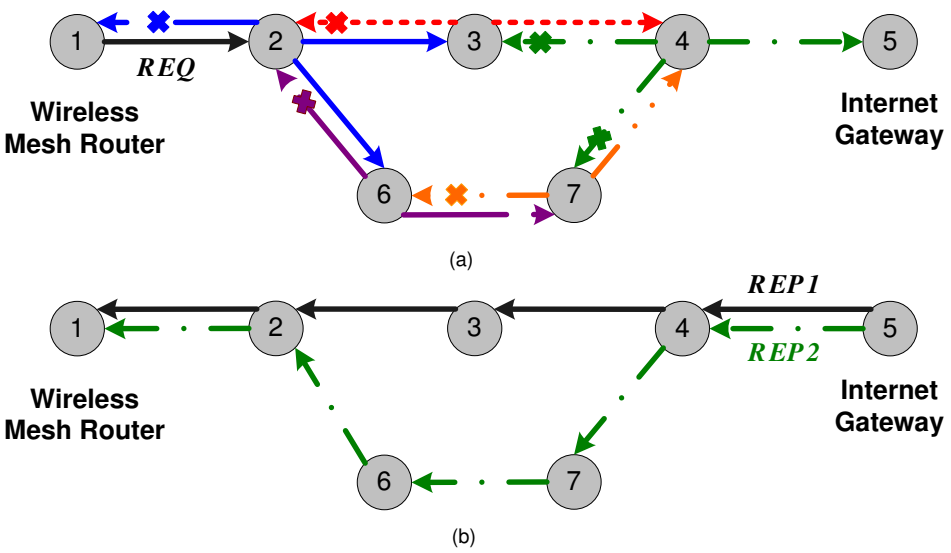

Figure 2.2: An example to demonstrate the route discovery procedures. (a) WMRs send *REQ* packets to their immediate neighbors, and (b) Gateway node sends *REP* packets back to source through the routes just found.

ticking the clock in WMR 1.

2.5 Distributed Opportunistic Proportional Fair Scheduling Algorithm

Once the route from the source to the destination is found to meet certain QoS requirements, the task is left for MAC layer, or the scheduling algorithm, to enforce the packet switching at a much finer time scale, *i.e.,* time slot by time slot. We assume that each node schedules one of the links associated with it in the control frame. Then the objective of our scheduling algorithm is to identify not only the duplex mode (transmitting or receiving) but also the specific direction (to or from which neighbour) of the next communication in an opportunistic manner (by "opportunistic", we mean by exploiting the multiuser diversity inherent in multiuser wireless networks to enhance total system throughput). For example, if a node is

receiving a great deal of interference, it may be more appropriate for the node to choose to transmit, provided that the intended receiver is expected to receive properly. On the contrary, if a node finds that one of its incoming links of the highest profit among all of its associated links, then the node may prefer to receive from that link. In our scheduling algorithm, every directional link is assigned with a utility representing the benefit of transmitting on this link in the next time frame, and hence the opportunistic approach is to choose a combination of concurrent links with the highest *aggregated* instantaneous utility.

On the other hand, uncertainty of link capacity in WMNs due to randomness of PHY layer protocols and wireless channel may degrade the performance of routing protocols. Furthermore, it is difficult to guarantee system performance if an opportunistic MAC layer is deployed, because opportunistic approaches usually introduce more fluctuating instantaneous performance at individual nodes. Therefore, it is important to propose a scheduling utility (or metric), denoted as U_{ij} for link (i, j), which not only achieves opportunistic gain but also supports QoS as committed by the routing algorithm in use. Otherwise, the QoS promised by the routing protocol to its applications cannot be guaranteed.

The proposed co-operation between the scheduling and routing algorithms is in a "request-enforce" manner. It is desirable for the routing layer to specify a *target* throughput allocation for each link, and then request the scheduling algorithm to enforce such throughput allocation. It is worth noting that rather than achieving the precise target throughput for each link, (or "hard" QoS), the objective of our scheduling algorithm is to achieve the *relative* target throughput scaled by a per-node (not per-link) proportional constant, (or "soft" QoS).

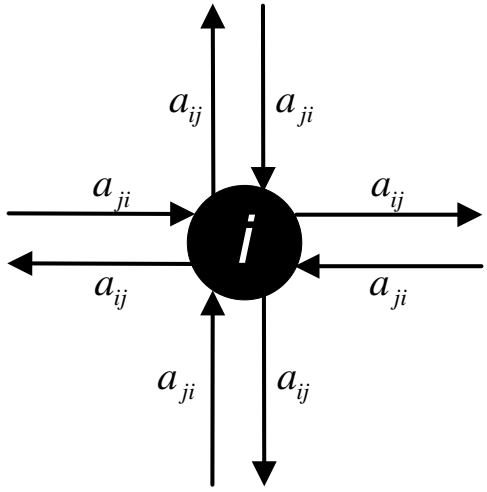

Figure 2.3: The routing demand for incoming and outgoing links of node i.

2.5.1 The Link Utility

Since we are aiming to propose a distributed scheduling framework, we shall focus on one individual node in the following derivation of a new utility definition. Here we treat the incoming and outgoing links equally as competitors. For an arbitrary node i with J_i neighboring nodes, it has maximum $2J_i$ candidate links to schedule in every time slot. The routing algorithm periodically estimates the throughput demand on each link associated with a pair of nodes in the next time frame, and provides the scheduler with a vector of target throughput to achieve, dented as $\underline{\tau}_i = (\tau_{i1}, \tau_{i2}, \ldots, \tau_{iJ_i}, \tau_{1i}, \tau_{2i}, \ldots, \tau_{J_i i})$, where τ_{ij} denotes the target throughout over link (v_i, v_j), as shown in Figure 2.3. In our proposed QoS routing algorithms, routing demand τ_{ij} is computed as,

$$\tau_{ij} = \sum_{\forall q} \beta_T T_q^r, \text{ if } (i,j) \in \Omega_{sg}^{k^*}(q), \tag{2.14}$$

where $\Omega_{sg}^{k^*}(q)$ denotes the chosen route by QoS routing algorithm for connection q, and τ_{ij} represents the accumulated throughput demands of all connections running through link (i, j).

Then[3] our goal here is to define an appropriate utility with which the scheduler's allocation of the long-run throughput $\underline{\phi}_i = (\phi_{i1}, \phi_{i2}, \ldots, \phi_{iJ_i}, \phi_{1i}, \phi_{2i}, \ldots, \phi_{J_i i})$ for all links is proportional to the target allocation $\underline{\tau}_i$, *i.e.*, $\underline{\phi}_i^* = c_i \underline{\tau}_i$, where c_i is a positive and proportional constant for node i and $\underline{\phi}_i^*$ is the *optimal* solution for node i. [74] proved that if the optimization problem for each node i is to maximize the objective function $f(\underline{\phi}_i)$ as,

$$\begin{aligned} \{\phi_i^*\} &= \arg\max_{\underline{\phi}_i} f(\underline{\phi}_i) \\ &= \arg\max_{\underline{\phi}_i} \sum_{j=1}^{J_i} (\tau_{ij} \log \phi_{ij} + \tau_{ji} \log \phi_{ji}) \end{aligned} \tag{2.15}$$

$$\text{subject to: } \sum_{j=1}^{J_i} (\phi_{ij} + \phi_{ji}) \leq C_i, \forall i,$$

where C_i denotes the node capacity, and under the assumptions of half-duplexing and single antenna model, the node capacity equals to the link capacity. Then the optimal solution $\underline{\phi}_i^* = (\phi_{i1}^*, \phi_{i2}^*, \ldots, \phi_{iJ_i}^*, \phi_{1i}^*, \phi_{2i}^*, \ldots, \phi_{J_i i}^*)$ is directly proportional to $\underline{\tau}_i$ element by element. Correspondingly, the physical meaning of the above proportionality is that the optimal solution $\underline{\phi}_i^*$ for the optimization problem is proportional to the target throughput allocation $\underline{\tau}_i$. Therefore, by using the scheduling utility (or metric) for all outgoing link $(i,j), \forall j = \{1, 2, \ldots, J_i\}$, and incoming link $(j,i), \forall j = \{1, 2, \ldots, J_i\}$ of node i as:

$$U_{ij} = \tau_{ij} \frac{\rho_{ij}}{\phi_{ij}}, \tag{2.16}$$

the routing demands is successfully enforced in the long-run. In (2.16), ρ_{ij} denotes the instantaneous link capacity for (i,j), which is calculated from Shannon's capacity

[3]For the completeness of this dissertation and for the purpose of enforcing the overall understanding of the proposed cross-layer approach, the rest of this subsection is necessary and thus referenced from Dr. Yun Hou's Ph.D. dissertation made in 2009 at Imperial College London [73]. However the author of this dissertation does not claim as his own contribution.

formula, as:

$$\rho_{ij} = W \log(1 + \kappa_{ij}^t \gamma_{ij}^t), \tag{2.17}$$

and ϕ_{ij} is the long-time average of ρ_{ij}. W is the system bandwidth, γ_{ij}^t is the receiving SINR and κ_{ij}^t captures the unpredicted interference effects.

2.5.2 The Scalability

The major concern of the scalability issues of the used distributed scheduling framework comes from the assumption of full time synchronization of the WMRs; however it has proven feasible in [63–65]. On the other hand, one may argue that the MAC layer of each WMR needs to maintain a set of queues for each direction of transmission/reception, which is feasible for static deployment of WMRs with sufficient power supply and computation capability. Other solutions may include using a prioritized queue for all directions of transmission/receptions and further algorithm to support this is needed.

2.6 Multi-Level QoS Management for GoS

In order to increase the flexibility of the network resource management to handle both the existing and the new connections, we introduce a novel multi-level QoS management scheme. The aim of this scheme is to reduce the blocking probability of the new connections, while at the same time maintaining a low outage probability for all existing ones, *i.e.,* we are trying to maximize the number of simultaneous connections with satisfactory QoS experience offered by the network. A typical application of the considered multi-level QoS scheme is the transmission of the hierarchically encoded video where the video bit stream is composed of a set of hierarchical sub streams, each one of which enhances the quality through different level of required bit streams (*e.g.,* in MPEG video).

However, we will not allow all connections to degrade their service quality to release network resources, but a certain level of GoS has to maintain. Under the proposed multi-level QoS context, we define GoS as the ratio of the number of accepted high-QoS (or, HQoS) connections over the overall number of the served connections in the network, if we assume another level of QoS is denoted as low-QoS (or, LQoS). This GoS definition can thus be translated to the probability a connection to be served in HQoS, which has to be higher than the GoS threshold G_0, *i.e.,*

$$\text{GoS} = \frac{\text{N}_{\text{HQoS}}}{\text{N}_{\text{HQoS}} + \text{N}_{\text{LQoS}}} \geq G_0. \tag{2.18}$$

It is worth noting that in real network scenarios, there is a mixed of connection where some applications may have strict QoS requirements, but not degradable; and this will be simulated later.

The novelty of the proposed algorithm is that given that at any time GoS $\geq G_0$ has to be satisfied, we allow the QoS routing algorithm to degrade the ongoing HQoS connections' service quality to the low level, or LQoS. In this way, it maximizes the number of simultaneous connections in the network while it optimizes the provided end-user QoS experience. The functionality of the proposed multi-level QoS management scheme (for simplicity, only two-levels of throughput have been considered) is described in the following steps:

1. The source node uses the proposed QoS routing algorithm to initiate the route discovery phase to collect statistics for each route, including ETE packet delay $D^a(k)$, throughput $T^a(k)$, and PER $E^a(k)$. These statistics are passed to the source node for satisfactory route selection.

2. Given the high QoS requirement of the new connection (D_q^r, T_q^r, E_q^r), the source node use (2.13) to find the best route to support. If it fails to find any route that provides HQoS requirement, it re-computes (2.13) trying to accommodate

the connection with low-level QoS (LQoS) requirement $(D_q^{r,l}, T_q^{r,l}, E_q^{r,l})$.

3. If it fails again to guarantee LQoS requirement, before performing the rejection, it tries to degrade the level of ongoing HQoS connections to LQoS requirements, given that the condition GoS $\geq G_0$ must be satisfied. Then, the routing procedure is called to accommodate LQoS requirement until one route k^* is found, and an admission signal is released; otherwise the new connection is unfortunately rejected.

Figure 2.4 depicts the real-time performance of the proposed scheme. The GoS threshold has been set to $G_0 = 0.9$ while the QoS requirements of LQoS connections are around half of that of the HQoS ones; for instance in video transmission, the throughput demand of HQoS connections is around $2Mbps$, while that of the LQoS connections is around $1Mbps$. It can be observed that the offered GoS converges to the GoS threshold as more connections arrive and are served by the network.

2.7 Simulation Results

2.7.1 OPNET Simulator

We uniquely develop a time-slotted, event-driven OPNET [75] simulator which comprises a number of randomly deployed clients/servers, WMRs, and IGWs in 2-D square in a way that no disconnected clusters of nodes exist in the network, as shown in Figure 2.7. A number of client and servers are attached to the backhaul network to emulate the access points, where traffic is generated from the clients according to the Poisson process to be routed to certain IGWs, or servers. For each WMR, we employ the conventional layered communication protocols including PHY, MAC, network, and application layers, but we allow information exchange among layers to facilitate the proposed cross-layer design.

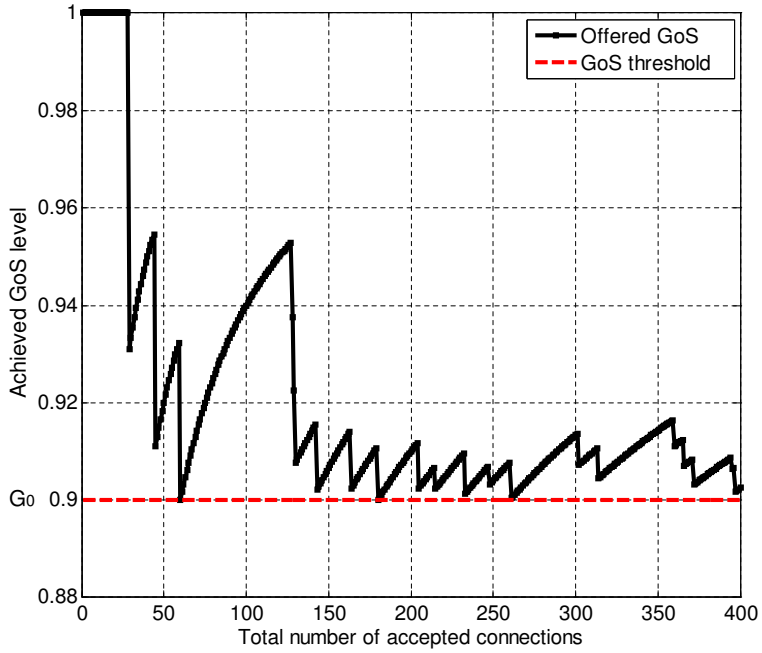

Figure 2.4: Real-time simulation of the offered GoS as a function of the number of accepted connections from time to time, where it can be seen that the offered GoS decreases and gradually approximates the predetermined GoS threshold value $G_0 = 0.9$ when more connections arrive.

The conventional OPNET PHY layer by default only has omni-directional antenna model and large scale path loss fading model, where we significantly enhance this model to include the adaptive modulation and coding (AMC) schemes, the Rayleigh fading channel model [76] and directional antenna with steerable beams. Figure 2.5 shows the way to compute the received PER with the corresponding received SINR value, where different combinations of modulation and coding schemes are used. Figure 2.6 shows the used directional antenna model where the main lobe is 30dB higher than the side lobe. The transmission power is computed to guarantee the reach of fixed transmission range without interference and the SNR should be above certain threshold. An important feature of this directional antenna model is the steerable beam, *i.e.,* when the scheduling decision is made by the MAC layer of each WMR, the beam of each antenna is switched pointing to the corresponding transmitter/receiver so that the amount of co-channel interference could be largely

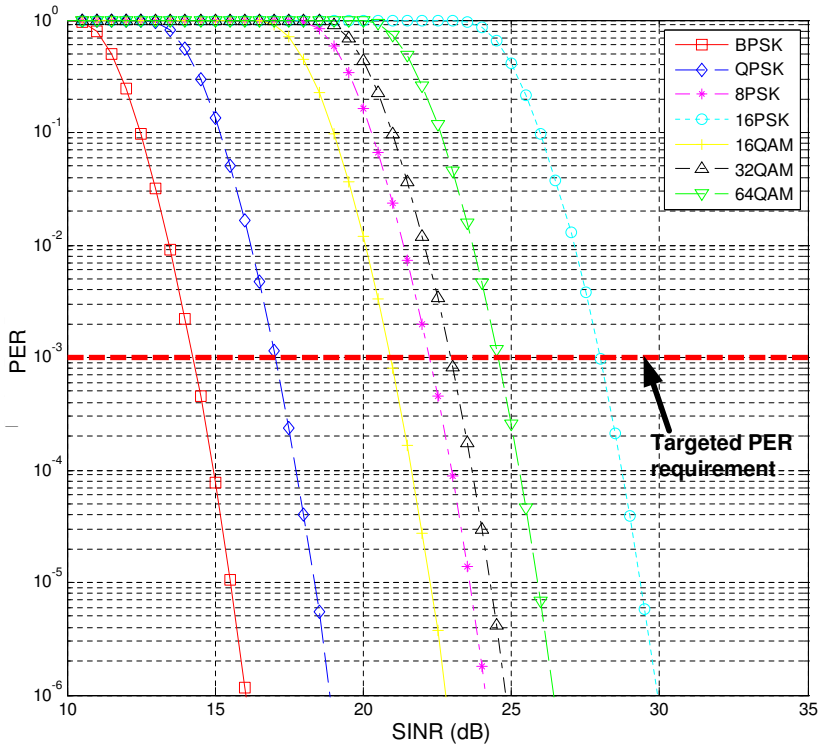

Figure 2.5: The used mapping from the received SINR (dB) to the PER with different combinations of modulation and coding schemes.

avoided and link-level throughput is increased.

The proposed distributed opportunistic proportional fair scheduling algorithm is implemented in each WMR's MAC layer, where it also maintains a set of queues for the incoming packets (the reception) to place in and for the outgoing packets (the transmission) to be removed. The proposed QoS routing algorithm is implemented in the network layer of each WMR where the WMR, if served as the source, sends out the REQ packet for find the satisfactory route.

Two application profiles are considered in this simulation, including VoIP and FTP, each of which is attached with three QoS constraints, *i.e.,* throughput, ETE packet delay and PER. Table 2.2 and Table 2.3 indicate the most important parameters. The simulation parameters are summarized in Table 2.4.

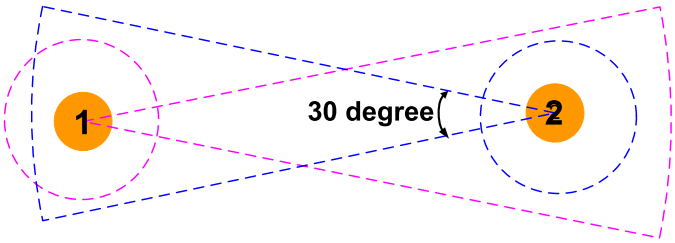

Figure 2.6: An example of the used directional antenna model where the main lobe is 30dB higher than the side lobe. When the scheduling decision is made by the MAC layer of each WMR, the beam of each antenna is switched to point to the corresponding transmitter/receiver so that the amount of co-channel interference could be largely avoided and link level throughput is increased.

Table 2.2: FTP application profiles

Parameter	**Value**
Inter-arrival Time	Poisson distribution with $1/\lambda = 20$ms, variable
File Size	Constant 1 MBytes
Type of Service	Best Effort
Delay Requirement	—
Throughput Requirement	100kbps-2Mbps
PER Requirement	0
GoS Levels	HQoS with average throughput 1Mbps and LQoS with average throughput 200kbps

The performance of the proposed cross-layer solution highly depends on the accurate estimation of multiple parameters in different protocol layers which are required for the QoS routing and distributed scheduling, which include real-time monitored per-link statistics like throughput on link (i, j), or $T^a(i, j)$, queuing and transmission delay on link (i, j), or $D^a(i, j)$, and PER on link (i, j), or $E^a(i, j)$. These statistics are tracked and updated periodically according to the scheduling and routing operational time scales to represent the most recent channel qualities and queue status.

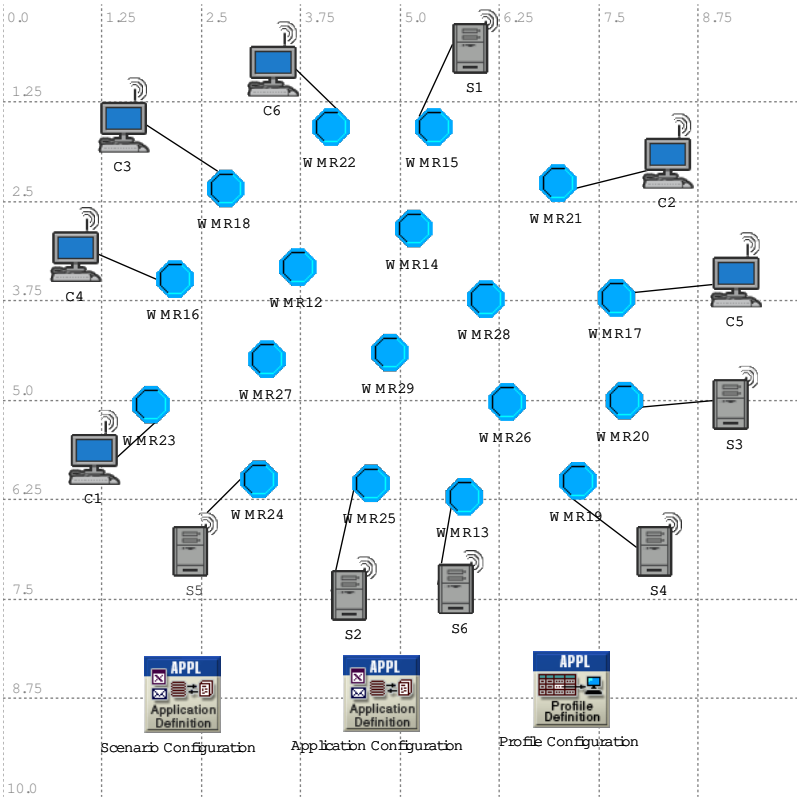

Figure 2.7: An example of the standard scenario used in our OPNET simulation platform. The WMN consists of eighteen WMRs with six client and server pairs to serve as the sources of traffic and the destinations.

2.7.2 Network Performance

The proposed integrated multi-constrained QoS routing (IQoSR) and distributed scheduling (Dist) algorithms are assessed here, compared with the combination of conventional Round Robin scheduler (RR, [77]) and AODV routing protocol. Table 2.5 summarizes these four comparisons, while Figure 2.8 and Figure 2.9 demonstrate the gateway goodput and the average QoS outage probability with different traffic loads, respectively. Gateway goodput is computed as the time average goodput of the gateway node counting in the successful connections but without those QoS is not guaranteed. The average QoS outage probability is the percentage of successful connections with satisfactory QoS experience of all completed connections.

Table 2.3: VoIP application profiles

Parameter	Value
Encoder scheme	G.729A
Voice frame per packet	1
Type of Service	Interactive Voice
Duration	10s-500s
Delay Requirement	100ms-300ms
Throughput Requirement	17kbps-106kbps
PER Requirement	0.1%
GoS Levels	Only HQoS is allowed, not degradable

Figure 2.8 shows that the "Dist OPF" scheduler in our framework can guarantee high gateway goodput even for the small traffic inter-arrival time when the offered network traffic is getting high. On the other hand, the "RR+AODV" scheme provides a much lower goodput compared with all other three schemes since the RR scheduler fails to exploit the multi-user diversity gain of wireless channel (or, channel resources are reserved and pre-allocated as a round robin fashion), and AODV routing protocol creates bottleneck links in the network by transporting traffic through always the route with the minimum hop. "Dist+AODV" and "RR+IQoSR" perform between the lower-bound performance of "RR+AODV" and the upper-bound performance of "Dist+IQoSR", because either they take advantage of wireless channel to provide high throughput or they manage to select the best candidate route to ensure QoS, but unfortunately not both.

The above judgement for the four schemes become even clearer in Figure 2.9 that demonstrates the average QoS outage probability for all completed connections. It can be seen that all four schemes successfully guarantee better QoS if we increase the traffic inter-arrival time, or less traffic load are offered in the network. However, when more traffic is injected into the network without using an efficient admission

Table 2.4: OPNET simulation parameters for network configurations

Parameter	Value
Channel Model	Rayleigh fading model
Path Loss Coefficient	3.5
Directional Antenna Pattern	Side lobe: -25dB, Main lobe: 30°
Adaptive Modulation and Coding	BPSK, QPSK, 8PSK, 16PSK 16QAM, 32QAM, 64QAM
Doppler Frequency	25Hz
System Bandwidth	50MHz
Slot Duration	$80G_0s$
Slots per Frame	100
Frame Duration	8ms
MAC Packet Length	1024 bytes
Number of WMR	5-35, Typical number 18
Number of Client/Server pair	6
Network Area	10 miles by 10 miles
Transmission Range	2 miles
Traffic Patterns	FTP and VoIP
Queue Length	Infinite

Table 2.5: Scheduling and routing performance comparisons

MAC	Routing	Cross-Layer Term
Distributed OPF	IQoSR	Dist+IQoSR
Distributed OPF	AODV	Dist+AODV
Round Robin	IQoSR	RR+IQoSR
Round Robin	AODV	RR+AODV

control scheme, the outage probability always increases due to the severe impacts of the new connections on the QoS of the existing running ones in the network. This proves the importance of using certain admission control scheme in the multi-hop wireless network, and drives the research in Chapter 3.

Figure 2.10 demonstrates the effect of the proposed scheme on the average number of admitted and successful connections in the network as a function of the GoS threshold. As expected, when GoS threshold increases the total number of connections with LQoS requirements decreases to allow more resources for connec-

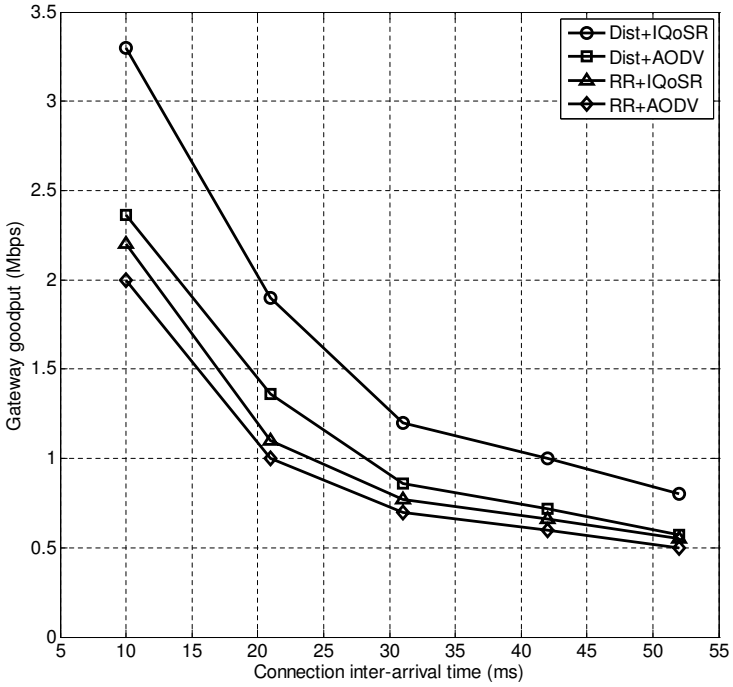

Figure 2.8: Simulation result of the average gateway goodput with respect to (w.r.t.) the different new connection inter-arrival time for different combinations of scheduling and routing algorithms.

tions with HQoS requirements. However, the total number of connections admitted is going down because more stringent QoS is expected. Overall, this figure shows the capability of our multi-level QoS management scheme to successfully maintain a few levels of QoS requirement given the satisfactory GoS threshold G_0.

2.8 Summary

Cross-layer designs to support QoS in wireless mesh networks have attracted much research interests from both academic and industrial communities. Unlike existing works that focus either on global optimization decomposition or barely information delivery among layers, in this chapter, we propose a novel cross-layer framework that includes a QoS routing algorithm in the network layer and a distributed opportunistic proportional fair scheduling algorithm in the MAC layer. We defined a novel

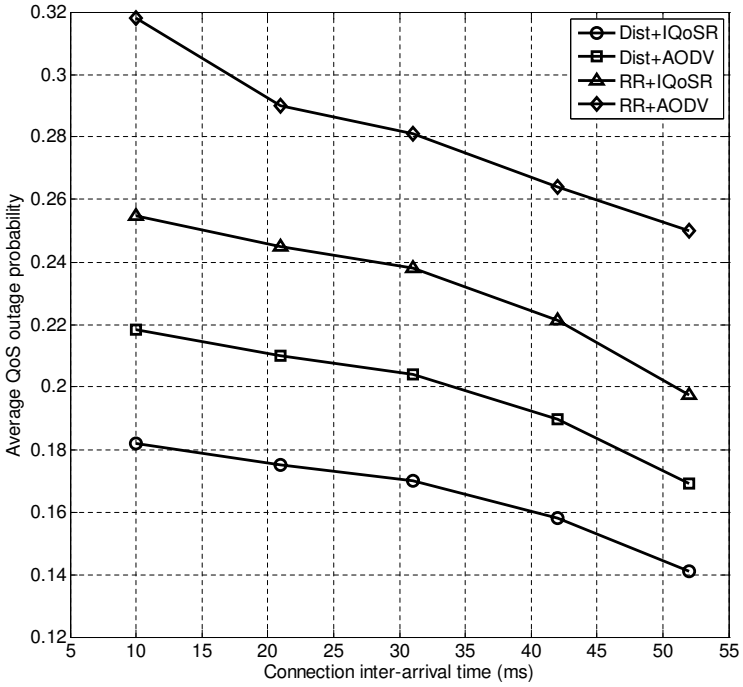

Figure 2.9: Simulation result of the average QoS outage probability w.r.t. the different new connection inter-arrival time for different combinations of scheduling and routing algorithms.

utility function that is requested by a multi-constrained QoS routing algorithm and finally enforced by an efficient distributed opportunistic proportional fair scheduler. Extensive simulation results and analysis shows the success of our framework to combine algorithms and techniques from three different layers and achieve the best overall performance as compared to other schemes.

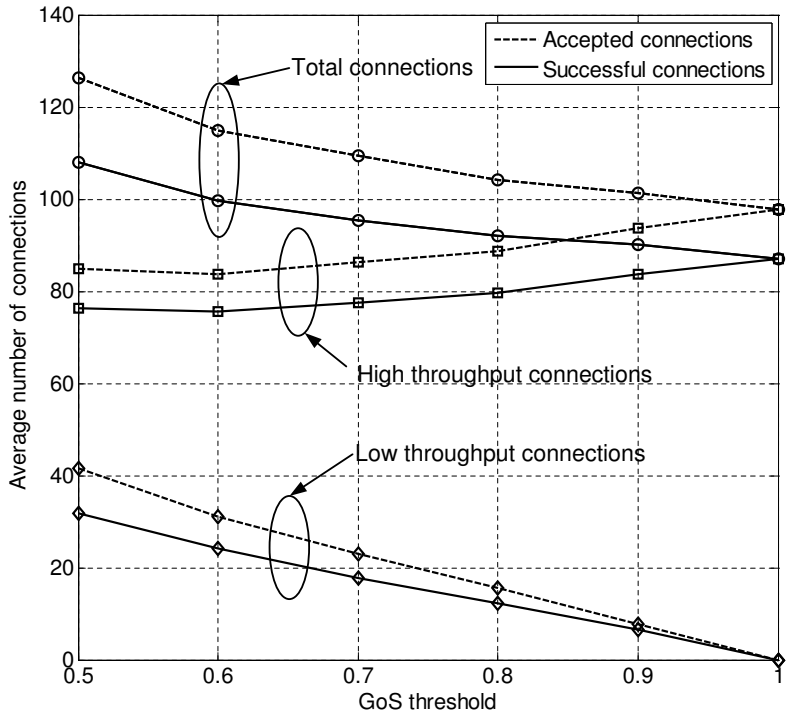

Figure 2.10: Simulation result of the average number of admitted and successful connections in the network as a function of the GoS threshold G_0.

Chapter 3

A Generic Admission Control Methodology for QoS

THIS Chapter is motivated by the research finding in Chapter 2 that one cannot allow arbitrary number of connection admitted to the network otherwise QoS experience cannot be guaranteed. This admission control decision is usually made by knowing the network capacity as *a priori*, however it is well-known to be very difficult to estimate due to the operational characteristics of a variety of communication protocols, dynamic connection behaviors, and their associated multiple QoS requirements. These make the network capacity generally cannot be easily parameterized by a single (or a set of) variable(s).

To address this challenge and facilitate the admission control, a packet network, which may consist of multiple heterogeneous subnetworks, between a pair of source and destination routers in the network is modelled as a "black box". A generic mathematical function is then used to map the multiple input variables to a single output parameter, called the *QoS performance index*. By using the Taylor expansion, we propose a generic AC (GAC) methodology to predict the impact of the potential admission of a new connection on the QoS experience of existing connections. The uniqueness of the proposed methodology is its wide applicabil-

ity to many type of packet networks, wired and/or wireless, independent of the communication protocols or standards in use at the lower protocol layers. Finally, extensive simulations have shown a significant network goodput increase and QoS outage probability decrease if compared with other statistical based methods and conventional network protocols.

3.1 Introduction

With the rapid development of Internet applications and wireless devices, networking has lately experienced unprecedented advances that have been pushing high-speed wired networking into new domains, making mobile and wireless networking much more ubiquitous, and driving the needs for all optical, 3G wireless, and QoS-based packet networks [78–80]. Moreover, the increase in processing power and memory availability of current user devices such as PDAs, game consoles and laptops, give rise to a new wave of bandwidth-hungry and delay-sensitive mobile services and applications that will push the quality-of-service (QoS) demands to their limits or beyond.

To satisfy the QoS of all connections in a packet network, we have identified the following four challenges when designing an efficient network architecture or network protocols.

First, the multi-dimensional QoS requirements of the multimedia traffic. The question of how to support the multimedia traffic, like voice-over-IP (VoIP), interactive video, and broadband data services, is always central for the design of efficient network architecture or protocols like routing and scheduling, which are always associated with different combination of QoS metrics to be enforced by the network. These metrics include but not limited to end-to-end (ETE) throughput, packet delay, packet-error-rate (PER), etc. It is proven NP complete in finding an optimal

route satisfying more than one metric simultaneously [39], and thus challenging to mathematically accurately quantify the overall QoS experience for the completed connections who have finished the service and for the newly arrived connections who will be admitted into the packet network in the near future.

Second, the limited amount of network resources cannot allow arbitrary large number of connections with strict and multiple QoS requirements admitted into the network, which can easily jeopardize ongoing connections' QoS experience in network and make the overall network inefficient and fragile. Furthermore, in wireless networks, due to the co-channel interference and channel fluctuations, the improper admission of the new connections can highly affect the resource availability of adjacent transmissions.

Third, estimating the network capacity (or the amount of available network resources) in a single wireless network is already very challenging. It is well known that the network capacity is one of the key network design parameters for QoS provisions that has different interpretations at different protocol layers and different networks. Nevertheless, it is known difficult to estimate even in a single wireless network [32]; especially for the support of multimedia traffic, connections with different QoS requirements will consume different amount of network resources, making the network capacity highly dynamic and the estimation of the remaining resources extremely difficult.

Finally, estimating the ETE network capacity is even more challenging. For connections going through several subnetworks from an ingress node of a packet network to an egress node, the heterogeneous features of network protocols in use in the individual subnetwork make the modelling the ETE QoS experience and the estimation of ETE network resource availability extremely difficult.

All these challenges drive us to redesign a new generic AC methodology

(GAC) to enforce the QoS control in packet networks where we make the following three contributions, followed by the description of the details of each contribution.

First is the proposal of aggregating multiple QoS metrics into one single performance index, called the *QoS performance index*. By using this index, the degree of QoS stratification a connection has received for multi-dimensional QoS requirements is quantified to a scalar, with the range from 0 to infinity, whereas the satisfactory bound for this index belongs to the range $[0, 1]$.

Second is the modelling of a packet network between an ingress node and an egress node as a "black box" for the amount of available network resources to share. The ingress node aggregates traffic as the input to the network, and an egress node serves as an intended destination. The input parameters to the black box (*i.e.,* network status, traffic pattern, QoS requirements) are mapped to a single output parameter (or the defined *QoS performance index*), where we adopt a runtime analysis for such black box using only the inputs and the output, without specifically knowing the detailed operational contexts of the communication protocols in different subnetworks as *a priori*. In other words, we hide the heterogeneous operational features of network protocols in several subnetworks and treat the packet network homogenously.

Third is the proposal of using Taylor approximation to estimate the QoS impact of admitting the new connection. To facilitate the AC decision, we characterize the potential new connection admission as the input change to the black box, and the change of output is estimated by Taylor approximation in a closed-form. The only unknown variables are partial derivatives which can be easily obtained from the shape of the curve produced by learning the packet network. Later, the AC is performed by comparing the new projected QoS performance index as the output with its satisfactory value 1.

Finally, we show that the proposed GAC methodology has wide applicability to any packet networks, wired and/or wireless. It is completely transparent to the lower protocol layers (*e.g.,* physical (PHY), medium access control (MAC) and network layers), *i.e.,* irrespectively of any advanced technology to use, the proposed algorithm can still efficiently collect statistics and make them applicable to the AC algorithm. However, this does not mean at all we are developing a layered approach, but later they key design element, QoS performance index, successfully integrate all lower layer parameters together, and thus as a cross-layer approach. We also discuss various important feasibility issues like the impact of large connection throughput requirement, statistics feedback delay, and statistics collection time.

The rest of this Chaper is organized as follows. After introducing the related work in Section 3.2, the system model is described in Section 3.3 and the QoS performance index is introduced in Section 3.4. Followed by the mathematical presentation of our system model in Section 3.5, Section 3.6 presents the methodology for estimating the impacts on QoS performance index by connection admission. Section 3.7 describes the steps of performing the proposed GAC methodology. Next, numerical results and detailed analysis are given in Section 3.8. Finally, Section 3.9 presents the discussions on the applicability and the feasibility issues, and a summary is drawn in Section 3.10.

3.2 Related Work

The great deal of research attention increases significantly recently for AC algorithm in a variety of packet networks [20–25], due to the growing popularity of multimedia applications (such as voice, video, and broadband data) and the central role AC scheme plays in QoS provisioning (in terms of the connection blocking probabilities, packet delay, and throughput etc.). The challenges come from the inefficiency of

lower layer protocols in use to provide satisfactory network resource management and QoS supports, not only for the individual per-node performance but also for the ETE network performance. Arriving new connections are granted/denied access to the network by the AC based on predefined criteria, taking the network loading conditions and resource availability into consideration. On the other hand, heterogeneous nature of multi-hop wireless networks require a degree of transparency of AC scheme to lower layer protocols, such that whatever advanced technologies to use, the AC scheme can always efficiently control the connection admission by estimating network resource availability.

Much research work has been done on AC algorithms in mobile ad hoc wireless networks [81], as well as in wireless mesh networks (WMNs). [82] proposed an algorithm to support rate and delay requirements, but it assumed no channel fading and co-channel interferences among wireless links, and uses a tree-structure [83] MAC scheduling. The distributed AC algorithm is proposed in [84] for each node to estimate the used bandwidth of all neighbours. A set of papers in [85–87] studied the design of optimal joint admission control and routing that can maximize the overall revenue while guaranteeing the QoS for multiple classes in mesh networks has not been addressed, where the AC problem is formulated as a semi-Markov decision process (SMDP), then solved by a linear programming (LP) based algorithm. Unfortunately, the notion of QoS is only denoted as the SNR constraint. In [88], an adaptive admission control (AAC) protocol estimates the resource availability in contention-based WLAN MAC layer to control QoS, however neither provide a degree of transparency to lower protocol layers nor the guarantee of multi-dimensional QoS requirements; followed by a similar approach in [89] and [90]. In [91], a joint centralized scheduling and time slot allocation based AC algorithm is proposed for WiMAX networks, which allowed to admit a connection if extra unused slots are sufficient to satisfy bandwidth requirement. The integrated framework of routing and

admission control for IEEE 802.16 distributed mesh networks was studied in [92]. It estimated available bandwidth in a token bucket to perform AC with minimum time slot requirement for each connection, and it used shortest-widest efficient bandwidth metric for route discovery. [93] is much related to our previous work in [94] that makes admission decision by estimating the achievable capacity between any pair of ingress and egress nodes with only packet loss constraint, assuming traffics arrive according to Gaussian distribution. Works [95] studied extensively on the integrated QoS routing protocol and the actual interface between the scheduling and routing schemes, to provide optimum routes that guarantee multiple QoS constraints.

In a summary, the existing AC algorithms in the literature do assume certain operational characteristics of the underlying communication protocols in use and AC schemes are developed for specific network settings. Furthermore, none of these have accurately estimate the admission impact in terms of QoS experience, nor the wide applicability to many other packet network; and these become the central of our Chaper and the research path in general.

3.3 System Model

Consider a packet network that comprises a set of subnetworks, as shown in Figure 3.1, each of which is further composed of a finite number of nodes. The individual subnetwork can be an access network, backhaul network, backbone network, etc. For the packet network, traffic comes from application layer and reaches the ingress node, and intends to be transferred to the egress node (potentially) across several subnetworks, as shown in Figure 3.1. Connection $q \in \mathcal{Q}$ (where $\mathcal{Q}$ denotes the connection set currently being served) is attached with a set of performance requirements, namely: ETE packet delay D_q^r, throughput T_q^r, and PER E_q^r, where

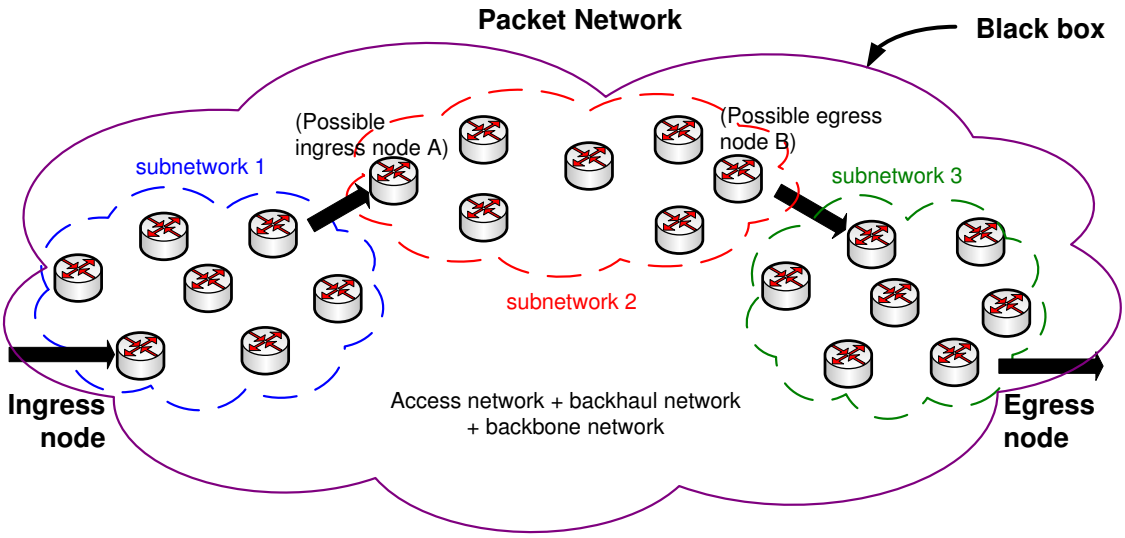

Figure 3.1: An illustrative example of a packet network consisting of three separate subnetworks. The packet network is modelled as a black box between a pair of ingress and egress nodes. Traffic starts from the ingress node, and intends to communicate with the egress node. This figure also shows that the black box model can be applied to one single subnetworks where possible ingress and egress node are denoted as A and B respectively in subnetwork 2.

superscript r denotes the *required* value.

Within the packet network, concurrent connections share limited amount of network resources, which include, but not limited to, buffer, bandwidth, transmission power, time slot, etc. To overcome the difficulties of dealing with a variety of operational communication protocols in different subnetworks, we treat the packet network between a pair of ingress and egress nodes as a "black box" as shown in Figure 3.1, where the detailed operational contexts for protocols beneath the application layer are not transparent to the ingress and egress nodes' application layers. We opt to go around the issue of modelling these communication protocols in some closed-form expressions, but we adopt a runtime analysis such that connections are probed (or monitored) by the ingress and egress nodes for the satisfactory QoS experience. In other words, the benefit a packet network can provide to each served connection across several heterogeneous subnetworks is constantly monitored at the

egress node and informs back to the ingress node through the feedback loop, so that the availability of the current packet network to support certain QoS is known.

The Inputs

Typical parameters include, but not limited to, the number of ongoing connections, multiple QoS requirements, etc.

The Output

One single output is used at the egress node to reflect the degree of QoS satisfaction and the service quality of the black box, which is the defined *QoS performance index*.

It is worth noting that our black model does not constrain in the considered packet network, but also applies to any single subnetwork, as shown in Figure 3.1 where a pair of possible ingress and egress nodes in subnetwork 2 is modelled in the same way.

3.4 The QoS Performance Index

Per-connection QoS Performance Index

The key for admission control is to identify how the new connection, if accepted, will experience and impact the QoS of existing connections in the considered packet network. To this end, a unique, time-varying QoS utility function based on the *QoS outage ratio*, dented as R, is defined for each QoS parameter, *i.e.,* $R_q^D(t)$ for ETE packet delay, $R_q^T(t)$ for throughput, and $R_q^E(t)$ for PER, $\forall q \in \mathcal{Q}$. These ratios are defined between the *attained* real-time parameter measurement denoted by superscript a, and the *required* (and fixed) QoS value denoted by superscript r,

i.e.,

$$R_q^D(t) = \frac{D_q^a(t)}{D_q^r}, \quad R_q^T(t) = \frac{T_q^r}{T_q^a(t)}, \quad R_q^E(t) = \frac{E_q^a(t)}{E_q^r}, \quad \forall q \in \mathcal{Q}, \forall t. \tag{3.1}$$

The advantage of using a set of QoS outage ratios, but not exploring the potential interrelations of each pair of parameters is mainly because of the nature of heterogeneous communication protocols in use in different network settings, which ultimately contribute to the relationship modelling of these parameters. However, our method uses relative degree of QoS satisfaction in a normalized form (as implied in the ratio), and easy extendable to other QoS parameters if not considered in this chapter.

Next, we briefly introduce the way to obtain the required values of these three performance metrics. Delay information is collected by stamping each packet of connection q when it arrives at the ingress node at time $t_q^{\text{send}}(j)$ and when it reaches the egress node at time $t_q^{\text{recv}}(j)$, and calculated as the difference; throughput is monitored by the amount of receiving packets at the egress node within an predetermined time interval t_{interval}; and finally PER information is collected by tracking the probability a packet to be erroneous when it is received by the egress node, and calculated as the ratio. Therefore, we have:

$$D_q^a(t) = \frac{1}{\mathrm{N}_e + \mathrm{N}_c} \sum_{j=1}^{\mathrm{N}_e+\mathrm{N}_c} \Big[t_q^{\text{recv}}(j) - t_q^{\text{send}}(j) \Big], \tag{3.2}$$

$$T_q^a(t) = \frac{l(\mathrm{N}_e + \mathrm{N}_c)}{t_{\text{interval}}}, \tag{3.3}$$

$$E_q^a(t) = \frac{\mathrm{N}_e}{\mathrm{N}_e + \mathrm{N}_c}, \tag{3.4}$$

where l denotes the packet length, $\mathrm{N}_e, \mathrm{N}_c$ denote the number of received erroneous and correct packets of connection q respectively.

Due to the multi-dimensional nature of the QoS requirements, even though

we use a set of QoS outage ratios $\left(R_q^D(t), R_q^T(t), R_q^E(t)\right)$ to denote the degree of satisfaction for each performance metric, it is also very difficult to judge the *overall* QoS experience any connection has received. Meanwhile, the degree of overall QoS satisfaction, but not the individual QoS performance metric, is most important for the end-user or service provider. Therefore, we are in need of a simple, but representative and quantitative scalar to uniquely denote the level of QoS satisfaction. These are primarily why we introduce a novel concept of per-connection *QoS performance index* $\theta_q(t), \forall q \in \mathcal{Q}$, which is defined as,

$$\theta_q(t) \triangleq \max\left(R_q^D(t), R_q^T(t), R_q^E(t)\right), \quad \forall q \in \mathcal{Q}, \forall t \tag{3.5}$$

where max operator combines multiple QoS parameters into one single performance index. Therefore, it follows immediately from the definition of QoS performance index that:

Lemma 3.4.1. *For any connection $q \in \mathcal{Q}$, its multi-dimensional QoS requirements are simultaneously satisfied within a packet network if and only if $\theta_q(t) \in [0, 1]$ at any time t.*

The uniqueness and main essence of using this QoS performance index are mainly two folds. First, this index hide the heterogeneous nature of network protocols to use in several subnetworks but only use attained and required QoS values to calculate a scalar, and thus t is completely transparent to lower protocol layers and technologies (*e.g.,* routing, scheduling, advanced PHY layer modulation, coding and antenna, etc.) in use. Second, although simple, but it always represents and reflects the degree of resource availability for maintaining certain QoS performance in real-time.

Network-wide QoS Performance Index

The per-connection QoS performance index $\theta_q(t), \forall t$ is constantly monitored by the egress node when any connection q is being serviced in the packet network. Furthermore, it will pass to the corresponding ingress node through a feedback loop, and thus we would expect the ingress node to consistently know the most recent real-time QoS performance index $\theta_q(t), \forall t$. Then, we define use a network-wide QoS Performance Index (or, simply QoS performance index used later in this chapter) $\mathrm{I}(t), \forall t$ to indicate the ETE network-wide QoS satisfaction that combines the most recent received QoS performance index of the completed connection. This parameter evolves over time through exponential smoothing as:

$$\mathrm{I}(t) = \gamma\theta_q(t) + (1-\gamma)\mathrm{I}(t-1), \forall t, \tag{3.6}$$

where $\gamma \in (0,1)$ is the weight factor and the initial value $\mathrm{I}(0) = 0$.

Lemma 3.4.2. *For all ongoing connections $q \in \mathcal{Q}$, their multi-dimensional QoS requirements are simultaneously satisfied within a packet network if and only if* $\mathrm{I}(t) \in [0,1]$ *at any time t.*

3.5 The Mathematical Representation of the Packet Network

After properly defining the inputs (*e.g.,* the number of current served connections and QoS requirements) and the output (the QoS performance index) of our black model, the key step is to use some mathematical functions to appropriate the resource space of the packet network, or to estimate the black box. Without loss of generality, we use a generic mathematical function f to represent the operational characteristics of the packet network, which maps M_b-dimensional input vector $\underline{x}(t) = (x^1(t), x^2(t), ..., x^{M_b}(t)) \in \mathbb{R}^{M_b}$ to a scalar output $y(t) \in \mathbb{R}$, which denotes

the degree of resource occupancy within the packet network (*i.e.,* the black box). In other words, we use the mapping

$$f : \mathbb{R}^{M_b} \rightarrow \mathbb{R}, \text{ or } y(t) = f(\underline{x}(t)) \tag{3.7}$$

to represent a (M_b+1)-dimensional space, where M_b input variables $\underline{x}(t)$ capture the system status at any given time t, like the number of connections and required QoS requirements. Furthermore, as we use the QoS performance index in the output, we have:

$$y(t) \triangleq \mathrm{I}(t). \tag{3.8}$$

Input Change by the Connection Admission

Next, the new connection admission is characterized as an input change $\Delta\underline{x}(t) = (\Delta x^1(t), \Delta x^2(t), ..., \Delta x^{M_b}(t))$ for each dimension of the input variables into the black box, which will result in a change of output to:

$$\tilde{y}(t) = f(\underline{x}(t) + \Delta\underline{x}(t)). \tag{3.9}$$

It is worth noting that the mapping f is usually quite complicated, which generally cannot be expressed in a closed-form expression. However, we can actually approximate this mapping by real-time measurements. When the network is just initialized and empty, the connections can be freely admitted and flow through the network to the egress node, and this is when the construction of the mapping f starts and repeats the following five steps as:

1. The admitted connections will cause the ingress node to update the M_b-dimensional input vector $\underline{x}(t)$. For instance, if the inputs are the number of served connections $N(t)$ and total served throughput $T(t)$, then the new

connection admission would increase the number of ongoing connections increased by 1, and the total served throughput is increased by the corresponding throughput requirement of the new connection.

2. For any running connection $q \in \mathcal{Q}$, the egress node constantly monitors its received QoS performance index calculated in Eqn. (3.5) and further exponentially smoothed by Eqn. (3.6) to obtain an updated network-wide ETE QoS performance index $\mathrm{I}(t)$.

3. ETE QoS performance index $\mathrm{I}(t), \forall t$ is passed to the ingress node through a feedback loop.

4. When the ingress node receives the new statistics $\mathrm{I}(t)$ and knows certain connection has just completed, it updates the input vector $\underline{x}(t)$ again. For instance, the new connection completion would cause the number of ongoing connections decreased by 1, and the total served throughput decreased by the throughput requirement of the finishing connection.

5. Go back to Step 2 and continue.

Therefore, iteratively, if enough connection performance is monitored and statistics are recorded, the shape of the curve produced by the mapping f could be approximated by a set of statistics $(\underline{x}(t), \mathrm{I}(t))$. Next, we need to estimate the potential admission impact on QoS performance index. It is not difficult to observe that we could use the Taylor expansion of the right hand side of (3.9) to approximate its left hand side. However, the accurate expansion requires infinite orders of Taylor series, which may not be needed in the engineering problems; we also notice that the first order derivatives usually represent the long-term average, and second order derivatives usually represent the fast change, or the variance. Then, it is natural to

take only the first and second order partial derivatives, as:

$$\begin{aligned}\tilde{y}(t) &= f(\underline{x}(t)+\Delta\underline{x}(t)) \approx f(\underline{x}(t)) + \sum_{i=1}^{M_b} \frac{\partial f}{\partial x^i(t)}\Delta x^i(t) \\ &+ \frac{1}{2}\left(\sum_{i=1}^{M_b} \frac{\partial^2 f}{\partial x^i(t)^2}\left(\Delta x^i(t)\right)^2 + \sum_{i=1}^{M_b}\sum_{j\neq i} \frac{\partial^2 f}{\partial x^j(t)\partial x^i(t)}\Delta x^i(t)\Delta x^j(t)\right).\end{aligned} \tag{3.10}$$

It is worth noting that we do not have explicit assumptions of the convexity/concavity of the mapping f, but this mapping is learned by system measurements in real-time. Furthermore, Taylor expansion does not at all require any convexity/concavity feature of the function, but uses the partial derivatives to approximate the value of it. Finally, in the above equation, as introduced later, $y(t), \Delta x(t)$ are also known, whereas the only unknown variable is $\tilde{y}(t)$.

3.6 Impacts on the QoS Performance Index by Admission

This section aims to estimate the impacts of the new connection admission on the QoS performance index, which is the central part to perform the accurate admission control.

3.6.1 The Scalar Input

For illustration purposes, we start by using only a scalar input, *i.e.,* $M_b = 1$, as an illustrative example to demonstrate the steps of estimating this impact, *i.e.,* $\underline{x}(t) \triangleq N(t) \in \mathbb{R}$, as shown in Figure 3.2. In other words, within the black box, the number of ongoing connections $N(t)$ is used; and we track this statistics from time to time. It is worth noting that the number of currently served connections cannot accurately capture of the overall operational status of the packet network,

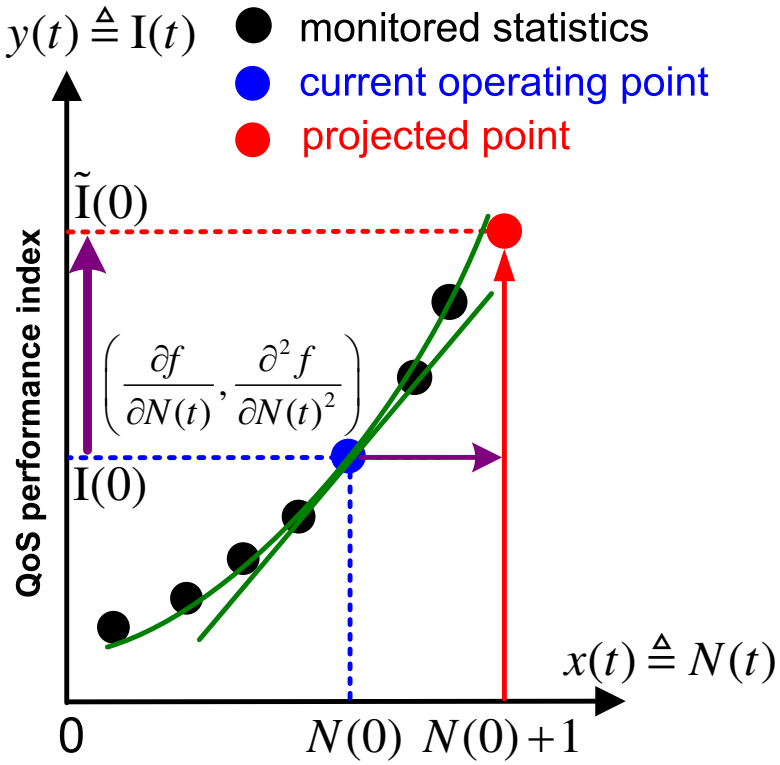

Figure 3.2: An illustrative example for admission estimation on the shape of curve produced by the mapping f, where scalar input $N(t)$ is considered, *i.e.*, $M_b = 1$.

since connections demand different amount of network resources with different QoS requirements; and we shall move to a more realistic case with $M_b = 2$ later. For now, the mapping f becomes,

$$\mathrm{I}(t) = f\Big(N(t)\Big). \tag{3.11}$$

We denote the current packet network status by the number of currently served connections $N(0)$, if we assume the current time is $t = 0$. At this time, we suppose a new connection q with a set of performance requirements denoted as $(D_q^r, T_q^r, E_q^r, L_q^r)$ arrives at the ingress node for admission.

As discussed earlier, the potential new connection admission with require-

ments $(D_q^r, T_q^r, E_q^r, L_q^r)$ would result in an input change of:

$$\Delta x(0) = \Delta N(0) = 1, \tag{3.12}$$

into the black box. As shown in Figure 3.3, once this input change $\Delta x(0)$ is incurred, it will result in an output change to,

$$\widetilde{\mathrm{I}}(0) = f\Big(N(0) + 1\Big). \tag{3.13}$$

With the reference of Taylor expansion in (3.10), we rewrite (3.13) as,

$$\widetilde{\mathrm{I}}(0) = \mathrm{I}(0) + \frac{\partial f}{\partial N(t)} + \frac{1}{2}\frac{\partial^2 f}{\partial N(t)^2}, \tag{3.14}$$

where both first and second order partial derivatives taken at the current operating point $x(0) = N(0)$, as shown in Figure 3.2. Since, $N(t)$ is discrete, we approximate its "derivatives" by the slopes of adjacent network measurements. For example, assume that at least two adjacent measurements $(N(0)|_1, \mathrm{I}(0)|_1), (N(0)|_2, \mathrm{I}(0)|_2)$ around the current state $N(0)$ are obtained; then, the first order partial derivative is computed as the average of two adjacent slopes of measurements,

$$\left.\frac{\partial f}{\partial N(t)}\right|_1 \approx \frac{\mathrm{I}(0)^1 - \mathrm{I}(0)}{N(0)|_1 - N(0)}, \quad \left.\frac{\partial f}{\partial N(t)}\right|_2 \approx \frac{\mathrm{I}(0) - \mathrm{I}(0)|_2}{N(0) - N(0)|_2}, \tag{3.15}$$

and the second order partial derivative is computed as the change of the above two slopes:

$$\frac{\partial^2 f}{\partial N(t)^2} \approx \frac{\frac{\partial f}{\partial N(t)}\big|_1 - \frac{\partial f}{\partial N(t)}\big|_2}{(N(0)|_1 - N(0)) - (N(0) - N(0)|_2)}. \tag{3.16}$$

It is interesting to observe that the potential connection admission actually updates the output of the mapping f from $\mathrm{I}(0)$ to $\widetilde{\mathrm{I}}(0)$ (and this would be the admission impact we are aiming to estimate), when the network status changes from $x(0) = N(0)$ to $x(0) + \Delta x(0) = N(0) + 1$. In other words, if this connection is physically accepted, the new operating point for the packet network would be at

state $(N(0)+1)$.

3.6.2 The 2-D Inputs

As discussed at the beginning of this section, scalar input variable the number of currently served connections $N(t)$ cannot capture of the overall operational statistics of the packet network. To this end, we now move to a more realistic case where two-dimensional input variables are extensively considered in the following black box model, where $\underline{x}(t) = (x^1(t), x^2(t)) \triangleq (N(t), T(t)) \in \mathbb{R}^2$, as shown in Figure 3.3. In other words, within the black box, the number of ongoing connections $N(t)$ and the total served throughput $T(t)$ are used; and we track these two statistics from time to time. Now, the mapping f becomes,

$$\mathrm{I}(t) = f\left(N(t), T(t)\right). \tag{3.17}$$

As discussed earlier, the potential new connection admission with requirements $(D_q^r, T_q^r, E_q^r, L_q^r)$ would result in an input change of:

$$\Delta\underline{x}(0) = (\Delta N(0), \Delta T(0)) = (1, T_q^r), \tag{3.18}$$

into the black box. As shown in Figure 3.3, once this input change $\Delta\underline{x}(0)$ is incurred, it will result in an output change to,

$$\widetilde{\mathrm{I}}(0) = f\left(N(0)+1, T(0)+T_q^r\right). \tag{3.19}$$

With the reference of Taylor expansion in (3.10), we rewrite (3.19) as,

$$\widetilde{\mathrm{I}}(0) = \mathrm{I}(0) + \frac{\partial f}{\partial N(t)} + T_q^r\frac{\partial f}{\partial T(t)} + \frac{1}{2}\frac{\partial^2 f}{\partial N(t)^2} + \frac{(T_q^r)^2}{2}\frac{\partial^2 f}{\partial T(t)^2} + T_q^r\frac{\partial^2 f}{\partial N(t)T(t)}, \tag{3.20}$$

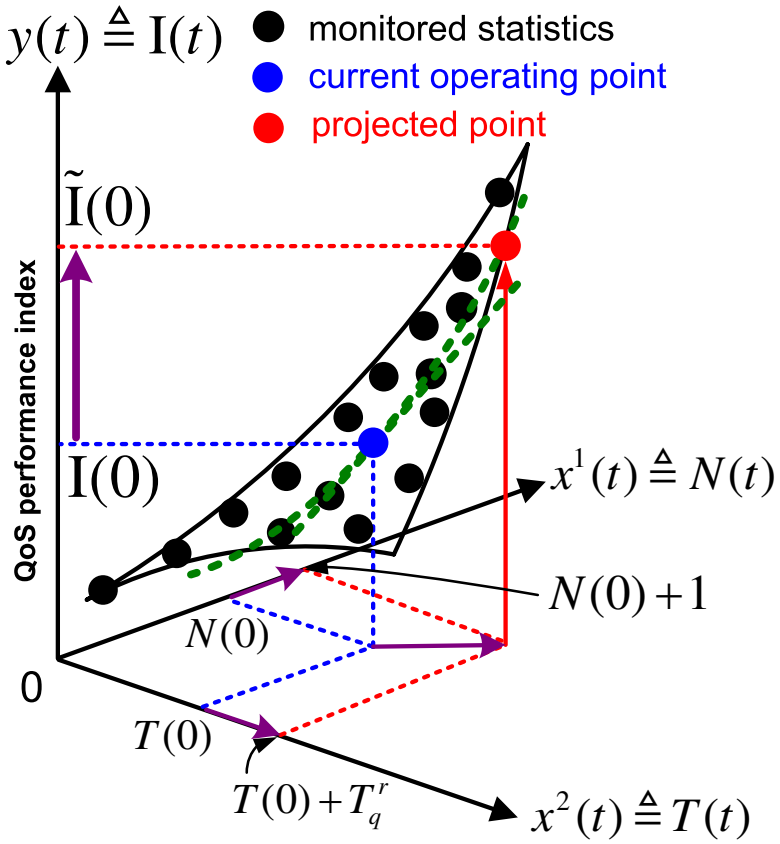

Figure 3.3: An illustrative example for admission estimation on the shape of curve produced by the mapping f, where two-dimensional inputs are considered, *i.e.*, $M_b = 2$.

where partial derivatives taken at the initial state $\underline{x}(0) = (N(0), T(0))$, as shown in Figure 3.3.

In order to get the mixed derivative $\frac{\partial^2 f}{\partial N(t) \partial T(t)}$, it is worth noting that the chosen two input variables are temporarily correlated, since the total served throughput is a function of the number of ongoing connections, as:

$$T(0) = \sum_{q=1}^{N(0)} T_q^r. \tag{3.21}$$

Then, by using the limit definition of the first-order partial derivative, we have,

$$\begin{aligned}\left.\frac{\partial T(t)}{\partial N(t)}\right|_{\underline{x}(0)} &= \lim_{\Delta N(0)\to 1}\frac{\sum_{q=1}^{N(0)+\Delta N(0)} T_q^r - \sum_{q=1}^{N(0)} T_q^r}{\Delta N(0)} \\ &= \lim_{\Delta N(0)\to 1}\frac{\sum_{q=N(0)+1}^{N(0)+\Delta N(0)} T_q^r}{\Delta N(0)} = T_q^r. \end{aligned} \tag{3.22}$$

Therefore,

$$\left.\frac{\partial^2 f}{\partial N(t)T(t)}\right|_{\underline{x}(0)} = \frac{\partial^2 f}{\partial T(t)^2}\left.\frac{\partial T(t)}{\partial N(t)}\right|_{\underline{x}(0)} = T_q^r\left.\frac{\partial^2 f}{\partial T(t)^2}\right|_{\underline{x}(0)}. \tag{3.23}$$

Substituting (3.23) into (3.20) yields,

$$\widetilde{\mathrm{I}}(0) = \mathrm{I}(0) + \frac{\partial f}{\partial N(t)} + T_q^r\frac{\partial f}{\partial T(t)} + \frac{1}{2}\frac{\partial^2 f}{\partial N(t)^2} + \frac{3}{2}(T_q^r)^2\frac{\partial^2 f}{\partial T(t)^2}. \tag{3.24}$$

It is interesting to observe that the potential connection admission actually updates the output of the mapping f from $\mathrm{I}(0)$ to $\widetilde{\mathrm{I}}(0)$ (and this would be the admission impact we are aiming to estimate), when the network status changes from $\underline{x}(0) = (N(0), T(0))$ to $(\underline{x}(0) + \Delta\underline{x}(0)) = (N(0)+1, T(0)+T_q^r)$. In other words, if this connection is physically accepted, the packet network would operate at state $(\underline{x}(0) + \Delta\underline{x}(0))$.

For some network scenarios, if the shape of the curve produced by the mapping f is smooth enough around current operating point $\underline{x}(0) = (N(0), T(0))$ so that the second order derivatives are negligible and the first order statistics are sufficient, we simplify (3.24) as:

$$\widetilde{\mathrm{I}}(0) = \mathrm{I}(0) + \frac{\partial f}{\partial N(t)} + T_q^r\frac{\partial f}{\partial T(t)}, \tag{3.25}$$

where partial derivatives are taken at state $\underline{x}(0) = (N(0), T(0))$.

To conclude this section, it is worth noting that although only two-dimensional input variables are demonstrated to show the steps of estimating the

impacts of admitting the new connection, it is easy to observe its applicability for M_b-dimensional inputs in general.

3.6.3 The Complexity

It is worth to highlight that the computation complexity to estimate the impacts of the new connection on QoS performance index is relatively very small. The first issue is the amount of overhead incurred by the feedback mechanism to pass a single statistics $\mathcal{T}$, since other statistics like $N(t), T(t)$ could be monitored at the ingress node side. We believe that this control overhead is relatively very small, and could be done by higher layer protocols like TCP acknowledgement packet; otherwise if TCP control is not available, we need to artificially send back a very small-sized packet.

Once the shape of the curve produced by the mapping f is approximated by real-time measurements and the current operating point is known, we could derive the first and second order statistics accordingly before plugging into (3.24). The only difficulty of deriving comes from the mixed derivative of f, *i.e.,* the entries off the main diagonal in the Hessian matrix; however due to the nice internal structure of $N(t)$ and $T(t)$, it can be approximated by (3.23). Therefore, we argue that the computation complexity of the proposed estimation method is small.

3.7 Admission Algorithm

Our proposed generic AC (GAC) methodology for QoS control is initialized when the new connection q arrives at an ingress node of the packet network with multiple QoS constraints, intended to communicate with the egress node (the connections may potentially go through several heterogeneous subnetworks). Without loss of generality, suppose the arrival time is $t = 0$. The following steps summarize and

describe the algorithm.

1. **Statistics Collection:** the ingress node consistently keeps track of two statistics at any time t within the considered black box, namely: (a) the number of ongoing connections $N(t)$, and (b) the total served throughout $T(t)$. The egress node constantly monitors the received QoS performance index $\theta_q(t), \forall q \in \mathcal{Q}, \forall t$ of all ongoing connections; and this index is exponentially smoothed by (3.6) to produce $\mathrm{I}(t)$ and informs back to the ingress node. Therefore, the ingress node is aware of pairs of statistics tuple $(N(t), T(t), \mathrm{I}(t))$.

2. **Derivatives Derivation:** once pairs of statistics tuple $(N(t), T(t), \mathrm{I}(t))$ are collected by the ingress node, we use them to approximate the shape of curve produced by the mapping $f : \mathrm{I}(t) = f(N(t), T(t))$, *i.e.,* the mapping f is not modelled by any closed-form expression, but learned through the real-time measurements. Then, partial derivatives around the current operating point are derived.

3. **Admission Control:** the impacts of admitting the new connection on the QoS performance index is characterized and approximated by Taylor expansion while taking both the first and second partial derivatives in (3.24), or only the first order derivatives in (3.25), as inputs. Then, the new QoS performance index $\widetilde{\mathrm{I}}(0)$ is calculated. Next step is to verify if there is enough network resources within the packet network available for the new connection, as:

$$\begin{cases} \text{Admission}, & \text{if } \widetilde{\mathrm{I}}(0) \leq 1 \\ \text{Rejection}, & \text{otherwise.} \end{cases} \tag{3.26}$$

where Lemma 3.4.1 has to be satisfied if the new connection is admitted to the network.

Table 3.1: MATLAB simulation parameters for network configurations

Parameter	Value
Channel Model	Rayleigh fading model
Path Loss Coefficient	3.5
Directional Antenna Pattern	Side lobe: -25dB, Main lobe: 30°
Adaptive Modulation and Coding	BPSK, QPSK, 16QAM, 64QAM
Doppler Frequency	25Hz
System Bandwidth	50MHz
Slot Duration	$80\mu s$
Slots per Frame	100
Frame Duration	8ms
MAC Packet Length	1024 bytes
Number of WMR	5-35, Typical number 15
Network Area	3 mile × 3 mile square
Transmission Range	1.25 mile
Traffic Patterns	FTP, VoIP, and Video
Queue Length	Infinite

3.8 Simulation Results

We developed a cross-layer MATLAB simulator to assess the proposed GAC algorithm. Connections are generated with Poisson distribution, and three QoS requirements, ETE packet delay, throughput, and PER, are attached. Wireless mesh networks (WMNs) are used as an evaluation platform where the integrated QoS scheduling and routing protocol (IQoSR, [95]) is used in network and MAC layers to provide sub-optimal solution for QoS. Rayleigh fading channel model [76], adaptive modulation and coding scheme, and directional antennas are used in PHY Layer to improve the channel capacity and the frequency reuse efficiency and to reduce the interference to adjacent concurrent transmissions. The simulation parameters are summarized in Table 3.1.

It is important to point out that the methodology is not constrained in WMNs, but rather has wide applicability in other wireless and wired IP networks as discussed in Section 3.9.

3.8.1 An Example: A Five Node WMN

We first assess our GAC methodology in a simple five-node WMN setting as shown in Figure 3.4, where node 1 serves as the source (ingress node) to generate connections and node 5 serves as the gateway (egress node) as the intended receiver. Connections are admitted into one of the three disjoint routes as shown. Table 3.2 summarizes the simulation results while varying two methods to estimate the impacts of admitting the new connection on QoS performance index, namely: to use only first order partial derivatives, and to use both first and second order partial derivatives. It is interesting to observe that second order statistics successfully improve the volume of maximum supported throughput and connection number by 13% and decrease the prediction error, QoS outage, and blocking probabilities. This is because the accurate prediction is achieved with the help of higher order statistics that shows the finer horizon of the shape of the curve produced by the mapping f, especially when the packet network operates around its capacity where one single connection admission with large throughput requirement may jeopardize all existing connections' QoS. In other words, second order statistics aid to admit the most *appropriate* connection (in term of throughput requirement) with the knowledge of the satisfactory QoS performance index range $\mathrm{I}(t) \in [0, 1], \forall t$, while maintaining QoS satisfactions to all ongoing connections.

3.8.2 The Overall Network Performance

The proposed algorithm, referred as "IQoSR+GAC", is compared with the existing work "IQoSR" that does not include the proposed GAC scheme, and also compared with the statistical admission control (SAC) algorithm in [93], referred as "IQoSR+SAC". The reason of chosing "SCAC" primarly because it is a statistics collection based admission control algorithm estimating the network capacity while

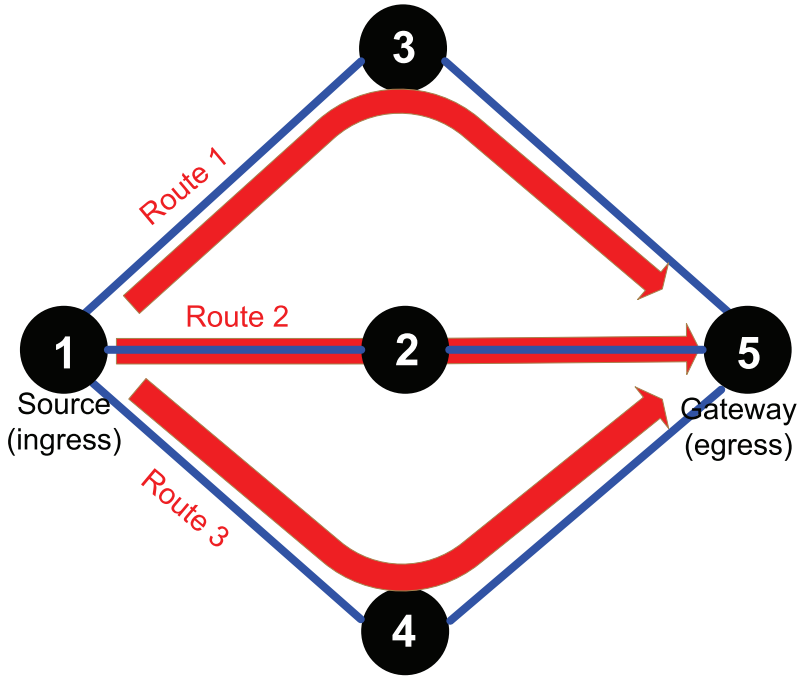

Figure 3.4: A five-node WMN setting, where node 1 serves as the source (an ingress node) to generate connections with multiple QoS requirements and node 5 serves as the gateway (an egress node). Three disjoint routes exist between the packet network to carry the traffic.

considering the packet loss ratio as the QoS requirement. We think it is comparable to our GAC methodology in many ways. Our algorithm is also compared with conventional protocol layer 2 and layer 3 techniques: the round robin scheduler (RR, [77]) and AODV routing protocol [96]. A complete simulation topology is shown in Figure 3.5. The overall performance is investigated in terms of the overall gateway goodput in Figure 3.6, and the average QoS outage probability of completed connections in Figure 3.7.

Figure 3.6 shows that "IQoSR+GAC" outperforms all other schemes in terms of the overall gateway goodput even for small traffic inter-arrival time (heavy load conditions). It is observed that 1.7 times, 2.5 times, and 4.1 times gains are achieved if our proposal is compared with "IQoSR+SAC", "IQoSR" schemes, and the lower-bound "RR+AODV" scheme respectively. This is due to the accurate justification

Table 3.2: Effects of using different combinations of partial derivatives for admission estimation

	First order statistics	**First and second order statistics**
Maximum Supported Throughput	22Mbps	25Mbps
Error Bound	±2Mbps	±500Kbps
Maximum Number of Connections	30	35
Error Bound	±5	±2
QoS Outage	≈ 13%	≈ 10%
Blocking Probability	≈ 12%	≈ 8%

of the packet network resource availability through Taylor expansion and the derived statistics like QoS performance index and associated partial derivatives. It is interesting to mention that the compared "SCAC" scheme, where the essence of this algorithm is to define the achievable capacity as the amount of bandwidth that can probabilistically guarantee the packet loss ratio to be smaller than a threshold, and using Central limit Theorem to approximate its closed-form by Gaussian process. In our simulation, we find that under high traffic load condition, the arrival process could be more accurately assumed to be Gaussian which helps "SAC" scheme achieves relatively high goodput. Nevertheless, when the traffic load is relatively low, "SAC" scheme makes wrong admission decisions which turns into less goodput and higher QoS outage. On the other hand, due to the multi-dimensional QoS requirements besides the packet loss ratio, "SCAC" scheme under-performs our proposal for delay-sensitive applications like VoIP. It is also interesting to observe that the gateway goodput saturates when the traffic load becomes higher. Finally, when we increase the number of nodes deployed in a fixed geographic area from 15 to 25, gateway goodput decreases by 20%. This is not only because of the sharing nature of network resources, but also much more co-channel interference created to, or from, the adjacent nodes.

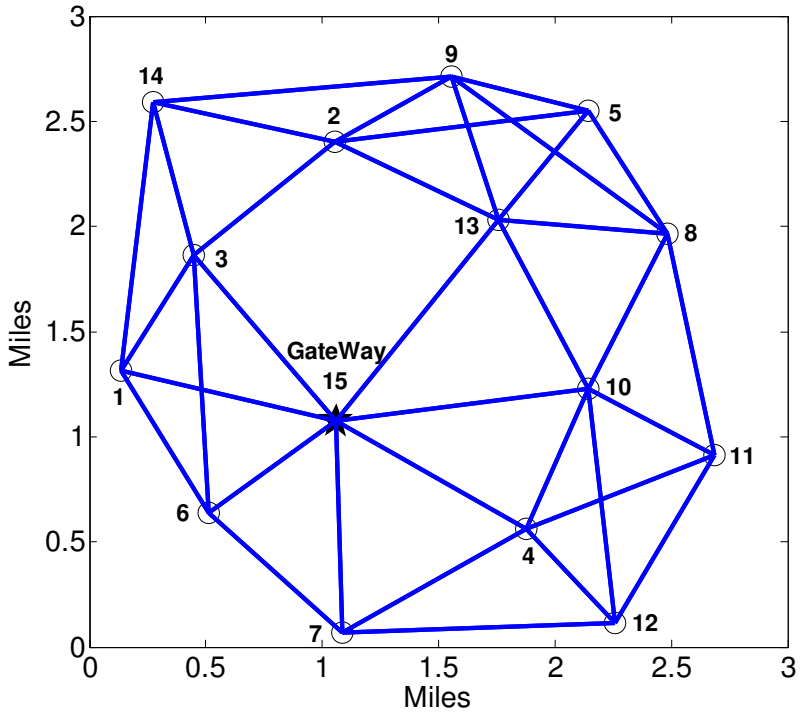

Figure 3.5: A complete simulation setting with 15 wireless mesh routers and one gateway node; they are all randomly deployed in a 2-D area.

Figure 3.7 illustrates the QoS outage probability, defined as the probability of any connection's QoS requirements to fail during their lifetime, or the condition $\theta_q(t) \leq 1, \forall q \in \mathcal{Q}$ at any time is not satisfied at all. It can be seen that the proposed QoS control scheme can even guarantee 85% of the QoS satisfaction for the underlying applications, as compared to only 81% if no AC is used, 82% if "SAC" scheme is used, and 58% if "RR+AODV" is employed. This is because the impact of the newly admitted connections on existing ones' QoS experience has been estimated and accurately reflected in the parameter of the updated QoS performance index $\widetilde{\mathrm{I}}(0)$ which reflects the new QoS experience the packet network can provide to all connections, new and old.

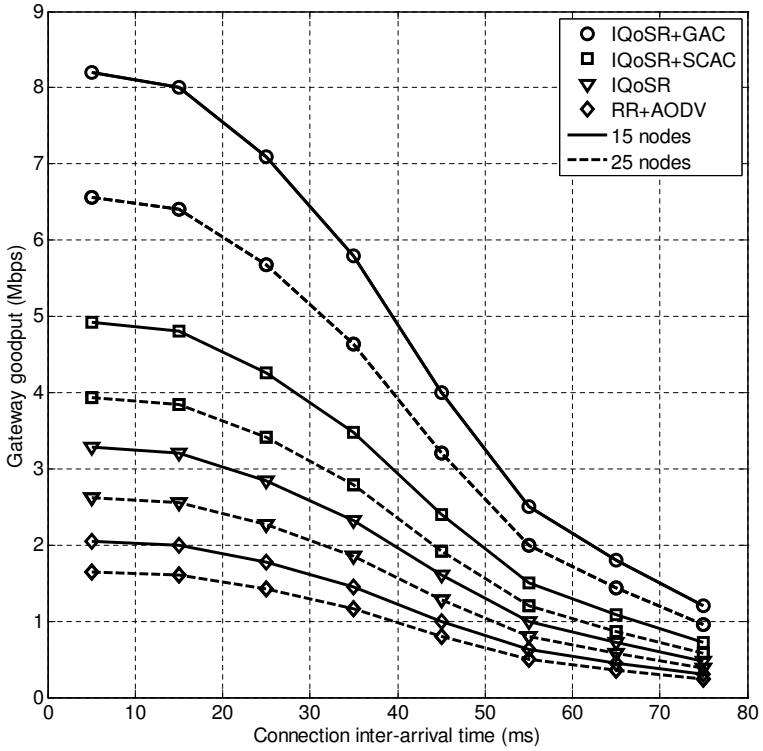

Figure 3.6: Simulation result of the overall gateway goodput with respect to (w.r.t.) the different new connection inter-arrival time and the number of nodes.

3.9 Discussions

This section provides extensive discussions on the applicability and feasibility issues of the proposed GAC methodology for QoS control.

3.9.1 The Applicability

The proposed GAC methodology has wide applicability to many packet networks, wired and wireless. The main essence of this methodology is to use a generic mapping f to represent the resource space between the ingress and the egress, or a packet network. By doing this, we hide the heterogeneous operational features of each subnetwork, and we are able to estimate the ETE potential impact the new connection admission would make on the QoS experience over the entire network. Furthermore,

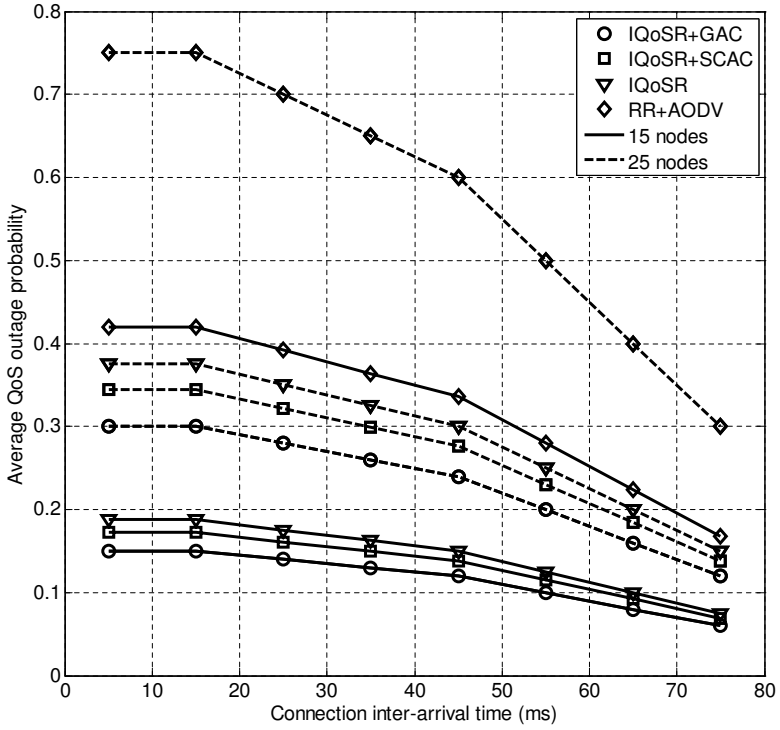

Figure 3.7: Simulation result of the average QoS outage probability w.r.t. the different new connection inter-arrival time and the number of nodes.

this methodology is applicable to many packet networks, since only the derivatives in Hessian matrix need to be computed at the egress node after approximating the shape of the curve produced by the mapping f through real-time measurements, and only ingress and egress nodes are involved in statistics reporting but not any intermediate node within any of the subnetworks within the packet network. Therefore, the method itself does not require any knowledge of detailed network protocols to use at any node within the packet network as *a priori*. However, the benefit each connection can receive is constantly probed by the egress node and feedbacks to the ingress node. Finally, when it is applied to wired networks, smoother channel fluctuation with no interference can be expected and we may only need first order statistics, since second order statistics are used to track the fast change, especially applicable to wireless networks.

3.9.2 The Scalability

The scalability issue is always a desirable property of the network design, which indicates its ability to handle the growing network size in a graceful manner. The proposed GAC methodology applies to many packet networks, wired and wireless, and can be scalable to different network sizes, however with some potential extensions. When the network size of each subnetwork is relatively very big which contains a large number of nodes, for the purpose of AC, the task of keeping the ETE operational statistics between each pair of the ingress and egress node is increasing exponentially w.r.t. the network size, which can be computationally very difficult. Therefore, we propose to adopt a *hierarchical* (or two-tier) solution. As shown in Figure 3.1, The first tier solution is the local AC decision for each subnetwork where we are able the to hide the scalability issues of each subnetwork, *i.e.,* however how many nodes are deployed in each subnetwork, from the whole packet network perspective, it can also observe a small set of ingress and egress node pairs at the edge of each subnetwork. Then, the second-tier solution is the AC decision at the edge of the packet network, same as the one proposed in this chapter. In this way, we believe our GAC is scalable to many packet networks whatever the network size would be.

3.9.3 The Feasibility

The Impact of Confections with Large Throughput Requirement

As mentioned earlier, the Taylor expansion is used as a tool to estimate the potential admission impact, however, it is feasible given that statistics are sufficiently collected within a relatively close region of the operating point. When incoming connections are associated with large throughput requirements, our algorithm may under-perform the optimum due to the discontinuous nature of the mapping f. This

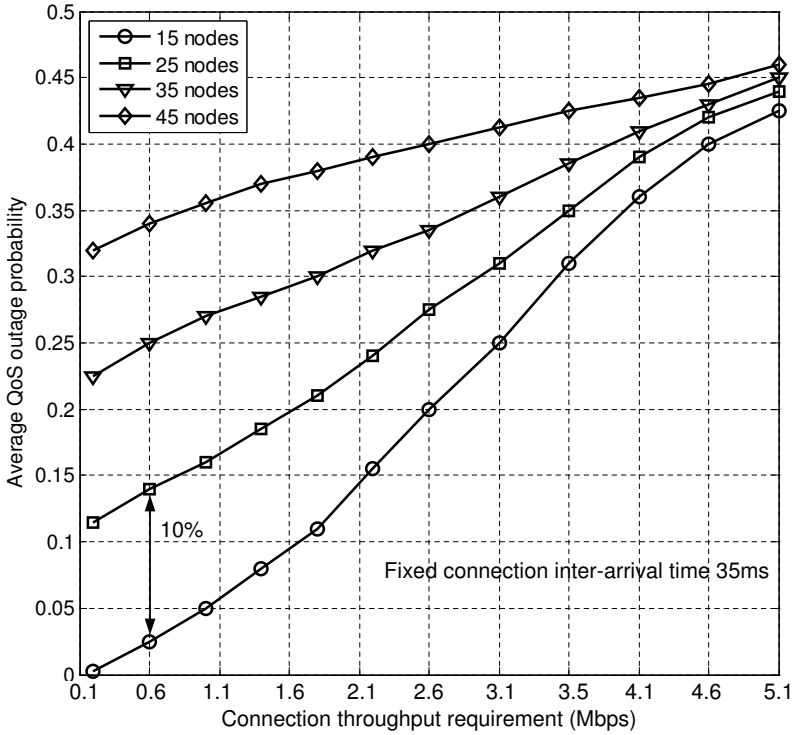

Figure 3.8: The impact of different throughput requirements on the estimation of the new QoS performance index if the new connection is admitted. The figure is plotted with different number of nodes in a fixed size area.

will impact the accuracy of the derivative calculations and ultimately the estimated QoS performance index $\widetilde{\mathrm{I}}(0)$. The effect is depicted in Figure 3.8 that the larger the $T_q^r, \forall q \in \mathcal{Q}$ is, the more severe impact on error accumulation and amplification for the estimation would be. Therefore, connections' QoS experience is more vulnerably to be violated, where the careless admission of one single large connection may jeopardize all existing connections' QoS. This is why QoS outage probability increases significantly w.r.t. larger $T_q^r, \forall q \in \mathcal{Q}$. On the other hand, when more nodes are deployed within the packet network, more interference will be created, and thus QoS outage probability is increased.

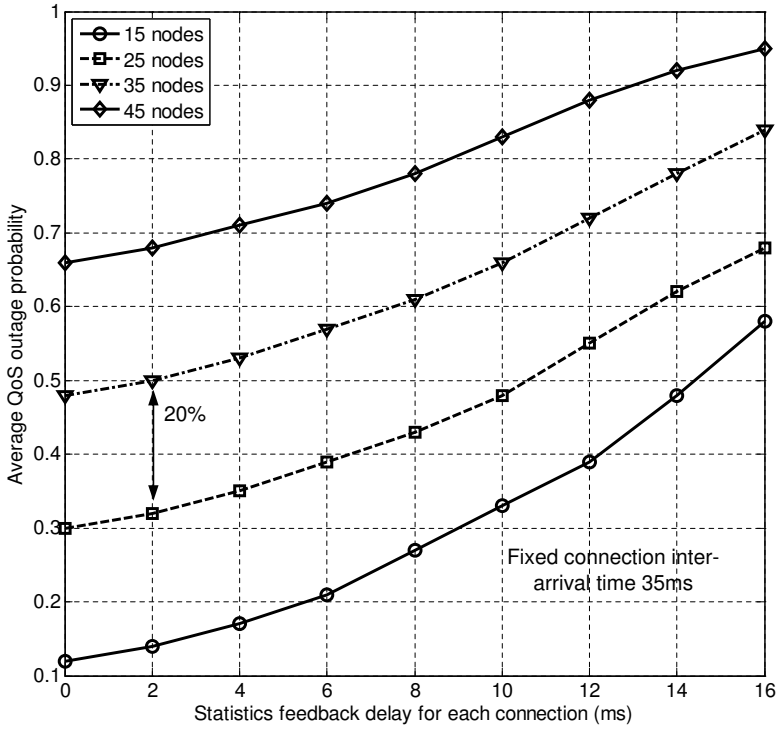

Figure 3.9: The impact of the statistics feedback delay on the estimation of the new QoS performance index if the new connection is admitted. The figure is plotted with different number of nodes in a fixed size area.

Statistics Feedback Delay

Figure 3.9 shows the impact of statistics feedback delay (from the egress node to the ingress node) on the average QoS outage probability. Outdated statistics incurred by higher feedback delay may result in the slow reaction of the ingress node to be aware of the packet network operational status. As a result, imperfect AC decisions would be made based on the inaccurate network status that can affect the QoS of all connections, new and existing. Therefore, slight modifications in existing protocols may be required to minimize the feedback delay by giving higher priority to packets carrying the required statistics.

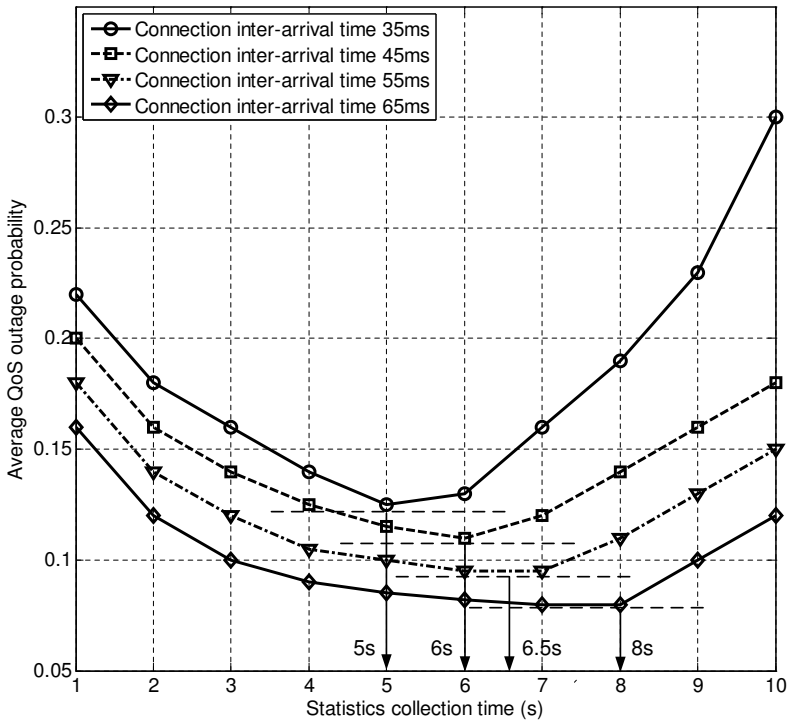

Figure 3.10: The impact of the statistics collection time on the estimation of the new QoS performance index if the new connection is admitted. The figure is plotted with different number of nodes in a fixed size area.

Statistics Collection Time

This corresponds to the time window at the ingress node that the collected statistics $(N(t), T(t), \mathrm{I}(t))$ are used or discarded to approximate the shape of the curve produced by the mapping f. If the size of the time window is relatively big, historical data may not represent the most recent packet network status given the highly dynamic nature of some packet networks. On the other hand, the relatively short collection time may lead to the insufficient number of statistics and inaccurate predictions, especially if the traffic is bursty. The effect of these can be seen from Figure 3.10 where for fixed connection inter-arrival time, there is an *optimal* statistics collection time that achieves the lowest QoS outage probability. Therefore, this simulation results can be used for optimal system design where a collection time window will provide the optimal system performance.

3.10 Summary

In this chapter, a GAC methodology for all packet networks, wired and/or wireless, is proposed. First, a novel concept of QoS performance index is introduced to integrate the multi-dimensional QoS requirements to indicate the degree of QoS satisfaction for any connection in the packet network. Second, the packet network (which may consist of multiple heterogeneous subnetworks) between a pair of ingress and egress nodes is modeled as a black box, and a generic mathematical function is used to represent it. In this way, the heterogeneous operational features of each subnetwork and network protocols in use are hidden, so that a degree of transparency the proposed AC methodology is successfully provided to end node upper layers. Third, AC decision is made for the new connection through estimating the admission impact of admitting the new connection on QoS performance index by Taylor expansion in limited orders. Last, the applicability, scalability and feasibility issues are discussed. Finally, extensive simulations are performed in WMNs to show that if compared with other statistics-based network capacity driven AC algorithm and other conventional network protocols, the proposed GAC methodology can efficiently accommodate higher number of connections with satisfactory QoS.

Chapter 4

Network Operations and Management through Negotiations

PREVIOUS Chapters would perform well in terms of network protocol designs of multi-hop wireless networks, however how to dynamically adjust network resource allocations (or "internal" operations) and adapt applications' service quality demands (or "external" operations) to achieve the optimal network operations, is still missing. This Chapter thus identifies such a research gap between the "external" and "internal" operations, which is presented as a *negotiation* process.

To demonstrate the proposed approach, we use wireless sensor networks (WSNs) as the illustrative example, where the notion of service quality is interpreted from the quality-of-information (QoI) aspects, which relate to the ability to judge available information *fit-for-use* for a particular purpose [97, 98] in general. Unlike QoS (*e.g.,* packet or connection level requirements such as delay, throughput, PER etc.) defined by telecommunications industry years ago, QoI has been sparsely studied in WSNs, normally characterized by a number of quality attributes for information, such as accuracy, latency, completeness (the degree of similarity of the received information compared with the original), and spatiotemporal relevancy [99]. For the illustration purposes, we only demonstrate the way handling the

information accuracy in this chapter.

The overall goal of this negotiation process is to control the overall QoI levels provided to the new and existing tasks through a runtime monitoring of the QoI levels provided to the completed tasks. Key design elements in support of the proposed negotiation approach include:

1. the *QoI satisfaction index*, which quantifies the degree to which the required QoI is satisfied by the WSN;

2. the *QoI network capacity*, which expresses the ability of the WSN to host a new task (with specific QoI requirements) without sacrificing the QoI of other currently hosted tasks;

3. a *negotiation process* that iteratively reconfigures and optimizes the usage of network resources and the degree of QoI acceptance of prioritized tasks.

4.1 Introduction

Continuing advances in sensor-related technologies, including those in pervasive computing and communications, are opening more and more opportunities for the deployment and operation smart autonomous WSNs [2]. A significant portion of research in the area of WSN deployment and operation, which we refer to as *operations and management* (O&M) of WSNs, focuses primarily on the "internal" aspects of WSNs such as energy-efficiency, coverage, routing topologies for efficient query and data dissemination, and so on [2]. The complementary area that considers the "external" relationships that WSNs have with the QoI needs of the tasks they support have experienced significantly less exposure. The novel study of O&M in WSNs for the efficient and effective support of the QoI needs of applications is central of this chapter.

Existing research usually uses network utility analysis techniques, striving to achieve desirable network operation by fine tuning both statically and dynamically configurable WSN resources, such as traffic flows, routing paths, transmission power, to maximize a network utility [33,34] curve that is assumed to be known as *a priori*. However, the design requirement of *a priori* knowledge of utility functions is very challenging, or even more if the utility comes to represent the entire network's behaviour when dealing with the multi-dimensionality of QoI attributes for the varying needs of on demand tasks. These challenges are further compounded when considering the time-varying radio, energy, and other network resource conditions, along with the stochastic nature of the task arrival and duration processes.

To address these challenges, we conduct a *runtime* learning of the QoI benefit provided by the WSN to the tasks it supports by monitoring the level of QoI satisfaction (or, the *QoI satisfaction index* of a task) they attain in relation to the QoI they request. This relaxes the requirement for the *a priori* knowledge of utility functions and facilitates the dynamic accommodation of tasks with heterogeneous requirements. Then, by proposing the concept of *QoI network capacity*, the ability of a WSN to host a new task (with specific QoI requirements) is expressed without sacrificing the QoI of existing tasks. Last, an adaptive *negotiation* process is proposed to dynamically configure the usage of network resources to best accommodate all tasks' QoI requirements.

The rest of Chapter 4 is organized as follows. In Section 4.2, related research activities are highlighted. Section 4.3 presents the system model. Section 4.4 describes three key design elements for the design. Numerical results and discussions are demonstrated in Section 4.5 through an evaluation of a dynamic multi-task intruder detection environment. Finally, we discuss the applicability and the complexity of the proposed approach in Section 4.6 and conclude in Section 4.7 with a summary.

4.2 Related Work

To the best of our knowledge, the proposed QoI-aware O&M framework represents the first such WSN application management solution of its kind. However, there is a related body of work that has motivated our current research path. Despite endeavours for defining QoI [97,98], it was not until recently that work in [100] proposed a conceptual framework to enable the dynamic binding of sensor information producers and consumers in a QoI-aware manner. The framework expresses information requirements and capabilities according to the 5WH (who, what, when, where, why) principle and enables information producers to categorize the quality attributes of their information in an application-agnostic manner while permitting information consumers to access QoI in application-specific way. Such principles largely enable the development of a framework such as ours.

The network utility maximization (NUM) framework has been recently extended to consider a unique aspect of WSNs: shared consumption of a single sensor data source by multiple tasks with different utility functions [33]. This is further addressed in [34], where NUM is used for jointly adapting source data rates and node transmission powers in a multicast, multi-hop wireless environment.

Other work has focused on modelling the state of the network with respect to supporting quality-related administrative decisions. This includes characterizing information loss due to network delays and buffer overflows to make task admission decisions [101] and monitoring network resource allocations and the status of sensed phenomena to determine available QoI [102] and sustain required QoI [103]. Sensor network management issues were studied in [104, 105], where in [105] information quality (completeness and accuracy) is supported by a dynamic Bayesian network model based constraint optimization problem which takes into account all the levels of information processing, from measurement to aggregation to data delivery

with predefined network utility. Similarly, [104] further compared the solution with Bayesian network model.

In closing we also mention here work on WSN middleware designs [106] to support some notion of information quality [107–109]; the latter work has inspired aspect of our research in the area.

4.3 System Model

We consider a WSN comprising a set $\mathcal{S}$ of sensor nodes, $\mathcal{S} = \{s_i; i = 1, 2, \ldots, N\}$, and a sink node (with sufficient information processing and energy capabilities). Let $\mathcal{Q}$ be the set of tasks the WSN currently services, i.e., tasks that currently are bound to the WSN and retrieve the sensed information from it. Let l_q be the duration of that service for task $q \in \mathcal{Q}$, and let $\mathcal{S}_q \subset \mathcal{S}$ be servicing task q. The arrival and service duration processes are in general stochastic in nature and their details will be specified as needed later on.

Task $q \in \mathcal{Q}$ requires the monitoring of specific feature(s) of interest such as temperature, event occurrences or locations, density of a hazardous chemical, and so on. Each feature is associated with one or more QoI attributes (members of the vector $\underline{z}$), such as accuracy and latency of the received information, whose desired values are declared by the tasks upon their arrival for service. We use the superscript r to denote a QoI attribute value as *required* (and declared) by a task and superscript a for the level of the QoI attribute *attained* by the WSN. For example, α_q^r and α_q^a may denote the required and attained probability of detection of an event, respectively. Finally, tasks belong to one of $|u|$ priority classes with higher priority tasks enjoying preferential treatment and higher guarantees for receiving satisfactory QoI levels. The set $\mathcal{Q}_u \subset \mathcal{Q}$ represents all the tasks of priority u, $u = 1, 2, \ldots, |u|$.

We close this section by highlighting the overall flow of the proposed QoI

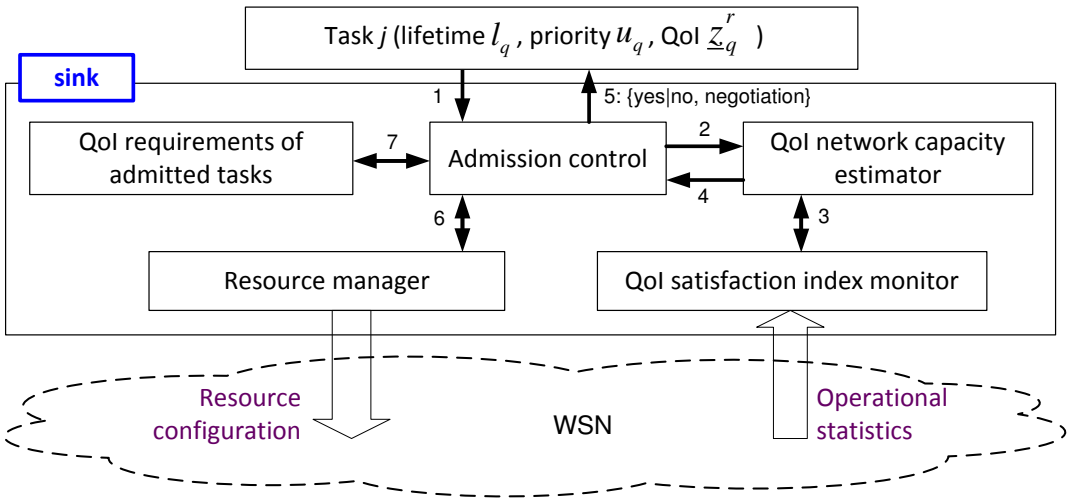

Figure 4.1: The overall flow of the negotiation process.

O&M approach, as shown in Figure 4.1. Tasks arrive at the sink for admission (arrow 1), upon which the *QoI network capacity* is measured (arrow 2, see Section 4.4.2). The capacity value is estimated by monitoring the *QoI satisfaction index* of completed tasks (arrow 3, see Section 4.4.1). Each of the QoI requirements of the new task is then compared with the QoI network capacity (arrow 4); if there are enough network resources to support the admission of the task (admission decision represented by arrow 5), then optimal resource allocation among all occupying tasks is calculated (arrow 6). Otherwise, a negotiation process may be called upon in an attempt to adjust the QoI requirements of existing tasks so that the new task will be accommodated (arrows 3, 4, 6, and 7; see Section 4.4.3). When a task completes, the resource allocation function is called again to re-optimize the allocation of limited network resources so that existing ongoing tasks' QoI will be improved.

4.4 Key Design Elements

In this section, we will describe the following three key design elements of our proposal, namely, (1) QoI satisfaction index, (2) QoI network capacity, and (3) a nego-

tiation process.

4.4.1 QoI Satisfaction Index

Similar to the QoS performance index introduced in Chapter 3, this index is used to describe the level of QoI satisfaction the tasks received from the WSN at any time t. It is applicable to each task q and QoI attribute z and is defined as:

$$\theta_q^z(t) \triangleq \tanh\Big(k \ln \frac{z_q^a(t)}{z_q^r}\Big), \quad \forall q \in \mathcal{Q}, \underline{z}, t, \tag{4.1}$$

where z is a member of vector $\underline{z} \in \mathbb{R}^{M_z}$, which represents one of the QoI attributes, could be α for the probability of detection, and k denotes a scaling factor with typical value $k = 1$.

The selection of the functions $\ln(\cdot)$ and $\tanh(\cdot)$ is rather arbitrary but result in the intuitively appealing and desirable behavior for satisfaction as shown in Figure 4.2. The QoI satisfaction index behaves symmetrically around the origin, rising from -1 to $+1$, with the value 0 signifying the case where the WSN satisfies exactly the QoI expectations of tasks (lower bound). The parameter k is a scaling factor that determines the "range" of the values for the ratio of attained and required ones. For example, when this ratio changes slightly while close to 1 (so its logarithm is close to 0), the QoI satisfaction index experiences the biggest variations. On the other hand, when the ratio is sufficiently away from 0, the QoI satisfaction index is less sensitive.

Per-task QoI Satisfaction Index

A per task QoI satisfaction index $\theta_q(t)$ can be defined by combining the per QoI attribute indexes above. In this chapter, we opt to use the minimum of these indexes,

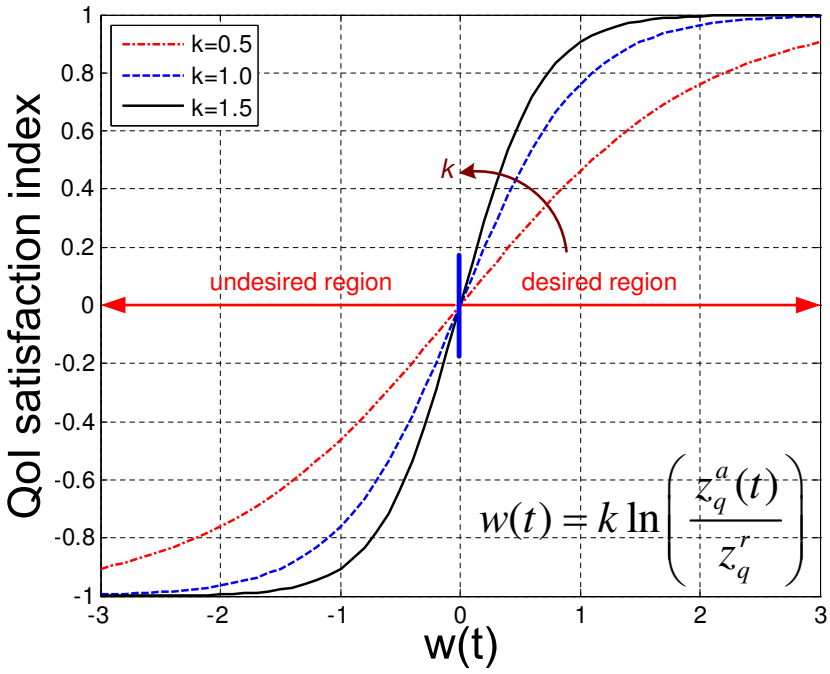

Figure 4.2: The illustrative example for the definition of QoI satisfaction index. It is desirable to have $z_q^a \geq z_q^r$ since it is assumed that the QoI attribute values should be at least as big as the required value to guaranteed the service quality.

i.e.,

$$\theta_q(t) = \min_{\forall z \in \underline{z}} \theta_q^z(t) \in (-1, 1), \quad \forall q \in \mathcal{Q}, \forall t. \tag{4.2}$$

Therefore, it follows immediately from the definition of satisfaction index that:

Lemma 4.4.1. *For any task $q \in \mathcal{Q}$, its multiple QoI requirements are simultaneously satisfied if and only if $\theta_q(t) \in [0, 1)$ at any time t.*

Network-wide QoI Satisfaction Index

Likewise, we can define the network-wide, instantaneous QoI satisfaction index (or, simply QoI satisfaction index later in this chapter) $\mathrm{I}(t)$ as the minimum of indexes of all tasks in service at time t:

$$\mathrm{I}(t) = \min_{\forall q \in \mathcal{Q}} \theta_q(t). \tag{4.3}$$

Lemma 4.4.2. *For all ongoing tasks $q \in \mathcal{Q}$, their multiple QoI requirements are simultaneously satisfied if and only if* $\mathrm{I}(t) \in [0, 1)$ *at any time t.*

Note that the QoI satisfaction index not only represents the sensing quality at a selected group of data sources $\mathcal{S}_q$, but also reflects the communications quality and potential data aggregation along the reporting route. This is important because successful QoI supports rely on two parts: information sensing of multiple data sources, and information reporting through multi-hop WSNs that may incur further packet loss, delay, or damage.

4.4.2 QoI Network Capacity

Before admitting a new task for service, it is important to identify the potentially limiting resources and estimate the maximum "capacity" a WSN can support at any given time t, denoted as $\underline{\mathcal{C}}(t) \in \mathbb{R}^{M_c}$. Thus, we define:

QoI network capacity indicates the time-varying capability a WSN can provide to any task with satisfactory QoI requirements, such that $\mathrm{I}(t) \in [0, 1), \forall t$. QoI network capacity $\underline{\mathcal{C}}(t)$ is a vector with dimension M_c such that each element $\mathcal{C}_i(t) \in \underline{\mathcal{C}}(t), \forall i = 1, 2, \ldots, M_c$, can represent any one of the following parameters (not exclusively though): the maximum probability of detection, the maximum information accuracy, the smallest information gathering delay, etc.

To illustrate the process for the estimation of QoI network capacity, consider a use case where event detection tasks ask service from the WSN declaring a required detection probability $\alpha_q^r, \forall q \in \mathcal{Q}$. In this case, the QoI network capacity reduces to a *scalar* representing the maximum probability of detection, $\underline{\mathcal{C}}(t) \triangleq \alpha_{\max}(t)$.

Similar to Chapter 3, we have opted to adopt a "black box" view for the WSN encompassing the sensor nodes and associated network resources. These sensors include data sources, relays, and sinks, which are involved in collecting and reporting

sensor measurements. Finite resources are shared by multiple tasks within the black box that include, but are not limited to, time, bandwidth, energy, etc. We assume that a new task arrives at $t = 0$, and input variables are the number of existing tasks $N(0)$ and the maximum required probability of detection $\alpha(0) = \max_{q\in\mathcal{Q}} \alpha_q^r$, *i.e.,*

$$\underline{x}(0) = \Big(N(0), \alpha(0)\Big) \in \mathbb{R}^2. \tag{4.4}$$

Then, the black box is represented by the mapping $f(\cdot)$:

$$\mathrm{I}(0) = f\Big(N(0), \alpha(0)\Big). \tag{4.5}$$

Given more stringent QoI requirement for the input variables, a lower QoI satisfaction index is expected. Meanwhile, Lemma 4.4.1 indicates that the shape of curve will reach a lowest satisfaction level when QoI satisfaction index $\mathrm{I}(0) = 0$, at which level the QoI network capacity is also defined, as shown in Figure 4.3.

Therefore, we assume a large new task is admitted in the WSN, which corresponds to an input change $\Delta\underline{x}(0) = \big(\Delta N(0), \Delta\alpha(0)\big) = \big(1, \alpha_{\max}(0) - \alpha(0)\big)$, and for the expected output change,

$$\tilde{\mathrm{I}}(0) = f\Big(n_{\max}(0), \alpha_{\max}(0)\Big) = 0. \tag{4.6}$$

Therefore, we rewrite the Taylor expansion in Eqn. (3.10), as:

$$\mathrm{I}(0) + \frac{\partial f}{\partial N(0)} + \Delta\alpha(0)\frac{\partial f}{\partial \alpha(0)} + \frac{1}{2}\frac{\partial^2 f}{\partial N(0)^2} + \frac{[\Delta\alpha(0)]^2}{2}\frac{\partial^2 f}{\partial \alpha(0)^2} + \Delta\alpha(0)\frac{\partial^2 f}{\partial N(0)\partial\alpha(0)} = 0, \tag{4.7}$$

where $\Delta\alpha(0) = \alpha_{\max}(0) - \alpha(0)$ and all partial derivatives are computed at current network state $\underline{x}(0) = (N(0), \alpha(0))$ at time $t = 0$. Since, $N(t)$ is discrete, we approximate its "derivatives" by the slopes of adjacent network measurements. For example, assume that at least two adjacent measurements $(N(0)|_1, \mathrm{I}(0)|_1), (N(0)|_2, \mathrm{I}(0)|_2)$

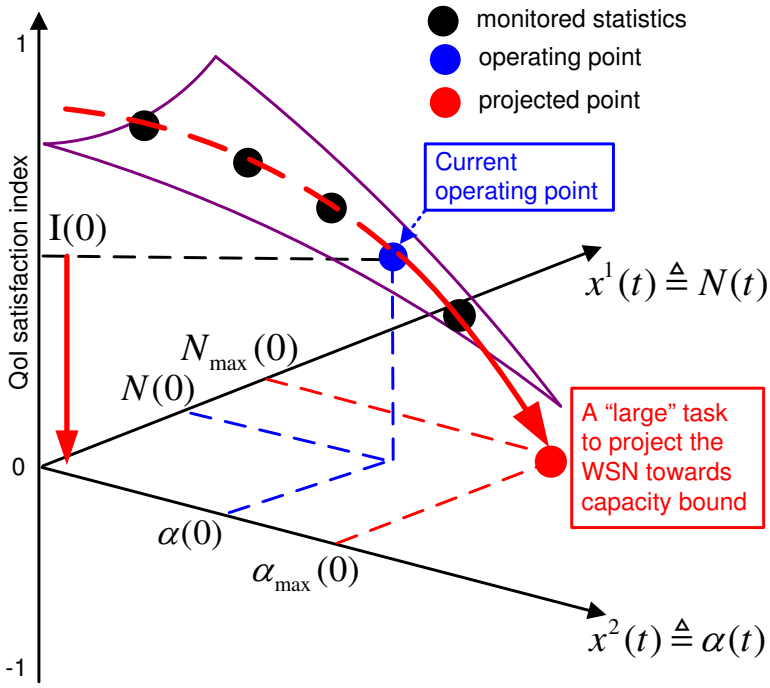

Figure 4.3: An example of the shape of curve produced by the mapping f to show how to obtain the QoI network capacity in term of the maximum probability of detection $\alpha_{\max}(t)$.

around the current state $N(0)$ are obtained; then, the first order partial derivative is computed as the average of two adjacent slopes of measurements,

$$\left.\frac{\partial f}{\partial N(t)}\right|_1 \approx \frac{\mathrm{I}(0)|_1 - \mathrm{I}(0)}{N(0)|_1 - N(0)}, \quad \left.\frac{\partial f}{\partial N(t)}\right|_2 \approx \frac{\mathrm{I}(0) - \mathrm{I}(0)|_2}{N(0) - N(0)|_2}, \tag{4.8}$$

and the second order partial derivative is computed as the change of the above two slopes:

$$\frac{\partial^2 f}{\partial N(t)^2} \approx \frac{\left.\frac{\partial f}{\partial N(t)}\right|_1 - \left.\frac{\partial f}{\partial N(t)}\right|_2}{(N(0)|_1 - N(0)) - (N(0) - N(0)|_2)}. \tag{4.9}$$

Expression (4.7) is a quadratic function with only decision variable $\alpha_{\max}(0)$. Therefore, since $\alpha_{\max}(0) > \alpha(0)$,

$$\alpha_{\max}(0) = \alpha(0) + \frac{-b + \sqrt{b^2 - 4ac}}{2a}, \tag{4.10}$$

where:

$$\begin{aligned} a &= \frac{1}{2}\frac{\partial^2 f}{\partial \alpha(0)^2}, \\ b &= \frac{\partial^2 f}{\partial N(0)\partial \alpha(0)} + \frac{\partial f}{\partial \alpha(0)}, \\ c &= \mathrm{I}(0) + \frac{\partial f}{\partial N(0)} + \frac{1}{2}\frac{\partial f}{\partial N(0)^2}. \end{aligned} \tag{4.11}$$

If, furthermore, the second order term is negligible around the current system operating point $\underline{x}(0) = (N(0), \alpha(0))$, we can further simplify (4.10):

$$\alpha_{\max}(0) = \alpha(0) - \frac{\mathrm{I}(0) + \frac{\partial f}{\partial N(0)}}{\frac{\partial f}{\partial \alpha(0)}}. \tag{4.12}$$

Figure 4.3 illustrates how this methodology is used, and Figure 4.4 depicts a 3-D real-time measurements (from a system simulation) of QoI satisfaction indexes collected. Each dimension of statistics collected is shown in Figure 4.5(a) and Figure 4.5(b), where QoI network capacity is estimated.

4.4.3 Negotiation Process

Following the estimation of the QoI network capacity, suppose a new task q' with priority $u_{q'}$ and QoI requirements $\underline{z}^r_{q'}$, arrives at the sink for the admission decision at time t. Before assigning the task to any sensor(s), admission control decision is made according to the following conditions,

$$\begin{cases} \text{Admission, if } \underline{\mathcal{C}}(t) \succeq \underline{z}^r_{q'}, \\ \text{Negotiation, otherwise,} \end{cases} \tag{4.13}$$

where notation $\succeq$ denotes the element-by-element comparison. Typically, an admission control scheme will outright ban the new task if some threshold condition was violated. However, here we opt first for a negotiation between all tasks, new and old,

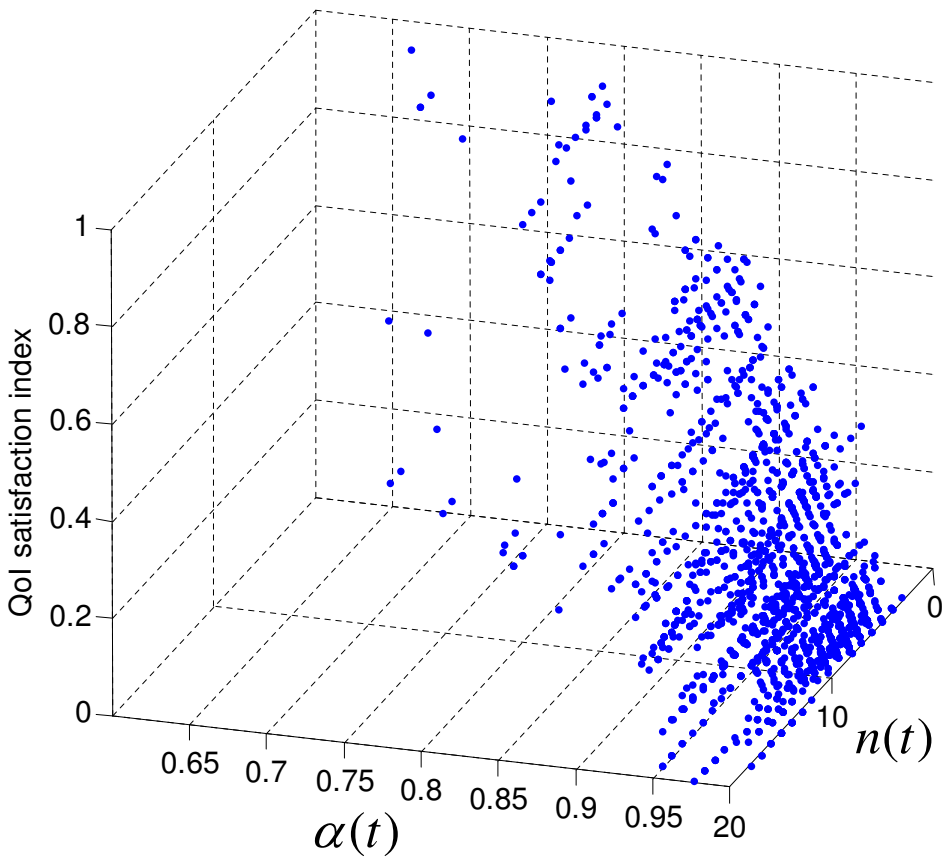

Figure 4.4: An example of obtaining the QoI network capacity through real-time system statistics.

in search of an acceptable (to the tasks) and attainable (by the network) compromise regarding the QoI satisfaction index delivered by the network. Resource management in this case includes scheduling, rate and power allocation, sensor selection, integration of data compression, etc.

Under the guidance of the resource optimization, ongoing tasks may reconfigure and reallocate network resources among themselves, so that the optimized network status will give the best achievable QoI for the new task. Nevertheless, sometimes the network might be overloaded operating near the capacity bound, *i.e.,* however the network resources are optimized and reconfigured, the required QoI will not be satisfied. Hence, the negotiation process is employed, *i.e.,* the new

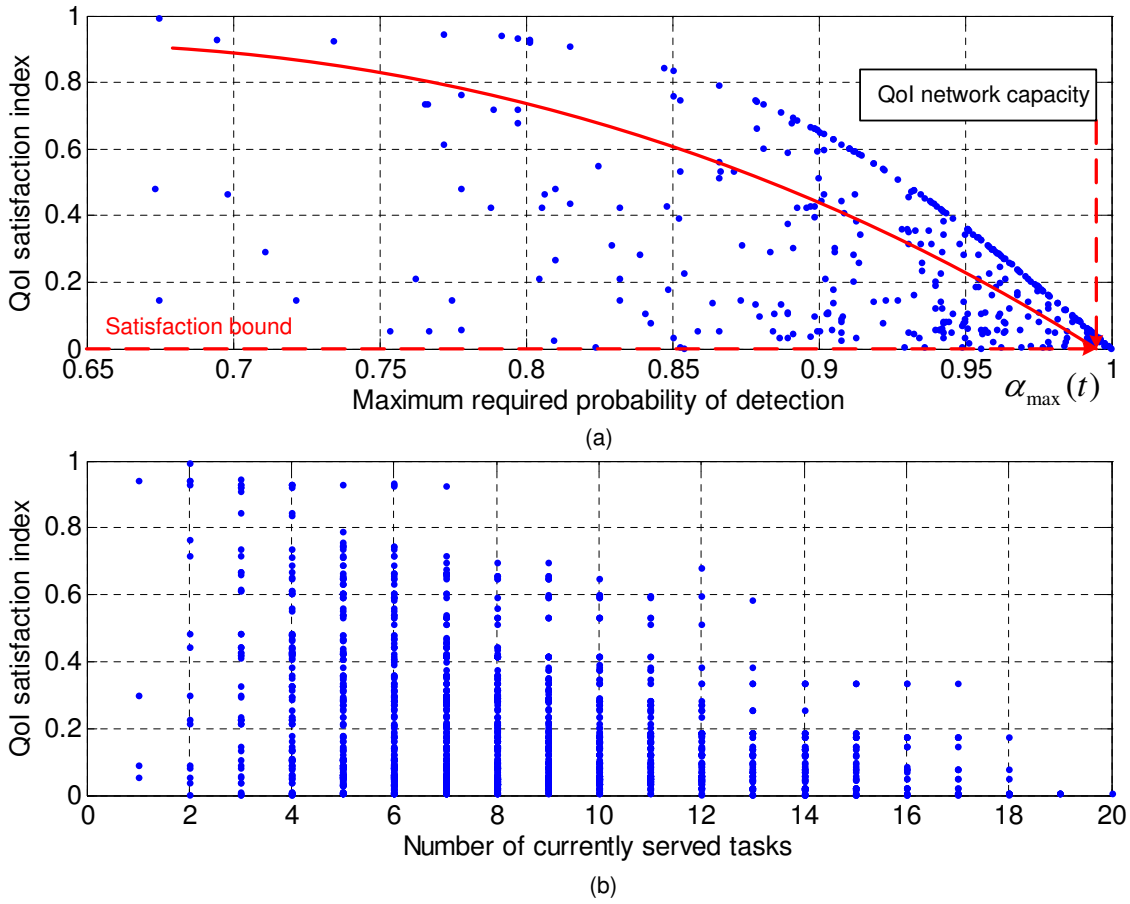

Figure 4.5: (a) System simulation to show $\alpha(t)$ dimension of the statistics. (b) System simulation to show $N(t)$ dimension of the statistics.

task may gradually *adapt* its QoI level in order to meet network capabilities, or existing tasks with lower priority levels may tune their QoI requirements and release resources for the new higher priority one. This information would feed to the admission control module for admission; if still unsuccessful, WSN will trigger the resource optimization module to further reconfigure the limited resources based on updated QoI levels. This is an *iterative* process, where task QoI, admission control, and resource optimization collaborate until satisfactory QoIs for all tasks are reached, or otherwise the new task is eventually rejected.

Mathematically, during the negotiation phase, the following optimization is

pursued:

$$\left\{\underline{\xi}_q^*(t)\right\}_{\forall q\in\mathcal{Q}\cup q'} = \arg\max \mathcal{F}\left(\left\{\underline{z}_q^r\right\}_{\forall u_q<u_{q'}}, \{\underline{\xi}_q(t)\}_{\forall q\in\mathcal{Q}\cup q'}\right) \tag{4.14}$$

$$\text{subject to: } \begin{cases} z_q^a(t) \geq z_q^r, \forall q \in \mathcal{Q}\cup q', \forall \underline{z}, \forall t \\ \sum_{\forall q\in\mathcal{Q}\cup q'} \underline{\xi}_q(t) \preceq \underline{\mathcal{P}}(t). \end{cases}$$

Recall that $u_{q'}$ denotes the priority of the new task. The objective function *Fairness* $\mathcal{F}$ is chosen as the optimization target since service degradation and adaptation for lower priority tasks may violate ongoing tasks' QoI satisfactions.

The arguments to this optimization problem are adaptable multiple QoI requirements $\left\{\underline{z}_q^r\right\}_{\forall u_q<u_{q'}}$ of those tasks with lower priority classes, and the resource occupancy $\{\underline{\xi}_q(t)\}$. The optimization is first constrained by the need to respect the QoI satisfaction for the task of different priority groups. Then it is further constrained by the resource allocation. Without loss of generality, let $\underline{\mathcal{P}}(t) \in \mathbb{R}^{M_r}$ describe the instantaneous remaining resources like energy, and let $\underline{\xi}_q^*(t) \in \mathbb{R}^{M_r}$ denote the corresponding optimal resource occupancy of each task $q, \forall q \in \mathcal{Q}\cup q'$, after the resource allocation. Then, $\underline{\eta}(t) \in \mathbb{R}^{M_r}$ represents the total resource occupancy for all ongoing tasks at time t, *i.e.,*

$$\underline{\eta}(t) = \sum_{\forall q\in\mathcal{Q}\cup q'} \underline{\xi}_q^*(t). \tag{4.15}$$

And this total resource demands have to be smaller or equal than the remaining resources $\underline{\mathcal{P}}(t)$ element by element.

4.5 Numerical Results

4.5.1 The Scenario

We assess the proposed scheme under an intruder detection scenario [110], where detection tasks arrive dynamically into a WSN with different QoI constraints (see Figure 4.6). Detection probability α_q^r for task q is the only parameter considered in the multi-dimensional QoI requirements, and 30 sensors are deployed randomly in a 2-D square 200×200 meters. Suppose that at the initial deployment, a cumulative reserve of energy at level $\mathcal{E}$ is equally assumed for each sensor, so that $N\mathcal{E}$ denotes the overall energy research for the entire network. Tasks arrive according to the Poisson process with rate λ and last for a random exponential time interval with average duration $1/\mu$, and let $l_q(t)$ denote the actual remaining lifetime of task $q \in \mathcal{Q}$ at time t. All tasks are categorized randomly into a high priority task set $\mathcal{Q}_1$ and a low priority task set $\mathcal{Q}_2$, or $\mathcal{Q} = \mathcal{Q}_1 \cup \mathcal{Q}_2$. While high priority users have guaranteed QoI requirements that are not negotiable, low priority users' QoI requirements are adaptable between least-satisfactory and most-satisfactory QoI levels, $\alpha_q^{r,l}$ and $\alpha_q^{r,h}$, respectively. Typical values for the probability of detection in the simulation is $\alpha_q^{r,l} = 0.75, \alpha_q^{r,h} = 0.95$. Sensors are equipped with antenna arrays such that at any given time one sensor could form multiple beams to service concurrent tasks and the strength of the beam is controlled by power allocated to each sensor (as sensor 8 shown in Figure 4.6).

The Detection Model:

We employ a simple detection model [111] using physical properties of the sensors, where the probability of detection p_{iq} for task q from sensor i is achieved assuming

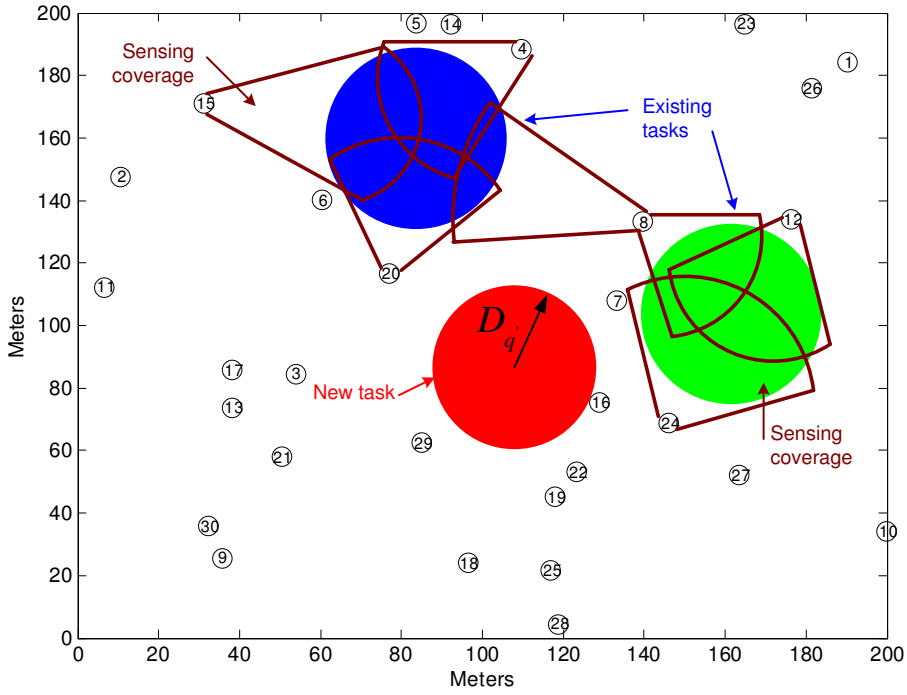

Figure 4.6: Simulation scenario for the considered intruder detection application. Two existing intruder detection tasks exist in the network (marked as the blue and green regions), while a new task (marked as red region) arrives for admission. Several sensors are selected per task as data sources (sensor 8 serves two tasks simultaneously by adjusting antenna beams).

using normalized full power $\gamma_q^*(t) = 1$, *i.e.*,

$$p_{iq} = \begin{cases} 1, & \text{if } r_{iq} < d_1, \\ e^{-\beta_1 (r_{iq} - d_1)^{\beta_2}}, & \text{if } d_1 < r_{iq} < d_2, \\ 0, & \text{if } r_{iq} > d_2 > d_1, \end{cases} \tag{4.16}$$

$\forall i \in \mathcal{S}_q$, and the question of how to determine the sensor sources $\mathcal{S}_q$ for task q will be discussed later. $\beta_1 = 0.12, \beta_2 = 0.8$, and $d_1 = 28$m, $d_2 = 58$m are typical parameters used, and r_{iq} denotes the sensor-to-target distance. The optimal resource occupancy vector $\underline{\xi}_q^*(t)$ is reduced to a scalar as the power in this case, *i.e.*, $\underline{\xi}_q^*(t) \triangleq \gamma_q^*(t)$, and the *attained* per-task QoI satisfaction index $\theta_q(t)$ can be explicitly expressed in the

following form,

$$\theta_q(t) = \tanh\left(k \ln \frac{\gamma_q^*(t) \times \min_{\forall i \in \mathcal{S}_q} p_{iq}}{\alpha_q^r}\right), \forall q \in \mathcal{Q}, \tag{4.17}$$

where attained probability of detection is computed as $\alpha_q^a(t) = \gamma_q^*(t) \min_{\forall i \in \mathcal{S}_q} p_{iq}$. Here we assume that the probability of detection it experiences is given by the smallest of all probabilities of detection attained by any of the the sensors that service the task ($\min_{\forall i \in \mathcal{S}_q} p_{iq}$). Furthermore, we assume that the QoI level received by task q, α_q^a, increases linearly with the corresponding power $\gamma_q^*(t), \forall q \in \mathcal{Q}$. Other assumptions are also applicable, but out of scope of this Ph.D. dissertation.

The Selection of Sensor Sources:

The question of how to determine the sensor sources $\mathcal{S}_q$ for task $q \in \mathcal{Q}$ is illustrated in this section. As introduced in (4.17) for the detection model, given the geographic location of sensors and tasks, one can assume the full power mode $\gamma_q^*(t) = 1, \forall q \in \mathcal{Q}$ to compute the best achievable probability of detection without the presence of the negotiation and optimization, and this could be done centrally at the sink. Therefore, the maximum set of sensor sources can be determined and $\mathcal{S}_q, \forall q \in \mathcal{Q}$ is obtained. Mathematically, we have:

$$\begin{cases} \text{selected, } i \in \mathcal{S}_q, & \text{if } p_{iq} \geq \alpha_q^r, \quad \forall q \in \mathcal{Q}, \\ \text{not selected ,} & \text{otherwise.} \end{cases} \tag{4.18}$$

The Optimal Power Allocation:

Optimal resource allocation among all existing and new tasks is performed such that all tasks' QoI requirements are successfully guaranteed and certain network objective is maximized. Here the fairness among all tasks is provided, given QoI

satisfaction achieved for all high priority and low priority users, we have:

$$\begin{aligned} \left\{\gamma_q^*(t)\right\}_{\forall q\in\mathcal{Q}} &\triangleq \arg\max \min_{\forall q\in\mathcal{Q}} \theta_q(t) \\ &= \arg\max \mathrm{I}(t) \end{aligned} \tag{4.19}$$

$$\text{subject to: } \begin{cases} \alpha_q^a(t) \geq \alpha_q^r, \forall q \in \mathcal{Q}, \forall t \\ \sum_{\forall q \text{ on } i} \left[\gamma_q(t) l_q(t)\right] \leq \zeta_i(t), \forall i \in \mathcal{S}_q, \end{cases}$$

where the design objective is chosen to balance the achieved QoI satisfaction indexes among all ongoing and new tasks. $\theta_q(t)$ is defined in (4.17) as a function of resource occupancy $\gamma_q(t)$. The first constraint represents the QoI satisfaction condition among all tasks, while the second constraint represents the energy reserve, and $\zeta_i(t)$ denotes the remaining energy for each sensor.

Without loss of generality, we assume that the equal power is allocated for all sensor sources of a particular task, the decision variable for this optimization problem is a set of power $\{\gamma_q^*(t)\}_{\forall q\in\mathcal{Q}}$.

The Negotiation Process:

When the network does not have enough network resources (energy in this case) supporting the new task, existing lower priority ones have to adapt, or degrade, their QoI levels to release resources for the new task. The optimization objective function for this process is to minimize the maximum percentage of QoI loss as a result of task negotiations, where the QoI loss is mathematically presented as the percentage of $\frac{\mathrm{I}_q^{\text{before}}(t)-\mathrm{I}_q^{\text{after}}(t)}{\mathrm{I}_q^{\text{before}}(t)}$. Therefore, we have:

$$\left\{\gamma_q^*(t)\right\}_{\forall q\in\mathcal{Q}} \triangleq \arg\min \max_{\forall q\in\mathcal{Q}_2} \frac{\mathrm{I}_q^{\text{before}}(t) - \mathrm{I}_q^{\text{after}}(t)}{\mathrm{I}_q^{\text{before}}(t)} \tag{4.20}$$

$$\text{subject to: } \begin{cases} \alpha_q^a(t) \geq \alpha_q^{r,h}, \forall q \in \mathcal{Q}_1, \\ \alpha_q^a(t) \geq \alpha_q^{r,l}, \forall q \in \mathcal{Q}_2, \\ \sum_{\forall q \text{ on } i} \Big[\gamma_q(t) l_q(t)\Big] \leq \zeta_i(t), \forall i \in \mathcal{S}_q, \end{cases}$$

where $\mathrm{I}_q^{\text{after}}(t)$ denotes the attained QoI level *after* negotiation by using power $\widetilde{\gamma}_q^*(t)$ in (4.17). While the first two constraints denote QoI requirement constraints for high and low priority users, the third constraint represents the per-sensor energy reserve for the sum of allocated energy among tasks. The solution of this optimization problem gives the best achievable QoI level for the new task by adapting existing ones' QoI requirements.

4.5.2 The Optimal Network Parameters

Given the proposed QoI-aware network management framework, we would like to explore the system limits under the conditions of constrained network resources and varying QoI requirements for different tasks, aiming at higher QoI network capacity, longer WSN lifetime, and increased task admission rate, while satisfying the required QoI of all admitted tasks. Particularly, for the considered intruder detection scenario, the WSN lifetime $T_{\max}$ is defined in a QoI-friendly fashion, as:

WSN lifetime is defined as the useful length of time for the WSN beyond which the amount of remaining energy reserve cannot guarantee a minimum probability of detection $\alpha_{\min}^r = \min_{\forall q \in \mathcal{Q}} \alpha_q^r$ for any task appearing at this time, located anywhere within the sensing field.

For this, we view the entire WSN system at time $t = 0$ before any task has been allocated, and view it as a service or "queuing" system where resources are not just the server and buffer capacities, but bandwidth, radio conditions, energy reserves of the system, etc. In this queuing system, the service capacity is not fixed or known *a priori*. It is represented by the *QoI network capacity*, which, as previously

discussed, is learnt at runtime from the QoI levels that the WSN delivered in the past. Given an average arrival rate of task λ, and an average task service duration $1/\mu$, questions of interest for such a system include:

(1) Given network load $\rho = \lambda/\mu$, what is the maximum WSN lifetime $T_{\max}$ given that all tasks accepted experience satisfactory QoI levels, *i.e.,* $\mathrm{I}(t) \geq 0$?

(2) Given minimum required WSN lifetime $T_{\min}$ and satisfactory QoI levels for all tasks, what is the region of admission rates $\lambda \leq \lambda_{\max}$ that the WSN can sustain as a function of task duration $1/\mu$?

The following Lemma broadly derives some expressions regarding the above questions under the considered intruder detection scenario. Recall that in this use case, the resource occupancy for each task q is reduced to a scalar as power $\gamma_q^*(t)$, and thus the relationship between $\gamma_q^*(t)$ and QoI satisfaction index $\theta_q(t)$ can be analytically represented by (4.17).

Lemma 4.5.1. *The task arrival rate λ vs. WSN lifetime T trade-off is of the form $\frac{\lambda T}{\mu} \leq \frac{N\mathcal{E}}{\beta\alpha_{\min}^r}$, where $\beta \triangleq \min_{\forall i \in \mathcal{S}_1} p_{i1}$ denotes a constant given geographic locations of sensors and tasks. Furthermore, the WSN lifetime and the maximum admission rate can be expressed as $T_{\max} = \frac{N\mathcal{E}}{\beta\alpha_{\min}^r \rho}$, and $\lambda_{\max} = \beta\frac{N\mathcal{E}\mu}{\beta\alpha_{\min}^r T_{\min}}$, respectively.*

Proof. Recall that for each task q, the amount of resource allocated is sufficiently reflected in (4.17). Or, we rewrite it as,

$$\gamma_q^*(t) = \alpha_q^r \frac{\exp\left(\frac{1}{k}\tan\theta_q(t)\right)}{\min_{\forall i \in \mathcal{S}_q} p_{iq}}, \quad \forall q \in \mathcal{Q}^T, t = 0, \tag{4.21}$$

where $\mathcal{Q}^T$ denotes the task set has been serviced during WSN lifetime T, and for brevity reasons, we drop the time index $t = 0$.

According to Lemma 4.4.1, the lower bound resource condition for satisfactory QoI is taken when $\theta_q = 0, \forall q \in \mathcal{Q}$ is used as the input, which produces the

minimum required power assumption $\gamma^*_{q,\min}$, as:

$$\gamma^*_q = \frac{\alpha^r_q}{\min_{\forall i \in \mathcal{S}_q} p_{iq}} \geq \frac{\alpha^r_{\min}}{\min_{\forall i \in \mathcal{S}_q} p_{iq}} = \gamma^*_{q,\min}, \tag{4.22}$$

where the inequality condition uses the notation $\alpha^r_q \geq \alpha^r_{\min}, \forall q \in \mathcal{Q}$.

At the same time though, resource constraints enforce the total amount of allocated network resources no more than the original total energy reserve level $N\mathcal{E}$ in the entire network at time $t = 0$, *i.e.,*

$$\sum_{\forall q \in \mathcal{Q}^T} \gamma^*_q l_q \leq N\mathcal{E}, \tag{4.23}$$

where N denotes the total number of sensor sources in the field, and l_q denotes the duration of certain task q that conforms to exponential distribution with parameter μ. Due to the stochastic nature of task arrival and departure processes, we use the conditions of expectation to approximate the LHS random variables of (4.23), as:

$$\begin{aligned} N\mathcal{E} &\geq \mathbb{E}\Big(\sum_{\forall q \in \mathcal{Q}^T} \gamma^*_q l_q \Big) = \mathbb{E}\Big(\mathbb{E}\Big(\sum_{\forall q \in \mathcal{Q}^T} \gamma^*_q l_q \Big| \mathcal{Q}^T \Big) \Big) \\ &= \mathbb{E}\Big(\sum_{\forall q \in \mathcal{Q}^T} \mathbb{E}\Big(\gamma^*_q l_q \Big) \Big) = \mathbb{E}\Big(\mathcal{Q}^T \mathbb{E}\Big(\gamma^*_1 l_1 \Big) \Big) \\ &= \mathbb{E}\Big(\mathcal{Q}^T \Big) \mathbb{E}\Big(\gamma^*_1 l_1 \Big) = \lambda T \mathbb{E}\Big(\gamma^*_1 \Big) \mathbb{E}\Big(l_1 \Big) \\ &= \frac{\lambda T}{\mu} \mathbb{E}\Big(\gamma^*_1 \Big), \end{aligned} \tag{4.24}$$

where we use the fact that the task's arrival process, departure process, and task optimal resource occupancies $\gamma^*_q, \forall q \in \mathcal{Q}^T$, are independent random variables. Furthermore, the average number of tasks $\mathbb{E}\left(\mathcal{Q}^T\right)$ admitted during WSN lifetime T can be approximated by the Little's theorem [112] as $\mathbb{E}\left(\mathcal{Q}^T\right) = \lambda T$, and average duration of task can be represented by $\mathbb{E}\left(l_1\right) = 1/\mu$. Therefore, we further simplify

(4.24) by using condition (4.22), as:

$$\begin{aligned} N\mathcal{E} &\geq \frac{\lambda T}{\mu}\mathbb{E}\Big(\gamma_1^*\Big) \geq \frac{\lambda T}{\mu}\mathbb{E}\Big(\gamma_{1,\min}^*\Big) \\ &\geq \frac{\lambda T}{\mu}\mathbb{E}\left(\frac{\alpha_{\min}^r}{\min_{\forall i\in\mathcal{S}_1} p_{i1}^d}\right) \\ &= \frac{\alpha_{\min}^r \lambda T}{\beta\mu}, \end{aligned} \tag{4.25}$$

where $\beta \triangleq \min_{\forall i\in\mathcal{S}_1} p_{i1}^d$ denotes a constant given geographic location of sensors and task. Hence, we rewrite (4.25) as,

$$\frac{\lambda T}{\mu} \leq \frac{N\mathcal{E}}{\beta\alpha_{\min}^r} \tag{4.26}$$

Finally, we derive the maximum network lifetime $T_{\max}$ and the maximum task admission rate $\lambda_{\max}$ as:

$$\begin{aligned} T_{\max} &= \frac{N\mathcal{E}}{\beta\alpha_{\min}^r\rho}, \\ \lambda_{\max} &= \frac{N\mathcal{E}\mu}{\beta\alpha_{\min}^r T_{\min}}. \end{aligned} \tag{4.27}$$

□

Lemma 4.5.1 proves that (4.26) serves as the principle worst-case (in terms of guaranteeing the minimum QoI requirement) WSN design criterion for this scenario, however it shows the fundamental trade-off among the WSN lifetime, the task arrival and departure rates, and the QoI requirement. For instance, higher QoI requirement ($\alpha_{\min}^r$) would constrain the energy usage for multiple tasks which in turn has impacts on the maximum admission rate $\lambda_{\max}$ and the WSN lifetime $T_{\max}$.

4.5.3 The Overall Network Performance

The proposed algorithm, referred as "AC+Negotiation", is compared with the scheme without negotiation process, referred as "AC only" and the traditional WSN management approach, referred as "Traditional".

The traditional WSN management approach is an one-off deployment process assuming "static" behaviors of system parameters, where sensors are positioned in the field of interests and set up their power consumptions in order to attain a particular level of probability of detection (e.g. $\alpha_q^r = 90\%$). Furthermore, the WSN does not adjust any of its operational parameters throughout its lifetime, no matter a task's needs. In contrast, the proposed negotiation-based network management approach allows parameters to be adjusted judiciously according to the task needs. In this simulation, we set the probability of detection in the traditional approach as the average of the probabilities of detection that the various tasks request in the proposed approach.

Figure 4.7 illustrates the average QoI outage probability of all completed tasks as a function of both task arrival rate λ and average task duration $1/\mu$. We define QoI outage as the portion of tasks (among all completed tasks) whose QoI requirements have failed. A QoI failure occurs for task q if $\theta_q(t) < 0, \forall q \in \mathcal{Q}$ occurs at least once during the task's lifetime. For fixed average task lifetime, it is interesting to observe the saturation feature of QoI outage probability for both "AC only" and "AC+Negotiation" schemes when we increase the arrival rate since admission controlling the new tasks helps maintain the QoI satisfaction of ongoing tasks. However, the saturation bounds of the two schemes vary significantly: for example, when $\lambda = 0.8$/min and $1/\mu = 20$ minutes, the proposed algorithm can guarantee 81% of QoI satisfaction for all tasks, as compared to 65% for "AC only" scheme. This is because the negotiation process helps optimize resource utilization to release some resources for higher priority tasks. On the other hand, when the average task duration is increased, the QoI outage probabilities of the three schemes increase by 20% proportionally. This is because the increasing network load $\rho = \frac{\lambda}{\mu}$ at any time in the network may jeopardize the QoI satisfaction of ongoing tasks, since finite network resources are shared by more tasks than before, which in turn

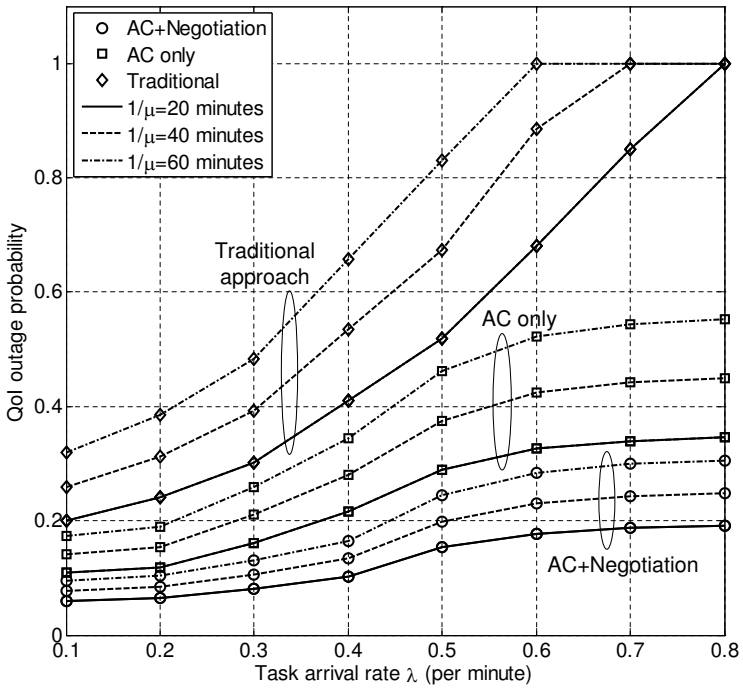

Figure 4.7: Simulation result of the average QoI outage probability among all completed tasks, w.r.t different task arrival rate λ and the average task lifetime $1/\mu$.

may violate the QoI network capacity bound.

The more detailed view of the average QoI outage probability for different priority user groups is shown in Figure 4.8, where only the "AC+Negotiation" scheme is plotted with a fixed average task lifetime $1/\mu = 40$ minutes. Interestingly, although similar behaviours for high and low priority user groups can be seen, the saturation speeds of their QoI outage probability differ significantly. This is primarily because our proposed negotiation process successfully guarantees non-negotiable QoI levels for high priority tasks, and adaptable QoI levels for low priority ones. On the other hand, successful task rejections help maintain low QoI outage probability and high QoI satisfaction for existing tasks in the network.

Figure 4.9 shows the behaviour of the average task blocking probability w.r.t. both task arrival rate and duration (the traditional case is not shown because rejection to new tasks is not applied). Given the fixed energy reserve for each sensor,

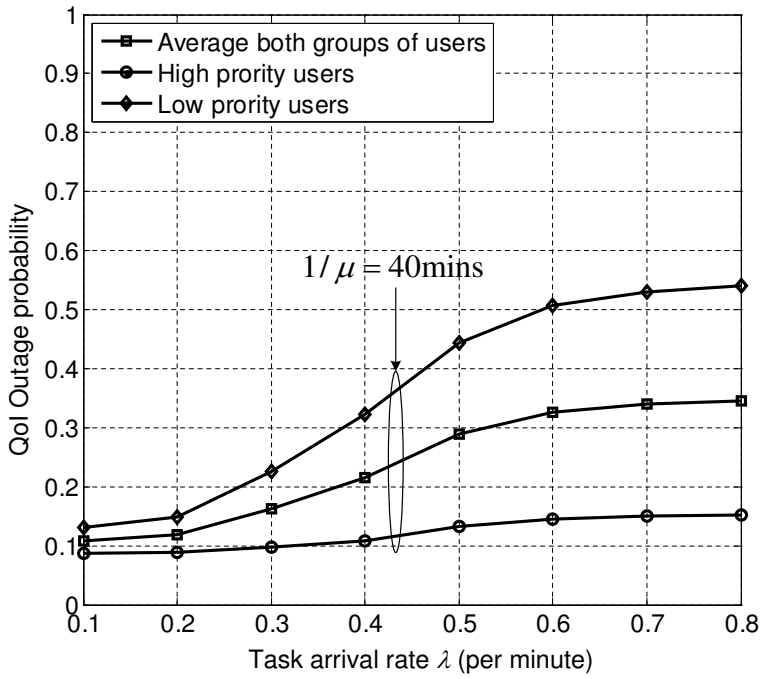

Figure 4.8: Simulation result of the average QoI outage probability among all completed tasks of two priority groups, w.r.t different task arrival rate λ and the average task lifetime $1/\mu$.

we see that the task blocking probability increases significantly when more tasks are offered (higher λ). This is because limited energy reserve is not able to support an increasing number of tasks arrived; however, these successful task rejections help maintain low QoI outage probability and high QoI satisfaction for existing ones in the network, as seen consistent with Figure 4.7. On the other hand, when network load ρ is increased by increasing task duration, the per-task resource availability decreases since the overall network resources are shared by higher number of concurrent tasks serviced. An interesting topic left here is to explore the energy consumption v.s. the decrease of blocking probability trade-off, *i.e.,* if the sensors are with power supply, ultimately in this intruder detection scenario, the blocking probability would be 0, however this is not the case in the real world, and thus certain compromise should have to be reached.

Table. 4.1 shows the average jitter performance of QoI satisfaction index

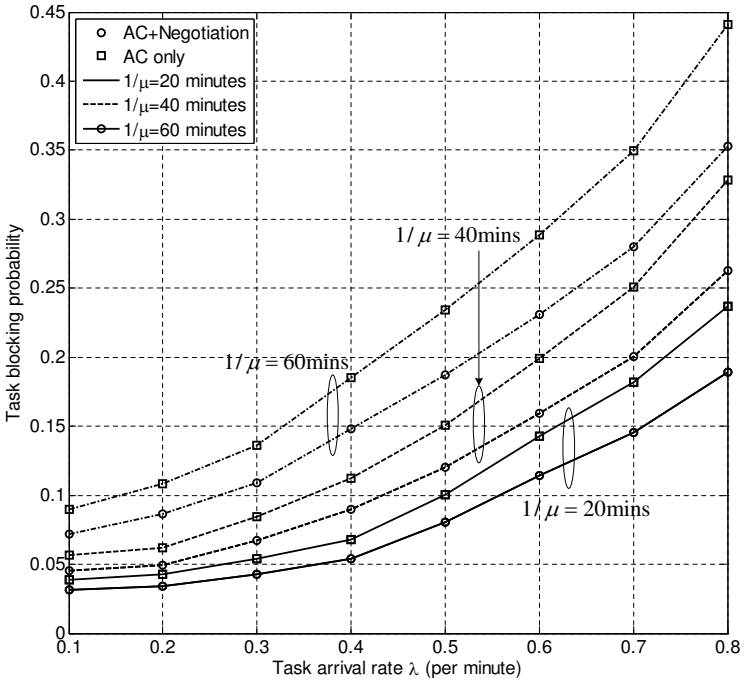

Figure 4.9: Simulation result of the average task blocking probability, w.r.t different task arrival rate λ and the average task lifetime $1/\mu$.

Table 4.1: Average jitter values of the received QoI satisfaction indexes among the low priority users, where the considered traffic has a fixed task arrival rate $\lambda = 0.5$ per minute

	AC+Negotiation	AC only	Traditional
$1/\mu = 20$ mins	0.16	0.21	0.27
$1/\mu = 40$ mins	0.17	0.22	0.28
$1/\mu = 60$ mins	0.18	0.24	0.29

among completed and satisfactory tasks, which is defined as the variance of the attained QoI satisfaction indexes. This performance metric directly reflects the range of QoI levels delivered to all tasks of the same priority group, which should be expected the smaller the better. For fixed average task lifetime $1/\mu$, we see 5% and 11% jitter decrease if the proposed "AC+Negotiation" scheme is compared with the "AC only" scheme and the "Traditional" scheme, respectively.

Figure 4.10 shows the normalized WSN lifetime w.r.t. different task arrival

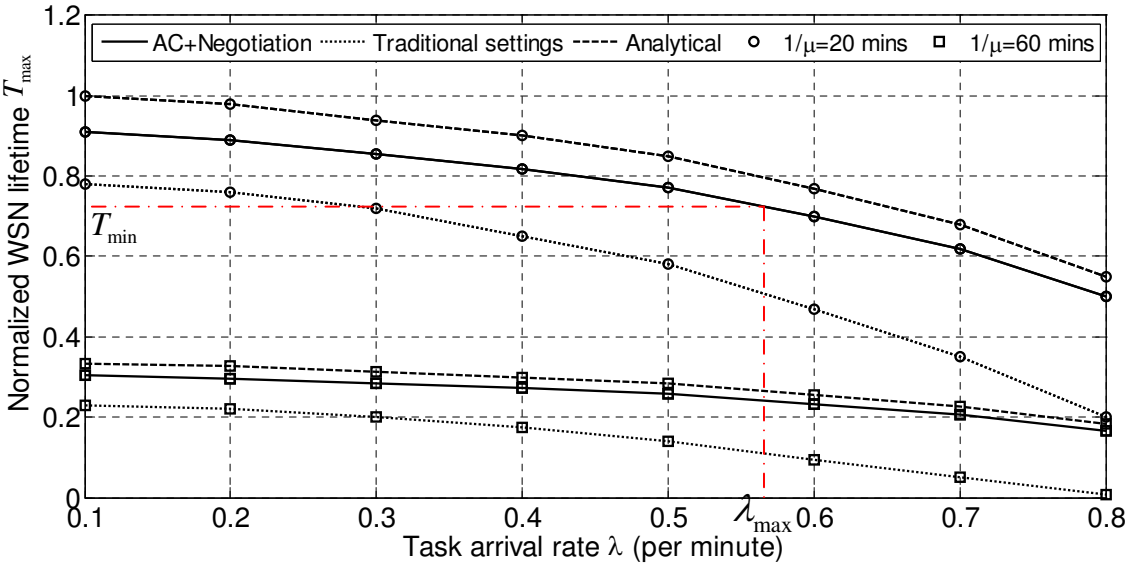

Figure 4.10: Simulation result of the normalized WSN lifetime w.r.t. the different task arrival rate λ and the task departure rate μ.

and departure rates. The minimum probability of detection is set to $\alpha^r_{\min} = 90\%$ to compute the analytical bound in (4.27), and in the simulation of the proposed "AC+Negotiation" scheme, the required probability of detection is set as $\alpha^r_q \in [\alpha^r_{\min}, 1], \forall q \in \mathcal{Q}$. For "Traditional" scheme, this QoI requirement is set as the average of that for "AC+Negotiation" scheme. We see significant improvement in the WSN lifetime when compared with the traditional scheme, and this improvement increases when tasks arrive more frequently (due to more efficient resource allocation among all tasks). Furthermore, the proposed approach successfully approximates the analytical results given in (4.27) while traditional settings perform far away behind. Meanwhile, given the desired WSN lifetime, this figure also shows the way to obtain the maximum admission rate $\lambda_{\max}$ the network can support given a minimum probability of detection $\alpha^r_{\min}$.

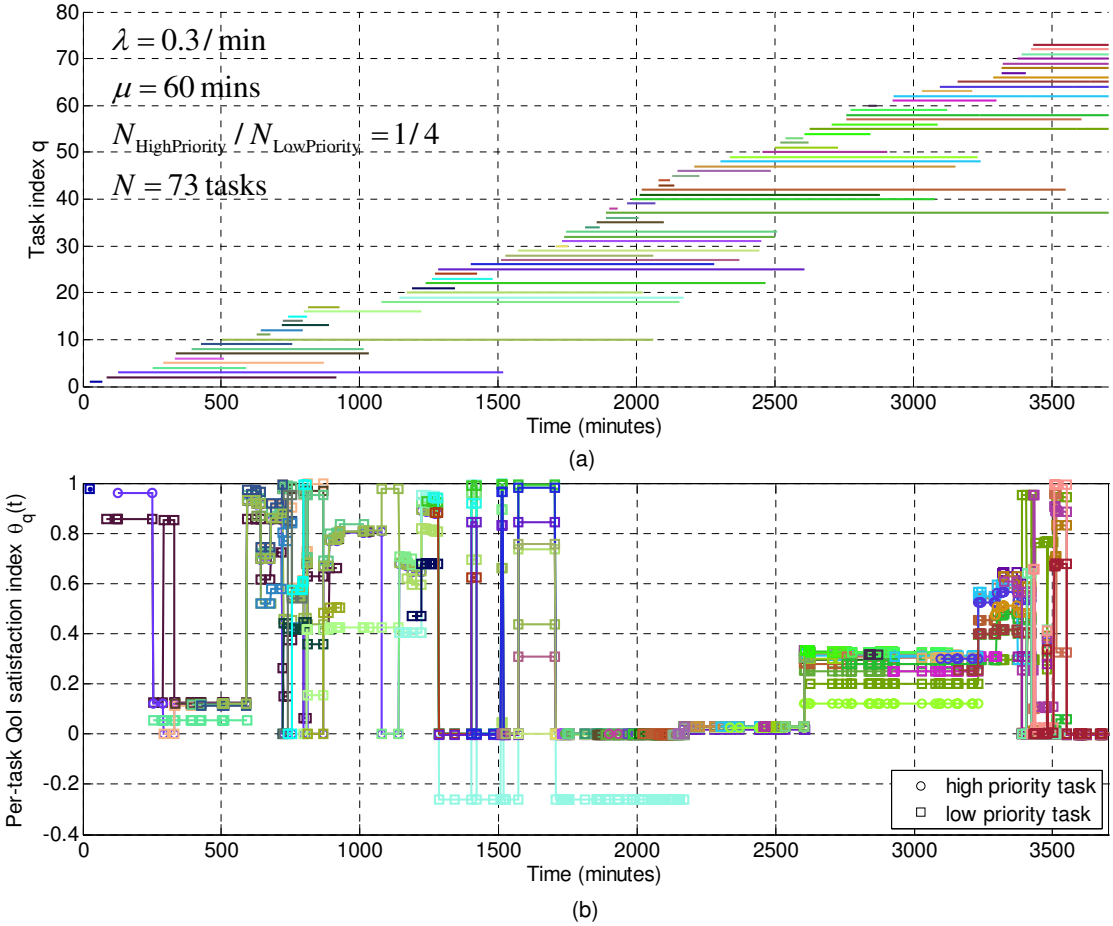

Figure 4.11: Simulation result for the system behavior as a result of resource optimizations and negotiations, where (a) shows the task arrival and departure time line, and (b) shows the per-task QoI satisfaction index change in real-time.

4.5.4 System Dynamic Behaviors

This Section aims to understand the detailed system behaviors due to dynamic task arrivals and departures, heterogeneous QoI requirements, and the resource optimizations and negotiations. Figure 4.11(a) illustrates the simulated traffic pattern (*i.e.,* the number of tasks, task arrival and departure processes, QoI requirements), and Figure 4.11(b) shows dynamic QoI experience received by 73 tasks.

Abrupt QoI changes can be seen under the relatively high traffic load conditions. When new task arrives, the negotiation process will attempt to accommodate

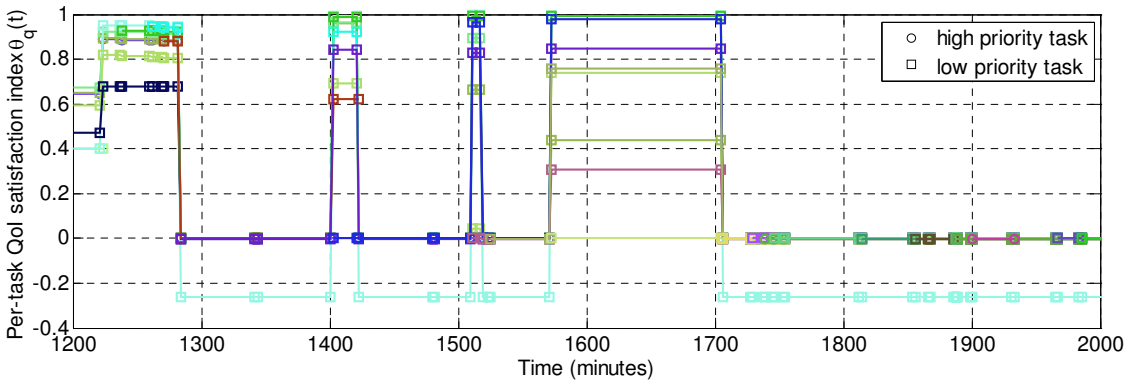

Figure 4.12: A finer view of the per-task QoI satisfaction index change from time 1200mins to 2000mins.

it while reasonably degrading the level of QoI satisfactions of existing tasks, but still maintaining the minimum required levels for all of them, *i.e.*, $\theta_q(t) = 0, \forall q \in \mathcal{Q}$. Meanwhile, when completed tasks are removed, pre-allocated network resources are released and reallocate to the existing ones so that the QoI levels of ongoing tasks are improved. However, our framework shows its capability to always optimize the resource utilization (power in this use case) in a way to maximize the QoI satisfaction whenever there is an opportunity. Meanwhile, when there is a sudden surge of task arrivals during a short period of time, or the tasks require very stringent QoI requirements (as shown from time 1200mins to 2300mins), some low-priority tasks would experience QoI failures as their QoI satisfaction levels cannot be satisfied in any meaningful anyway; but nevertheless the WSN always successfully guarantees the QoI levels of a portion of other low-priority tasks, *i.e.*, not all low-priority tasks suffer failure, but there are still portions of tasks successfully maintain the minimum level, *i.e.*, $\theta_q(t) \geq 0, \forall q \in \mathcal{Q}, \forall t$, to utilize the limited network resources. A finer view of what had happened from time 1200mins to 2000mins has been shown in Figure 4.12. In summary, network management with the presence of multi-task operations is more like a game, where tasks compete for limited network resources according to the relative compatibility of their priority and requested QoI require-

ments with dynamic network status. Therefore, not necessarily in the extreme case all tasks give up execution, but some low priority tasks with low QoI requirements may successfully survive.

4.6 Discussions

4.6.1 The Applicability

It is worth to highlight that the applicability of the proposed negotiation process is not restricted in WSN settings, but it rather has wide applicability to any other multi-hop wireless networks. This approach does not require specific protocols like scheduling or routing, nor PHY layer communications technologies to use. It provides a fundamental view for network management that application requirements can be negotiable with network operations such that service quality is maximized for all tasks, and blocking probability is minimized.

Furthermore, a possible extension of the existing model also applies to the case where sensors are with some degree of mobility. This is because that by modelling the WSN as a black box, we hide the heterogeneous operational status of each sensor and the communications quality among them; therefore, although the mobility may contribute to some deterioration of the communications links among sensors, but it would later be fully reflected by the defined QoI satisfaction index, or the quality aspect of the received information. Later, the proposed QoI network capacity estimation and negotiation optimization could be operated as normal.

4.6.2 The Scalability

We have identified that the proposed negotiation-based network O&M framework may not be scalable to large networks, especially for those there might not be a

central controller, like the sink node in WSNs. Therefore, we are very interested in extending the current work to a distributed solution to the proposed network O&M framework, and will be discussed in Chapter 6 as the future work.

4.6.3 The Complexity

Readers may argue that the proposed negotiation-based network management approach is mainly performed at the sink node, which is a centralized solution for WSNs and requires a powerful sink node with sufficient power supply and computation ability to negotiate and optimize among all tasks. Nevertheless, it is shown in Figure 4.11(b) that system adaptation and negotiation are usually happened in the time scale of several minutes, and thus it will not generate much overhead to the WSNs periodically for the control purpose. Furthermore, due to the distinct nature of the WSNs, collected information is always (partly) processed at some control center with sufficient information processing capability, and therefore to assume a centralized sink is appropriated.

4.7 Summary

In this chapter, a negotiation process is proposed, between the negotiable QoI requirements of the applications and the adaptable resource allocation of the network, where the benefit a network can provide to tasks is learned through the runtime monitoring of the satisfaction levels of completed tasks. To facilitate the negotiation process, QoI satisfaction index, QoI network capacity, and optimal system design parameters like network lifetime are defined and analyzed. Finally, extensive numerical results on a complete intruder detection user scenario show the proposed framework can successfully guarantee satisfactory QoI while maintaining low blocking probability and jitter.

Chapter 5

Data Ferrying Among Multiple Disconnected Mobile Subnetworks

PREVIOUS Chapters mainly focus on the design aspects of a single multi-hop wireless network, or, a subnetwork, from network protocol designs like QoS routing, distributed scheduling, and admission control algorithms, to the negotiation-based network management approach. Furthermore, the research so far has proved that the proposed algorithms, models, and protocols could successfully support some notion of quality for one single subnetwork; however, service quality requirements are sometimes required to extend beyond one single subnetwork, but among *multiple*, *disconnected*, or even *mobile* subnetworks. The solution for each of these subnetworks could be the one designed in previous Chapters; but the research issue of bridging communications among multiple subnetworks is the primary motivation for the research presented in this chapter.

We propose to use controlled, unmanned, and sensor-mounted mobile helper nodes, which are called "data ferries". In order to facilitate the packet exchange among subnetworks, we assume data ferries are equipped with wireless sensors, which could collect the data from one subnetwork (served as the source) and deliver the data to the other subnetwork (served as the destination). However, the sensors

have only limited sensing range, which forms a region that could be sensed, as shown in Figure 5.1, compared with the entire scope of one subnetwork. While existing work has explored various trajectory designs for the data ferry by assuming either static subnetworks or full observations at the data ferry, the problem still remains open when the nodes are mobile in each subnetwork, and when the data ferry only has partial observations. In this chapter, we investigate the problem of dynamic data ferry mobility control design under limited sensing capabilities. Assuming the data ferries are capable of sensing node presence within certain range and adjust their movements dynamically, we aim to design *control policies* that maximize the number of effective contacts (defined later), or the overall network throughput.

We investigate a comprehensive model of the control framework using Partially Observable Markov Decision Process (POMDP), based on which we study the structure of the optimal policy and propose an efficient heuristic policy which shows significant improvement over the predetermined benchmark. To the best of our knowledge, this is the first data ferry control mechanism that can handle both run-time randomness and incomplete observations.

Finally, it is worth noting that the "cross-layer" issues in this chapter is mainly from the design goal of maximizing the link-level throughput from the application layer's perspective, *i.e.,* the control policy, which is new of its kind.

5.1 Introduction

Continuing advances in sensor technologies and pervasive computing brings in new perspectives to solving challenging communication problems. Consider a group of moving nodes in a subnetwork, as illustrated in Figure 5.1 (where sufficient connectivity within a group is assumed). Due to rough terrains (*e.g.,* obstacles or danger zone in between) or application requirements, these subnetworks do not have di-

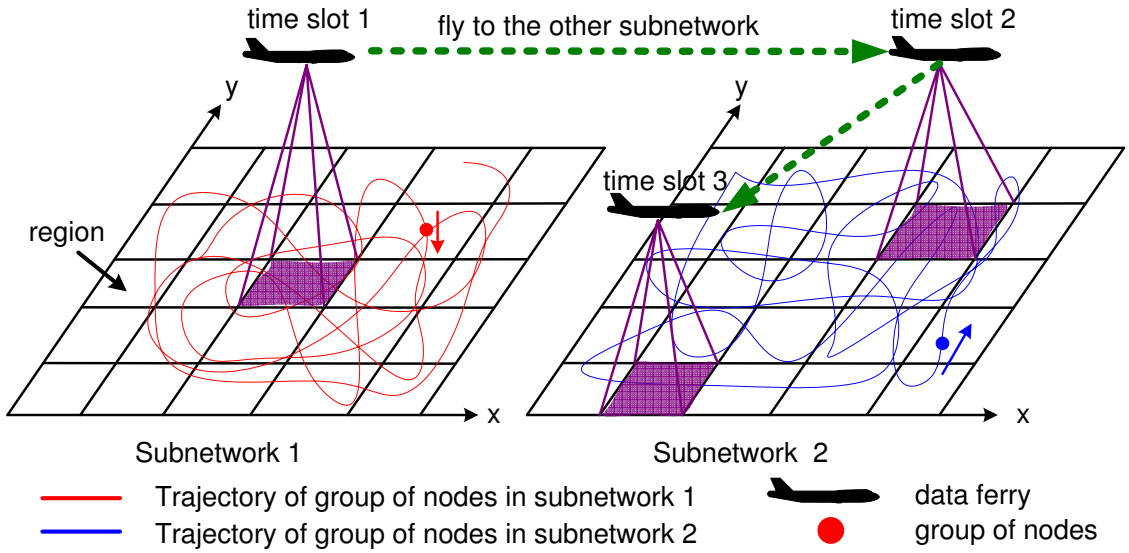

Figure 5.1: An example of how to bridge the communications between two disconnected subnetworks using a unmanned, sensor-mounted data ferry, where two groups of nodes move on disjoint trajectories and the data ferry has only limited (square as shown in this illustrative example) sensing range. The movement of the data ferry within three time slots are demonstrated.

rect contacts, and thus the mobile nodes in each subnetwork cannot communicate. Yet, they may have occasional communication needs. Applications of this kind can be found in military coalition networks, emergency response scenarios, and other challenged scenarios. In such circumstances, helper nodes mounted on controllable mobile platforms such as UAVs [35] have been proposed to assist with the communications in a *load-carry-and-deliver* manner. If the sensing range of the data ferry covers the entire network domain (which we refer here to as the *fully* observable case), the problem is straightforward and has been extensively addressed in the literature [35, 113, 114]. In practice, however, complete sensing coverage may not always be possible due to ground obstacles, vast network area, limitations of the sensors, or simply because of the need of keeping the UAVs from being exposed to the adversary. In this chapter, we study in detail how to bridge communications in such challenged scenarios using dynamically controlled, autonomous data ferries.

In this chapter, we explicitly consider the case where the sensing range of a data ferry only covers a subset of the entire subnetwork, and thus the data ferry does not know the exact locations of nodes once they are out of range (referred to as the *partially* observable case). For control purpose, we partition the entire domain of one subnetwork into regions as shown in Figure 5.1 (in 1-D case, a region refers to a segment). Furthermore, once the data ferry locates above certain region, it has full observations within the region, so that it can sense and communicate with a group of nodes in it. Each data ferry is equipped with certain sensing, communications, and storage capabilities, and most importantly, with a programmable control logic which can navigate it among local regions. Periodically, the data ferry senses the presence of nodes and uploads/downloads data upon contact, after which it will move to the next sensing point specified by the control logic and repeat the process. Meanwhile, the nodes may move among regions of their local subnetwork constantly according to their mission needs. Although it is possible to infer statistically properties of their movements, it is often impractical to accurately predict how nodes will move due to run-time randomness. The questions we investigate are: how should one control the data ferries to move intelligently based on the prior knowledge of node movements and the real-time (partial) observations? To our best knowledge, this is the first effort to address both run-time randomness and incomplete observations in data ferry control.

We are interested in the design of control policies for autonomous data ferries in delay tolerant networks (DTNs). To address the challenges of run-time randomness and incomplete observations, we take the approach of dynamic control, instead of designing fixed trajectories, and we design control policies that dynamically map available information to navigation actions at run time. Our specific contributions are three folds:

Comprehensive Control Framework:

We develop a comprehensive framework for the design of control logic using the tool of Partially Observable Markov Decision Process (POMDP). The framework incorporates both the prior knowledge of node movements, modelled by Markov chains on the partitioned subnetwork, and the design criteria, modelled by a payoff function and a reward structure. For concrete analysis, we aim to maximize the total number of effective contacts with an exponential discount.

Efficient Policy Computation Algorithm:

Due to the well-known curse of history and dimensionality, POMDP problems are generally difficult to solve exactly. We address this issue by developing an efficient policy computation algorithm based on belief space quantization. Moreover, we show that due to a special property of our problem, we can limit the belief points to subspaces one dimension smaller than the original simplex and significantly improve the performance.

Numerical Studies:

The proposed policy is evaluated numerically on well-known random walk mobility model. The results show strong correlation between the randomness in node mobility and that in the mobility of the data ferry. The dynamic policies computed by the proposed algorithm yield 30% more contacts than the predetermined policy, and the proposed belief sampling strategy improves the performance by 15% compared with sampling on the entire belief simplex.

Our goal in this chapter is to explore a new approach that can handle uncertainties in ferry mobility control systematically. Although the specific results are limited by the models, initial study has shown promising performance compared

with the switching policy as a benchmark, and a different belief selection algorithm. Further investigation in more practical scenarios will be highly desirable and is left for future work.

The rest of the Chapter is organized as follows. After summarizing the related work in Section 5.2, Section 5.3 presents the control framework based on POMDP. Section 5.4 formulates the problem and presents the optimal control policy. Hardness results and efficient alternative policies are presented in Section 5.5, which are evaluated numerically in Section 5.6. Finally, Section 5.7 summarizes this chapter.

5.2 Related Work

Recently, the idea of using designated mobile nodes to support communications in poorly connected networks is emerging [113–116], where a mobile backbone is constructed to cover all the task nodes if sufficient helper nodes are available [115, 116], or the helper nodes will move between task nodes as data ferries otherwise. The main assumption of existing work on data ferries is that nodes are slow-moving, or the network state is fully observable. These assumptions can be too idealistic in applications involving complicated terrains, limited visibility, and highly mobile task nodes. In contrast, we aim to explicitly model and design control policies to deal with these scenarios.

Technically, our problem belongs to the family of stochastic control problems with partial observations, first proposed in [117]. Although extensively studied in operation research and robotics, to our best knowledge its application on mobility control in communication networks has not been explored before. Recent work [118] claims to use Markov Decision Process (MDP) to select routes of data ferries, although their solution is for stationary nodes and full observations.

5.3 Control Framework

5.3.1 Network Model

Given a number of n_s disconnected subnetworks, we assume each subnetwork contains a group of nodes moving within the local regions according to certain group mobility patterns. Furthermore, we select one node per group to perform as the gateway for communications across groups. We assume that nodes have sufficient contacts within a group, and the selected group heads have sufficient storage to buffer data while waiting for contacts with the data ferry.

Each data ferry is assigned to serve multiple subnetworks, which periodically senses the presence of a group of nodes in one subnetwork within a certain sensing range, uploads the data with nodes upon contacts, buffers the data in the data ferry, and finally deliver the data to the other group of nodes in another subnetwork domain. Moreover, it has a controller which can dynamically navigate the ferry among multiple subnetworks.

The rest of this Section specifies the control framework based on the POMDP.

5.3.2 State Space and Mobility Model

To facilitate control, we partition each subnetwork field into regions, denoted as $\mathcal{L}_i = \{0, 1, \ldots, |\mathcal{L}_i|\}, \forall i = 1, 2, \ldots, n_s$, for subnetwork i, so that the data ferry is able to sense a group of nodes and exchange data with it once they are in the same region. In this chapter, we keep a minimum state space where the data ferry only remembers the region index of the subnetwork that it is trying to contact in the current slot, but not specifically which subnetwork to navigate to. Therefore, we manage to focus on the issue of stochastic control of the data ferry within each subnetwork, leaving the choice of which subnetwork to go to for the existing solutions

in the literature. Mathematically, if the current subnetwork the ferry belongs to in slot t is subnetwork i, then the state is denoted as $s_t \in \mathcal{L}_i = \{0, \ldots, |\mathcal{L}_i|\}$. Other network characteristics, such as traffic demands, QoS requirements, and buffer size constraints, are also information of interest and will be explored in future work.

We model the mobility of a group of nodes in one subnetwork i by a Markov chain on the quantized space derived from the above partition. Let $P^i = \{p^i_{jk}\}_{j,k\in\mathcal{L}_i}$ be the transition matrix for the mobility in subnetwork i, where each element $p^i_{jk} \triangleq \Pr\{s_{t+1} = k | s_t = j\}$. For simplicity, we will assume i.i.d. mobility for all groups of nodes in multiple subnetworks, and we therefore drop the superscript i in the sequel, but the framework can be easily amended for heterogeneous cases.

5.3.3 Action Space

The action a_t specifies the region the data ferry will move to in the coming time slot $t+1$, and the action space defines the set of movements feasible to the data ferry. In general, the action space is the union of all the regions a data ferry may visit across n_s subnetworks, *i.e.,* $a_t \in \bigcup_{\forall i} \mathcal{L}_i$, which may grow linearly with the number of subnetworks. As mentioned before, to keep the policy scalable, we consider a hierarchical design where the action only specifies a particular *region* of one subnetwork where should the data ferry be navigated to; however the subnetwork index is controlled by an upper layer policy. For example, one can use existing ferry route design algorithms [113] to obtain a subnetwork sequence that optimizes certain performance metrics.

Under such an hierarchical design, the action space is divided into two subsets: "follow" actions $\mathcal{A}_f = \{$"follow" 0, "follow" $1, \ldots,$ "follow" $n\}$ and "switch" actions $\mathcal{A}_s = \{$"switch" 0, "switch" $1, \ldots,$ "switch" $n\}$, where "follow" j means to navigate to region j of the current subnetwork to follow the current group of nodes, and

"switch" j means to switch to region j of a new subnetwork specified by the upper layer policy. The reason to allow for switching is that occasionally, the group of nodes may wander away from their usual regions, which will cause consecutive contact misses, and it may be more efficient to move on to other subnetworks first and revisit this subnetwork later. In the case where skipping a node is undesirable (*e.g.,* data have hard deadlines), one can simply remove the switch actions from the action space. The set of feasible policies allowing switching includes the set of policies not allowing switching, and thus the optimal performance of the former gives an upper bound of the latter. Therefore, the overall action space is given by $\mathcal{A} = \mathcal{A}_f \cup \mathcal{A}_s$.

5.3.4 Observation Model

The onboard sensor produces a binary observation per time slot. Let $o_t \in \mathcal{O} = \{0, 1\}$ denote the observation in slot t, where $o_t = 1$ means "contact successful" and $o_t = 0$ means "contact miss". Then, under the perfect sensing, $o_t = 1$ if and only if $a_{t-1} = s_t$, *i.e.,* the region the ferry decides to navigate to (one slot earlier) coincides with the region the node should move to. Note that in general, the data ferry may miss a node even if it is within the sensing range, which can be modeled by a randomized observation model $o_t = 1$ with certain probability if $a_{t-1} = s_t$.

5.3.5 Payoff Function

The payoff function represents the goal of control. Since the job of a data ferry is to ferry data among different groups of nodes in multiple subnetworks, an intuitive goal is to maximize the total number of *effective contacts*, which is defined as the *first* contact after switching the targeted subnetwork domain. Note that this effectively prevents the trivial case of being stuck with the same group of nodes in one subnetwork, since only the first of each run of consecutive contacts within the same

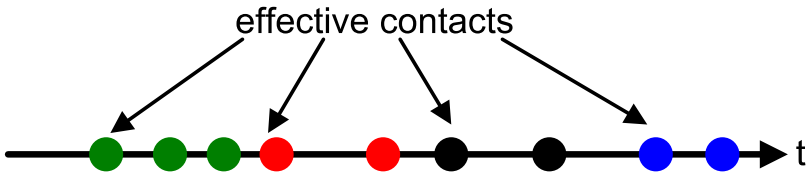

Figure 5.2: An example of the defined effective contacts. Marks of the same color represent consecutive contacts with the same group of nodes in one subnetwork.

subnetwork counts, as illustrated in Figure 5.2. Under this payoff structure, we note that it is not sufficient to only know whether the data ferry meets the targeted group of nodes or not, but we also need to know whether the contact is effective or not. Let $1_t \in \{0,\ 1\}$ denote such an indicator, where $1_t = 1$ means effective contact and $1_t = 0$ otherwise. The payoff function $r(o_t,\ 1_t)$ is given by:

$$r(o_t,\ 1_t) = o_t \cdot 1_t = \begin{cases} 1 & \text{if } o_t = 1 \cap 1_t = 1, \\ 0 & \text{if } o_t = 0 \cup 1_t = 0, \end{cases} \tag{5.1}$$

which gives unit reward if and only if there is an effective contact.

Based on the payoff function, we can calculate the cumulative payoff over a *design horizon* H, which is a period of time the data ferry is optimized upon. In this chapter, we consider the discounted reward over the horizon H as:

$$R_H \quad \triangleq \quad \mathbb{E}\left[\sum_{t=1}^{H} \kappa^t r(o_t,\ 1_t)\right], \tag{5.2}$$

where $\kappa \in (0,\ 1)$ is a discount factor specified by the application. Note that this payoff function has not taken into account the punishment. We can generalize it to include a cost for each action to study performance-energy tradeoff for instance, which will be left for future work.

5.4 Problem Statement and Optimal Policy

Given the framework developed in Section 5.3, the problem is to design a control policy for dynamic mobility control. A policy π is a sequence of mappings that map all the available information, including past actions $\underline{a}_{1:t-1}$ and observations $\underline{o}_{1:t-1}$, to a new action a_t in each time slot, *i.e.,* $\pi = \{\pi_t\}|_{t=1}^{H}$, and

$$\pi_t\big(\underline{o}_{1:t-1},\, \underline{a}_{1:t-1}\big) = a_t, \tag{5.3}$$

where we use $\underline{x}_{t_1:t_2}$ to denote the vector $(x_{t_1}, \ldots,\, x_{t_2})$ where $t_1 \leq t_2$. We further denote the expected cumulative payoff under a policy π by

$$R_H^{\pi} \;\triangleq\; \mathbb{E}\Big[\sum_{t=1}^{H} \kappa^t r(o_t,\, 1_t)\big|a_t = \pi_t\big(\underline{o}_{1:t-1},\, \underline{a}_{1:t-1}\big)\Big], \tag{5.4}$$

where the expectation is taken over all possible data ferry movements and observations.

Now, the problem is to find a policy π that maximizes R_H^{π} over all feasible policies for a predetermined design horizon H. In the following discussions, we introduce the belief updates within each time slot and the way to compute the optimal policy through value iterations.

5.4.1 Belief Updates

It is known that the sufficient statistics for the past information is the *belief vector* (also called belief state, or simply belief) $\underline{b}_t$, which is the posterior distribution of the movement of the (current) group of nodes given all their past observations, *i.e.,*

$$\underline{b}_t = \Big\{ \Pr\big(s_t = j|\underline{o}_{1:t-1},\, \underline{a}_{1:t-1}\big) \Big\}_{j\in\mathcal{L}}. \tag{5.5}$$

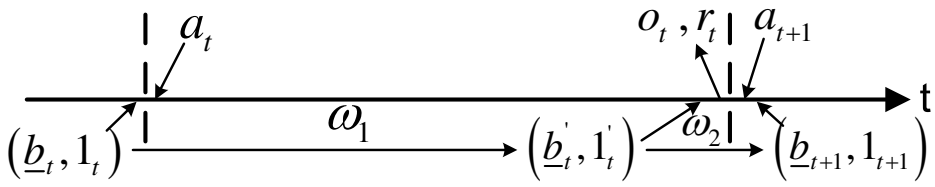

Figure 5.3: The order of action, state transition, and observation during one time slot (*i.e.,* one sensing period).

Enriched with the indicator 1_t, we obtain a new state $(\underline{b}_t,\ 1_t)$ which is the state the data ferry will act upon.

To understand the control process, let us zoom in to one slot starting with some initial state $(\underline{b}_t,\ 1_t)$, as illustrated in Figure 5.3. At the beginning of the t^{th} time slot, the data ferry chooses the next action based on its policy $a_t = \pi_t(\underline{b}_t,\ 1_t)$ and moves accordingly. At the same time, the group of nodes in each subnetwork also move according to its own mobility pattern represented by state transition matrix P, and they will reach a new belief state $P^H\underline{b}_t$ by the end of the t^{th} time slot. However, if the action for the data ferry at the beginning of the t^{th} time slot is to choose to switch to a new group of nodes in a new subnetwork domain, then the targeted subnetwork, or the "current group of nodes", changes, and its belief of the new group of nodes is reset to the limiting distribution $\underline{b}_0$ (assume it exists) while the indicator 1_t is also reset to 1. Combining these two cases give the state transition specified by function ω_1 as:

$$(\underline{b}_t',\ 1_t') = \omega_1\Big(\underline{b}_t,\ 1_t|a_t\Big) = \begin{cases} (P^H\underline{b}_t,\ 1_t) & \text{if } a_t \in \mathcal{A}_f, \\ (\underline{b}_0,\ 1) & \text{if } a_t \in \mathcal{A}_s, \end{cases} \tag{5.6}$$

where the fist condition exists if the action for the data ferry at the beginning of the t^{th} time slot is to "follow" the same group of node, and the second condition is to ask the data ferry to "switch" the targeted subnetwork domain.

At the end of the t^{th} time slot, the data ferry takes an observation o_t and

earns a unit reward if effective contact occurs. The expected payoff is given by

$$r_t = r(\underline{b}'_t,\ 1'_t,\ a_t) = 1'_t b'_t. \tag{5.7}$$

In addition, the data ferry will update its belief vector according to the Bayesian rule. The Bayesian update $(\underline{b}''_t,\ 1''_t) = \omega_2(\underline{b}'_t,\ 1'_t|a_t,\ o_t)$ is given by:

$$(\underline{b}''_t,\ 1''_t) = \omega_2\Big(\underline{b}'_t,\ 1'_t|a_t,\ o_t\Big) = \begin{cases} (\mathbf{e}_{a_t},\ 0) & \text{if } o_t = 1, \\ \Big([\underline{b}'_t]_{\backslash a_t},\ 1'_t\Big) & \text{if } o_t = 0, \end{cases} \tag{5.8}$$

where let $\mathbf{e}_a$ denote the unit vector with 1 at the a^{th} element and 0 elsewhere, and $[\underline{b}'_t]_{\backslash a}$ be the belief vector derived by setting the a^{th} element of $\underline{b}'_t$ to 0 followed by vector normalization. The physical meaning of this update comes from the observation o_t, where the successful contact $o_t = 1$ will give the data ferry complete knowledge of the location of the group of nodes, *i.e.,* the specific region of the subnetwork, therefore setting the corresponding element of the belief vector to 1. However, if the data ferry fails to catch the group of nodes, it will also know the target group of nodes is not in the region where the ferry is currently located, however it is not aware of the exact location of the group of nodes, and therefore normalizing the rest of the belief vector elements.

The updated state is $(\underline{b}''_t,\ 1''_t)$ then used as the new state for the next time slot:

$$(\underline{b}_{t+1}, 1_{t+1}) = (\underline{b}''_t,\ 1''_t), \tag{5.9}$$

and the whole process repeats. Note that instead of one belief update per step (which combines the Bayesian update at the end of a slot with the state transition in the next slot) as in classic POMDP, there are two updates in our problem because the overall belief update consists of two parts: the self- belief update of the data ferry based on the current action (ω_1), and the observation of the data ferry with/without

the presence of the target group of nodes (ω_2).

5.4.2 Optimal Policy and Value Iteration

Let value function $V_H(\underline{b}_t,\ 1_t)$ denote the cumulative payoff over horizon H starting from state $(\underline{b}_t,\ 1_t)$. Then the optimal value function must be the solution of the *value iteration* (VI) in the following form [119]:

$$V_H(\underline{b}_t,\ 1_t) = \kappa \max_{a_t \in \mathcal{A}} \left[r\left(\underline{b}'_t,\ 1'_t,\ a_t\right) + \sum_{o_t} \Pr\left(o_t|\underline{b}'_t,\ a_t\right) V_{H-1}\left(\underline{b}''_t, 1''_t\right) \right], \tag{5.10}$$

where the probability of contact successful is given by the value of element a_t in belief vector $\underline{b}'_t$ after the transition of mobility matrix P, and the probability of contact miss is given by that of one minus the value of element a_t in belief vector $\underline{b}'_t$, as:

$$\Pr(o_t|\underline{b}'_t,\ a_t) = \begin{cases} b'_t(a_t) & \text{if } o_t = 1, \\ 1 - b'_t(a_t) & \text{if } o_t = 0. \end{cases} \tag{5.11}$$

And the optimal policy must be the one achieving the optimal value function, *i.e.,*

$$\pi_H(\underline{b}_t,\ 1_t) = \arg\max_{a \in \mathcal{A}} \left[r\left(\underline{b}'_t,\ 1'_t,\ a_t\right) + \sum_{o_t} \Pr(o_t|\underline{b}'_t,\ a_t) V_{H-1}\left(\underline{b}''_t, 1''_t\right) \right]. \tag{5.12}$$

For infinite horizon $H = \infty$, it is known that the VI will converge as long as the chosen discount factor $\kappa \in (0, 1)$, and the limit V_∞ gives the optimal stationary policy that maximizes the long-term total discounted payoff (the proof of convergence could be found in conventional POMDP problem [114]). Therefore, we are particularly interested in stationary policies, *i.e.,* $\pi_t \equiv \pi, \forall t$, which maximizes the long-term payoff when $H \to \infty$. In later simulations, we are using finite design horizon $H = 20$ to approximate the infinite case, due to the simulation finding of policy convergence shown in Figure 5.4. It demonstrates the impact of discount

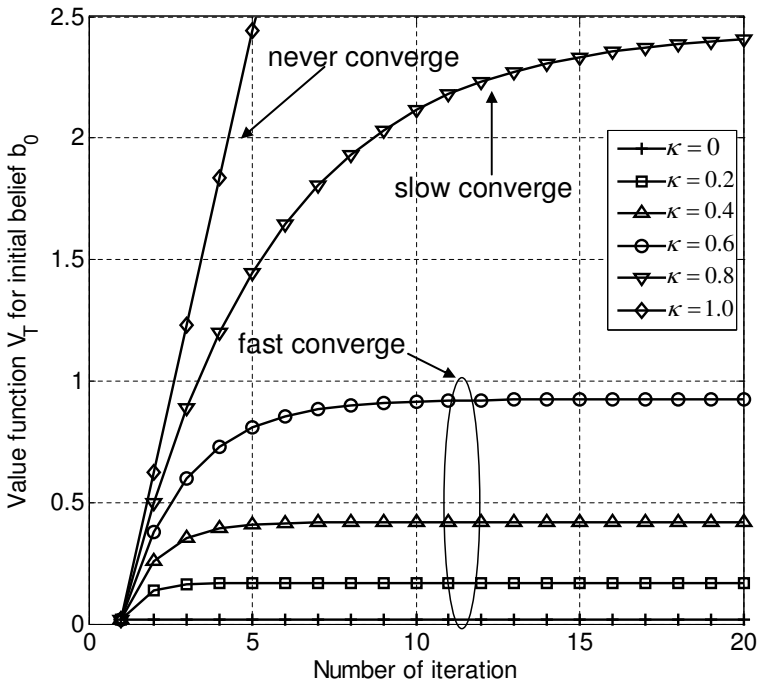

Figure 5.4: Simulation result of the impact discount factor κ on the speed of convergence of VI.

factor κ on the speed of convergence for VI. It is interesting to observe that for relatively small $\kappa < 0.5$, design horizon $H = 10$ successfully guarantees relatively fast convergence, compared with slow convergence $H = 20$ if κ is increased. Case $\kappa = 1$ yields non-convergence case of the VI because the total discounted reward becomes unbounded as design horizon increases.

5.5 Hardness Result and Efficient Heuristic Policies

It is known that to compute π_H for arbitrary initial state is PSPACE-hard [120]. Without loss of generality, we assume the initial state is set as $(\underline{b}_0, 1)$, (*i.e.,* the data ferry is aware of limiting distribution of the movement for the a group of nodes), then it can be shown that the complexity is reduced to $O(|\mathcal{A}|^H)$, where $|\mathcal{A}|$ denotes the

number of actions. This is primarily because the number of reachable states grows as $|\mathcal{A}|^{H-1}$ when value function (5.10) iterates at each step, and we need to optimize value functions over $|\mathcal{A}|$ actions. Therefore, in order to compute the optimal policy, the main difficulty comes from the fact that there is an infinite number of reachable belief vectors which grows exponentially with the length of design horizon H.

The uniqueness of the proposed state transition structure allows the computation of belief points not on the entire belief simplex, but on the belief simplex one dimension smaller, due to data ferry overstrains $o_t = \{0, 1\}$. As seen in (5.8), the successful contact will give the complete knowledge of the location of the group of nodes and the unsuccessful contact (or miss) will inform the data ferry that the target is not in the current region. Therefore, the belief simplex where belief points may possibly appear will be reduced to one dimensional smaller than the entire space.

As an illustrative example, Figure 5.5 shows a case with random walk mobility model where forward/backward parameters are set to $p = 0.1, q = 0.2$ and each subnetwork is partitioned in to 3 regions. Therefore, the belief simplex is a triangle. From the simulation, it can be seen that sampled belief points only appear on vertexes and edges of the belief simplex (however with one exception of the limiting distribution which lies in the middle), which proves that observation o_t can limit the belief points to subspaces one dimension smaller than the original simplex.

It is worth noting that the policy the data ferry would follow is computed off-line. This is primarily because that policy computation requires two major elements: one is the space quantization, *i.e.,* how each subnetwork domain is quantized to a few regions with the size of the sensing rage of the data ferry, and the other one is the selection of the belief set, *i.e.,* the *representative* beliefs need to be selected from the multi-dimensional belief simplex, which serve as the foundation for policy computation, or value iteration in particular.

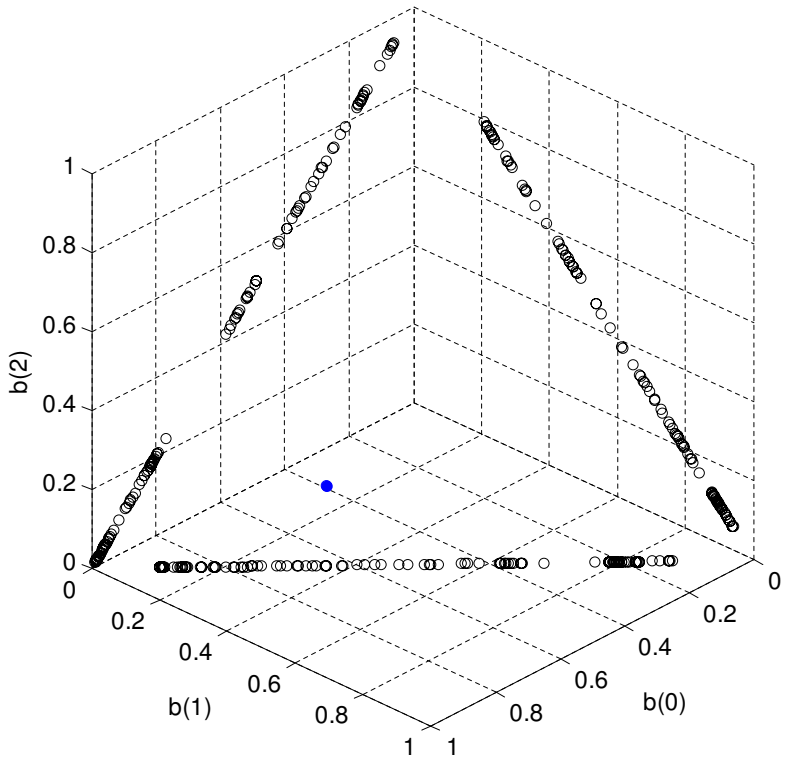

Figure 5.5: The set of reachable belief vectors in 3-D belief simplex where considered parameters are $n = 2$, $p = 0.1, q = 0.2$, and 6 iterations for VI. Legend $\circ$ represents a reachable belief vector.

5.5.1 Algorithm Description

Since the policy is computed off-line, how well this algorithm approximates the optimal policy highly depends on the selection of the belief set $\mathcal{B}$, which includes both the dimension $|\mathcal{B}|$ and the sampling mechanism for each belief vector. Ideally, $\mathcal{B}$ should represent all the reachable belief vectors during the online data ferry navigation. Nevertheless, as introduced earlier, computing the policy over design horizon H requires the value iteration of all previous value functions, *i.e.,* the number of reachable belief vectors grows exponentially over time. This imposes the big challenge for optimal policy calculation and proves PSPACE-hard [120]. Therefore, certain approximation method has to be used to select *representative* beliefs over the entire belief simplex. Grid-based algorithms [121] have been proposed which limit

VI to only a set of belief points sampled from the belief simplex. Nevertheless, it does not specify how to obtain the corresponding value function of the un-sampled belief points, which would possibly happen when the data ferry is online navigated.

To this end, we propose a *nearest-neighbour* quantization approach which approximates the values of un-sampled beliefs in (5.10) by the values of the nearest belief samples, *i.e.,*

$$\hat{\underline{b}}''_{k,o} \leftarrow \arg \min_{j=1,2,\ldots,|\mathcal{B}|-1} \|\underline{b}_j - \underline{b}''_{k,o}\|, \forall \underline{b}_j \in \mathcal{B}, \tag{5.13}$$

where $\underline{b}_j$ belongs to a predetermined set of belief points $\mathcal{B} = \{\underline{b}_0, \underline{b}_1, \ldots, \underline{b}_{|\mathcal{B}|-1}\}$ with dimension $|\mathcal{B}|$, and 2-norm is used to calculate the distance between two beliefs. Algorithm 1 shows the flow of computing suboptimal policy π_H, where ϵ is the application defined parameter for convergence condition and time index t is implied.

5.5.2 The Complexity

The complexity of the proposed heuristic VI algorithm for policy computation is relative small, due to the following reason. Given the finite design horizon $H = 20$, the value iteration will cease within finite steps, and thus although the number of belief points, and later the number of the value functions needs to be computed will grow exponentially, given the finite number of sampled beliefs, this would still be of the reasonable size. Mathematically, the complexity of the algorithm, *i.e.,* the number of basic calculations, is given by:

$$O\Big(2H|\mathcal{B}||\mathcal{A}|\Big), \tag{5.14}$$

where in the following simulations $|\mathcal{B}| = 50, |\mathcal{A}| = 5, H = 20$, the typical computation will be around 10000 times basic calculations.

Algorithm 1 : Approximated VI based on belief sampling on the reduced the simplex

Initialize: $H = 1, \epsilon > 0, \kappa \in (0,1)$, $\hat{V}_H \leftarrow 0$
for all $\underline{b}_k \in \mathcal{B}$, where $k = 0, 1, \ldots, |\mathcal{B}| - 1$ **do**
  for all $1 \in \{0, 1\}$ **do**
    for all $a \in \mathcal{A}$ **do**
      compute $\underline{b}''_{k,o}$ in (5.8) for $o = 0$ and $o = 1$
      if $\nexists \underline{b}''_{k,o} \in \mathcal{B}$ **then**
        $\hat{\underline{b}}''_{k,o} \leftarrow \arg\min_j \|\underline{b}_j - \underline{b}''_{k,o}\|, \forall \underline{b}_j \in \mathcal{B}$.
      end if
      update:

$$Q_H(\underline{b}_k,\ 1, a) = r\left(\underline{b}'_k,\ 1',\ a\right) + \sum_o p(o|\underline{b}'_k, 1')\hat{V}_{H-1}\left(\underline{b}''_{k,o},\ 1''\right)$$

    end for
    $\hat{V}_H(\underline{b}_k, 1) \leftarrow \max_{a \in \mathcal{A}} \kappa Q_H(\underline{b}_k, 1, a)$
    if $H \geq 20$ **then**
      continue in Step 3;
    end if
    if $\|\hat{V}_H(\underline{b}_k,\ 1) - \hat{V}_{H-1}(\underline{b}_k,\ 1)\| < \epsilon$ **then**
      continue in Step 3;
    else
      $H \leftarrow H + 1$
    end if
  end for
end for
Return: policy π_H.

5.6 Simulation Results

We assess the performance of the proposed POMDP model for mobile data ferry control problem by a case study, where there are $n_s = 3$ disjoint subnetworks (range 50 miles), each of which contains a group of moving nodes. Each subnetwork is further partitioned into five regions, which corresponds to five action states for the data ferry, *i.e.,* $\mathcal{A} = \{0, 1, 2, 3, 4\}$ and $|\mathcal{A}| = 5$, which will be tuned later. Design parameters include discount factor $\kappa = 0.7$ and design horizon $H = 20$. 1-D random walk [122] mobility model is used with forward/backward parameters p, q, as shown in Figure 5.6. An example of 3-state transition matrix P under this mobility model

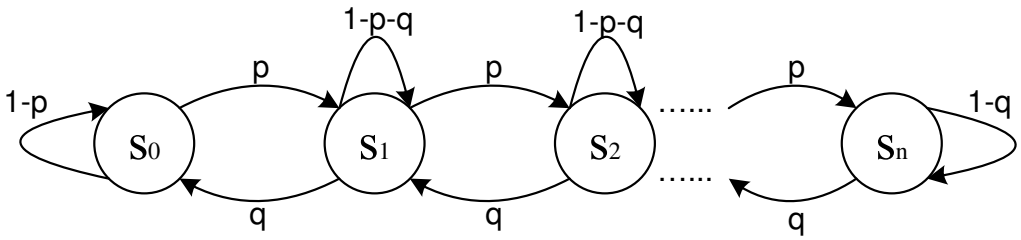

Figure 5.6: The state transition diagram for the considered random walk mobility model, with forward parameter p and backward parameter q. $\mathbf{S}_0, \ldots, \mathbf{S}_n$ denote the state of each group of nodes, or the geographic location within each subnetwork, where the data ferry should navigate to.

is like:

$$P = \begin{pmatrix} 1-p & p & 0 \\ q & 1-p-q & p \\ 0 & q & 1-q \end{pmatrix} \tag{5.15}$$

Two illustrative examples to demonstrate the real-time data ferry navigation with different mobility patterns are shown in Figure 5.7(a) and Figure 5.7(b), where mobility parameters are set differently to distinguish two special cases, (a) symmetric mobility pattern, and (b) bias mobility pattern. The more bias mobility patterns $p = 0.1, q = 0.8$ gives higher probability of contact, and the other case of $p = 0.3, q = 0.3$, the relatively symmetric mobility, leads to more ineffective navigations/misses. Furthermore, the plot exhibits positive correlation between the randomness of the movements of the group of nodes, and that of the movement of the data ferry, suggesting that the proposed controller is indeed able to control the data ferry to navigate among groups of nodes and successfully adapt to group mobility patterns.

We compare our proposed algorithm with two other benchmark policies. The first one is the predetermined switching policy, referred as "Switching policy" in the simulation, where the data ferry keeps switching among the most likely regions (which is the location with the maximum value of the limiting distribution,

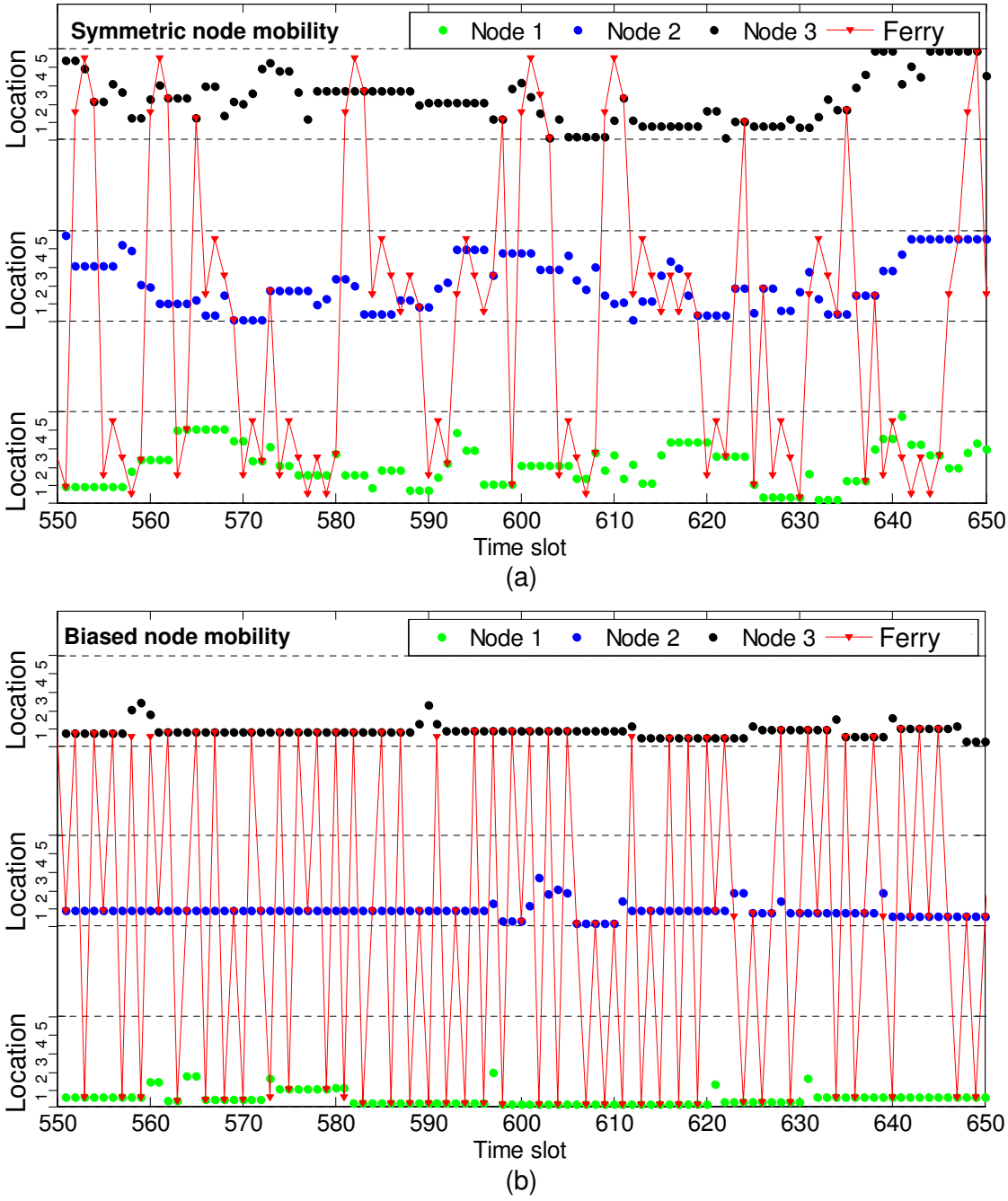

Figure 5.7: Two simulated trajectories of the data ferry, where three groups of nodes move within three disjoint 1-D subnetworks. Conspired mobility patterns include (a) $p = q = 0.3$, and (b) $p = 0.1, q = 0.8$.

or $\max \underline{b}_0$) of different subnetworks that groups of nodes belong to, regardless of the observation. The other one is to use the similar VI algorithm specified in Algorithm 1, whereas the random sampling mechanism of representative belief points on the entire simplex is enforced, referred as "Random sampling" in the simulation.

Figure 5.8 illustrates the impact of different belief sampling techniques on the policy calculation, by comparing the proposed reduced subspace belief sampling with the one on the entire simplex, together with switching policy, w.r.t. the number of samples $|\mathcal{B}|$ under two different mobility models (localized case and relatively random case). Compared with sampling on the entire simplex with no prior-knowledge on belief state, our scheme outperforms by 15% on both mobility patterns. This gain increases to 30% if compared with switching policy. The limit gain for the localized mobility pattern is 20.5% and 33.6% if our scheme is compared with the random sampling scheme and the switching policy; meanwhile, the limit gain for the random mobility pattern is 25.5% and 38.6% respectively. The reason of lower limit gain for localized mobility pattern is because of the potential benefits our POMDP model is lack of for the degree of node randomness. On the other hand, for a fixed scheme, a larger number of samples result in higher reward due to less error incurred in belief quantization, but this increase is slight which suggests the proposed policy computation algorithm is robust to the selection of representative beliefs. To achieve these gains, it is worth noting that our proposed algorithm does not require any further computation complexity compared with "Random sampling", and will be performed off-line, same as "Switching policy".

Figure 5.9 shows the impact of state partitioning on total discounted reward, w.r.t. different mobility models and the number of actions $|\mathcal{L}|$, while changing mobility parameters p, q. Two cases are compared, bias mobility pattern ($p = 0.8, q = 0.1$), and symmetric mobility pattern ($p = 0.3, q = 0.3$). For fixed geographical area, a larger number of partitioned regions, or larger $|\mathcal{L}|$, represents a

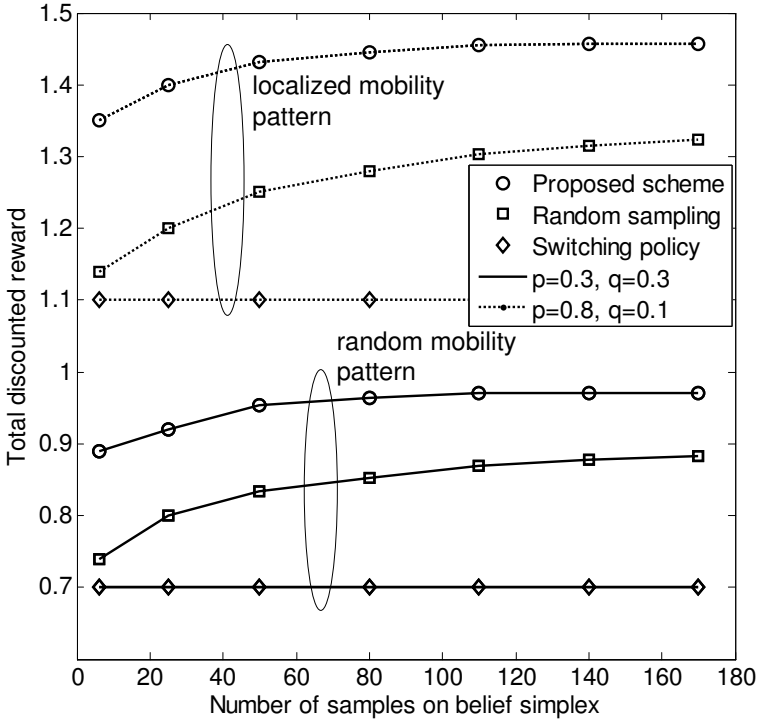

Figure 5.8: Simulation result of the impact of different sampling techniques on the belief simplex w.r.t. different mobility models and number of samples.

smaller sensing range of the data ferry; however, for fixed sensing range, it represents a larger field of subnetwork. Nevertheless, both above scenarios correspondingly increase the obscurity of data ferry navigation, or beliefs. This is because the larger dimension of belief vector, the more obscurity the data ferry will encounter if it misses the targeted group of nodes, so that the next control policy is not wise enough to make effective navigation decisions. This will decrease the value of total discounted rewards. However, if we make the mobility model more bias by using more divergent values for p, q, the decay with $|\mathcal{L}|$ greatly slows down. This is because under more bias mobility, the controller can predict the location of the group of nodes relatively more accurately within a small neighbourhood, and the size of the entire subnetwork no longer matters as much. Again, switching policy performs the worst in both cases compared with our proposed scheme.

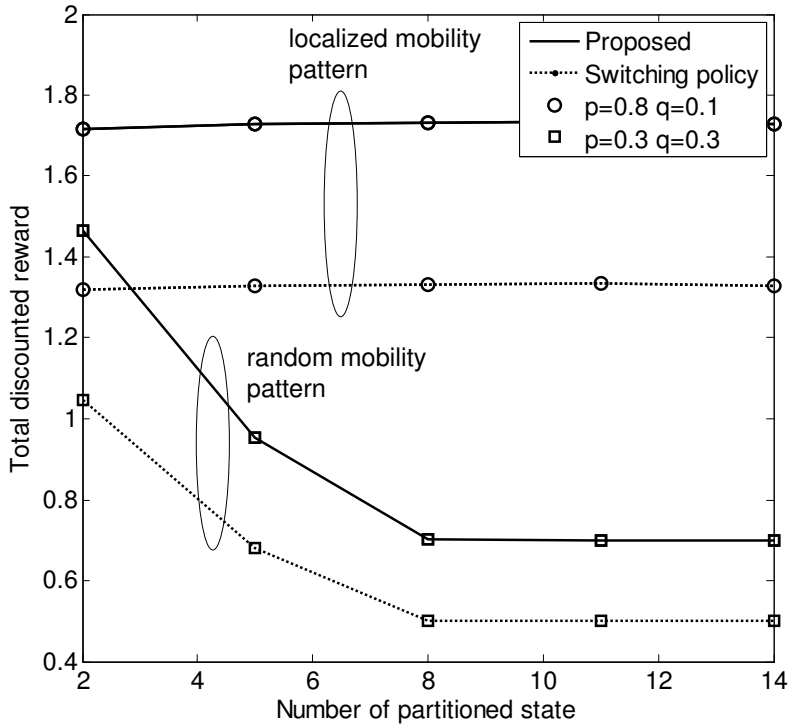

Figure 5.9: Simulation result of the impact of state partitioning w.r.t. different mobility models and the number of state partitions.

5.7 Summary

In this chapter, the problem of dynamic control of data ferries under partial observations is investigated with the goal of bridging communications between multiple disconnected mobile subnetworks. A comprehensive model of the control framework using POMDP is proposed based on which the structure of the optimal policy is studied and an efficient heuristic policy is proposed. Simulation results show that the proposed scheme achieves significantly more successful contacts when compared with the switching policy as the bottom line.

Chapter 6

Conclusions and Future Work

MULTI -hop wireless networks are usually defined as a collection of nodes equipped with radio transmitters, which not only have the capability to communicate each other in a multi-hop fashion, but also to route each others' data packets. The idea of multi-hop wireless networking is sometimes also called infrastructure-less networking, since nodes in the network dynamically establish routing among themselves to form their own network "on the fly."

6.1 Conclusions

This Ph.D. dissertation mainly investigates two important aspects of research issues for multi-hop wireless networks, namely: (1) network protocols and (2) network management. All research work have been conducted under some cross-layer design paradigms to ensure the notion of service quality, for instance the quality of service (QoS) in WMNs for backhaul applications and the quality of information (QoI) in WSNs for sensing tasks. Throughout the presentation of this Ph.D. dissertation, different network settings have been used as illustrative examples, however the proposed algorithm, methodologies, protocols, and models are not restricted in the considered networks, but rather have wide applicability, as discussed in each

Chapter.

Chapter 2 proposed a novel cross-layer design solution integrating the distributed scheduling and QoS routing algorithms, while using WMNs as an illustrative example. It has been shown in extensive simulations that the proposed approach has significant performance gain compared with conventional network protocols and other recent research outputs. This heuristic approach successfully guarantees QoS supports, and at the same time it opened up another dimension of research to perform admission control since arbitrary large number of connections are not allowed in any multi-hop wireless networks, from which Chapter 3 is motivated. Chapter 3 proposed a generic admission control methodology, where the network is modelled as a black box and potential admission impact on existing connections' QoS experience is accurately estimated. Next, this Ph.D. research extends its contribution in Chapter 4, where a negotiation-based network management framework is introduced, bridging applications' service quality demands and the network resource management, while using WSNs as an illustrative example. Finally, this dissertation extends its focus from how to maintain service quality in one single subnetwork to multiple subnetworks, when they are disconnected or even mobile. Therefore, the issue of *inter*-domain communications for multiple, disconnected, mobile subnetworks were addressed in Chapter 5 to maintain the service quality by using controlled, sensor-mounted data ferries to maximize the overall network throughput.

In conclusion, this Ph.D. dissertation focused on maintaining service quality in several cross-layer design solutions for multi-hop wireless networks, *i.e.,* the proposed models, algorithms, methodologies are not restricted in using information from a single protocol layer, but touch upon multiple layers to improve the overall design efficiencies.

6.2 Future Work

Following the investigations described in this Ph.D. dissertation, a number of research topics could be taken up; and these topics include but are not limited to:

1. For problems of the admission control algorithm, questions still remain include:

   (a) the stability and robustness issues of the proposed GAC methodology, *i.e.,* how long does it take for the system to stabilize and does the system support?

   (b) the impacts of the dynamics of the problem, *i.e.,* different network setting, on the propose GAC methodology;

   (c) the analytical model to capture the impacts of statistics feedback delay and statistics collection time on network performance, *i.e.,* borrowing ideas from the theoretical control research to quantify the feedback delay in a mathematical way.

2. For the network O&M research, questions still remain including:

   (a) to study the impacts of distributed duty-cycling policies on defining QoI network capacity and facilitating negotiation. The primary reason facilitating such research directions is because that duty-cycling changes the capacity of the network beyond that is caused by the number of tasks and their respective QoI requirements;

   (b) to design the duty-cycling policy in an intelligent manner in consider the QoI network capacity a time-varying variable, to tune the duty-cycling algorithm as a part of performing negotiation, and finally to tune the duty-cycling behavior on a spatiotemporal (stressing the "spatio-" part) basis given the QoI required by the waiting task;

(c) to explore the energy consumption v.s. the decrease of blocking probability trade-off, *i.e.,* if the sensors are with power supply, ultimately in this intruder detection scenario, the blocking probability would be 0, however this is not the case in the real world, and thus certain compromise should have to be reached.

3. For problems within the areas of inter-domain communications by mobile data ferries, questions still remain including:

   (a) to improve and analyze the proposed heuristic policies;

   (b) to analyze how far away the proposed heuristic policies can achieve compared with the optimal policy;

   (c) to extend the exiting model to capture the more actual realistic scenarios, like limited buffer size of the data ferry;

   (d) to evaluate the proposed policies on real mobility traces;

   (e) to embed other design goals beside the overall network throughput, like the delay bound, into the control framework.

All these research areas are very interesting and important as further extension to this Ph.D. dissertation, and the author would be very interested to vividly explore for the future career path.

Bibliography

[1] I. F. Akyildiz and X. Wang, "A survey on wireless mesh networks," *IEEE Comm. Mag.*, vol. 43(9), pp. S23–S30, Sept. 2005.

[2] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Comm. Mag.*, vol. 40, no. 8, pp. 102–114, Aug 2002.

[3] L. Pelusi, A. Passarella, and M. Conti, "Opportunistic networking: data forwarding in disconnected mobile ad hoc networks," *IEEE Comm. Mag.*, vol. 44, no. 11, pp. 134–141, November 2006.

[4] M. Lima, A. dos Santos, and G. Pujolle, "A survey of survivability in mobile ad hoc networks," *IEEE Comm, Surveys & Tutorials*, vol. 11, no. 1, pp. 66–77, Quarter 2009.

[5] S. Marwaha, J. Indulska, and M. Portmann, "Challenges and recent advances in QoS provisioning, signaling, routing and MAC protocols for manets," in *Australasian Telecomm. Networks and Applications Conf. (ATNAC) 2008*, Dec. 2008, pp. 97–102.

[6] L. Junhai, Y. Danxia, X. Liu, and F. Mingyu, "A survey of multicast routing protocols for mobile ad-hoc networks," *IEEE Comm, Surveys & Tutorials*, vol. 11, no. 1, pp. 78–91, Quarter 2009.

[7] R. Bruno, M. Conti, and E. Gregori, "Mesh networks: commodity multihop ad hoc networks," *IEEE Comm. Mag.*, vol. 43, no. 3, pp. 123–131, 2005.

[8] T. Braun, "Wireless mesh networks for meteorological monitoring," in *IEEE ICDCS Workshops 2009*, June 2009, pp. 425–425.

[9] H. Song, B. C. Kim, J. Y. Lee, and H. S. Lee, "IEEE 802.11-based wireless mesh network testbed," in *16th IST Mobile and Wireless Communications Summit, 2007*, July 2007, pp. 1–5.

[10] Y. Yan, H. Cai, and S.-W. Seo, "Performance analysis of IEEE 802.11 wireless mesh networks," in *IEEE ICC'08*, May 2008, pp. 2547–2551.

[11] T.-W. Wu and H.-Y. Hsieh, "Interworking wireless mesh networks: Performance characterization and perspectives," in *IEEE GLOBECOM'07*, Nov. 2007, pp. 4846–4851.

[12] P. Gajbhiye and A. Mahajan, "A survey of architecture and node deployment in wireless sensor network," in *First Int'l Conf. on the Applications of Digital Information and Web Tech. (ICADIWT) 2008*, Aug. 2008, pp. 426–430.

[13] S. Krco, V. Tsiatsis, K. Matusikova, M. Johansson, I. Cubic, and R. Glitho, "Mobile network supported wireless sensor network services," in *IEEE MASS 2007*, Oct. 2007, pp. 1–3.

[14] D. Niculescu, "Communication paradigms for sensor networks," *IEEE Comm. Mag.*, vol. 43, no. 3, pp. 116–122, March 2005.

[15] S. Dai, X. Jing, and L. Li, "Research and analysis on routing protocols for wireless sensor networks," in *Int'l Conf. on Comm., Circuits and Sys., 2005*, vol. 1, May 2005, pp. 407–411.

[16] S. Kumar, V. S. Raghavan, and J. Deng, "Medium access control protocols for ad hoc wireless networks: A survey," *Ad Hoc Networks*, vol. 4, no. 3, pp. 326 – 358, 2006.

[17] T.-J. Tsai and J.-W. Chen, "IEEE 802.11 mac protocol over wireless mesh networks: problems and perspectives," in *IEEE AINA 2005*, vol. 2, March 2005, pp. 60–63.

[18] A. Iwata, C.-C. Chiang, G. Pei, M. Gerla, and T.-W. Chen, "Scalable routing strategies for ad hoc wireless networks," *IEEE JSAC*, vol. 17, no. 8, pp. 1369–1379, Aug 1999.

[19] M. Haenggi, "Routing in ad hoc networks - a wireless perspective," in *First Int'l Conf. on Broadband Netw. (BroadNets) 2004*, Oct. 2004, pp. 652–660.

[20] M. Naghshineh and M. Schwartz, "Distributed call admission control in mobile/wireless networks," *IEEE JSAC*, vol. 14, no. 4, pp. 711–717, May 1996.

[21] A. Djouama, M. Abdennebi, L. Mokdad, and S. Tohme, "Lifetime aware admission control for infrastructure-less wireless networks," in *IEEE Symp. on Computers and Comm. (ISCC) 2009.*, July 2009, pp. 67–72.

[22] F. Didi, M. Feham, H. Labiod, and G. Pujolle, "Dynamic admission control algorithm for WLANs 802.11," in *3rd Int'l Conf. on Information and Comm. Tech.: From Theory to Applications (ICTTA) 2008.*, April 2008, pp. 1–6.

[23] O. Baldo, "A cross-layer distributed call admission control," in *IEEE WIMOB 2009.*, Oct. 2009, pp. 441–446.

[24] E. Stevens-Navarro, A. H. Mohsenian-Rad, and V. Wong, "Connection admission control for multiservice integrated cellular/wlan system," *IEEE Trans. on Vehicular Tech.*, vol. 57, no. 6, pp. 3789–3800, Nov. 2008.

[25] M. H. Ahmed, "Call admission control in wireless networks: a comprehensive survey," *IEEE Comm. Surveys & Tutorials*, vol. 7, no. 1, pp. 49–68, Qtr. 2005.

[26] B. Zhang and G. Li, "Survey of network management protocols in wireless sensor network," in *Int'l Conf. on E-Business and Information Sys. Security (EBISS '09)*, May 2009, pp. 1–5.

[27] ——, "Analysis of network management protocols in wireless sensor network," in *Int'l Conf. on MultiMedia and Information Tech. (MMIT) 2008*, Dec. 2008, pp. 546–549.

[28] V. Aseeja and R. Zheng, "Meshman: A management framework for wireless mesh networks," in *IFIP/IEEE International Symposium on Integrated Network Management, 2009*, June 2009, pp. 226–233.

[29] N. Parameswarany, S. Srivathsan, and S. S. Iyengar, "A framework for application centric wireless sensor network management," in *IEEE COMSNETS 2009*, Jan. 2009, pp. 1–7.

[30] M. S. Siddiqui, S. O. Amin, and C. S. Hong, "An efficient mechanism for network management in wireless mesh network," in *IEEE ICACT 2008*, vol. 1, Feb. 2008, pp. 301–305.

[31] H. Li and G. Chen, "Wireless lan network management system," in *IEEE Int'l Symp. on Industrial Electronics 2004*, vol. 1, May 2004, pp. 615–620.

[32] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Trans. on Inf. Theory*, vol. 46(2), pp. 388–404, March 2003.

[33] S. Eswaran, A. Misra, and T. La Porta, "Utility-based adaptation in mission-oriented wireless sensor networks," in *IEEE SECON 2008*, June, pp. 278–286.

[34] Y. Hou, K. K. Leung, and A. Misra, "Joint rate and power control for multicast sensor data dissemination in wireless ad-hoc networks," in *PIMRC 2009*.

[35] T. X. Brown, B. Argrow, C. Dixon, S. Doshi, R.-G. Thekkekunnel, and D. Henkel, "Ad hoc uav-ground network (AUGNET)," in *AIAA 3rd Unmanned Unlimited Technical Conf.*, Chicago, IL, Sept. 2004.

[36] Q. Zhang and Y.-Q. Zhang, "Cross-layer design for QoS support in multihop wireless networks," *The Proceedings of IEEE*, vol. 96(1), pp. 64–76, Jan. 2008.

[37] L.Wang and W. Zhuang, "A call admission control scheme for packet data in cdma cellular communications," *IEEE Trans. on Wireless Comm.*, vol. 5(2), pp. 406–416, 2006.

[38] S. A. AlQahtani and A. S. Mahmoud, "Call admission control scheme with gos guarantee for wireless ip-based networks," in *IEEE 61st VTC-Spring, 2005*, vol. 4, 2005, pp. 2172–2175.

[39] Z. Wang and J. Crowcroft, "Quality-of-service routing for supporting multimedia applications," *IEEE JSAC*, vol. 14(7), pp. 1228–1234, Sept. 1996.

[40] H. Jiang, W. Zhuang, and X. Shen, "Cross-layer design for resource allocation in 3g wireless networks and beyond," *IEEE Comm. Mag.*, vol. 43(12), pp. 120–126, Dec. 2005.

[41] M. Cao, X. Wang, S.-J. Kim, and M. Madihian, "Multi-hop wireless backhaul networks: a cross-layer design paradigm," *IEEE JSAC*, vol. 25(4), pp. 738–748, May 2007.

[42] I. F. Akyildiz and X. Wang, "Cross-layer design in wireless mesh networks," *IEEE Trans. on Vehicular Tech.*, vol. 57(2), pp. 1061–1076, March 2008.

[43] R. Bhatia and M. Kodialam, "On power efficient communication over multihop wireless networks: Joint routing, scheduling, and power control," in *IEEE INFOCOM*, 2004, pp. 1457–1466.

[44] U. C. Kozat, I. Koutsopoulos, and L. Tassiulas, "A framework for crosslayer design of energy-efficient communication with QoS provisioning in multi-hop wireless networks," in *IEEE INFOCOM*, 2004, pp. 1446–1456.

[45] M. Chiang, S. H. Low, A. R. Calderbank, and J. C. Doyle, "Layering as optimization decomposition: A mathematical theory of network architectures," *The Proceedings of IEEE*, vol. 95(1), pp. 255–312, Jan. 2007.

[46] M. R. Garey and D. S. Johnson, "Computers and intractability: A guide to the theory of np-completeness," *Freeman*, 1979.

[47] Z. Wang, "On the complexity of quality of service routing," *Information Processing Letters*, vol. 69, pp. 111–114, 1999.

[48] X. Yuan, "Heuristic algorithms for multiconstrained quality-of-service routing," *IEEE/ACM Trans. on Netw.*, vol. 10(2), pp. 244–256, Apr. 2002.

[49] J. M. Jaffe, "Algorithms for finding paths with multiple constraints," *IEEE Netw.*, vol. 14, pp. 95–116, Apr. 1984.

[50] P. V. Mieghem and F. A. Kuipers, "On the complexity of QoS routing," *Computer Comm.*, vol. 26(4), pp. 376–387, Mar. 2003.

[51] T. Korkmaz and M. Krunz, "Bandwidth-delay constrained path selection under inaccurate state information," *IEEE/ACM Trans. on Netw.*, vol. 11(3), pp. 384–398, Jun. 2003.

[52] Y. Zhang and T. Gulliver, "Quality of service for ad hoc on-demand distance vector routing," in *IEEE WiMob'2005*, vol. 3, Aug. 2005, pp. 192–196.

[53] C. R. Lin and J. Liu, "QoS routing in ad hoc wireless networks," *IEEE JSAC*, vol. 17(8), pp. 1426–1438, 1999.

[54] C. R. Lin, "On-demand QoS routing in multihop mobile networks," in *IEEE INFOCOM 2001*, vol. 3, Apr. 2001, pp. 1735–1744.

[55] E. Felemban, C. G. Lee, R. Boder, and S. Vural, "Probabilistic QoS guarantee in reliability and timeliness domains in wireless sensor networks," in *IEEE INFOCOM 2005*, vol. 4, Mar. 2005, pp. 2646–2657.

[56] R. Draves, J. Padhye, and B. Zill, "Comparisons of routing metrics for static multi-hop wireless networks," in *ACM Annual Conf. Special Interest Group on Data Communication (SIGCOMM)*, Aug. 2004, pp. 133–144.

[57] S. Ramanathan, "Scheduling algorithms for multihop radio networks," *IEEE/ACM Trans. on Netw.*, vol. 1(2), pp. 166–177, 1993.

[58] K. Jain, "Impact of interference on multi-hop wireless network performance," *Wireless Networks*, vol. 11(4), pp. 471–487, 2005.

[59] L. Lovasz, *Matching theory*. North-Holland, 1986.

[60] M. R. Garey, *Computers and intractability: a guide to the theory of NP-completeness*. W. H. Freeman, 1979.

[61] "IEEE std 802.11-1997 information technology- telecommunications and information exchange between systems-local and metropolitan area networks-specific requirements-part 11: Wireless lan medium access control (MAC) and physical layer (PHY) specifications," IEEE Std 802. 11-1997, Tech. Rep., 1997.

[62] "IEEE std. 802.16-2001 IEEE standard for local and metropolitan area networks part 16: Air interface for fixed broadband wireless access systems,," IEEE Std 802. 16-2001, Tech. Rep., 2002.

[63] F. Sivrikaya and B. Yener, "Time synchronization in sensor networks: a survey," *IEEE Network*, vol. 18, no. 4, pp. 45 – 50, July-Aug. 2004.

[64] K. Tabata, Y. Kishi, S. Konishi, and S. Nomoto, "A study on the autonomous network synchronization scheme for mesh wireless network," in *IEEE PIMRC'03*, vol. 1, 7-10 2003, pp. 829 – 833.

[65] A. Tyrrell, G. Auer, and C. Bettstetter, "Emergent slot synchronization in wireless networks," *IEEE Trans. on Mobile Computing*, vol. 9, no. 5, pp. 719 –732, May 2010.

[66] S. Chen and K. Nahrstedt, "Distributed qos routing with imprecise state information," in *IEEE ICCN'98*, 12-15 1998, pp. 614 –621.

[67] S. Chen, "Routing support for providing guaranteed end-to-end quality-of-service," Ph.D. dissertation, University of Illinois at Urbana-Champaign, Champaign, IL, USA, 1999.

[68] S. Chen and K. Nahrstedt, "Distributed quality-of-service routing in ad-hoc networks," *IEEE JSAC*, vol. 17, pp. 1488–1505, 1999.

[69] K. Oida and M. Sekido, "Ars: an efficient agent-based routing system for qos guarantees," *Computer Comm.*, vol. 23, no. 14-15, pp. 1437 – 1447, 2000.

[70] S. Tadrus and L. Bai, "A qos network routing algorithm using multiple pheromone tables," in *IEEE/WIC Int'l Conf. on Web Intelligence'03*, 13-17 2003, pp. 132 – 138.

[71] W. Usaha and J. Barria, "A reinforcement learning ticket-based probing path discovery scheme for manets," *Ad Hoc Networks*, vol. 2, no. 3, pp. 319 – 334, 2004.

[72] E. Gelenbe, R. Lent, M. Gellman, P. Liu, , and P. Su, "Cpn and qos driven smart routing in wired and wireless networks," *Lecture Notes in Computer Science-Performance Tools and Applications to Networked Systems*, vol. 2965, pp. 68–87, 2004.

[73] Y. Hou, "Distributed scheduling and power control for wireless mesh networks," Ph.D. dissertation, Imperial College London, U.K., 2009.

[74] Y. Hou and K. K. Leung, "A novel distributed scheduling algorithm for mesh networks," in *IEEE Globecom 2007*, U.S.A., 2007.

[75] OPNET Inc., OPNET Inc., http://www.opnet.com/.

[76] B. Sklar, "Rayleigh fading channels in mobile digital communication systems .I. Characterization," *IEEE Comm. Mag.*, vol. 35, no. 7, pp. 90–100, 1997.

[77] X. Yuan and Z. Duan, "Frr: a proportional and worst-case fair round robin scheduler," in *IEEE INFOCOM 2005*, vol. 2, U.S.A., 2005, pp. 831–842.

[78] M. El-Sayed and J. Jaffe, "A view of telecommunications network evolution," *IEEE Comm. Mag.*, vol. 40, no. 12, pp. 74–81, December 2002.

[79] U. Varshney, "Recent advances in wired networking," *Computer Journal*, vol. 33, no. 4, pp. 107–109, April 2000.

[80] H.-L. Lu and I. Faynberg, "An architectural framework for support of quality of service in packet networks," *IEEE Comm. Mag.*, vol. 41, no. 6, pp. 98–105, June 2003.

[81] J. Ratica and L. Dobos, "Mobile ad-hoc networks connection admission control protocols overview," in *17th Int'l Conf. Radioelektronika*, 2007, pp. 1–4.

[82] L. Seungjoon, G. Narlikar, M. Pal, G. Wilfong, and L. Zhang, "Admission control for multihop wireless backhaul networks with QoS support," in *IEEE WCNC*, vol. 1, 2006, pp. 92–97.

[83] G. Narlikar, G. Wilfong, and L. Zhang, "Designing multihop wireless backhaul networks with delay guarantees," in *IEEE INFOCOM 2006*, 2006, pp. 1–12.

[84] A. Herms, S. Ivanov, and G. Lukas, "Precise admission control for bandwidth reservation in wireless mesh networks," in *IEEE MASS'07*, Pisa, Italy, October 2007, pp. 1–3.

[85] F. Yu, V. Krishnamurthy, and V. C. M. Leung, "Cross-layer optimal connection admission control for variable bit rate multimedia traffic in packet wireless cdma networks," *IEEE Trans. on Signal Processing*, vol. 54, no. 2, pp. 542–555, Feburary 2006.

[86] ——, "Cross-layer optimal connection admission control for variable bit rate multimedia traffic in packet wireless cdma networks," in *IEEE GLOBECOM'04*, vol. 5, Dallas, TX, 29 2004, pp. 3347 – 3351.

[87] S. Zhang, F. R. Yu, and V. Leung, "Joint connection admission control and routing in ieee 802.16-based mesh networks," in *IEEE Trans. on Wireless Comm.*, vol. 9, no. 4, Beijing, China, April 2010, pp. 1370 –1379.

[88] R. de Renesse, V. Friderikos, and H. Aghvami, "Cross-layer cooperation for accurate admission control decisions in mobile ad hoc networks," *IET Comm.*, vol. 1, no. 4, pp. 577 –586, August 2007.

[89] Q. Shen, X. Fang, P. Li, and Y. Fang, "Admission control for providing QoS in wireless mesh networks," in *IEEE ICC 2008*, pp. 2910–2914.

[90] S.-L. Su, Y.-W. Su, and J.-Y. Jung, "A novel QoS admission control for ad hoc networks," in *IEEE WCNC'07*, Hong Kong, 2007, pp. 4193–4197.

[91] D. Ghosh, A. Gupta, and P. Mohapatra, "Admission control and interference-aware scheduling in multi-hop wimax networks," in *IEEE MASS 2007*, 2007, pp. 1–9.

[92] T.-C. Tsai and C.-Y. Wang, "Routing and admission control in ieee 802.16 distributed mesh networks," in *IFIP Int'l Conf. on Wireless and Optical Comm. Networks (WOCN '07)*, 2007, pp. 1–5.

[93] H. Zhu, V. O. K. Li, Z. Ma, and M. Zhao, "Statistical connection admission control framework based on achievable capacity estimation," in *IEEE ICC 2006*, vol. 2, June 2006, pp. 748–753.

[94] C. H. Liu, A. Gkelias, and K. K. Leung, "Connection admission control and grade of service for QoS routing in mesh networks," in *IEEE PIMRC 2008*, Sept.

[95] ——, "A cross-layer framework of QoS routing and distributed scheduling for mesh networks," in *IEEE VTC 2008 Spring*, Singapore, 2008, pp. 2193–2197.

[96] C. Perkins and E. Royer, "Ad-hoc on-demand distance vector routing," in *IEEE WMCSA'99*, San Jose, CA, USA, 1999, pp. 90–100.

[97] R. Y. Wang and D. M. Strong, "Beyond accuracy: what data quality means to data consumers," *J. Manage. Inf. Syst.*, vol. 12, no. 4, pp. 5–33, 1996.

[98] M. E. Johnson and K. C. Chang, "Quality of information for data fusion in net centric publish and subscribe architectures," in *FUSION 2005*, vol. 2, 25-28 July, pp. 8–.

[99] C. Bisdikian, L. M. Kaplan, M. B. Srivastava, D. J. Thornley, D. Verma, and R. I. Young, "Building principles for a quality of information specification for sensor information," in *FUSION 2009*, July.

[100] C. Bisdikian, J. Branch, K. K. Leung, and R. I. Young, "A letter soup for the quality of information in sensor networks," in *IEEE Inf. Quality and Quality of Service (IQ2S) Workshop (in IEEE PerCom'09)*, Galveston, Texas, USA, 9-13 March 2009, pp. 1–6.

[101] A. Tolstikov, J. Biswas, and C.-K. Tham, "Data loss regulation to ensure information quality in sensor networks," in *ISSNIP 2005*, pp. 133–138.

[102] A. Tolstikov, C.-K. Tham, and J. Biswas, "Quality of information assurance using phenomena-aware resource management in sensor networks," in *IEEE Int'l Conf. on Networks 2006*, vol. 1, Sept., pp. 1–7.

[103] Y. Zhang and Q. Ji, "Active and dynamic information fusion for multisensor systems with dynamic bayesian networks," *IEEE Trans. on Syst., Man, and Cybernetics, Part B*, vol. 36, no. 2, pp. 467–472, April 2006.

[104] A. Tolstikov, W. Xiao, J. Biswas, S. Zhang, and C.-K. Tham, "Information quality management in sensor networks based on the dynamic bayesian network model," in *ISSNIP 2007*, Dec., pp. 751–756.

[105] A. Tolstikov, C.-K. Tham, W. Xiao, and J. Biswas, "Information quality mapping in resource-constrained multi-modal data fusion system over wireless sensor network with losses," in *Int'l Conf. on Inf., Comm. & Signal Processing, 2007*, Dec., pp. 1–5.

[106] K. Henricksen and R. Robinson, "A survey of middleware for sensor networks: state-of-the-art and future directions," in *Int'l Workshop on Middleware for Sensor Networks*, New York, USA, 2006, pp. 60–65.

[107] H. Alex, M. Kumar, and B. Shirazi, "Midfusion: middleware for information fusion in sensor network applications," in *IEEE ISSNIP 2004*, Dec., pp. 617–622.

[108] W. Heinzelman, A. Murphy, H. Carvalho, and M. Perillo, "Middleware to support sensor network applications," *IEEE Network*, vol. 18, no. 1, pp. 6–14, Jan/Feb 2004.

[109] J. W. Branch, J. S. D. II, D. M. Sow, and C. Bisdikian, "Sentire: A framework for building middleware for sensor and actuator networks," in *IEEE PerSeNS'05 Workshop*, vol. 0, pp. 396–400.

[110] E. Onur, C. Ersoy, H. Delic, and L. Akarun, "Surveillance wireless sensor networks: Deployment quality analysis," *IEEE Network*, vol. 21, no. 6, pp. 48–53, 2007.

[111] S. S. Iyengar and A. Elfes, "Occupancy grids: a stochastic spatial representation for actie robot perception," *Autonomous Mobile robots: Perception, Mapping, and Navigation*, vol. 1, pp. 60–70, 1991.

[112] L. Kleinrock, *Queueing Systems: Volume* I - *Theory*. New York: Wiley Interscience, 1975.

[113] W. Zhao, M. Ammar, and E. Zegura, "Controlling the mobility of multiple data transport ferries in a delay-tolerant network," in *IEEE INFOCOM 2005*, Miami, FL, March.

[114] D. Henkel and T. Brown, "On controlled node mobility in delay-tolerant networks of unmannned aerial vehicles," in *Int'l Symposium on Advance Radio Technologies*, 2006.

[115] A. Srinivas, G. Zussman, and E. Modiano, "Mobile backbone netowkrs—construction and maintenance," in *ACM MobiHoc 2006*, Florence, Italy, May.

[116] P. Basu, J. Redi, and V. Shurbanov, "Coordinated flocking of uavs for improved connectivity of mobile ground nodes," in *IEEE MILCOM 2004*, Monterey, CA, October.

[117] E. Sondik, "The optimal control of partially observable markov decision processes," Ph.D. dissertation, Stanford University, CA, 1971.

[118] D. Henkel and T. Brown, "Towards autonomous data ferry route design through reinforcement learning," in *IEEE/ACM WoWMoM*, Newport Beach, CA, June 2008.

[119] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley, 2005.

[120] C. Papadimitriou and J. Tsitsiklis, "The complexity of markov decision processes," *Mathematics of Operations Res.*, no. 3, pp. 441–450, 1987.

[121] M. Hauskrecht, "Value-function approximations for partially observable markov decision processes," *J. of Artificial Intelligence Res.*, vol. 13, pp. 33–94, 2000.

[122] L. Lovsasz, "Random walks on graphs: a survey," *Combinatorics. Paul Erdaos is Eighty*, vol. 2, pp. 353–397, 1993.