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
February 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 relating to the efficient multi-hop wireless networks design, namely: (a) network protocols and (b) network operations and management (O&M), both in a cross-layer design paradigm to ensure the notion of service quality, such as quality of service (QoS) in wireless mesh networks (WMNs) and quality of information (QoI) in wireless sensor networks (WSNs). 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 distributed proportional-fair scheduling, QoS routing, and connection admission control algorithms, while using WMNs as an illustrative example. The proposed approach has significant performance gain compared with conventional network protocols and other recent research outputs. Second, this dissertation proposes a runtime, quantitative feedback control methodology to estimate the network capacity for any packet network, by modeling the network as a black box, where a generic admission control scheme is also developed to perform QoS control for the new connection. 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. Instead of making immediate rejection decision to the new connection, this approach allows the negotiation among different service classes to adjust their own resource utilization and adapt their service quality requirements. Finally, the guarantees of the service quality is extended to the environment of multiple, disconnected, mobile multi-hop wireless networks, where the question of how to maintain communications using dynamically controlled, unmanned data ferries is investigated.

Acknowledgment

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 leadership, support, attention to detail, and hard work have set an example I hope to match some day. Over the years, I have enjoyed the aids of several scholarships which have supported me while I completed my Ph.D. 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 with me a golden opportunity to collaborate with world-class researchers on wireless sensor networks 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.

Finally, I really thank my parents for instilling in me confidence and a drive for pursuing my Ph.D.; I own them a lot.

List of Publications

The following publication has been written during the course of this work:

- 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.

- T. He, C. H. Liu, K. W. Lee, K. K. Leung, and A. Swami, "Flying in the Dark: Dynamic Control of Mobile Gateways," submitted to *IEEE/ACM MobiHoc 2010*, Chicago, USA.

- C. H. Liu, C. Bisdikian, J. Branch, and K. K. Leung, "QoI-Aware Wireless Sensor Network Management for Dynamic Multi-Task Operations," submitted to *IEEE SECON 2010*, Boston, USA; also in *IBM Res. Tech. Rep.* RC24933, January, 2010.

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

- 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*, 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*, Hawaii, USA.

- 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*, 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*, 12-14 Oct., 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*, France, 2008, pp. 1-5.

- A. Gkelias, B. Federico, 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*, Singapore, 2008, 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 Int'l Conf. on Information Networking 2008 (ICOIN 2008)*, Busan, Korea, 2008, pp. 1-5.

Contents

List of Figures

List of Tables

Statement of Originality

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

1. A novel cross-layer design solution for QoS supports in multi-hop wireless networks is proposed, where:

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

   (b) QoS routing and connection admission control algorithms are coupled by a unified optimization criterion "QoS performance index" that combines multiple QoS constraints to indicate the QoS experience of each route.

   (c) Multi-level QoS for grade of service (GoS) is proposed, allowing network resources to be organized and used in an optimal way to maximize the network resource utilization.

2. A generic network capacity estimation and QoS control methodology is proposed for any packet networks, wired and/or wireless, completely transparent to the lower protocol layers, where:

   (a) Any particular packet network is modeled as a set of subnetworks, each of which is modeled as a "black box" for the amount of available network resources.

   (b) Subnetwork capacity is analyzed to indicate the capability the subnetwork can provide, in terms of the amount of time-varying available resources, to any connection with any QoS constraints.

(c) A generic admission control (GAC) methodology is proposed for QoS control so that preferential treatments to different service classes are allowed such that a portion of subnetwork capacity is reserved for users of higher service classes.

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-centric sensor 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-based admission control process, which iteratively reconfigures and optimizes the usage of network resources and the degree of QoI acceptance of prioritized tasks;

   (d) A resource allocation method, which optimally allocates network resources for both the running and the new tasks.

4. The design of control policies for unmanned data ferries to maintain communications among multiple, disconnected, mobile multi-hop wireless networks are proposed, where:

   (a) A comprehensive framework for the design of control logic is developed, using the tool of Partially Observable Markov Decision Process (POMDP). The framework incorporates both the prior knowledge of node movements, modeled by Markov chains on the partitioned space, and the design criteria, modeled by a payoff function with a reward structure.

   (b) The model is proposed to maximize the total number of effective contacts with exponential discounts, although other criteria can also be used.

   (c) An efficient policy computation algorithm is developed, based on the belief space quantization.

(d) It has been shown that the number of belief points can be limited to subspaces one dimension smaller than the original simplex and significantly improve the performance.

Abbreviations

AMC: Adaptive Modulation and Coding Scheme

AP: Access point

CAC: Connection admission control

GAC: Generic admission control

GoS: Grade of service

IGW: Internet gateway

IQoSR: Integrated quality of service routing

MAC: Medium access control

O&M: Operations and management

PER: Packet-error-rate

PHY Layer: Physical layer

SINR: Signal to noise plus interference ratio

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 transmission. 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 issues 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 and according to the targeted applications, multi-hop wireless networks can be further classified into three categories: mobile ad hoc networks (MANETs, [1]), wireless mesh networks (WMNs, [2]), and wireless sensor networks (WSNs, [3]).

1.1 Background Research

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 [1,4,5]. 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 that are 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 [1, 6, 7].

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 network, where the typical upgrade of wired lines to high-speed fibre networks is not always an available and/or economically attractive solution [2, 8–12]. 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 user traffic, (b) a set of guaranteed 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 connection admission control (CAC) 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.

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 [3, 13–15]. 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 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 [3, 16] 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 operations and management (O&M).

1.2.1 Network Protocols

Medium Access Control (MAC) Protocols

MAC protocols, including scheduling algorithms, are responsible for coordinating the access from active nodes [17,18]. 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 [19, 20]. Conventional routing protocols operate in a best-effort manner, where all nodes within the communication range compete for the shared medium, or network resources. No guarantees or predictions can be given here on when a node is allowed to

send. For QoS routing, it is not sufficient to only find a route from a source to one or multiple destinations. This route also has to satisfy one or more QoS constraints. 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.

Connection Admission Control (CAC) Schemes

The great deal of research attention for CAC increases significantly recently [21–26], due to the growing popularity of multimedia applications (such as voice, video, and broadband data) and the central role CAC 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, which 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 CAC scheme based on predefined criteria, such as the network loading conditions and resource availabilities. On the other hand, the heterogenous nature of the protocols to use in any multi-hop wireless networks does require a degree of transparency of the CAC scheme to lower layer protocols, so that whatever advanced technologies to use, the CAC scheme can always efficiently control the connection admission by estimating network resource availabilities.

1.2.2 Network Operations and Management (O&M)

Building usually on top of the commination protocol layers, the core functionalities of the network O&M include network planning, deployment, configuration, operation, monitoring, tuning, repairing, and changing communication networks [27–32]. The difficulties

of performing the network O&M increase dramatically along with the complexity of the network structure and stochastic traffic pattern. 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 cover hundreds of nodes in a metropolitan area. Controlling such a large-scale network is a challenging issue.

Limited Network Resources:

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 O&M 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 task dynamically come and go (which happens in most network scenarios), the resource availabilities also change 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, and/or mobile, multi-hop wireless networks, which have not direct contact due to territory obstacles or extreme scenarios. How to perform the network O&M within such context to guarantee a notion of service quality is still an open issue.

1.3 Thesis Structure

1.3.1 The Overall Structure

This Ph.D. dissertation mainly address the afore-mentioned challenges of designing efficient multi-hop wireless networks on two aspects, namely: (a) network protocol designs and (b) network O&M designs. All research work has been addressed in several cross-layer design solutions to ensure the notion of service quality, for instance the QoS supports in WMNs for backhaul applications and QoI supports in WSNs for sensing tasks.

First, Chapter 2 proposes a cross-layer framework integrating the distributed scheduling, QoS routing, and CAC algorithms, while using WMNs as an illustrative example. This research identifies the importance of the capacity estimation in any multi-hop wireless networks, which motivates the research in Chapter 3. Chapter 3 proposes a generic methodology to estimate the network capacity, or the amount of available network resources, for any packet network. QoS control under admission control scheme is also developed based on the estimated network capacity. Next, questions would still remain even though a fully cross-layer protocol design approach is adopted to guarantee QoS, due to the lack of overall design perspective, bridging applications' service quality demands (or, the external operations) and the network resource management (or, the internal operations). Therefore, the research issue of how to connect the "internal" operations of multi-hop wireless networks and the "external" aspects of service quality demands motivates the research in Chapter 4, where a *negotiation* process is proposed and evaluated on a WSN intruder-detection scenario. Finally, thinking in the complete network setting where applications may require service delivery among multiple wireless networks, Chapter 5 is thus motivated to research the issues and challenges of maintaining communications among *multiple*, *disconnected*, *mobile* network components. The question of how to bridge communications using dynamically controlled, unmanned data ferries to maximize the overall throughput is investigated. Details of the motivations for each Chapter is introduced in the following descriptions.

1.3.2 Chapter Motivations and Connections

Within the context of multi-hop wireless networks, 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 (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 transceivers but they can also generate new bottlenecks in several routes in the network. Moreover, the imprecise estimation of the impact of the newly admitted connections on existing ones running in the network may jeopardize all connections' QoS, etc.

These are primarily why a *cross-layer* design approach is highly needed and where Chapter 2 is motivated. In Chapter 2, a heuristic low-complexity cross-layer framework is proposed, including a CAC scheme, a multi-constrained QoS routing algorithm, and a distributed proportional-fair scheduling algorithm. The solution aims to tackle the aforementioned challenges and provide QoS support. Directional antenna model with adaptive modulation and coding (AMC) schemes have been considered in the PHY layer while channel prediction in different time-scales is included to assist and guide the optimal operations of the overlying layers and algorithms.

Chapter 3 is motivated by the well-known difficulties of estimating the network capacity, or the amount of available network resources, from which the acceptance of the new connection can be judged to enforce the QoS control. Network capacity is one of the key parameters to design an efficient network architecture for QoS provisions that has different interpretations at different layers and different networks. Nevertheless, it is known

to be difficult to estimate in multi-hop wireless networks [33] and conventional IP networks. 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 channel fluctuations, the uncontrollable admission of the improper new connection can highly affect the resources of adjacent transmissions. Precise knowledge of the available network capacity will allow the network operator to perform optimal admission control to the new connection without jeopardizing the proper operations of the existing connections in terms of their QoS.

In order to estimate the network capacity and perform QoS control, Chapter 3 make three contributions. One, any packet network is generically modeled as a set of subnetworks, each one of which is considered as a "black box". The ingress node aggregates traffic as the input to the black box with parameters of traffic statistics and QoS requirements, and an egress node serves as an output from the black box with a single defined parameter called *QoS performance index*. Two, to facilitate the QoS control, the *subnetwork capacity* between any pair of ingress and egress nodes is analyzed, to indicate the amount of time-varying available resources the subnetwork can provide to any connection with any QoS constraints. These resources can be interpreted as a function of the maximum cardinality of the connection set, the maximum supportable throughput, the minimum supportable packet delay, etc. Three, to allow preferential treatments to different service classes, the portion of the subnetwork capacity is reserved for higher service class users. Without loss of generality, we show that the proposed capacity estimation and QoS control methodology has wide applicability to any packet networks, wired and/or wireless, and we discuss various feasibility issues like connection throughput versus subnetwork capacity, statistics feedback delay, and statistics collection time.

Questions would still remain even though a cross-layer solution is adopted from protocol design perspectives to guarantee the service quality. This is due to the lack of overall design framework bridging applications' service quality demands (or, the "external" operations) and the network resource allocations (or, the "internal" operations). In other

words, how to optimally manage the network resources to satisfy the applications' service quality requirements (bottom-up), and how to adapt the applications' service quality demands according to different service grades to fit in the network status (top-down), should be addressed 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[1] in WSNs, rather than the traditional QoS for backhaul applications.[2]

Chapter 4 is thus motivated by the significant research need of network O&M for WSNs, which so far focuses primarily on the "internal" aspects of WSNs such as energy-efficiency, coverage, routing topologies for efficient data transportation, etc. The complementary area that considers the "external" relationships that WSNs have with the service quality requirements of the applications they support have experienced significantly less exposure. Existing approaches strive to achieve desirable network operations by fine tuning both statically and dynamically configurable WSN resources, such as traffic flows, routing paths, transmission power, to maximize a network utility [34, 35] curve that is assumed to be known as *a priori*. However, *a priori* knowledge of the utility functions is very challenging, or even more challenging if the utility comes to represent the entire network's behavior. These challenges are further compounded with the multi-dimensionality of QoI attributes for the varying needs of on demand tasks, and when considering the time-varying radio, energy, and other network resource conditions, along with the stochastic nature of the task arrival and departure processes.

To address these challenges, we separate the process of calculating the QoI performance of the network at large from that of calculating utility resulting from allocating network resources to individual tasks. 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 the *a priori* knowledge of utility functions and

[1] or simply tasks to use in the rest of the dissertation

[2] QoI and QoS may have slightly different interpretations for their targeted applications, but they both focus on the broader aspects of service quality.

facilitates the dynamic accommodation of tasks with heterogenous 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. Third, an adaptive, *negotiation*-based admission control mechanism is proposed to dynamically configures the usage of network resources to best accommodate all tasks' QoI requirements.

Chapter 5 is motivated by maintaining communications among multiple disconnected mobile network components, or the inter-domain communications. 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 helper nodes mounted on controllable mobile platforms such as UAVs [36]) to assist the 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 navigate it among sensing points. Periodically, the data ferry senses the presence of nodes and uploads/downloads messages 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, this dissertation starts from the protocol designs including the integrated distributed scheduling, QoS routing, and admission control algorithms, and further investigates the admission control the QoS control by estimating the time-varying network capacity. Then, this dissertation aims to further improve the design efficiency by proposing a negotiation framework among applications' service quality demands and the network resource management, or the external and the internal operations respectively. Finally, to maintain the service quality among multiple disconnected network domains, this dissertation explores the design issues of inter-domain communications using unmanned data ferries. This dissertation has shown extensive simulations 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

Cross-Layer Protocol Designs to Support QoS

IN this Chapter, this dissertation 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 connection admission control, 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 [37]. 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 [2]. The destination can be an internet gateway 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 of all, they must meet the high capacity needs of the access nodes that have to forward the accumulated traffic of their underling users. Furthermore, they have to cope with multiple strict QoS requirements of the end user applications. QoS requirements can be divided into different groups, *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 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,

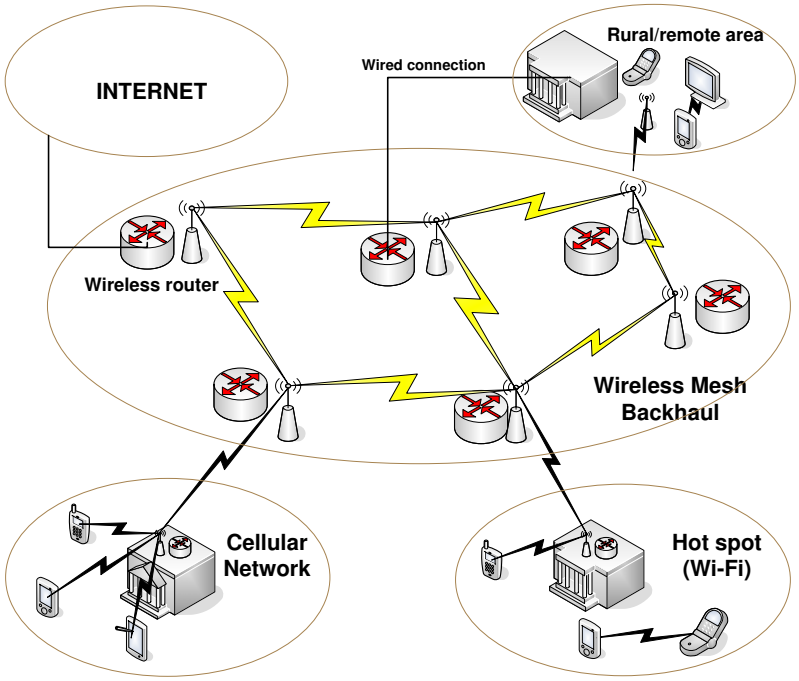

Figure 2.1: A typical wireless mesh network scenario.

and packet error rate (PER). Throughput (concave) denotes the amount of traffic along a certain route and is limited by the link with the lowest throughput along this route. Delay (additive) indicates the time between sending out a packet from the source node and 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]. In this paper, we consider three QoS constants, *i.e.,* ETE packet delay, throughput, and PER. Finally they 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),

[1]Besides these constraints, there are other interesting metrics for WMNs. The number of hops (additive) represents the number of links in a path. Jitter (additive) denotes the variation between expected and actual reception time of a packet. Energy (additive) takes the energy needed to send a packet from source to destination into account. Alternatively, energy can also be handled as a concave metric, where a (mobile) node has to provide a certain energy level, to be considered as part of a route. Further QoS metrics include, *e.g.,* , signal strength (concave) and distance (additive). QoS routing protocols utilize subsets of these metrics. In many cases, only single metric like bandwidth or delay or specific groups of metrics, *e.g.,* additive metrics, are taken into account.

routing and connection admission control algorithms.

In traditional cellular network settings, the grade-of-service (GoS) has been a fundamental parameter to define the quality of voice services [38,39], 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 QoS connection. Therefore, the GoS threshold can highly affect the connection admission control scheme by controlling the number of connections that can be allowed at each level. The GoS is usually closely related to the billing system of the telecommunication service provider since higher GoS can be obtained for premium users at a higher cost.

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 *sub-optimum* for the overall network performance. For instance, bad resource scheduling in MAC layer can lead to interference that affects the performance 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. Moreover, imprecise estimation of the impact of newly admitted connections on existing ones running in the network may jeopardize all connections' QoS.

These are primarily why cross-layer design for improving the network performance has been a focus of much recent work. In a cross-layer paradigm, the joint optimization of control over two or more layers can yield significantly improved performance. Caution needs to be exercised though, since cross-layer design has the potential to 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. For instance, additive (cost, delay, jitter, etc.), multiplicative (PER and path break probability), and concave (throughput, etc.) metrics have to be jointly taken into account which has been proven NP-complete [40]. Therefore, most routing algorithms that consider multiple constraints do not try to find the optimal path but rather any path satisfying all constraints.

In this Chapter, we propose a cross-layer design paradigm to support QoS in WMNs that includes a connection admission control scheme together with 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 (*i.e.,* delay, throughput and PER) in a unified utility function.

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

3. A tightly coupled design framework, combining routing and admission control, is proposed where a unified optimization criterion "QoS performance index" that combines multiple QoS constraints to indicate the QoS experience of each route is used. The proposed connection admission control scheme is fully distributed and capable of estimating the impact of new connections on existing connections' QoS to strictly prevent the new connection consuming much network resources. Meanwhile, multi-level QoS allows network resources to be organized and used in an optimal way to maximize the network resources utilization.

The remainder of the Chapter is organized as follows. After summarizing the related work in Section 2.2, the cross-layer 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 connection admission control scheme with both QoS and GoS provisions 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 [41–45] that generally includes two aspects of design methods: theoretical mathematical modeling and practical protocol design.

Layered protocol architecture is one of the most important factors that has 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 [46] 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. On the other hand, 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 MAC and routing protocols exchanges information in packet level like delay, SINR, PER etc. and connection level QoS requirements, but at the same time, routing and connection admission control algorithms are determined by one single design criterion "QoS performance index" (tightly coupled). Due to optimization execution across layers, we can expect that a better performance improvement can be achieved by both the loosely and the tightly coupled cross-layer designs than only one of them is used. Furthermore, the advantage of adopting both schemes for cross-layer design is that it 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. [47, 48] have addressed extensively on multi-constrained QoS routing algorithms in wired network based on network state [49, 50] to overcome the NP-complete difficulties of providing optimum routes that guarantee multiple QoS constraints [40]. Meanwhile, QoS routing algorithms for wireless ad-hoc networks have been previously explored in [51–55]. However, they either overlook the multi-hop queueing 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 [56–58] 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 adhoc networks. Moreover, the election-based scheduling algorithm specified in the IEEE 802.16 standard [62] or [63] for multi-hop mesh networks are some other scheduling schemes. 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.

As for connection admission control, much research work has been done in ad hoc wireless networks [64], as well as in WMNs. [65] proposed a algorithm to support rate and delay requirements, but it assumed no channel fading and co-channel interference among wireless links, and uses a tree-structure [66] MAC scheduling. [67] addressed a similar problem while contention-based MAC protocol is used to derive the bandwidth estimation for the new connection. In [68], 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 [69]. 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. [70] makes admission decision by estimating the achievable capacity between any pair of ingress and egress nodes with only packet loss constraint, assuming traffic arrive according to Gaussian distribution.

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 K_i one-hop neighbors within fixed transmission range, where these neighbors are denoted as $\{k = 1, 2, ..., K_i\}$. Each node maintains separate queues for each direction of transmission, and newly arrived packets will be placed into the corresponding queue according to the pre-determined route that they belong.

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. 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. Power control is not considered in this phase, *i.e.,* all the nodes have the same fixed transmission power.

Each WMR independently generates traffic, or connections, according to the Poisson distribution. 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 subscript r denotes the *required* value. We denote this QoS requirement set as (D_q^r, T_q^r, E_q^r). Let π_{sg} further denote the route set from a source WMR s to a particular IGW g. A route π_{st}^k from a source WMR with index s to a destination IGW indexed g within the route set π_{sg} is concatenated by a set of links $\{(v_i, v_j)\}$, for all $v_i, v_j \in V_R \bigcup V_G$. 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,

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

where $k = 1, 2, \ldots, m$. 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 the problem of providing optimum routes that guarantee multiple QoS constraints has been proven to be NP-complete [40]. Therefore, in order to facilitate the information delivery and exchange (loosely-coupled cross-layer design) 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 ration between the *attained* parameter measurement and the requested value. 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 $\mathrm{R}_q^D(k)$ on route π_{sg}^k is defined as the actual delay measurement, $\sum_{(i,j)\in\pi_{sg}^k} D_{ij}^a$, over the QoS delay requirement D_q^r, *i.e.,*

$$\mathrm{R}_q^D(k) = \frac{\sum_{(i,j)\in\pi_{sg}^k} D_{ij}^a}{(1-\beta_D)D_q^r}, \tag{2.3}$$

where subscript a denotes the *attained* value.

Throughput Outage Ratio

For connection q, throughput outage $\mathrm{R}_q^T(k)$ on route π_{sg}^k is formulated as the ratio between the throughput requirement T_q^r and actual *bottleneck* link throughput, $\min_{(i,j)\in\pi_{sg}^k} T_{ij}^a$, *i.e.,* the minimum of all one-hop throughputs along route π_{sg}^k, as:

$$\mathrm{R}_q^T(k) = \frac{(1+\beta_T)T_q^r}{\min_{(i,j)\in\pi_{sg}^k} T_{ij}^a}, \tag{2.4}$$

where subscript a denotes the *attained* value.

PER Outage Ratio

For connection q, PER outage $\mathrm{R}_q^E(k)$ on route π_{sg}^k is defined as the multiplication of all one-hop error rate, $1-\prod_{(i,j)\in\pi_{sg}^k}(1-E_{ij}^a)$, over PER requirement E_q^r, since this is a multiplicative constrain, *i.e.,*

$$\mathrm{R}_q^E(k)=\frac{1-\prod_{(i,j)\in\pi_{sg}^k}(1-E_{ij}^a)}{(1-\beta_E)E_q^r}, \tag{2.5}$$

where subscript a denotes the *attained* value.

A resource reservation margin factor has been introduced as β_D, β_T and β_E for delay, throughput and PER respectively. In other words $\beta_{(\cdot)}$ represents the additional resources that we reserve beyond the QoS requirements in order to provide a safe guard for imperfect resource estimations, and the system/wireless channel fluctuations. This is a free parameter that can be defined and modified by the network operator, or administrator, based on the network requirements. Some results and discussions on the impact of the parameter $\beta_{(\cdot)}$ on the QoS outage probability, channel resources and connection blocking probability is given in [71].

Since a connection has to fulfil the set of QoS requirements, a source-to-gateway route will be feasible if and only if all defined outage ratios are less than one, *i.e.,*

$$\Big(\mathrm{R}_q^D(k),\mathrm{R}_q^T(k),\mathrm{R}_q^E(k)\Big)\leq 1. \tag{2.6}$$

However, 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

$$1_p=\begin{cases}1 & \text{if parameter } p \text{ is critical for connection } q,\\ 0 & \text{otherwise.}\end{cases} \tag{2.7}$$

An example of the resource reservation margin factors and indication functions chosen for three types of traffic in the network, namely, voice-over-IP, interactive-video

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

	voice-over-IP	Interactive-video	Broadband Data
$1_D, \beta_D$	1, variable	1, variable	0, —
$1_T, \beta_T$	1, variable	1, variable	1, variable
$1_E, \beta_E$	0, —	1, variable	1, variable

and broadband data services respectively, is demonstrated in Table 2.1.

2.4.1 QoS Route Selection

Our multi-constrained *QoS performance index* for route π_{sg}^k can be formulated as:

$$\mathrm{I}_{sg}^k(q) = \max\left[1_D \mathrm{R}_q^D(k), 1_T \mathrm{R}_q^T(k), 1_E \mathrm{R}_q^E(k)\right], \tag{2.8}$$

and the proposed multi-objective routing decision function in order to take an optimum heuristic decision is given by,

$$k^* \longleftarrow \min_{\forall \pi_{sg}^k \in \pi_{sg}} \mathrm{I}_{sg}^k(q), \tag{2.9}$$

where route $\pi_{sg}^{k^*}$ is chosen. In other words, we are choosing the route with the minimum overall QoS outage probability.

2.4.2 Routing Procedures

Routing discovery phase requires each receiving node on one side of edge (v_i, v_j) records one-hop delay, link throughput and PER information, which are to be used later. 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, at given time, WMR 1 serves as the source. It generates a request packet *REQ* containing the underlying QoS constraints, and a clock before sending it through the allocated time slot in the control frame to its one-hop neighbors. Before this clock expires, if WMR 1 does not receive a reply message *REP*, it will regenerate

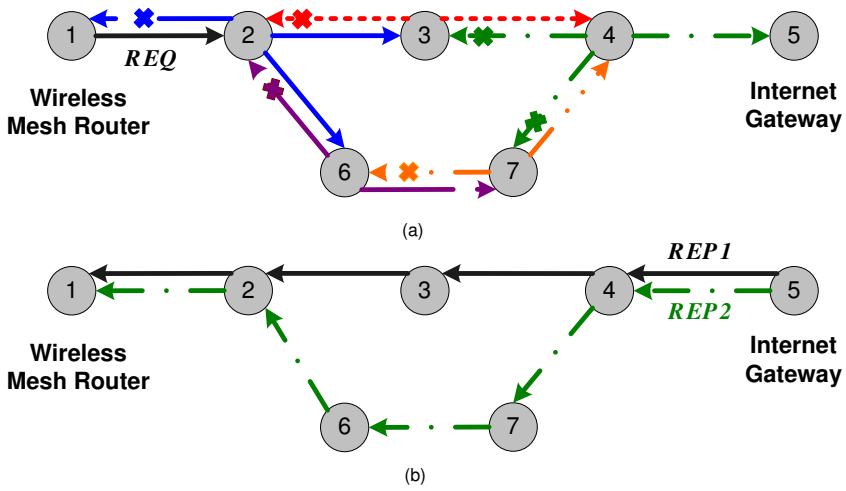

Figure 2.2: An example of the proposed QoS routing algorithm discovery procedures. (a) WMRs send REQ packets to their immediate neighbors, and (b) Gateway node sends REP packets back to source through the routes just found.

a request packet and broadcasts it to the whole network due to possible packet loss. In Figure 2.2(a), when WMR 2 receives *REQ*, it averages previous one-hop delay D_{12}^a, link throughput T_{12}^a and PER E_{12}^a measurements, which are recorded in WMR 2, then piggybacks this information in *REQ* and send it out to WMR 1, WMR 3 and WMR 6 through the allocated time slots in control frame. Nevertheless, only WMR 3 and WMR 6 need to forward this message to their neighbors after piggybacking D_{ij}^a, T_{ij}^a and E_{ij}^a for corresponding link (v_i, v_j) in *REQ* packet.

All other nodes in the network repeat these procedures until the gateway WMR 5 receives the request message *REQ*. Afterwards, the reply procedure is initialized in Figure 2.2(b). WMR 5 sends a reply packet *REP* back through two different routes to WMR 1, *i.e.*, $5 \to 4 \to 3 \to 2 \to 1$ and $5 \to 4 \to 6 \to 7 \to 2 \to 1$. By calculating the QoS performance index in (2.8), WMR 1 chooses the best route obtained before the clock expires. It is also worth noting that WMR 1 does not need to wait for all reply messages come back, not only because of the exponential number of possible routes even for reasonable network size, but also we only need to meet certain QoS requirements but unnecessarily find the optimal one if generating too much overheads. This is controlled by the clock in source router WMR 1.

2.5 Distributed Opportunistic Proportional Fair Scheduling Algorithm

[2] 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 duplexing mode (transmitting or receiving) but also the specific direction (to which neighbor) of the next communication in an opportunistic manner. 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 in link capacity of WMNs due to randomness of lower-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 utility function, or scheduling metric, which not only achieves opportunistic gain but also supports quality of service as committed by the routing algorithm in use. Otherwise, the QoS promised by the routing protocol to its applications cannot be guaranteed.

2.5.1 Utility Definition

The proposed co-operation between the scheduling and routing algorithms is in a "request-enforce" manner. Multi-constrained QoS routing algorithm introduced in Section 2.4 needs to estimate the future long-term link capacity as it is crucial in order to maintain an

[2]For the completeness of this dissertation and for the purpose of enforcing the overall understanding of the proposed cross-layer approach, this section of research on the distributed scheduling algorithm is necessary and thus referenced from Dr. Yun Hou's Ph.D. dissertation made in May 2009 at Imperial College London, U.K. However the author of this dissertation does not claim as his own contribution.

effective statistics table for various source and destination node pairs. As a result, it is desirable for the routing layer to specify a *target* throughput allocation among links on each node along a route 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 algorithm is to achieve the relative target throughput for each link scaled by a per-node (not per-link) proportionality constant, or, "soft" QoS. Thus achieving the relative target by the proposed scheduler effectively yields the actual throughput target.

Since the scheduling framework is fully distributed, we 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 K_i neighboring nodes, it has maximum $2K_i$ candidate links to schedule in every time slot. The routing algorithm periodically estimates the throughput demand on each link associated with each node in the next time frame, and provides the scheduler with a target throughput allocation $\underline{a}_i = (a_i(1), a_i(2), \ldots, a_i(2K_i))$[3] to achieve the desired QoS. In our proposed QoS routing algorithms, routing demand $a_i(k)$ associated with with link (i,k) is computed as,

$$a_i(k) = \sum_{\forall q} (1+\beta_T) T_q^r, \text{ if } (i,k) \in \pi_{sg}^{k^*}(q), \tag{2.10}$$

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

Then our goal here is to define an appropriate utility with which the scheduler's allocation of the long-run throughput $\underline{\phi}_i = (\phi_i(1), \phi_i(2), \ldots, \phi_i(2K_i))$ for all links is proportional to the target allocation $\underline{a}_i$, *i.e.*, $\underline{\phi}_i^* = c_i \underline{a}_i$, where c_i is a positive and proportional constant for node i and $\underline{\phi}_i^*$ is the *optimal* solution for node i. [72] proved that if the optimization problem for each node i is to maximize the objective function $f(\underline{\phi}_i)$ as,

$$\max_{\underline{\phi}_i} f(\underline{\phi}_i) = \max_{\underline{\phi}_i} \sum_{k=1}^{2K_i} a_i(k) \log \phi_i(k) \tag{2.11}$$

[3]The underlined notation signifies a vector quantity in this Chapter.

such that,

$$\sum_{k=1}^{2K_i} \phi_i(k) \leq C, \tag{2.12}$$

Then the optimal solution $\underline{\phi}_i^* = (\phi_i^*(1),_i^* (2), \cdots, \phi_i^*(2K_i))$ is directly proportional to $\underline{a}_i = (a_i(1), a_i(2), \ldots, a_i(2K_i))$ element by element. Correspondingly, the optimal solution $\underline{\phi}_i^*$ for the optimization problem is proportional to the target throughput allocation $\underline{a}_i$. In other words, the scheduling utility (or metric) for all link (i,k) from 1 to $2K_i$ of node i is,

$$M_i(k) = a_i(k)\frac{\rho_i(k)}{\phi_i(k)} \tag{2.13}$$

where $\rho_i(k)$ is the instantaneous supportable data rate for link (i,k) and $\phi_i(k)$ is the long-time average of $\rho_i(k)$. $\rho_i(k)$ is calculated from Shannon's capacity formula,

$$\rho_i(k) = W \log(1 + \beta_{ik}^t \gamma_{ik}^t) \tag{2.14}$$

where W is the system bandwidth, γ_{ik}^t is the receiving SINR and β_{ik}^t captures the unpredicted interference effects.

2.5.2 Distributed Framework

Since it has been shown that a collision free method for utility exchange is feasible, we assume here that utility values of both incoming and outgoing links is available to the node, and the two ends of each link keep the same latest utility value to make scheduling decisions. Figure 2.3 shows an illustrative example to show the distributed framework given certain utility functions (as shown the values on top of each arrow), where node 5 drop the incoming link from node 4 to avoid potential collision to schedule the outgoing link to node 1 since the latter obtains the higher link utility. The detailed distributed framework is summarized as follows.

The first stage of the framework for each node is to choose the link with the highest utility among all the incoming and outing links to activate for the next time frame. Then in the ideal case, $N/2$ links with the highest utilities will be chosen to activate in an N-node WMN.

However, the main difficulty in implementing this idea in a *distributed* way is the possibility that a node makes a decision conflicting with neighbors in terms of duplexing mode. It is difficult to improve the scheduling on all the nodes in the network in order to find a conflict-free solution that yields the best performance because fundamentally with a distributed algorithm, nodes have no prior knowledge about its neighbors' duplexing status at this decision making stage. Therefore, we retain the conflict-free decisions, and add one round of control exchange to solve the conflicts locally. Simply, our solution is to exchange the initial decision made among neighboring nodes and let nodes with a collided destination give up the intended transmission

A formal description of our distributed scheduling framework is as follows. It is composed of two control phases:

Utility Exchange and Initial Decision Making:

Each node exchanges the utility function of each of its incoming and outgoing links with its neighbors. After that, each node chooses the link with the best utility to be the initial decision of the next transmission.

Initial Decision Exchange and Final Decision Making:

Each node exchanges the initial decision to all its neighbors, including the IDs of the associated transmitting (origin) and receiving (destination) nodes. Based on the initial decision exchanges, each node with an initial decision of "transmit" checks if the desired receiving node is having the same "transmit" initial decision. If so, the node gives up the intended transmission. Otherwise, the node starts transmission in that direction in the next slot. Each node with an initial decision of "receive" also find outs the best transmitter based on the initial decision exchanges, and configures its physical layer to receive data from that direction in the next time slot.

To sum up, the proposed framework has demonstrated following merits:

1. It is fully distributed without deadlock. Nodes make scheduling decisions simulta-

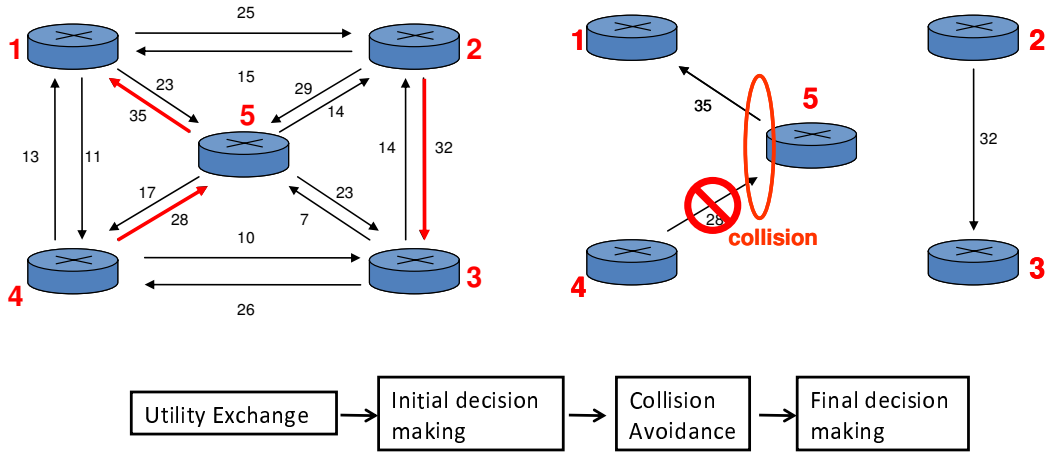

Figure 2.3: An illustrative example to show the distributed scheduling framework given certain utility functions (as shown the values on top of each arrow), where node 5 would drop the incoming link request from node 4 to avoid potential collision from the outgoing link to node 1, since the latter obtains the higher link utility.

neously, and do not need to wait for other nodes' decisions to make its own decision.

2. It exploits multi-user diversity. Although in mesh networks, it is very likely that the fluctuation of wireless links is weak, the multi-user diversity can be realized with other aspects such as differences in propagation loss (with random node layout), independent incoming and outgoing channel qualities and dynamic interference.

3. It tends to generate smooth interference, compared to random schedulers. Since the scheduling decisions are related to the instant utility, as long as the utility function is with strong time coherence, the link schedule shall generate interference with reasonably strong temporal correlation.

2.6 Connection Admission Control Algorithm

Researchers have so far developed various admission control schemes to provide decisions on connection admission before routing discovery is performed. This is of critical importance because newly admitted connections will change the traffic conditions across the network that will affect the co-channel interference after the scheduling decision is made,

and thus the signal quality of the existing links may deteriorate. Therefore, the resource allocation decisions among all connections have to be altered accordingly. The impacts of such changes on the existing traffic and the overall network performance have not yet been well studied in the literature, considering time-varying wireless channel conditions, multiple QoS requirements among different connections, etc.

Moreover, the distributed, proportional-fair scheduler proposed in Section 2.5 has been proven a promising technology enabler for WMNs since it can take advantage of the multi-user diversity and the dynamic nature of the wireless channel. However, it comes with a certain drawback, *i.e.,* while it maximizes the overall network throughput it cannot perform hard resource reservation that is required to provide strict QoS. This has as a result of an increased outage probability of the ongoing QoS connections. Therefore, a scheme is required to provide connection admission control to new connections by predicting their impact on the QoS of the connections already running in the network.

2.6.1 Admission Estimation

In our proposed admission control model, every WMR in the network keeps tracks the statistics of each packet going through each particular route. For instance, consider a node s, serves as the source expected to route data to the gateway node g, where m number of candidate routes exist between $s-g$ pair. Meanwhile, some connections started with source s have already finished transmission through different routes within the route set π_{sg}. We keep the updated information of QoS performance index values for each route $\pi_{sg}^k \in \pi_{sg}, \forall k = 1, 2, ..., m$, which are the maximum of three QoS utilities defined in (2.8). It is worth noting that this time-varying quantity represents the route quality for dynamic QoS requirements of various connections. Later, we shall use an aggregated, time-varying *resource utilization index*, $\mathrm{U}_{sg}^k(t)$, between s and gateway g as in (2.15) to denote these QoS constraints, as:

$$\mathrm{U}_{sg}^k(t) = \sum_{\forall q \in \pi_{sg}^k} Q_q, \tag{2.15}$$

where Q_q is denoted as the required goodput after the amount of packet loss is deducted from the throughput, *i.e.*, $Q_q = T_q^r(1 - E_q^r)$.

Connection admission control scheme is initialized when the new connection q arrives in the WMN with multiple QoS constraints at time t. Next, we propose a *per-route* based QoS performance index estimation scheme to try to accommodate this connection without violating the QoS of any ongoing connections.

Because node s has already kept some information about the quality on the k^{th} route, based on the statistics "resource utilization index" levels $\mathrm{U}_{sg}^k(t)$ and the corresponding QoS performance index $\mathrm{I}_{sg}^k(t)$ value at time t. Based on this, we can form a $\mathrm{U}_{sg}^k(t) \sim \mathrm{I}_{sg}^k(t)$ curve, from which the resource estimation is performed for the new connection q with the new "resource utilization index" Q_q if we assume it is accepted. The easiest way to do this is to use polynomial curve fitting method taking $\mathrm{U}_{sg}^k(t)$ as the input and $\mathrm{I}_{sg}^k(t)$ as the output. For instance, the transition function obtained is denoted as the mapping f, or:

$$\mathrm{I}_{sg}^k = f(\mathrm{U}_{sg}^k). \tag{2.16}$$

If the connection q is admitted into the network, this new input would incur the change of the output accumulated resource utilization index $\widetilde{\mathrm{U}}_{sg}^k(t)$ to,

$$\widetilde{\mathrm{U}}_{sg}^k(t) = \mathrm{U}_{sg}^k(t) + Q_q. \tag{2.17}$$

Now, we are able to estimate the k^{th} route quality based on the change of QoS performance index $\widetilde{\mathrm{I}}_{sg}^k(t)$ after the mapping f is taken:

$$\widetilde{\mathrm{I}}_{sg}^k(t) = f\left(\widetilde{\mathrm{U}}_{sg}^k(t)\right), \tag{2.18}$$

if we assume that the new connection is admitted.

According to the definition of QoS performance index with bounded threshold value 1, above which any connection will not be able to maintain satisfactory QoS requirements. Therefore, this new connection q could be accepted by the route π_{sg}^k and goes around the

QoS routing procedure if and only if it satisfies the condition (2.19),

$$\widetilde{\mathrm{I}}_{sg}^{k}(t) \leq 1, \tag{2.19}$$

otherwise the k^{th} route is *partially* rejected for the reason that our scheme may allow multi-level QoS and GoS resource management which will be introduced later. This scheme will release some resources of the existing low priority connections to maintain certain GoS for multiple service classes, and thus it is possible that the k^{th} route is feasible for the new connection q.

Similar steps should be performed for all routes within the route set π_{sg}, where the existing statistics of "resource utilization index" $\mathrm{U}_{sg}^{k}(t)$ and the corresponding QoS performance index $\mathrm{I}_{sg}^{k}(t)$ are used to form the mapping f, and estimated new QoS performance index will be computed and compared with value 1, until one of them is found as the feasible route for the new connection. From simulations, we observe that by interpolation and prediction on the mapping $f : \mathrm{U}_{sg}^{k}(t) \sim \mathrm{I}_{sg}^{k}(t)$, this connection admission control scheme is able to obtain a fair estimate of the QoS performance index with higher than 95% confidence bound.

It is interesting to see that from the mapping f, the time-varying "route capacity" on the route π_{sg}^{k} is changing constantly with the multi-dimensional QoS requirements of the dynamic arriving connections. This is because the required QoS metrics are served as the input while the attained QoS performance is used as the output, and thus the mapping $f : \mathrm{U}_{sg}^{k}(t) \rightarrow \mathrm{I}_{sg}^{k}(t)$ could completely reflect the change of QoS performance with respect to (w.r.t.) the change of QoS requirements. Therefore, it is safe to argue that the impacts of the new connections on the exiting ones running on the route is sufficiently reflected by the mapping f and thus the admission estimation is accurate. Furthermore, with regard to the route capacity, the maximum QoS performance index (equivalent to value 1) on a particular route corresponds to the capacity of the resource on that route , *i.e.,*

$$\mathrm{U}_{sg}^{k,\max}(t) = f^{-1}(1), \tag{2.20}$$

where f^{-1} denotes the inverse function of the mapping f and due to the knowledge of the curve f, the "route capacity" $\mathrm{U}_{sg}^{k,\max}(t)$ can also be computed by interpolation and/or estimation.

2.6.2 Multi-Level QoS and GoS Resource Management

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 requirements 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, in order to guarantee the satisfactory QoS experience for all users, this algorithm has to maintain a certain level of GoS. Under the proposed multi-level QoS context, we define GoS as the ratio of the number of 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 μ, *i.e.,*

$$\mathrm{GoS} = \frac{\mathrm{N_{HQoS}}}{\mathrm{N_{HQoS}} + \mathrm{N_{LQoS}}} \geq \mu \tag{2.21}$$

The novelty of the proposed algorithm is that not only it successfully manages the incoming connections but also allows the admission control scheme to degrade the ongoing HQoS connections' service quailing to the low level LQoS, given that at any time $\mathrm{GoS} \geq \mu$ has to be satisfied, so that network resources will become available for the new connections. 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 algorithm (for simplicity, only two-levels of throughput have

been considered) is described in the following steps:

1. The source node uses the admission prediction scheme described in Section 2.6 to check if any of the existing routes can provide high-throughput QoS. If not, it initiates route discovery phase to search for new possible routes that can provide high-throughput to the connection.

2. If it fails to find any route that provides high-throughput QoS requirement, it firstly repeats the prediction scheme trying to accommodate the connection with low-throughput QoS requirement; and if it fails again the IQoSR procedure is called with the low-level throughput QoS requirement as the input.

3. If it fails again to guarantee low-level throughput QoS requirement, before performing the rejection, it tries to degrade the level of ongoing HQoS connections to LQoS requirements, given that the condition GoS $\geq \mu$ must be satisfied at any time. The prediction scheme is repeated until the route k^* can accommodate the connection, and an admission signal is released; otherwise the new connection is rejected.

The pseudo-code of the proposed algorithm is given by Algorithm 1[4]. Moreover, Figure 2.4 depicts the real-time performance of the proposed algorithm. The GoS threshold has been set to $\mu = 0.9$ while the low-throughput QoS requirement values are one-third of the high-throughput ones. 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

We develop a time-slotted, event-driven OPNET [73] simulator which comprises PHY, MAC and network layers, where antenna and wireless channel model, adaptive modulation and coding (AMC) schemes, different MAC scheduling and routing algorithms are implemented respectively. WMRs and IGWs are randomly deployed in 2 D square in a

[4]We use T^k_{avail} to denote the supportable throughput requirement on the k^{th} route, use $T^{r,H}_q$ and $T^{r,L}_q$ to denote the connection's HQoS and LQoS throughput requirements respectively, and use π^{on}_{sg} to denote the route set who have already carried on some ongoing connections.

Algorithm 1 : Multi-level QoS and GoS resource management algorithm

New connection q arrives in the network
Call admission control for all route $k \in \pi_{sg}^{on}$
if $\exists k \in \pi_{sg}^{on}, s.t., T_{avail}^{k} > T_q^{r,H}$ **then**
  Accept the connection in the route k with $T_q^{r,H}$ (FINISH)
else
  Call IQoSR procedure
  if $\exists k^* \in (\pi_{sg} - \pi_{sg}^{on}), s.t., T_{avail}^{k^*} > T_q^{r,H}$ **then**
    Accept the connection q in route k^* with $T_q^{r,H}$ (FINISH)
  else
    Estimate GoS level G
    if $G \geq \mu$ **then**
      if $\exists k \in \pi_{sg}^{on}, s.t., T_{avail}^{k} > T_q^{r,L}$ **then**
        Accept the connection in the route k with $T_q^{r,L}$ (FINISH)
      else
        Call IQoSR procedure
        if $\exists k^* \in (\pi_{sg} - \pi_{sg}^{on}), s.t., T_{avail}^{k} > T_q^{r,L}$ **then**
          Accept the connection q in route k^* with $T_q^{r,L}$ (FINISH)
        else
          $i = 1$
          while $G = \frac{\mathrm{N_{HQoS}} - i}{\mathrm{N_{HQoS}} + \mathrm{N_{LQoS}}} \geq \mu$ **do**
            Degrade one connection q with HQoS requirement to LQoS requirement *i.e.,* HQoS $\rightarrow$ LQoS
            Call admission control $\forall k \in \pi_{sg}^{on}$
            if $\exists k \in \pi_{sg}^{on}, T_{avail}^{k} > T_q^{r,L}$ **then**
              Accept the connection q in route k with $T_q^{r,L}$ (FINISH)
            end if
            $i \leftarrow i + 1$
          end while
          Reject the connection q (FINISH)
        end if
      end if
    else
      Reject the connection q (FINISH)
    end if
  end if
end if

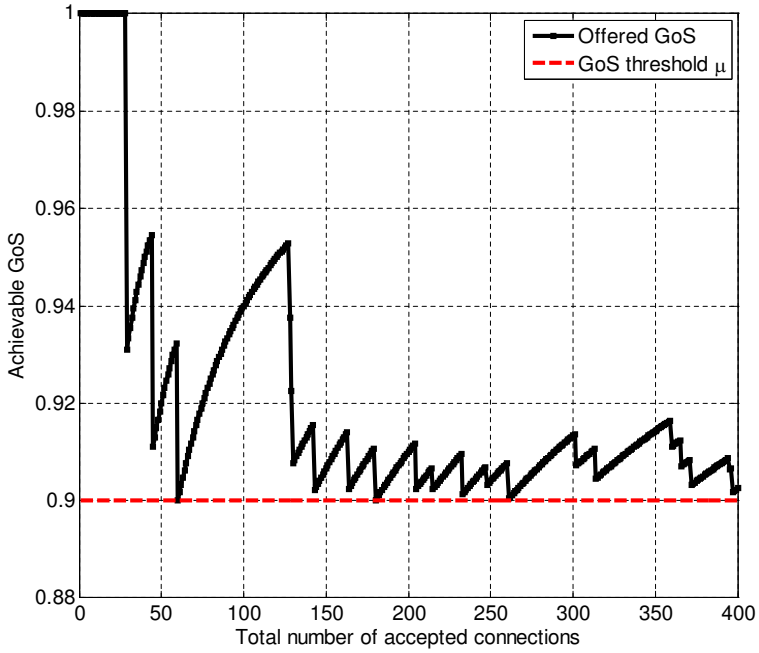

Figure 2.4: Real-time simulation of the offered GoS as a function of the number of accepted connections from time to time, where it can be seen that the offered GoS decreases and gradually approximates the predetermined GoS threshold value $\mu = 0.9$ when more connections arrive.

way that no disconnected clusters of nodes exist in the network, as shown in Figure 2.5. 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. Different traffic patterns are considered in this simulation, including VoIP, video and broadband data services (like FTP), each of which is attached with three QoS constraints, *i.e.,* throughput, ETE packet delay and PER, where ETE packet delay consists of both the queuing and the transmission delays. In PHY layer, the Rayleigh fading channel model [74] is used to generate the wireless link characteristics among WMRs. PER is simulated based on the SINR curve for the used AMC scheme. Infinite-persistent automatic retransmission request scheme (IP-ARQ) in MAC layer is assumed in case of the packet failure. The simulation parameters are summarized in Table 2.2.

The performance of the proposed cross-layer solution highly depends on the accurate estimation of multiple parameters in different protocol layers which are required

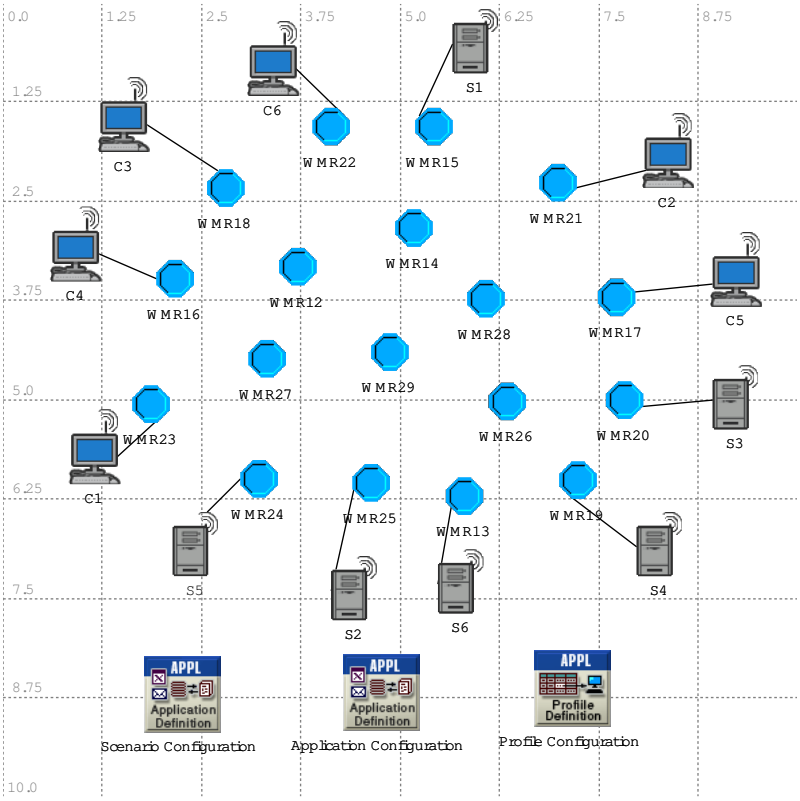

Figure 2.5: An example of the standard scenario used in our OPNET simulation platform. The WMN consists of eighteen WMRs with six client and server pairs to generate traffic.

for the QoS routing, distributed scheduling and the admission control decisions, which include real-time monitored per-link statistics like throughput between link (i, j), or T_{ij}^{a}, queuing and transmission delay between link (i, j), orD_{ij}^{a}, and PER between link (i, j), or E_{ij}^{a}. These statistics are stored and updated periodically according to the scheduling and routing operational time scales to represent the most recent channel qualities and queue status.

2.7.1 The Overall Network Performance

In this section, we assess the overall performance of the proposed cross-layer design solution to support QoS in WMNs that comprises a distributed opportunistic proportional fair scheduling algorithm (denoted as "Dist") in MAC layer, a multi-constrained QoS routing (denoted as "IQoSR") and the admission control algorithm used in network layer (denoted

Table 2.2: 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-1/2, QPSK, 16QAM, 64QAM, 128QAM
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 18
Number of Client/Server pair	6
Network Area	10 km × 10 km square
Transmission Range	2 km
Traffic Patterns	FTP, voIP and Video
Queue Length	Infinite

Table 2.3: Cross-layer performance comparisons

MAC	Routing	Admission Control	Cross-Layer Term
Round Robin	AODV	-	RR+AODV
Distributed OPF	IQoSR	RC-CAC	Dist+IQoSR+RC-CAC

as "RC-CAC"). The compared benchmark protocol is Round Robin scheduler (denoted as "RR" [75]) and AODV routing scheme, denoted as the combination"RR+AODV", as shown in Table 2.3. The performance is investigated in terms of the gateway goodput in Figure 2.6, and the average QoS outage probability among all connections in Figure 2.7, both are plotted with different network sizes and traffic loads.

Figure 2.6 depicts the behaviors of the overall gateway goodput w.r.t. different network sizes (*i.e.,* placing different number of nodes in a fixed network area) and traffic inter-arrival times. It is interesting to observe that there is an optimal network size in terms of the node density for a given traffic inter-arrival time for both schemes 'Dist+IQoSR+RC-CAC" and "RR+AODV". The reasons are two folds. One, distributed opportunistic scheduler takes advantage the multi-user diversity gain by always selecting the best wire-

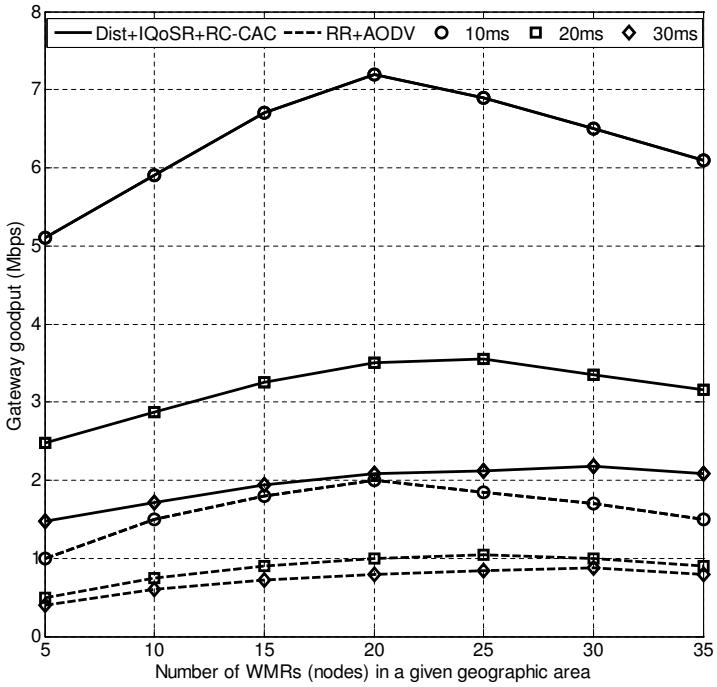

Figure 2.6: Simulation result of the average gateway goodput w.r.t. the different network sizes and the different new connection inter-arrival time.

less channel among all neighbors. Two, QoS routing and admission control algorithms can not only successfully select the best candidate route in a pool of candidate routes, but also accurately predict the impact of the new connection admissions on the existing ones, and thus QoS is guaranteed. However, since network resources (*e.g.,* time-slot, codes, power etc.) are limited and shared, the achieved gain decreases when the network size is larger. Meanwhile, the increasing node density may potentially create more co-channel interference due to concurrent transmissions, and thus deteriorating the per-node gateway goodput, as known by Gupta et. al's work in [33]. Furthermore, if WMRs are sparsely distributed, *i.e.,* only small number of nodes exist in the network, the opportunistic gain could not be fully exploited by the distributed scheduler, and the selected route may not be good enough to provide the QoS. These are primarily why there is an optimal operational point for the number of nodes in a given network to maximize the overall gateway goodput. Finally, as we increase the traffic input rate, higher goodput is expected, but always twice higher than the "RR+AODV" scheme.

Figure 2.7 demonstrates the average QoS outage probability of all completed connections as a function of both the traffic load and the network size. QoS outage probability is defined as the probability of any of the QoS requirements of a connection to fail during its lifetime; in other words, the condition $I_{sg}^{k} \leq 1$ does not hold at all. It can be seen that the proposed cross-layer solution "Dist+IQoSR+RC-CAC" always achieves 15% lower outage probabilities than the combination "RR+AODV", due to the accurate resource estimation of the network layer routing and admission control schemes to prevent new connections from consuming too many network resources of the existing connections in the network. Meanwhile, the distributed scheduler interacts with the routing algorithm to provide long-term throughput guarantees as well as fully exploited multi-user diversity gain. However, as we increase the node density in a given network area, potentially we may generate more co-channel interference due to more concurrent transmissions, and thus deteriorating the connection quality (higher QoS outage), however, for a fixed traffic load, the multi-user diversity gain of wireless channel boosts the gateway throughput although for any single connection the potential QoS failure may increase.

2.7.2 Performance Evaluations on the Scheduling and Routing Algorithms

The loosely-coupled cross-layer design solution for distributed opportunistic scheduling (Dist) and integrated multi-constrained QoS routing (IQoSR) algorithms are assessed here, compared with the conventional Round Robin scheduler (RR, [75]) and AODV routing protocol. Table 2.4 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.

Figure 2.8 shows that the opportunistic scheduler considered 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 scheduler fails to exploit the multi-user diversity gain of wireless channel (or, channel resources

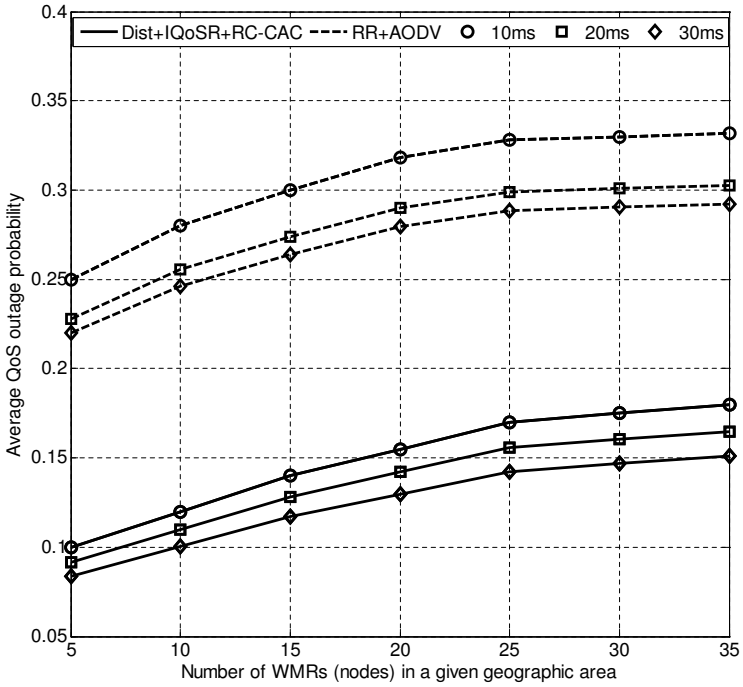

Figure 2.7: Simulation result of the average QoS outage probability w.r.t. the different network sizes and the different new connection inter-arrival time.

Table 2.4: Scheduling and routing performance comparisons

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

are reserved and pre-allocated for RR scheduler as a round robin fashion), and AODV routing protocol creates bottleneck links in the network by transporting traffic through always the shortest route. "Dist+AODV" and "RR+IQoSR" run between the lower-bound performance of "RR+AODV" and the upper-bound performance of "Dist+IQoSR", since 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

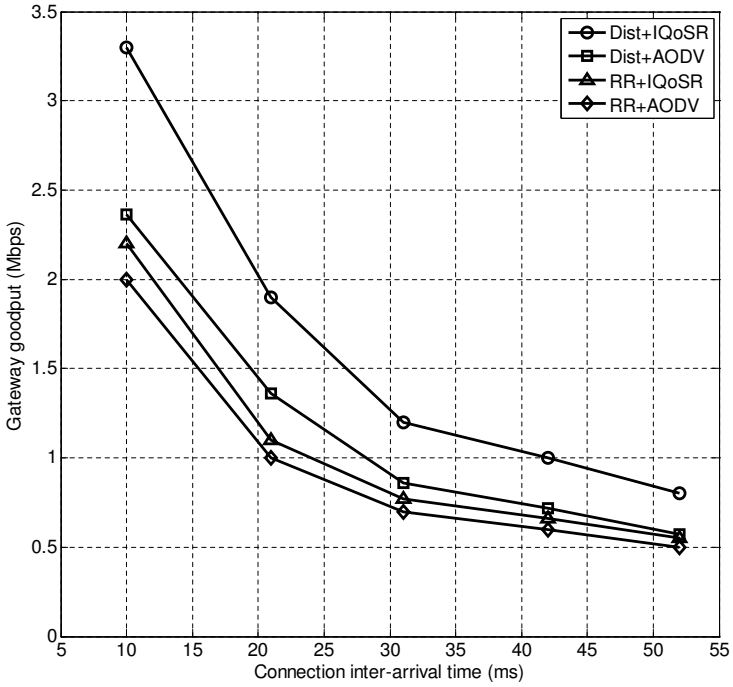

Figure 2.8: Simulation result of the average gateway goodput w.r.t. the different new connection inter-arrival time for different combinations of scheduling and routing algorithms.

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 connection admission 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 connection admission control scheme in multi-hop wireless network from another perspective.

2.7.3 Performance Evaluations on the Connection Admission Control Algorithm

Now, we turn our attention to the performance of the connection admission control algorithm as a outcome of the tightly-coupled cross-layer design approach with QoS routing. The proposed algorithm ("Dist+ IQoSR+RC-CAC") is compared with the "Dist+IQoSR"

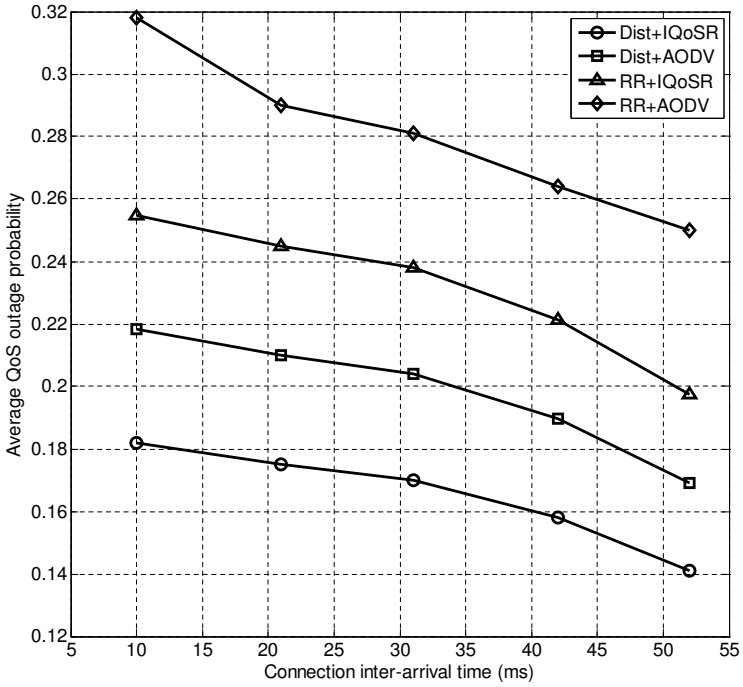

Figure 2.9: Simulation result of the average QoS outage probability w.r.t. the different new connection inter-arrival time for different combinations of scheduling and routing algorithms.

that does not include an efficient prediction scheme for admission control. We also compare our scheme with benchmark protocol "RR+AODV", and with a statistical admission control ("SCAC", [70]) algorithm in the literature. The performance is investigated in terms of the overall gateway goodput in Figure 2.10, and the average QoS outage probability of all completed connections in Figure 2.11.

Figure 2.10 shows that the cross-layer approach"Dist+IQoSR+RC-CAC" outperforms all other schemes in terms of the overall gateway goodput. An important observation is that the proposed framework can successfully achieve high goodput even for small traffic inter-arrival time (heavy load conditions), *i.e.,* 1.4 times more than "Dist+IQoSR+SCAC", 2.2 times more than "Dist+IQoSR", and 3.2 times more than "RR+AODV". This is primarily because that the connection admission control scheme can admit or reject new connections wisely to maximize the ETE resource utilization in the whole network scale by predicting the route capacity. By monitoring the resource occupancies along each route, it accurately identifies links and routes with potentially limiting

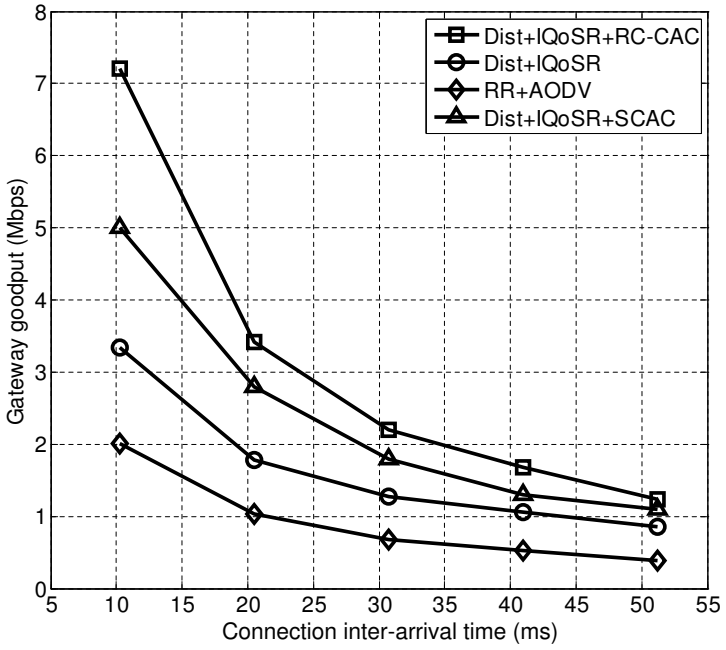

Figure 2.10: Simulation result of the average gateway goodput w.r.t. the new connection inter-arrival time for different connection admission control algorithms.

resources, and captures the impact of arrival connections on the existing ongoing connections. This will result in lower QoS outage probability as shown in Figure 2.11, especially when the network operates at heavy traffic conditions. Meanwhile, the GoS management allows certain bandwidth resources preserved for high QoS users through maintaining the GoS threshold μ. On the other hand, the "SCAC" scheme achieves high goodput when the traffic load is high due to its Gaussian traffic arrival assumption, but when the traffic load is relatively low, it fails to accurately estimate the achievable capacity region, thus make wrong decisions on connection admission which turns into lower goodput and higher QoS outage probability.

Figure 2.11 illustrates the probability of QoS outage of all completed connections as a function of the traffic load. This is defined as the probability of any of the QoS requirements of a connection to fail during its lifetime; in other words, $\mathrm{I}_{sg}^{k} > 1$. It is interesting to observe that even for high network loading conditions, our proposed algorithm can guarantee 85% of all connections satisfying their all QoS requirements of the underlying application, as compared to 81% if no admission control is used, 82% if "SCAC" is used,

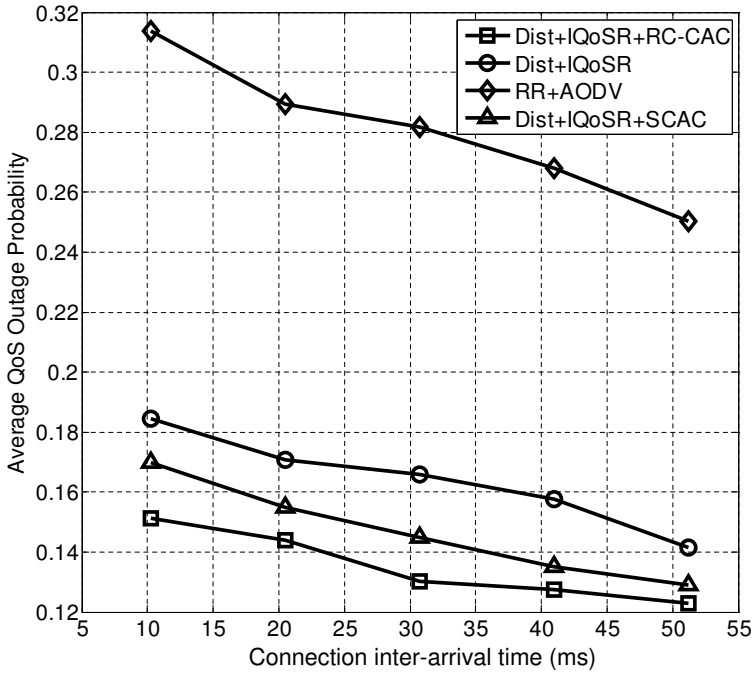

Figure 2.11: Simulation result of the average QoS outage probability for all completed connections, w.r.t. the new connection inter-arrival time for different connection admission control algorithms.

and 68% for "RR+AODV". This is because the impact of the new admitted connection on existing connections has been estimated and accurately reflected during the route capacity estimation phase.

Figure 2.12 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 admitted low-level throughput connections decreases to allow more resources for high-level throughput connections. However, the total number of connections admitted is going down because more stringent end-user quality is expected. Overall, it can be seen that the end user QoS experience can be significantly improved by reducing the blocking probability while maintaining a reasonable outage probability and a high GoS of the admitted connections.

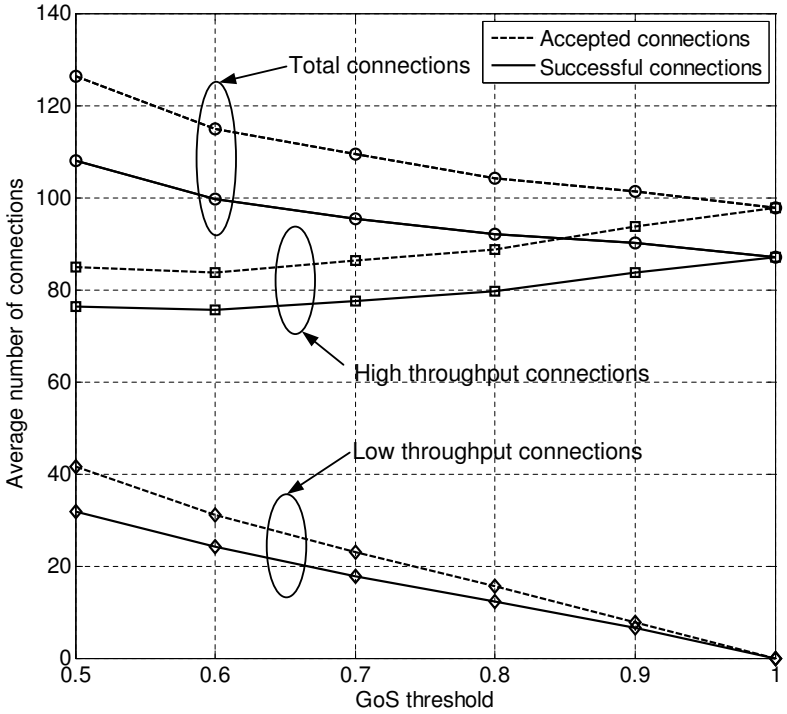

Figure 2.12: Simulation result of the average number of admitted and successful connections in the network as a function of the GoS threshold μ.

2.8 Summary

Cross-layer designs to support QoS in wireless mesh networks has 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 connection admission control together with QoS routing in the network layer and distributed opportunistic proportional fair scheduling in MAC layer. We defined a novel utility function that is exchanged between an efficient distributed opportunistic proportional fair scheduler and a multi-constrained QoS routing algorithm. Furthermore, a novel tightly-coupled design method for joint routing and admission control has been demonstrated, where a unified optimization criterion "QoS performance index" that combines multiple QoS constraints to indicate the QoS experience of each route has been proposed. 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.

Chapter 3

Network Capacity Estimation and QoS Control

IN this Chapter, the dissertation further extends the contributions of Chapter 2 to tackle the main challenges of estimating the time-varying network capacity with the presence of multiple QoS requirements, to facilitate the QoS control of the new connection. The well-known difficulties of the capacity estimation mainly come from the dynamic connection behaviors and the associated multiple QoS requirements, which make the network capacity generally cannot be easily parameterized by a single (or a set of) variable(s). However, researchers are always highly interested in estimating it since the accurate network capacity estimation is required for the network optimization and planning, QoS control through admission control (AC) of the new connection, and the optimal operations of the served connections.

To address this challenge, the whole network is modeled as a set of subnetworks, each one of which lies between a pair of source and destination network routers. The subnetwork is further represented by a "black box". Several network-level and traffic-level parameters are used as inputs. A generic mathematical function is then used to map the multiple input variables to a single output parameter, called *QoS performance index*. The bounded value of this index corresponds to the maximum available network resources, defined as the *subnetwork capacity*, under the given traffic volume and requirements. By

using the Taylor expansion, we are able to predict the impact of new connections on the subnetwork capacity, which will be later used by a generic AC (GAC) methodology to maximize the resource utilization while the QoS requirements from multiple service classes are guaranteed. The uniqueness of the proposed methodology is its wide applicability to any type of packet networks, wired and/or wireless, independent of the communication protocols or standards to use in lower protocol layers.

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 [76]. 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.

The challenges of these technology advances are mainly driven by the question of how to support the multimedia traffic, like voice-over-IP (VoIP), interactive video, and broadband data services. The large number of connections with strict and multiple QoS requirements including end-to-end (ETE) throughput, packet delay, and packet error rate (PER) can make all these networks inefficient and fragile. On the other hand, preferential treatments for users of different service classes (such as premium and regular services), or generally different grade-of-services (GoS), are required by most service providers to be adaptable with various billing systems in order to improve user experiences and maximize revenue.

Furthermore, it is well known that the network capacity, or the amount of available network resources, is one of the key parameters for designing such an efficient network architectures (for both QoS and GoS provisions) that has different interpretations at dif-

ferent protocol layers and different networks. Nevertheless, it is also known to be difficult to estimate the network capacity in both ad hoc wireless networks [33] and conventional IP networks. 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. Furthermore, in wireless networks, due to the co-channel interference and channel fluctuations, the uncontrollable admission of the improper new connection can highly affect the resource availability of adjacent transmissions. Hence, the precise knowledge of the available network capacity will allow the network operator to perform optimal admission control (AC) to new connections without jeopardizing the proper operations of the existing ones in terms of their QoS and GoS.

All these challenges drive us to redesign a new methodology for capacity estimation and QoS control to enforce the AC decision in packet networks where we have made the following three contributions.

One, we generically model any packet network as a set of subnetworks, each of which is modeled as a "black box" for the amount of available network resources. The ingress node aggregates traffic as the input to the subnetwork, and an egress node serves as an intended destination. The input parameters (*i.e.,* network status, traffic pattern, QoS requirements) to the black box are mapped to a single output parameter (or later 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 operational contexts of the communication protocols as *a priori*.

Two, to facilitate the AC decision, we uniquely introduce a novel concept of QoS-aware *subnetwork capacity* between any pair of ingress and egress nodes in the network to indicate the capability the subnetwork can provide, in terms of the amount of time varying available resources, to any connection with any QoS constraints. This capacity is defined as a multi-dimensional vector (for certain network-defined dimension P), envisioned as a "pool" of parameters, including but not limited to, the maximum cardinality of the connection set, the maximum supportable throughput, the minimum supportable delay

requirement, etc. Subnetwork capacity proves to be the key design parameter for any combination of connections with satisfactory QoS before admission.

Three, different GoS are successfully maintained by a generic AC (GAC) methodology allowing preferential treatments to different service classes such that the portion of subnetwork capacity is reserved for higher service class users. Without loss of generality, we show that the proposed capacity estimation and QoS control methodology has wide applicability to any packet networks, wired and/or wireless, and we discuss various important feasibility issues like the ratio between connection throughput and subnetwork capacity, statistics feedback delay, and statistics collection time.

Finally, this methodology is completely transparent to the lower protocol layers, 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.

The rest of this Chapter is organized as follows. The system model is introduced in Section 3.2. Section 3.3 presents the methodology for subnetwork capacity estimation. Next, the capacity estimation and QoS control algorithm description is introduced in Section 3.4, and numerical results and detailed analysis are given in Section 3.5. Finally, Section 3.6 presents the discussions on the applicability and the feasibility issues, and a summary is drawn in Section 3.7.

3.2 System Model

Consider a generic packet network that comprises a finite number of nodes, a set of which are n_i ingress nodes, denoted as a set $V_I = \{v_i | i = 1, 2, \ldots, n_i\}$, and n_e egress nodes, denoted as a set $V_E = \{v_e | e = 1, 2, \ldots, n_e\}$. Furthermore, to support GoS among different users, suppose service provider supports a finite set of service classes $\mathcal{J} \triangleq \{1, 2, \ldots, |\mathcal{J}|\}$, where higher service class is denoted as higher number in set $\mathcal{J}$. Traffic comes from application layer drills down to the ingress node, and intends to transfer to the egress node, however multicasting is out of the scope of this Chapter. Moreover, connections

are generated independently among all ingress nodes with arrival probability p_j, where $\sum_{\forall j \in \mathcal{J}} p_j = 1$, for service class $j \in \mathcal{J}$. Connection $q \in \mathcal{Q}$ is attached with a set of constraints, namely: ETE packet delay D_q^r, throughput T_q^r, and PER E_q^r, where subscript r denotes the *required* value.

To facilitate our capacity estimation and QoS control methodology, or generic admission control design in general, we model the whole packet network as multiple pairs of ingress-to-egress subnetworks as shown in Figure 3.1, where different subnetworks may share the common nodes and the network resources. Furthermore, within each subnetwork, multiple routes connect an ingress-to-egress node pair with limited amount of network resources shared by concurrent connections, which include, but not limited to, buffer, bandwidth, transmission power, codes, etc. As introduced earlier, network capacity, or the amount of available resources for the new connections, has been well-known very difficult to estimate even for ad hoc wireless networks [33] and conventional IP networks, it nevertheless serves as the key parameter for designing an efficient network architecture that coordinates PHY, MAC, network layers and the above. To overcome this difficulty, we treat each subnetwork 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 end-node application layers[1], but we adopt a runtime analysis such that connections are probed (or monitored) by the black box for the satisfactory levels of QoS, *i.e.,* the benefit packet network can provide to each served connection is constantly monitored at the egress node and informs the ingress node through the feedback loop as shown in Figure 3.1, so that whenever the new connection arrives at the ingress node, the availability of current subnetwork to support certain QoS in terms of the amount of the network resources is known.

The Inputs:

Two types of black box inputs are considered from the ingress node, including system (or network) level parameters, like the number of ongoing connections, and traffic-level

[1] which is usually the reality

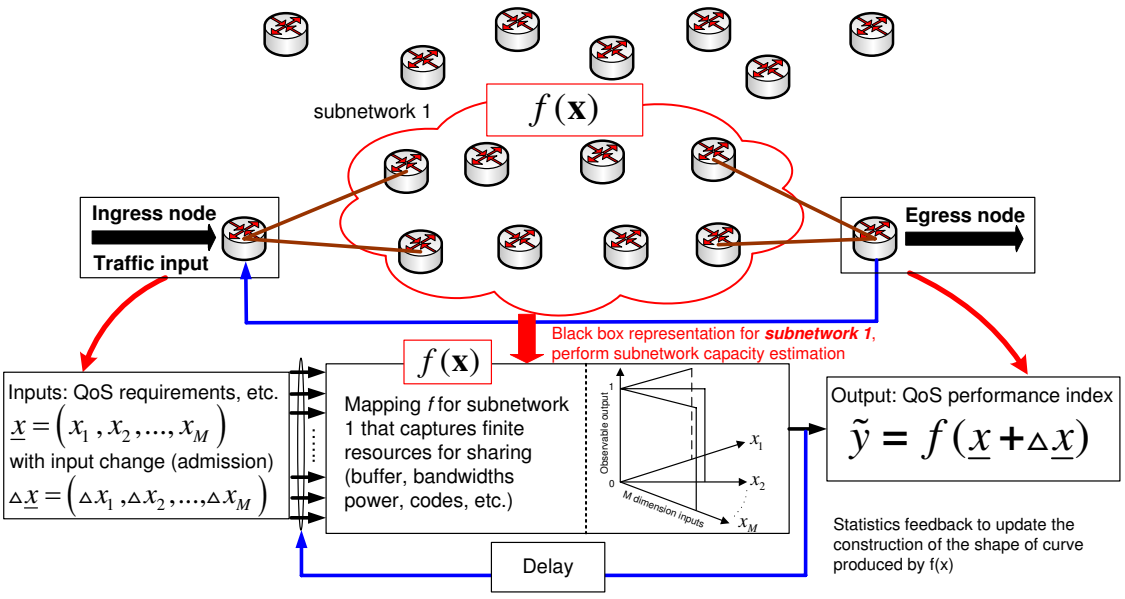

Figure 3.1: A mathematical representation for the subnetwork resources between an ingress node and an egress node. M dimension input variables are used to represent the traffic statistics like multiple QoS requirements, and the observable QoS performance index is used as a single output. These inputs and oupt construct a mapping f for the black box.

parameters, like multiple QoS requirements.

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, whcih is defined the *QoS performance index*.

By monitoring the resource occupancy at the egress node, we want to identify the potentially limiting resources within the black box and estimate the capacity a subnetwork can support (or later defined *subnetwork capacity*), to decide if the new connection can be admitted with satisfactory QoS. The uniqueness and advantage of this black box representation come from the fact that it is completely transparent to lower protocol layers such that whatever technologies (*e.g.,* routing, scheduling, advanced PHY layer modulation, coding and antenna, etc.) to use, it always reflects the amount of available resources for maintaining certain QoS performance in real-time.

3.2.1 QoS Performance Index

We are interested in identifying the degree of QoS satisfaction any connection has experienced in the considered packet network, which quantifies the degree to which a network can be judged sufficient for servicing a particular connection given its QoS requirements. To this end, a unique QoS utility function based on the *dissatisfaction ratio* R^2 is defined for each QoS parameter, *i.e.,* R_q^D for ETE packet delay, R_q^T for throughput, and R_q^E for PER, $\forall q \in \mathcal{Q}$, between the *attained* parameter measurement denoted by subscript a in (3.1), and *required* QoS value denoted by subscript r, *i.e.,*

$$R_q^D = \beta_D \frac{D_q^a}{D_q^r}, \quad R_q^T = \beta_T \frac{T_q^r}{T_q^a}, \quad R_q^E = \beta_E \frac{E_q^a}{E_q^r}, \quad \forall q \in \mathcal{Q}, \tag{3.1}$$

where $\{\beta_D, \beta_T, \beta_E\} \geq 1$ are a set of error margins introduced for delay, throughput, and PER respectively to provide a safe guard for imperfect resource estimations and system fluctuations. Hence, a connection maintains satisfactory QoS if and only if,

$$\left\{R_q^D, R_q^T, R_q^E\right\} \preceq 1, \forall q \in \mathcal{Q} \tag{3.2}$$

where notation $\preceq$ denotes the element-by-element comparison.

Due to the multi-dimensional nature of the QoS requirements, even though we use a set of dissatisfaction ratios to denote the degree of satisfaction for each metric, it is also very difficult to judge the *overall* QoS experience any connection has received. Meanwhile, the degree of QoS satisfaction is very important for the end-user or service provider, and thus we want a simple, representative, and quantitative scaler to uniquely denote the level of QoS satisfaction. These are primarily why we introduce a novel concept of the *QoS performance index* $\mathrm{I}(q), \forall q \in \mathcal{Q}$, which is defined as,

$$\begin{aligned} \mathrm{I}(q) &= g\left(R_q^D, R_q^T, R_q^E\right) \\ &\triangleq \max\left(R_q^D, R_q^T, R_q^E\right) \leq 1, \end{aligned} \tag{3.3}$$

[2] as the same way that is defined in Chapter 2

where $g(\cdot) \triangleq \max(\cdot)$ is one of many realizations for generic function $g(\cdot)$ that integrates multiple QoS parameters into one single performance index, and used later.

As shown in Figure 3.1, when connections complete transmission through the subnetwork from an ingress node v_i to an egress node v_e, the egress node keeps track of a set of statistics. These statistics include traffic-level measurements, like instant served throughout T_t within the subnetwork, and system-level parameters, like the number of ongoing connections N_t, and the QoS performance index $\mathrm{I}(q)$. All these statistics will pass to the ingress node v_i through a feedback loop on higher layer network protocols. Suppose at any given time t, connection q completes its transmission within the subnetwork and the egress node probes its quantitative QoS performance index $\mathrm{I}(q)$, then we use a parameter I_t to indicate the subnetwork-wide QoS satisfaction that combines all QoS performance indexes of the completed connections. This parameter evolves over time through exponential smoothing as:

$$\mathrm{I}_t = \gamma \mathrm{I}(q) + (1-\gamma)\mathrm{I}_{t-1}, \forall q \in \mathcal{Q}, \gamma \in (0,1). \tag{3.4}$$

where $\gamma \in (0,1)$ is the weight factor and $\mathrm{I}_0 = 0$. Figure 3.2(a) shows conceptually how each statistics $\mathrm{I}(q)$ is processed through exponential smoothing to get the long-term average value I_t in Figure 3.2(b).

3.2.2 The Mathematical Representation of the Black Box

Without loss of generality, we use a generic mathematical function f to represent the operational characteristics of the black box, which maps M-dimensional input variables $\underline{x}_t = (x_t^1, x_t^2, ..., x_t^M) \in \mathbb{R}^M$[3] to a scalar output $y_t \in \mathbb{R}$, which denotes the degree of resource occupancy within the black box. In other words, we use the mapping

$$f : \mathbb{R}^M \to \mathbb{R}, \text{ or } y_t = f(\underline{x}_t) \tag{3.5}$$

[3]The underlined notation signifies a vector quantity in this Chapter unless otherwise stated.

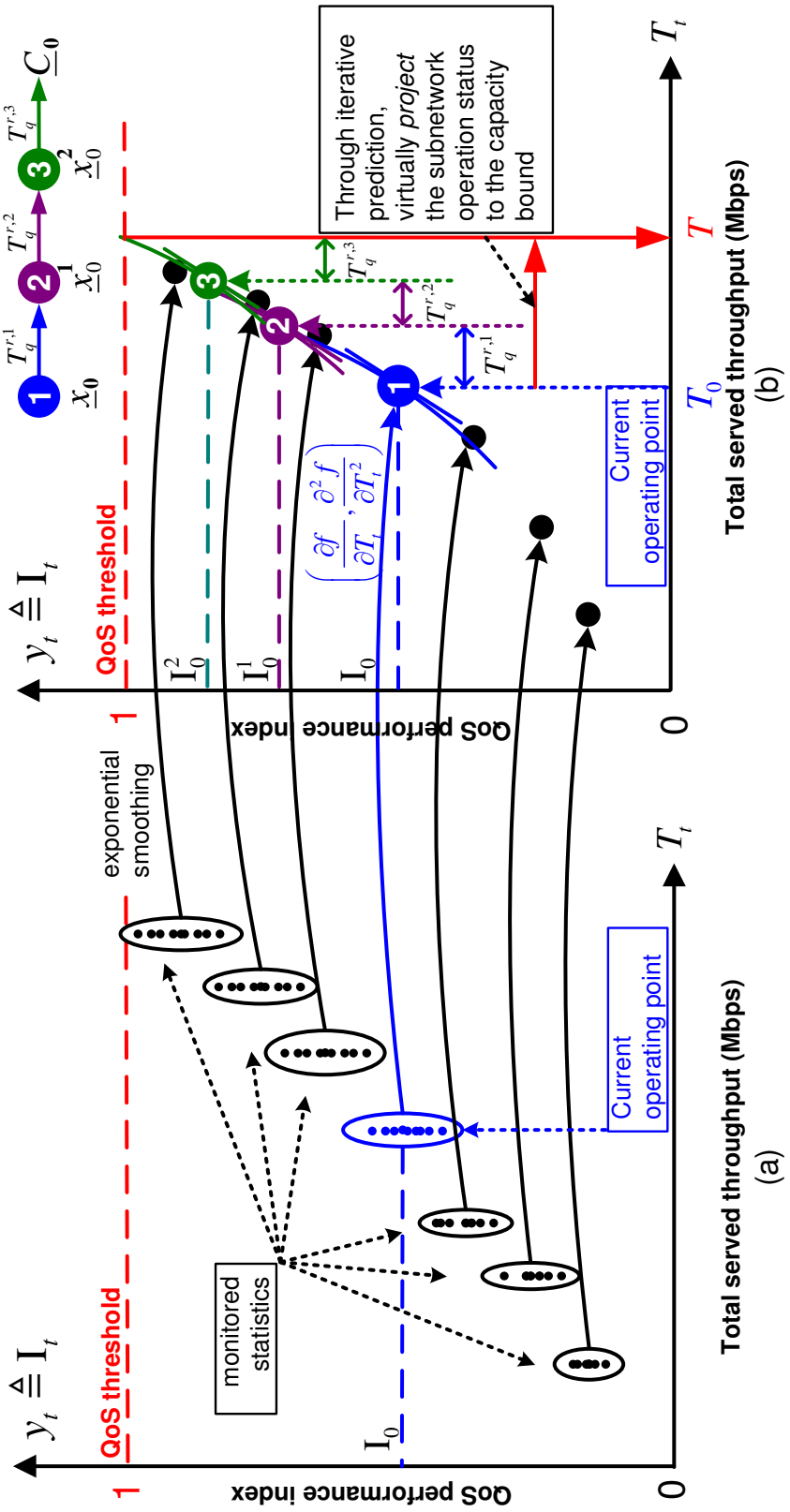

Figure 3.2: (a) An example of using exponential smoothing method to average all previous observed per-connection QoS performance index $\mathrm{I}(q)$, and to obtain a single output I_t. (b) An example of using the iterative estimation method to predict the subnetwork capacity on throughput dimension.

to represent a $(M+1)$-dimensional space as shown in Figure 3.1, where M input variables $\underline{x}_t$ capture the system status at any given time t, like the number of connections and required traffic QoS requirements. Furthermore, in our derived capacity estimation and QoS control methodology, we use the defined QoS performance index as the output, *i.e.,*

$$y_t \triangleq \mathrm{I}_t. \tag{3.6}$$

Input Change: the Admission

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

$$\tilde{y}_t = f(\underline{x}_t + \Delta\underline{x}_t). \tag{3.7}$$

It is worth to note that the mapping f is usually quite complicated, which generally cannot be expressed in a closed-form expression. Furthermore, the expression (3.7) implies that some mathematical method could be used to estimate the mapping f, instead of accurately model it in a closed-form. It is not difficult to observe that the Taylor expansion [77] is the one implied in (3.7), and could be used as a tool to approximate the shape of curve produced by the mapping f. 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:

$$\tilde{y}_t = f(\underline{x}_t + \Delta\underline{x}_t) \approx f(\underline{x}_t) + \sum_{i=1}^{M} f'_{x_t^i}\Delta x_t^i + \frac{1}{2}\left(\sum_{i=1}^{M} f''_{x_t^i}\left(\Delta x_t^i\right)^2 + \sum_{i=1}^{M}\sum_{j\neq i} f''_{x_t^j x_t^i}\Delta x_t^i \Delta x_t^j\right) \tag{3.8}$$

where we denote $f'_{x_t^i} = \frac{\partial f}{\partial x_t^i}$, $f''_{x_t^i} = \frac{\partial f^2}{\partial x_t^{i,2}}$, and $f''_{x_t^j x_t^i} = \frac{\partial f^2}{\partial x_t^i \partial x_t^i}$.

Now we elaborate how the mapping f is obtained through 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 three steps follow as:

1. The admitted connections will incur the ingress node to update the subnetwork operating status $\underline{x}_t$ (*i.e.,* the M input variables) correspondingly.

2. When there is a connection completion event occurs at the egress node, it receives certain QoS performance index calculated in (3.3) and further exponentially smoothed by (3.4) to obtain an updated value I_t.

3. Statistics I_t is passed to the ingress node through a feedback loop[4].

4. When the ingress node receives the new statistics I_t and knows certain connection has just completes, it updates the current network status $\underline{x}_t$. Iteratively, if enough connection completion events are monitored and statistics are recorded, the mapping f is represented by a set of statistic pairs $(\underline{x}_t, \mathrm{I}_t)$.

Figure 3.2 shows an illustrative example, where Figure 3.2(a) shows how statistics are collected and processed. Each dot in the plot represents one such performance observation $\mathrm{I}(q)$, and we apply exponential smoothing method with parameter α as a weight factor on the observations, in order to get a long-term average estimation I_t as shown in Figure 3.2(b).

Physical Meaning of the Taylor Series

It is very interesting to observe that the Taylor series imply useful physical meanings as the first order partial derivatives actually means the *average* QoS performance index change if certain connection is admitted, whereas the second order partial derivatives illustrate the *finer* QoS performance index change which cannot be represented by the first order partial derivatives. However, the change incurred by the second order partial derivatives will never exceed the amount of change incurred by the first order partial derivatives, *i.e.,* for certain circumstances second order statistics can be negligible.

[4]Practically, this feedback loop can be achieved through higher layer protocols e.g. to piggyback this information in the TCP acknowledgement packet.

3.3 Subnetwork Capacity

3.3.1 The Definition

We formally define a P-dimensional *subnetwork capacity* $\underline{C}_t = \{\mathcal{C}_t^1, \mathcal{C}_t^2, \ldots, \mathcal{C}_t^P\} \in \mathbb{R}^P$ within a particular subnetwork lies between a pair of ingress and egress nodes referring to the instant amount of available resources, as:

Subnetwork Capacity indicates the time varying capability a subnetwork can provide to any combination of connections with satisfactory QoS requirements from time to time, such that the condition $\mathrm{I}(q) \leq 1, \forall q \in \mathcal{Q}$ always satisfies. Subnetwork capacity $\underline{C}_t$ is a multi-dimensional vector with network defined dimension P such that each element $\mathcal{C}_t^p \in \underline{C}_t, \forall p = 1, 2, \ldots, P$ can represent any one of the following parameters (not exclusively though): the maximum cardinality of the connection set $\mathcal{Q}$, the maximum supportable throughput, the minimum supportable delay requirement, etc.

3.3.2 The Estimation Process

In order to demonstrate how the subnetwork capacity is estimated in realtime, without loss of generality, we will use two-dimensional input variables as an illustrative example, *i.e.*, $\underline{x}_t = (x_t^1, x_t^2) \triangleq (N_t, T_t) \in \mathbb{R}^2$, as shown in Figure 3.3. In other words, the number of currently ongoing connections N_t is used as a system-level parameter and the currently served throughput T_t is used as a traffic-level parameter, at any given time t. Now, the mapping f becomes,

$$y_t \triangleq \mathrm{I}_t = f\left(N_t, T_t\right). \tag{3.9}$$

Then, the current subnetwork status can be further denoted by the input vector $\underline{x}_0 \triangleq (N_0, T_0)$, if we assume the current time is $t = t_0$, when a new connection arrives for admission at the ingress node. Later, the key design parameter, P-dimensional subnetwork capacity $\underline{C}_0$, is reduced to a two-dimensional vector as:

$$\underline{C}_0 \triangleq (\mathcal{N}, \mathcal{T}) \in \mathbb{R}^2, \tag{3.10}$$

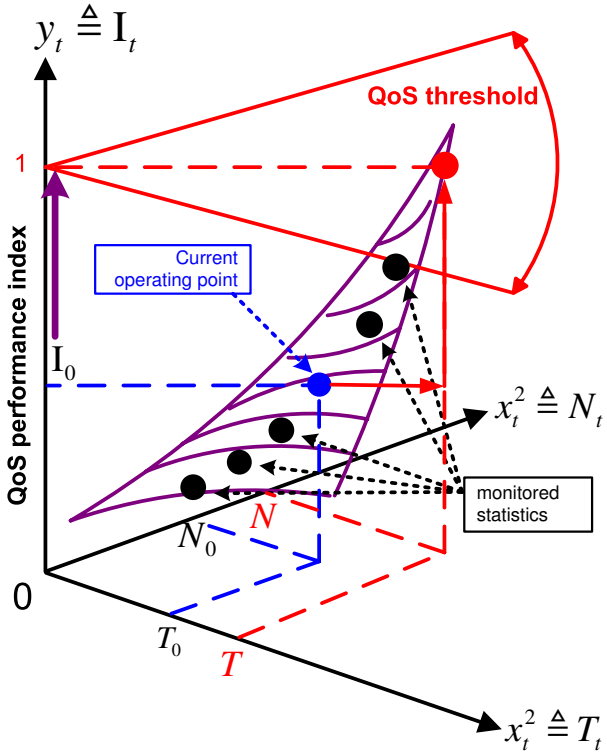

Figure 3.3: An illustrative example for the shape of curve produced by the mapping f, where the inputs are considered as two dimentions, *i.e.,* $M = 2$**.**

where $\mathcal{N}$ denotes the maximum number of ongoing connections and $\mathcal{T}$ denotes the maximum throughput the subnetwork can support. The remainder of this section aims to describe the methodology for estimating such subnetwork capacity by "virtually" *projecting* the operational status of a particular subnetwork towards the capacity bound, *i.e.,*

$$\underline{x}_0 \to \underline{C}_0, \tag{3.11}$$

as shown in Figure 3.2(b), beyond which at least one of the existing connections cannot maintain satisfactory QoS. This capacity bound $\underline{C}_0$ is achieved when the maximum tolerable QoS performance index approaches the value 1, *i.e.,*

$$\widetilde{\mathrm{I}}_0 = 1. \tag{3.12}$$

3.3.3 Second Order Capacity Approximation

We propose an *iterative* methodology for subnetwork capacity estimation, projecting multiple *virtual* connections into the considered subnetwork to force the system operation to change from the current operating point $\underline{x}_0 \triangleq (N_0, T_0)$ to the capacity bound $\underline{C}_0 \triangleq (\mathcal{N}, \mathcal{T})$ through several steps, as shown in Figure 3.2(b). Each step k corresponds to an input change of:

$$\Delta \underline{x}_0^k = \left(\Delta N_0^k, \Delta T_0^k\right) = (1, T_j^{r,k}) \tag{3.13}$$

by injecting one virtual connection with throughput requirement $T_j^{r,k}$ of service class $j \in \mathcal{J}$. By "virtual", we mean the connection is not physically requested by any application, but used for computation purpose only. The stochastic nature of the connection characteristics (*e.g.,* the arrival process, the departure process, the throughput requirement, etc.) make this method quite appealing and suitable for Taylor expansion, where it assumes the accurate expansion within the neighboring regions of the operating point. As shown in Figure 3.2(b), suppose that at the first step $k = 1$, a virtual connection is generated such that an input change $\Delta \underline{x}_0^1 = \left(\Delta N_0^1, \Delta T_0^1\right) = (1, T_j^{r,1})$ is incurred, resulting in an output change to,

$$\widetilde{\mathrm{I}}_0^1 = f\left(N_0 + 1, T_0 + T_j^{r,1}\right), \forall j \in \mathcal{J}. \tag{3.14}$$

We rewrite (3.8) as,

$$\widetilde{\mathrm{I}}_0^1 = \mathrm{I}_0 + f'^{,1}_{N_t} + T_j^{r,1} f'^{,1}_{T_t} + \frac{1}{2} f''^{,1}_{N_t} + \frac{(T_j^{r,1})^2}{2} f''^{,1}_{T_t} + T_j^{r,1} f''^{,1}_{N_t T_t}, \tag{3.15}$$

where $f'^{,1}$ and $f''^{,1}$ denote partial derivatives taken at the initial state $\underline{x}_0 = (N_0, T_0)$ for the first step, as shown Figure 3.2(b).

In order to get the second order partial derivative $f''^{,1}_{N_t T_t}$, it is worth to note that the chosen two input variables are dependent with each other, since the total currently served throughput is a function of the number of ongoing connections, as $T_0 = \sum_{q=1}^{N_0} T_q^r$.

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

$$\left.\frac{\partial T_t}{\partial N_t}\right|_{\underline{x}_0} = \lim_{\Delta N_0^1 \to 1} \frac{\sum_{q=1}^{N_0+\Delta N_0^1} T_q^r - \sum_{q=1}^{N_0} T_q^r}{\Delta N_0^1} = \lim_{\Delta N_0^1 \to 1} \frac{\sum_{q=N_0+1}^{N_0+\Delta N_0^1} T_q^r}{\Delta N_0^1} = T_j^{r,1}. \tag{3.16}$$

Therefore,

$$\left. f''^{,1}_{N_t T_t} \right|_{\underline{x}_0} = f''^{,1}_{T_t} \left.\frac{\partial T_t}{\partial N_t}\right|_{\underline{x}_0} = T_j^{r,1} \left. f''^{,1}_{T_t} \right|_{\underline{x}_0}, \forall j \in \mathcal{J}. \tag{3.17}$$

Replace (3.17) back to (3.15) and we rewrite it as,

$$\widetilde{\mathrm{I}}_0^1 = \mathrm{I}_0 + f'^{,1}_{N_t} + T_j^{r,1} f'^{,1}_{T_t} + \frac{1}{2} f''^{,1}_{N_t} + \frac{3}{2} (T_j^{r,1})^2 f''^{,1}_{T_t}. \tag{3.18}$$

So far, step $k = 1$ demonstrates the process of updating the output of the mapping f from I_0 to $\widetilde{\mathrm{I}}_0^1$, when the network status is updated from $\underline{x}_0 = (N_0, T_0) \in \mathbb{R}^2$ to $\underline{x}_0^1 = (N_0^1, T_0^1) = (N_0 + 1, T_0 + T_j^{r,1}) \in \mathbb{R}^2$. In other words, the subnetwork is "virtually" operating at state $\underline{x}_0^1$. However, the QoS performance index $\widetilde{\mathrm{I}}_0^1$ may not reach its upper bound 1 when the considered subnetwork reaches its capacity bound, and thus we need to repeat the same projection process in a best effort based on current state $\underline{x}_0^1 = (N_0^1, T_0^1)$, as the step $k = 2$:

$$\widetilde{\mathrm{I}}_0^2 = \mathrm{I}_0^1 + f'^{,2}_{N_t} + T_j^{r,2} f'^{,2}_{T_t} + \frac{1}{2} f''^{,2}_{N_t} + \frac{3}{2} (T_j^{r,2})^2 f''^{,2}_{T_t}, \tag{3.19}$$

where $\Delta \underline{x}_0^2 = (\Delta N_0^2, \Delta T_0^2) = (1, T_j^{r,2}), \forall j \in \mathcal{J}$, and all partial derivatives are computed at $\underline{x}_0^1 = (N_0^1, T_0^1)$.

Without loss of generality, we assume that we are able to repeat this estimation process $K+1$ times and find $\widetilde{\mathrm{I}}_0^{K+1} > 1$, however within the previous K steps the condition $\widetilde{\mathrm{I}}_0^k \leq 1, \forall k = 1, 2, \ldots, K$ always satisfies. In other words, the *projected* status of subnetwork $\underline{x}_0^K$ well closes to the defined subnetwork capacity $\underline{C}_0$, *i.e.*, $\underline{C}_0 \approx \underline{x}_0^K$. While each projection

step produces one equation like (3.15) and (3.19), K steps produce a set of equations as:

$$\begin{cases} \widetilde{\mathrm{I}}_0^K = \widetilde{\mathrm{I}}_0^{K-1} + f'^{,K}_{N_t} + T_j^{r,K} f'^{,K}_{T_t} + \frac{1}{2} f''^{,K}_{N_t} + \frac{3}{2}(T_j^{r,K})^2 f''^{,K}_{T_t} \\ \cdots\cdots \\ \widetilde{\mathrm{I}}_0^2 = \widetilde{\mathrm{I}}_0^1 + f'^{,2}_{N_t} + T_j^{r,2} f'^{,2}_{T_t} + \frac{1}{2} f''^{,2}_{N_t} + \frac{3}{2}(T_j^{r,2})^2 f''^{,2}_{T_t} \\ \widetilde{\mathrm{I}}_0^1 = \mathrm{I}_0 + f'^{,1}_{N_t} + T_j^{r,1} f'^{,1}_{T_t} + \frac{1}{2} f''^{,1}_{N_t} + \frac{3}{2}(T_j^{r,1})^2 f''^{,1}_{T_t} \end{cases}$$

By solving the above set of equations, we have,

$$\mathrm{I}_0 + \sum_{k=1}^{K} \left(f'^{,k}_{N_t} + T_j^{r,k} f'^{,k}_{T_t} + \frac{1}{2} f''^{,k}_{N_t} + \frac{3}{2}(T_j^{r,k})^2 f''^{,k}_{T_t} \right) \leq 1, \tag{3.20}$$

where partial derivatives $f'^{,k}, f''^{,k}$ at taken at state $\underline{x}_0^k$.

Remark 3.3.1. ***The Essense of the Proposed Capacity Estimation***

The above inequality (3.20) uniquely captures the essence for this subnetwork capacity estimation through iteratively and virtually generating traffic to project *the status of subnetwork from the current operating point all the way towards the capacity bound, as shown in Figure 3.2(b):*

$$\underline{x}_0 \xrightarrow{\Delta \underline{x}_0^1} \underline{x}_0^1 \xrightarrow{\Delta \underline{x}_0^2} \underline{x}_0^2 \ldots \xrightarrow{\Delta \underline{x}_0^K} \underline{x}_0^K \longrightarrow \underline{C}_0. \tag{3.21}$$

Correspondingly, the output of the mapping f, i.e., I_0^k, evolves as:

$$\mathrm{I}_0 \xrightarrow{f'^{,1}, f''^{,1}} \widetilde{\mathrm{I}}_0^1 \xrightarrow{f'^{,2}, f''^{,2}} \widetilde{\mathrm{I}}_0^2 \ldots \xrightarrow{f'^{,K}, f''^{,K}} \widetilde{\mathrm{I}}_0^K \longrightarrow 1. \tag{3.22}$$

In order to solve the inequality (3.20), we notice that partial derivatives $f'^{,k}_{T_t}, f''^{,k}_{T_t}$ are random variables due to the stochastic nature of traffic throughput requirement $T_j^{r,k}$ that is virtually projected at each step k. However, if we look at partial derivatives $f'^{,k}_{N_t}, f''^{,k}_{N_t}$, interestingly they are observed as *fixed* inputs, which can be directly derived from the mapping f since for each step k and for each virtual connection input, $\Delta N_0^k = 1$ is fixed.

Now, we are able to derive the subnetwork capacity $\underline{C}_0 = (\mathcal{N}, \mathcal{T})$ when the state

of subnetwork operates in the capacity bound of step K, as:

$$\begin{cases} \mathcal{T} = T_0 + \sum_{k=1}^{K} T_j^{r,k}, \forall j \in \mathcal{J}, \\ \mathcal{N} = N_0 + K. \end{cases} \tag{3.23}$$

Replace (3.23) back to (3.20), and we take the expectation as the first order approximation, we have:

$$\mathrm{I}_0 + \sum_{k=1}^{\mathcal{N}-N_0} f_{N_t}^{\prime,k} + \mathbb{E}\left(\sum_{k=1}^{\mathcal{N}-N_0} T_j^{r,k} f_{T_t}^{\prime,k}\right) + \frac{1}{2}\sum_{k=1}^{\mathcal{N}-N_0} f_{N_t}^{\prime\prime,k} + \frac{3}{2}\mathbb{E}\left(\sum_{k=1}^{\mathcal{N}-N_0}\left(T_j^{r,k}\right)^2 f_{T_t}^{\prime\prime,k}\right) \leq 1. \tag{3.24}$$

We further apply Schwarz's Inequality [77] to approximate:

$$\begin{cases} \mathbb{E}\left(\sum_{k=1}^{\mathcal{N}-N_0} T_j^{r,k} f_{T_t}^{\prime,k}\right) < \sum_{k=1}^{\mathcal{N}-N_0} \mathbb{E}^{\frac{1}{2}}\left(T_j^{r,k}\right)^2 \mathbb{E}^{\frac{1}{2}}\left(f_{T_t}^{\prime,k}\right)^2, \\ \mathbb{E}\left(\sum_{k=1}^{\mathcal{N}-N_0} (T_j^{r,k})^2 f_{T_t}^{\prime\prime,k}\right) < \sum_{k=1}^{\mathcal{N}-N_0} \mathbb{E}^{\frac{1}{2}}\left(T_j^{r,k}\right)^4 \mathbb{E}^{\frac{1}{2}}\left(f_{T_t}^{\prime\prime,k}\right)^2, \end{cases} \tag{3.25}$$

where the higher order statistics of the throughput requirement $T_j^{r,k}$ for each step k could be derived using the connection throughput distribution ϕ_j^T for the service class j, as:

$$\begin{cases} \mathbb{E}\left(T_j^{r,k}\right)^2 = \sum_{\forall j \in \mathcal{J}} p_j \int (T_j^r)^2 \phi_j^T dT_j^r, \\ \mathbb{E}\left(T_j^{r,k}\right)^4 = \sum_{\forall j \in \mathcal{J}} p_j \int (T_j^r)^4 \phi_j^T dT_j^r. \end{cases} \tag{3.26}$$

We assume ϕ_j^T is the probability density function of throughput requirement for service class j, and known as *a priori* from empirical analysis. Then, our problem is reduced to,

$$\begin{aligned} \frac{1}{2}\sum_{k=1}^{\mathcal{N}-N_0} f_{N_t}^{\prime\prime,k} \quad &+ \quad \sum_{k=1}^{\mathcal{N}-N_0} \mathbb{E}^{\frac{1}{2}}\left(T_j^{r,k}\right)^2 \mathbb{E}^{\frac{1}{2}}\left(f_{T_t}^{\prime,k}\right)^2 \\ &+ \quad \sum_{k=1}^{\mathcal{N}-N_0} f_{N_t}^{\prime,k} + \sum_{k=1}^{\mathcal{N}-N_0} \mathbb{E}^{\frac{1}{2}}\left(T_j^{r,k}\right)^4 \mathbb{E}^{\frac{1}{2}}\left(f_{T_t}^{\prime\prime,k}\right)^2 \leq 1 - \mathrm{I}_0. \end{aligned} \tag{3.27}$$

Iteratively examine $k = 1, 2, 3, \cdots$ till (3.27) does not hold anymore, then we obtain the step index K used for the upper bound solution of subnetwork capacity dimension one

$\mathcal{N} = N_0 + K$. After, we show that the subnetwork capacity $\underline{C}_0$ is approximated by:

$$\begin{cases} \mathcal{T} \approx T_0 + (\mathcal{N} - N_0) \sum_{\forall j \in \mathcal{J}} p_j \int T_j^r \phi_j dT_j^r, \\ \mathcal{N} = N_0 + K. \end{cases} \tag{3.28}$$

as the expectation approximation of (3.23).

3.3.4 First Order Capacity Approximation

If the shape of curve for 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, we simplify (3.15) as:

$$\widetilde{\mathrm{I}}_0^1 = \mathrm{I}_0 + f_{N_t}^{\prime,1} + T_j^{r,1} f_{T_t}^{\prime,1}, \forall j \in \mathcal{J}. \tag{3.29}$$

Again we perform the iterative estimation method to project the status of subnetwork towards the capacity bound. As a result, our problem is reduced to,

$$\sum_{k=1}^{\mathcal{N}-N_0} f_{N_t}^{\prime,k} + \sum_{k=1}^{\mathcal{N}-N_0} \mathbb{E}^{\frac{1}{2}} \left(f_{T_t}^{\prime,k} \right)^2 \mathbb{E}^{\frac{1}{2}} \left(f_{T_t}^{\prime,k} \right)^2 \leq 1 - \mathrm{I}_0, \tag{3.30}$$

where partial derivatives $f_{T_t}^{\prime,k}, f_{N_t}^{\prime,k}$ at taken at state $\underline{x}_0^k = (N_0^k, T_0^k)$. Finally, we iteratively examine $k = 1, 2, 3, \cdots$ till (3.30) does not hold, then we obtain the upper bound solution for $\mathcal{N} = N_0 + K$, and subnetwork capacity $\underline{C}_0$ as,

$$\begin{cases} \mathcal{T} \approx T_0 + (\mathcal{N} - N_0) \sum_{\forall j \in \mathcal{J}} p_j \int T_j^r \phi_j dT_j^r, \\ \mathcal{N} = N_0 + K. \end{cases} \tag{3.31}$$

3.4 QoS Control Algorithm

Our proposed generic AC (GAC) scheme for QoS control is initialized when the new connection q arrives at any ingress node v_i with multiple QoS constraints, intended to communicate with the egress node v_e. The functionality is summarized and described as follows.

1. **Statistics collection:** the egress node v_e periodically keeps track of system-level and traffic-level statistics within the considered $v_i - v_e$ subnetwork from time to time, including the number of ongoing connections N_t, total served throughout T_t, and observable QoS performance index $\mathrm{I}(q), \forall q \in \mathcal{Q}$. Then, by using exponential smoothing in (3.4), the shape of curve is constructed and I_t is obtained.

2. **Subnetwork capacity estimation:** $\underline{C}_t$ can be P-dimensional vector, but for simplicity and concrete analysis, it may reduce to a two-dimensional vector referring to the maximum served throughput $\mathcal{T}$ and the number of ongoing connections $\mathcal{N}$ a subnetwork can support, and it is approximated by taking both the first and second partial derivatives in (3.28), or only the first order derivatives in (3.31), at iterative status points as inputs. These derivatives are computed from the mapping $y_t = f(\underline{x}_t)$.

3. **GoS management:** to differentiate different GoS, preferential treatments to both regular and premium services are provided, *i.e.,* $\mathcal{J} \triangleq \{1, 2\}$. We introduce a concept of *capacity usage indicators* $\{\overline{\alpha}_1, \underline{\alpha}_1, \overline{\alpha}_2, \underline{\alpha}_2\}$ to denote higher ($\overline{\alpha}$) and lower ($\underline{\alpha}$) usage bounds of overall subnetwork capacity for each service class in set $\mathcal{J}$. Especially lower bound $\underline{\alpha}_2$ reserves a portion of subnetwork capacity for premium users to guarantee QoS as shown in Figure 3.4. The typical values are

$$\overline{\alpha}_1 = 0.8, \quad \underline{\alpha}_1 = 0, \quad \overline{\alpha}_2 = 0.5, \quad \underline{\alpha}_2 = 0.2. \tag{3.32}$$

4. **Admission control:** this step is to verify if there is enough network resource within the subnetwork available for the new connection, when it arrives at time $t = t_0$. If the new connection belongs to premium service class, the amount of remaining capacity $\mathcal{T}^p$ can be computed as:

$$\mathcal{T}^p = \overline{\alpha}_2 \mathcal{T} - T_0^p, \tag{3.33}$$

where T_0^p denotes the total served throughput for premium connections. Similarly, if

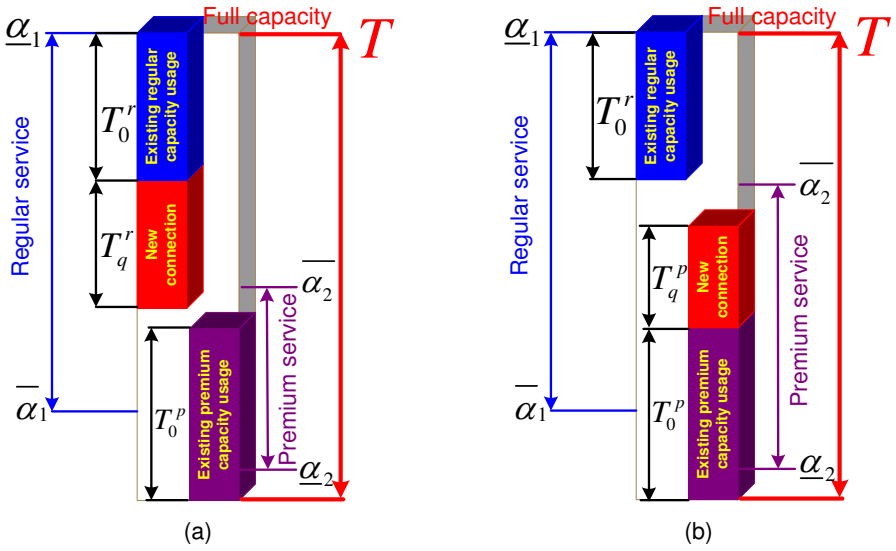

Figure 3.4: Two illustrative examples for subnetwork capacity usage indicators, where we suppose two service classes, *i.e.*, $\mathcal{J} = \{1, 2\}$, coexist in the subnetwork, where (a) shows that the case that the newly arrived connection belongs to the regular service class, and (b) hows that the case that the newly arrived connection belongs to the premium service class.

the new connection belongs to regular service class, the amount of remaining capacity $\mathcal{T}^r$ can be computed as:

$$\mathcal{T}^r = \overline{\alpha}_1 \mathcal{T} - T_0^r. \tag{3.34}$$

Then, the admission decision variable $\tau(q)$ is chosen by,

$$\tau(q) = \begin{cases} 1, & \text{if } T_q^r \leq \mathcal{T}^p, \mathcal{N} > N_0, \text{ premium service,} \\ 1, & \text{if } T_q^r \leq \mathcal{T}^r, \mathcal{N} > N_0, \text{ regular service,} \\ 0, & \text{otherwise.} \end{cases}$$

If admission decision variable $\tau(q) = 0$, the subnetwork fails providing enough network resources for satisfactory QoS and the connection is unfortunately rejected. Otherwise, the connection is admitted.

Since each ingress node (*e.g.*, the network router) maintain its own QoS performance index and subnetwork capacity estimation for each corresponding egress node in the network, the proposed methodology for QoS control is fully distributed. Moreover, it can be easily

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	2 km
Traffic Patterns	FTP, VoIP, and Video
Queue Length	Infinite

extended to other input variables to cast in more accurate representations of traffics, such as traffic delay distributions, jitter, packet length distributions, etc.

3.5 Simulation Results

We developed a cross-layer MATLAB simulator to assess the proposed capacity estimation and QoS control algorithm. Two service classes, regular and premium, *i.e.,* $\mathcal{J} \triangleq \{1, 2\}$, are assumed in the considered network to guarantee two levels of GoS, where connections for each service class $j \in \mathcal{J}$ 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, [71]) is used in network/MAC layers to provide sub-optimal solution for QoS. Rayleigh fading channel model [74], adaptive modulation and coding scheme, and directional antennas are used in PHY Layer to improve channel capacity and frequency reuse efficiency and to reduce the interference to adjacent concurrent transmissions. The simulation parameters are summarized in Table 3.1.

We need to highlight that the methodology is not constrained in WMNs, but rather

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

	First order only statistics	**First and second order statistics**
Average $\mathcal{T}$	22Mbps	25Mbps
Error Bound	±2Mbps	±500Kbps
Average $\mathcal{N}$	30	35
Error Bound	±5	±2
QoS Outage	≈ 13%	≈ 10%
Blocking Prob.	≈ 12%	≈ 8%

has wide applicability in other wireless and wired IP networks as discussed in Section 3.6.

3.5.1 An Example: Five Node WMN

We first assess our proposed capacity estimation and QoS control algorithm in a simple five-node WMN setting in Fig. 3.5, 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 for subnetwork capacity estimation, namely: to use (1) only first order partial derivatives, and (2) both first and second order partial derivatives. It is interesting to see that second order statistics successfully improve the volume of estimated subnetwork capacity 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 mapping f, especially when the state of subnetwork operates around the capacity bound where one single large connection admission 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 upper bound capacity, while maintaining QoS satisfactions to all ongoing connections.

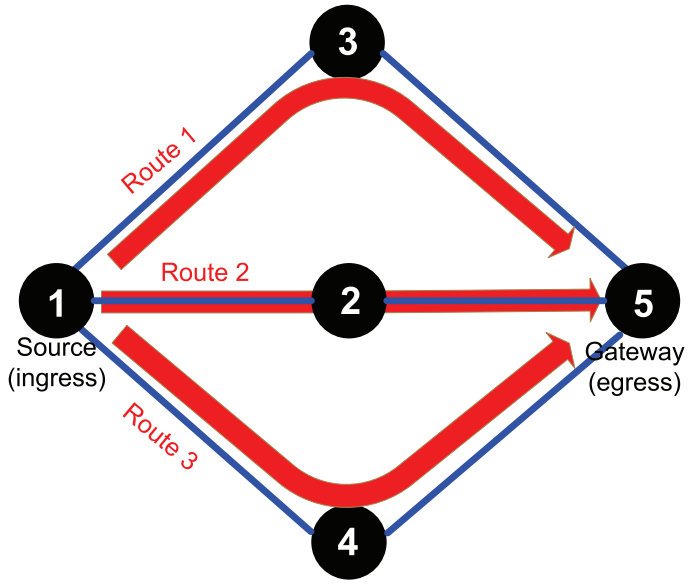

Figure 3.5: A five-node WMN setting, where node 1 serves as the source (an ingress node) to generate connections with multiple QoS requriemetns and node 5 serves as the gateway (an egress node). Three disjoint routes exist between then ingress-to-egress node pair to carry the traffic.

3.5.2 Overall Performance Evaluations

The proposed algorithm, referred as "IQoSR+GAC", is compared with existing work "IQoSR" [71] that does not include different GoS treatments and AC, and compared with the statistical admission control algorithm in the literature (SAC, [70]), referred as "IQoSR+SAC". It is also compared with conventional protocol layer 2 and layer 3 techniques: the round robin scheduler (RR, [75]) and AODV routing protocol [78]. The overall performance are investigated in terms of the overall gateway goodput in Figure 3.7, the average QoS outage probability of completed connections in Figure 3.8, and the connection blocking probabilities for premium and regular users in Figure 3.9. A complete simulation topology is shown in Figure 3.6.

Figure 3.7 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 compared with "IQoSR+SAC", "IQoSR" schemes, and the lower-bound "RR+AODV" scheme, thanks to the accurate justification of the network resource status through subnetwork capacity

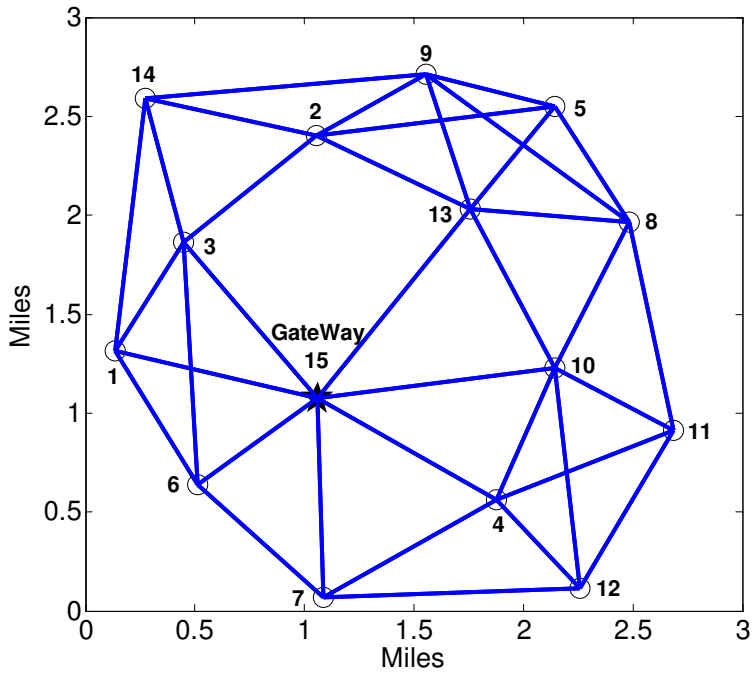

Figure 3.6: A complete simulation setting with 15 mesh routers and one gateway node randomly deployed.

estimation and the preferential treatments (or GoS management) for different service classes. On the other hand, 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. It is also interesting to observe that the gateway goodput saturates when the traffic load becomes higher. Finally, when we increase the node density in a fixed network area, 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 3.8 illustrates the QoS outage probability, defined as the probability of any connection's QoS requirements to fail during their lifetime, or the condition $\mathrm{I}(q) \leq 1, \forall q \in \mathcal{Q}$ will not satisfied at all. It can be seen that the proposed QoS control scheme can

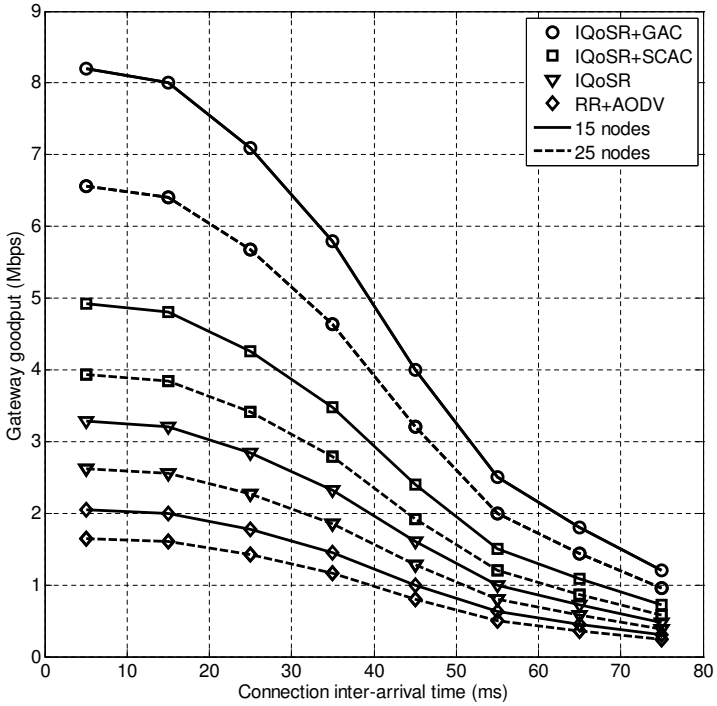

Figure 3.7: Simulation result of the overall gateway goodput w.r.t. the different new connection inter-arrival time and the number of nodes.

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 has been estimated and accurately reflected in the parameter of subnetwork capacity which reflects the status of subnetwork to support satisfactory QoS. Meanwhile, since no resource reservation scheme is provided for premium users in all other schemes, admitted premium connections with higher QoS requirements may easily exceed the capacity bound that in turn jeopardizes ongoing ones' QoS.

Figure 3.9 shows how preferential treatments for premium and regular users impact on the behavior of the average connection blocking probability, as we change the lower bound capacity usage $\underline{\alpha}_2$ for premium users. We fix the traffic arrival rate in this simulation. As $\underline{\alpha}_2$ increases, more resources are reserved, and lower connection blocking probability is expected. It is very interesting to see an *optimum*, or the lowest connection blocking portability point for all connections, around 15% when $\underline{\alpha}_2 \in [0.4, 0.5]$. We also

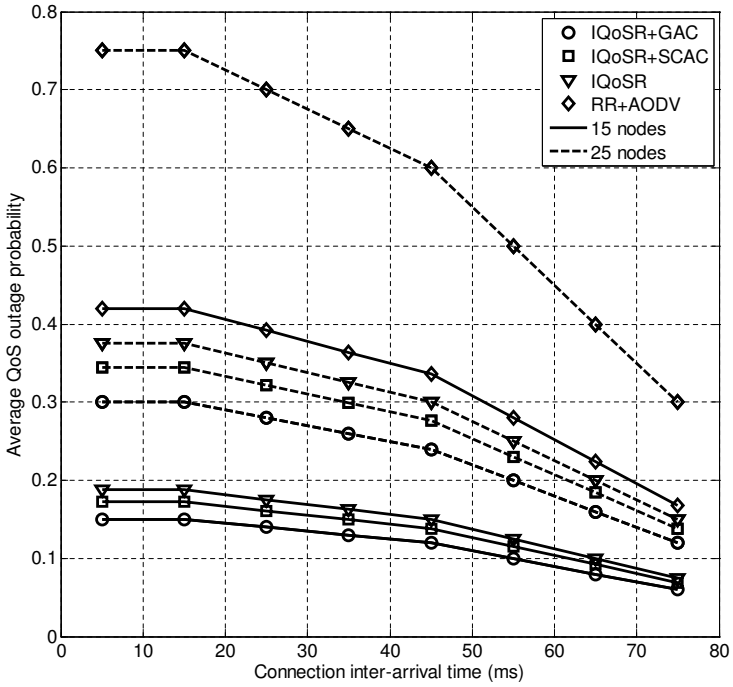

Figure 3.8: Simulation result of the average QoS outage probability w.r.t. the different new connection inter-arrival time and the number of nodes.

prove our justification from this figure that node density indeed impacts on QoS outage, due to the nature of resource sharing and unavoidable co-channel interference, as seen consistent with Figure 3.7 and Figure 3.8.

3.6 Discussions

This section provides a discussion on the applicability and feasibility issues of the proposed capacity estimation and QoS control methodology.

3.6.1 The Applicability

The proposed GAC methodology has wide applicability to many packet networks, wired and/or wireless. The main essence of this methodology is to use a generic mapping f to represent the resource space between the source and destination, so that we are able to predict the time varying *subnetwork capacity* $\underline{C}_t \in \mathbb{R}^P$ at runtime. The method itself does not require any knowledge of detailed network protocols to use at any node within

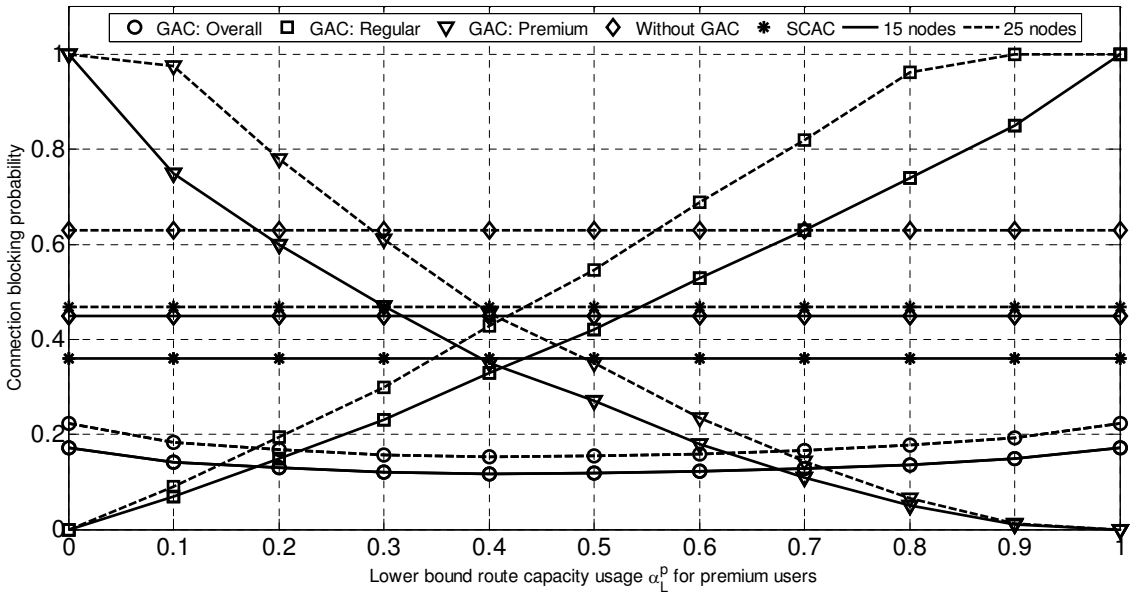

Figure 3.9: Simulation result of the connection blocking probability (for the fixed the traffic arrival rate), w.r.t. the lower capacity usage bound reserved for premium users.

the network as *a priori*, however the benefit each connection can receive is constantly probed by the egress node and feedbacks to the ingress node. The feedback loop itself dynamically adapts the system behavior and the network status from time to time, and the key parameter generated will be used for QoS control.

The procedure is to consider traffic-level and system-level parameters as inputs and the observable QoS performance index as an output, and then we use a closed-form Taylor approximation to estimate. In order to facilitate this estimation, only first and second order partial derivatives need to be computed at the egress node when a connection completes, which could be easily obtained through limitation expressions, *i.e.,* first order derivatives can be derived by using three adjacent measurements on the shape of curve produced by the mapping f, while second order statistics can be derived by four adjacent measurements. If the mapping f is smooth enough around the operating point so that the second order statistics are negligible, the estimation process could further be simplified. Furthermore, this methodology provides full transparency for lower layer technologies to use, for instance, routing, scheduling, and PHY layer techniques. 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.

Generalization of the Algorithm:

In the previous sections our modeling is primarily focused on a single subnetwork which is represented by a black box with a mapping f to map multiple inputs to a single output. However, a packet network comprises several pairs of ingress and egress nodes, corresponding to multiple subnetworks. Since those subnetworks have to share the available network resources, the AC in one subnetwork may impact the subnetwork capacities of the adjacent subnetworks. Nevertheless, this impact is sufficiently reflected on the QoS performance index of each subnetwork and can also be sufficiently addressed by updating this index in each subnetwork just before the AC procedure takes place. However, response lags in the subnetwork for any traffic changes (for instance, sometime it is required for the buffer sizes to change, corresponding to the new queueing and ETE delay), can lead to inaccurate measurements and estimations of the QoS performance index.

3.6.2 The Feasibilities of the Proposed Approach

Subnetwork Capacity versus Connection Throughput Requirement:

As mentioned before, the Taylor expansion is used as a tool to estimate the subnetwork capacity, however, it is feasible given that statistics are sufficiently collected within a relatively close region of the operating point. Although *iterative* estimation steps are proposed to tackle this problem (*i.e.,* to virtually inject only one small connection), 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 will impact the accuracy of the partial derivative derivations and ultimately the estimation of subnetwork capacity. The effect is depicted in Figure 3.10 where the ratio between the average connection throughput requirement T_q^r and the subnetwork capacity on throughput dimension $\mathcal{T}$, *i.e.,* $\mathbb{E}(T_q^r)/\mathcal{T}$, is plotted against the QoS outage probability. The larger

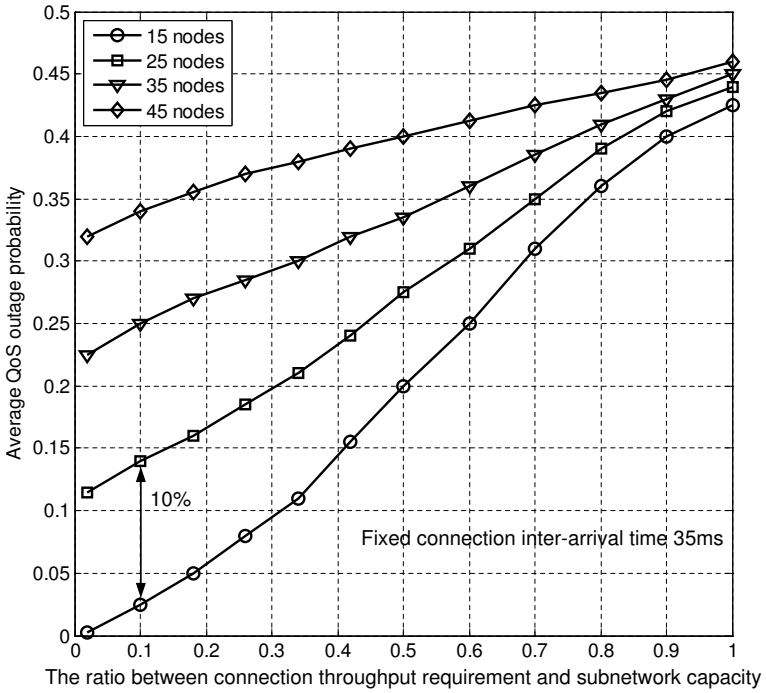

Figure 3.10: The impact of the connection throughput requirement on the estimation of subnetwork capacity, w.r.t. either different new connection inter-arrival time or node densities.

the ratio is, the more severe impact on error accumulation and amplification for subnetwork capacity estimation will be. Therefore, connections' QoS are more vulnerably to be violated, especially when the subnetwork is operating near the capacity bound where the careless admission of one single large connection may jeopardize all existing connections' QoS. This is why QoS outage probability increases significantly when the ratio goes to 1. On the other hand, when more ingress-to-egress node pairs are deployed in a given network area, more interference will be created among multiple subnetworks, and thus QoS outage probability is increased as a result of the decrease of subnetwork capacity.

Statistics Feedback Delay:

Figure 3.11 shows the impact of statistics feedback delay (from the egress node to the ingress node) on the average QoS outage probability. Higher layer protocol (*e.g.,* through TCP acknowledgement packets) helps to pass the required statistics within half of the round trip time. Nevertheless, outdated statistics incurred by higher feedback delay may

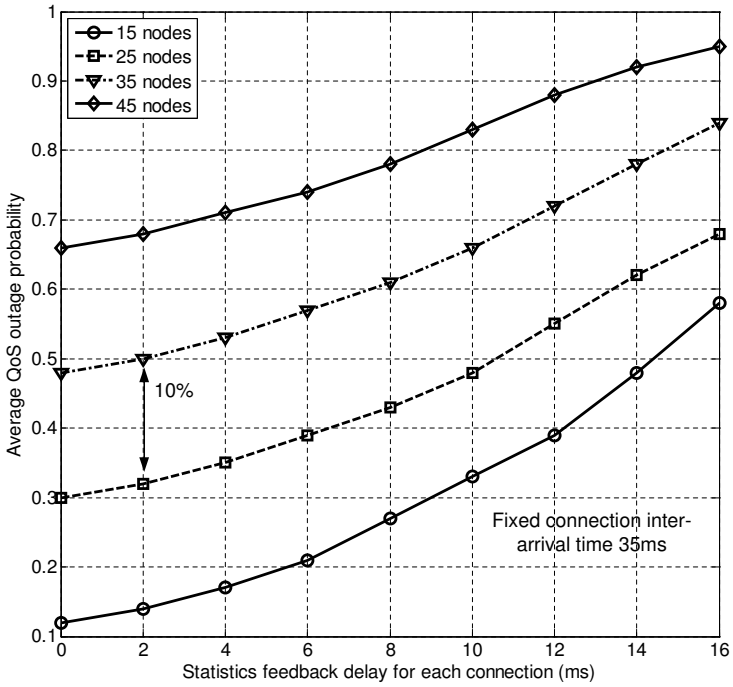

Figure 3.11: The impact of the statistics feedback delay on the estimation of subnetwork capacity, w.r.t. either different new connection inter-arrival time or node densities.

result in the slow reaction of the ingress node to be aware of the subnetwork status. As a result, imperfect AC decisions would be made based on the inaccurate network status that can effect the QoS of the 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.

Statistics Collection Time:

This corresponds to the time window at the ingress node that the collected statistics are periodically collected or discarded to construct the mapping f. If the collection time is relatively long, historical data may not represent the most recent subnetwork 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 3.12 where for fixed connection inter-arrival time, there is an *optimal* statistics collection time that

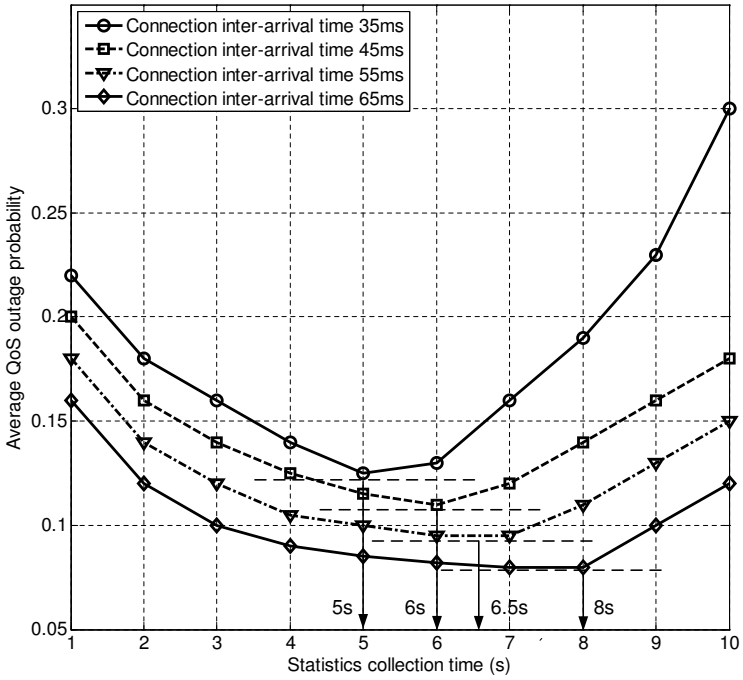

Figure 3.12: The impact of the statistics collection time on the estimation of subnetwork capacity, w.r.t. either different new connection inter-arrival time or node densities.

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.

Lightly Loaded System:

The proposed capacity estimation and QoS control algorithm relies on the continuous statistics collection for the construction of the mapping $y_t = f(\underline{x}_t)$. However, when the network is lightly loaded, there might not be enough statistics on the observation space for subnetwork capacity estimation. This can be solved by injecting test packets into the subnetwork to obtain necessary statistics. This is important because current lightly loaded system cannot guarantee future traffic arrival process, particularly when the bursty traffic may make the whole system malfunction. Nevertheless, in backhaul networks, a reasonable loaded system can be assumed since the network routers will serve the aggregated traffic from multiple sources.

3.7 Summary

In this Chapter, a generic capacity estimation and QoS control methodology for all packet networks, wired and/or wireless, is proposed. First, a novel concept of QoS performance index is introduced to indicate the degree of QoS satisfaction for any connection in the network. Second, the subnetwork capacity is defined to uniquely indicate the degree of resource occupancy between a pair of ingress and egress nodes, or within the black box. Third, a generic mathematical function is used to represent this black box, and the time varying subnetwork capacity is analyzed. Fourth, an AC scheme for QoS control is proposed to make admission decision for the new connection while preferential treatments for different user groups are introduced to maintain GoS. Last, the applicability and feasibility issues for any given packet networks are also discussed. Finally, extensive simulations are performed in WMNs to show that the proposed AC framework can successfully coordinate PHY, MAC and network layer resources to efficiently accommodate higher number of connections with satisfactory QoS and GoS. Future work includes the quantification of the performance due to feedback delay and statistics collection time.

Chapter 4

Network Operations and Management through Negotiations

AS presented in Chapter 2 and Chapter 3 where the dissertation mainly tackles the network protocol designs of multi-hop wireless networks, in this Chapter, this dissertation investigates the research issue of network operation and management (O&M). The proposed network O&M framework builds on top of the network protocols, to provide the optimal overall network performance through resource allocations and negotiations to support the service quality.

To demonstrate the proposed approach, we use wireless sensor networks (WSNs) as the illustrative example, where the notion of the 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 [79, 80] 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, and spatiotemporal relevancy [81].

Specifically, this Chapter is primarily motivated by the novel research area of how to support service quality in terms of QoI in WSNs to dynamically adjust network resource allocations (or "internal" operations) and adapt applications' service quality demands (or

"external" operations) to achieve the optimal task operations. This Chapter thus identifies such a research gap between the "external" and "internal" operations which are modeled as a *negotiation* process. The overall goal of this negotiation process is to control the overall QoI levels provided to 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*-based admission control process that relies on iteratively reconfiguring and optimizing the usage of network resources and the degree of QoI acceptance of prioritized tasks;
4. a resource allocation scheme, which optimally allocates network resources for both the existing and new 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 [3]. A significant portion of research in the area of WSN deployment and operation, which we refer to as *operation 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 [3]. 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 are central of our broader research pursues in general and this Chapter in particular.

Motivating our work in bridging between the afore-mentioned internal and external aspects of WSNs is the increasing body of research in the O&M area that uses network utility analysis techniques. These techniques strive 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 [34,35] 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 behavior 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.

We address the afore-mentioned challenges by proposing a QoI-aware O&M framework for WSNs, a novel research path in its own right. Our general approach is to separate the process of calculating the QoI performance of the network at large from that of calculating utility resulting from allocating network resources to individual tasks. First, 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 heterogenous 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. Third, an adaptive, negotiation-based admission control mechanism is proposed to dynamically configures the usage of network resources to best accommodate all tasks' QoI requirements. Finally, an evaluation of the WSN QoI performance at runtime in a dynamic multi-task environment is presented.

The rest of Chapter 4 is organized as follows. In Section 4.2, related research activities are highlighted. Section 4.3 describes the deployment view of WSNs, which is

the basis for this Chapter, and Section 4.4 presents the system model. Section 4.6 describes four key design elements for such a QoI-aware O&M framework. Numerical results and discussions are demonstrated in Section 4.7. Finally, we conclude in Section 4.8 with a summary.

4.2 Related Work

To the best of our knowledge, the proposed QoI-aware O&M framework for WSNs represents the first such WSN application management solution of its kind. However, there is related work that has motivated our current research path. Despite of endeavors for defining QoI [79, 80], it was not until recently that work in [82] proposed a conceptual framework to enable the dynamic binding of sensor information producers and consumers in QoI-aware manner. The framework expresses information requirements and capabilities according to the 5WH principle and enables information producers to categorize the quality attributes of their information in an application-agnostic manner while permitting information consumers to calculate 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 [34]. This is further addressed in [35], 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 modeling 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 [83] and monitoring network resource allocations and the status of sensed phenomena to determine available QoI [84] and sustain required QoI [85]. Sensor network management issues were studied in [86, 87], where in [87] 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, [86] further compared the solution with Bayesian network model.

In closing we also mention here work on WSN middleware designs [88] to support some notion of information quality [89–91]; the latter work has inspired aspect of our research in the area. In closing, we note that early thoughts behind this research path were reported in [92], but without any of the technical depth and numerical results included in this Chapter.

4.3 The Deployment View of WSNs

Figure 4.1 presents two deployment views of a WSN, where a functional view of a real WSN deployment is illustrated to the left, and a layered stack of operations is presented to the right. Both views are applicable likewise to an end-to-end sensor/transport/fusion/application system.

On the layered stack, the *data acquisition* and *application(s)/middleware* layers describe optional operations at a particular layer. The *data transport* and *system level fusion* layers represent optional layers that might not exist for one-off WSN solutions, where a single sensor system is integrated with its own application (*e.g.,* a simple robot), in this case, the entire stack collapses down to two layers, data acquisition and the application layers–where the application may be responsible for any sensor data manipulation/fusion. However, for highly *distributed*, multi-sensor systems feeding information to multiple applications/missions, every layer stack shown should be present. In this set-up, the term "measurements" applies to the sensed data collected by the sensors serving as data sources. Nevertheless, the information leaving these data sources and transported through the multi-hop WSNs are called "sensor observations." A sensor observation does not necessarily represent raw measurements but could be the outcome of the fusion of sensor measurements at various sensors. Sensor observations are processed at "system level" fusion sinks (as shown in Figure 4.1) to produce system-level sensor reports, which

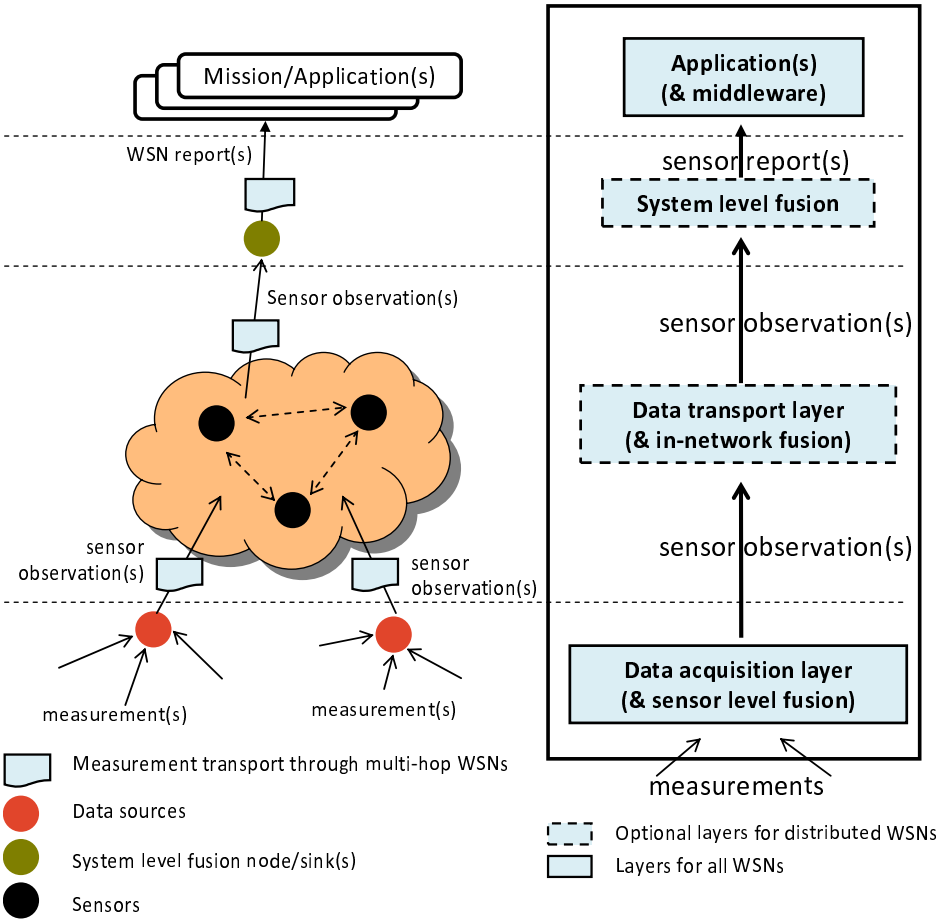

Figure 4.1: The deployment views of a WSN, where a functional view of a real WSN deployment is illustrated to the left, and a layered stack of operations is presented to the right.

are what the missions see coming (or reported) from the WSNs. As expected, a report may look exactly like an observation and exactly like a measurement, depending on how many layers of fusion take place between the sensors and the applications. The fact that these different items may look exactly the same should not deter us from thinking that sensors produce "observations" and applications consume "reports".

4.4 System Model

We consider a WSN comprising a set of sensor nodes, $\mathcal{S} = \{s_i; i = 1, 2, \ldots, N\}$ and a sink node (of sufficient information processing and energy capabilities). Tasks arrive at the

WSN and request service (*i.e.,* retrieve sensed information) to last some period of time l_q, where $\mathcal{Q}$ represents the set of tasks currently serviced by the WSN and sensors in $\mathcal{S}_q \subset \mathcal{S}$ 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 occurrence or location, density of a hazardous chemical, and so on. Each feature is associated with one or more QoI attributes, such as accuracy and latency in 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 *required* (and declared) by a task and superscript a for the level of of the QoI attribute *attained* by the WSN, *e.g.,* τ_q^r and τ_q^a will denote the probability of detection of an event (an accuracy attribute) or likewise d_q^r and d_q^a for the latency. Finally, tasks belong to one of U priority classes with higher priority tasks enjoy 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, $u = 1, 2, \ldots, U$. Task admission control is performed at the sink node before being assigned to any sensor node.

From the overall WSN resource allocation perspective, let $\underline{\mathcal{R}}(t) = \left(R^1(t), R^2(t), \ldots, R^P(t)\right)^T \in \mathbb{R}^{P1}$ denote a P-dimensional column vector describing the instantaneous remaining resources, like energy, bandwidth, buffer size, etc., and $\underline{\xi}_q^*(t) = \left(\xi_q^{1,*}(t), \xi_q^{2,*}(t), \ldots, \xi_q^{P,*}(t)\right)^T \in \mathbb{R}^P$ denote the corresponding optimal resource occupancy of each task $q, \forall q \in \mathcal{Q}$, after the resource allocation. Then, column vector $\underline{\eta}(t) = \left(\eta^1(t), \eta^2(t), \ldots, \eta^P(t)\right)^T \in \mathbb{R}^P$ represents the total resource occupancy for all ongoing tasks at time t, *i.e.,*

$$\underline{\eta}(t) = \sum_{\forall q \in \mathcal{Q}} \underline{\xi}_q^*(t). \tag{4.1}$$

4.5 The Flow of the Proposed Scheme

It is interesting to see that the QoI levels attained by tasks are the result of multiple operations spanning several protocol layers (*e.g.,* physical, MAC, network, information

[1]The underlined notation signifies a vector quantity in this Chapter unless otherwise stated.

processing, etc.) whose interrelations are too complex to describe effectively in any meaningful way. Therefore, similar to the techniques we developed in Chapter 3[2], we have opted to go around this issue by adopting 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, devices, time, buffers, bandwidth, energy, etc.

The I/O behavior of the black box is not known exactly but estimated at runtime. Without loss of generality, let this I/O behavior be represented by the mapping $f(\cdot)$, where:

$$f : \mathbb{R}^M \rightarrow \mathbb{R}\big(\underline{x}(t) \rightarrow y(t)\big)^3, \tag{4.2}$$

as shown in Figure 4.2.

The Inputs:

We consider two types of input variables: $\underline{x}(t) = \big(\underline{x}^1(t), \underline{x}^2(t)\big)$, where $\underline{x}^1(t) = \big(x_1^1(t), x_2^1(t), \ldots, x_{M_1}^1(t)\big)$ denotes M_1 dimension system-level parameters, like the number of ongoing tasks and the buffer size of each sensor, and $\underline{x}^2(t) = (x_1^2(t), x_2^2(t), \ldots, x_{M_2}^2(t))$ denotes M_2 dimension tasks' QoI requirements, like accuracy and latency, as shown in Figure 4.2; $M = M_1 + M_2$.

The Output:

The output $y(t)$ reflects the overall system utilization, denoted as QoI satisfaction index, see next section.

[2]For the completeness of the presentation, although the black box view is similar to the one presented in Chapter 3, here we readdress the importance of this model through concise presentations.

The Admission:

We characterize the potential new task admission as an input change $\Delta\underline{x}(t) = \big(\Delta\underline{x}^1(t), \Delta\underline{x}^2(t)\big) = \big(\Delta x_1^1(t), ..., \Delta x_{M_1}^1(t), \Delta x_1^2(t), ..., \Delta x_{M_2}^2(t)\big)$ into the black box, which will result in change of output to:

$$\tilde{y}(t) = f\big(\underline{x}(t) + \Delta\underline{x}(t)\big). \tag{4.3}$$

Next, we describe the overall flow of the proposed O&M framework, as shown Figure 4.2; details of the key concepts introduced are presented in the Section 4.6. The region above the dashed red line pertains to the external QoI (task-oriented) black box behavior of WSN, while below the line relates to the internal operation of the WSN. The mapping $f(\cdot)$ is derived by smoothly interpolating across the QoI levels delivered so far by the network to various tasks it has serviced and this is used to estimate the *QoI network capacity*, see Section 4.6.2, that is used to decide whether to admit the new task. The new task's QoI requirements are then compared with the QoI network capacity element-by-element such that if there is enough network resources to support, optimal resource allocation, see Section 4.6.4, runs to seek for optimal resource occupancy among all tasks, and $\underline{\xi}_q^*(t), \forall q \in \mathcal{Q}$, is obtained. Otherwise, a negotiation process is called such that existing tasks' QoI requirements are adapted to release some resources for the new task, see Section 4.6.3. When task completes, the resource allocation function is called again to re-optimize the distribution of limited network resources so that existing ongoing tasks' QoI will be improved.

Figure 4.3 illustrates the flow of the negotiation process, and key processes are summarized as follows.

1. Task *tsk* provides l, u, and $\underline{z}$ to WSN admission control (AC) upon its arrival
2. AC consults with QoI capacity estimator (QoI CE) whether can accommodate new task
3. QoI CE provides best capacity estimate in the presence of the new task and admission

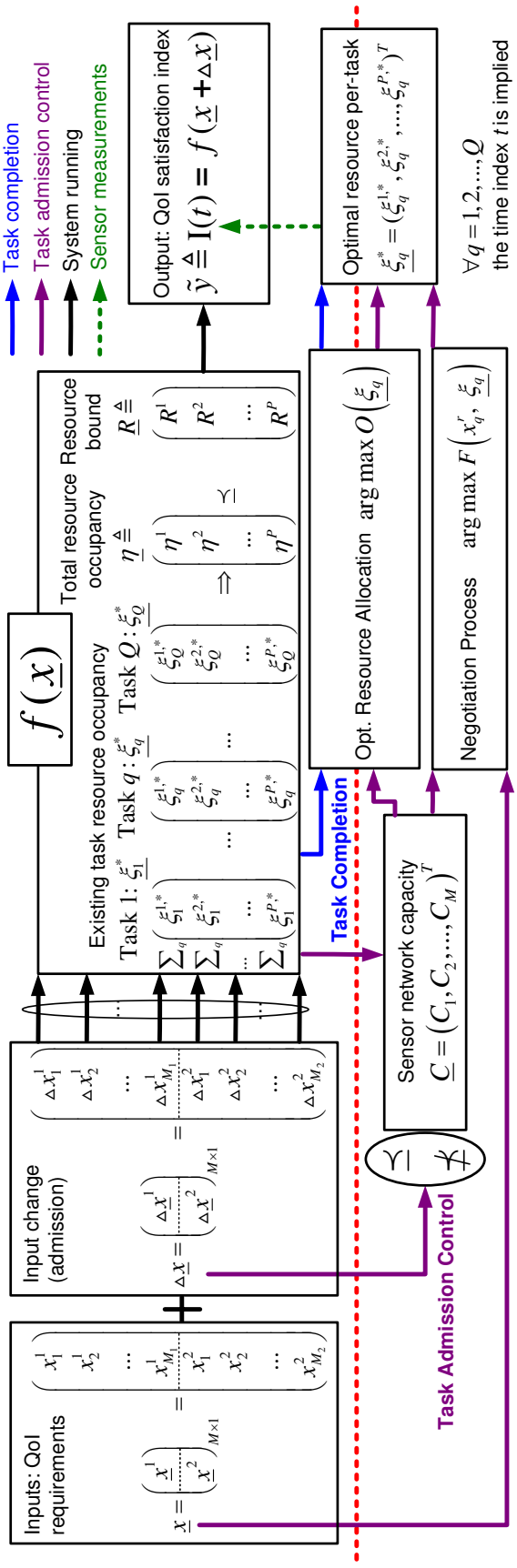

Figure 4.2: The flow of the proposed QoI-aware network O&M framework for any WSN. The system continuously measures the degree of resource occupancy at the output (*i.e.*, the QoI satisfaction index), and updates its own knowledge of the state of system to the inputs (*i.e.*, the QoI requirements). When the new task arrives for admission, QoI network capacity is obtained from the estimation that aids admission decision. Then, optimal resource allocation, or negotiation process if necessary, seeks for the optimal resource occupancy for all tasks. When any task completes, the resource allocation function is called again to re-optimize the allocation of limited network resources so that ongoing tasks' QoI are improved.

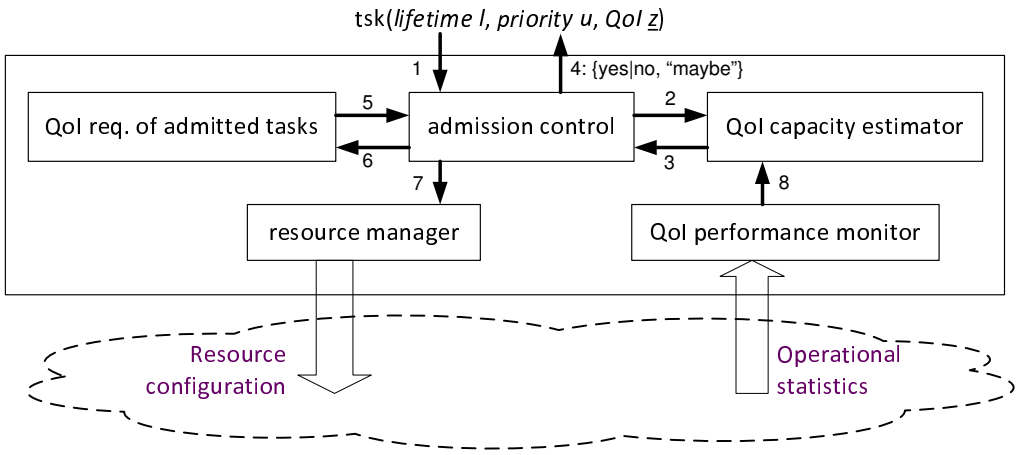

Figure 4.3: The overall flow of the negotiation process.

control make a "go/no go (yet)" decision

4. AC responds {yes, no} to task *tsk*

5. If not accepted, they enter a negotiation phase which includes adjusting the QoI levels of existing tasks to accommodate the QoI requirements of the new task.

6. Receive new acceptable QoI level for an already admitted task

7. Resource configuration commands are sent to WSN to configure resources as needed

8. Keep track of performance status

4.6 Key Design Elements

In this section, we will elaborate on the following four key design elements of our proposal, namely, (1) QoI satisfaction index, (2) QoI network capacity, (3) a negotiation-based admission control process, and (4) optimal resource allocation.

4.6.1 QoI Satisfaction Index

Similar to the QoS performance index introduced in Chapter 3, as its name implies, this index is used to describe the level of QoI satisfaction the tasks received from the WSN. It

is applicable to each task q and QoI attribute z and is defined as:

$$\mathrm{I}_q^z \triangleq \tanh\Big(k \ln \frac{z_q^a}{z_q^r}\Big), \quad \forall q \in \mathcal{Q}, \tag{4.4}$$

where z, which represents elements of the $\underline{x}^2(t)$ vector, could be τ or d for accuracy or latency, respectively, and k denotes a scaling factor. 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 4.4. The QoI satisfaction index behaves symmetrically around the origin, raising 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. A per task QoI satisfaction index I_q can be defined by combining the per QoI attribute indexes above. In this Chapter, we opt to use the minimum of these indexes, *i.e.,*

$$\mathrm{I}_q = \min\big(\mathrm{I}_q^z\big) \in (-1, 1), \forall q \in \mathcal{Q}. \tag{4.5}$$

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

Lemma 4.6.1. *For any task $q \in \mathcal{Q}$, its (multiple) QoI requirements are simultaneously satisfied if and only if $\mathrm{I}_q \in [0, 1)$.*

Likewise, we can define the instantaneous QoI satisfaction index $\mathrm{I}(t)$ as the minimum of indexes of all tasks in service at time t:

$$\mathrm{I}(t) = \min_{\forall q \in \mathcal{Q}} \mathrm{I}_q. \tag{4.6}$$

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 of the multi-hop WSNs for the reporting route, when the data is measured at the sink side. This is important because successful QoI supports rely on two parts: information sensing of

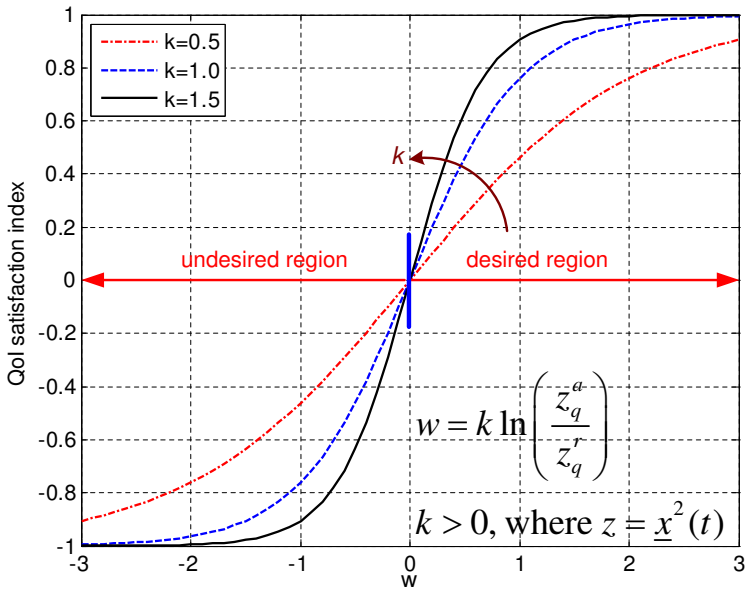

Figure 4.4: The illustrative example for the definition of QoI satisfaction index. It is desirable to have $z_q^a \geq z_q^r$ since it is assumed that the QoI attribute values should be at least as big as the required value to guaranteed the service quality.

multiple data sources, and information reporting through multi-hop WSNs that may incur further packet loss, delay, or damage.

4.6.2 QoI Network Capacity

Before admitting a new task for service, we would like to identify the potentially limiting resources and estimate the maximum "capacity" $\underline{\mathcal{C}}(t) = \big(\mathcal{C}_1(t), \mathcal{C}_2(t), \ldots, \mathcal{C}_P(t)\big)^T \in \mathbb{R}^P$ a WSN can support at any given time t. Thus, we define:

QoI network capacity indicates the time-varying capability a WSN can provide to any task with satisfactory QoI requirements, such that $\mathrm{I}_q \in [0, 1), \forall q \in \mathcal{Q}$. QoI network capacity $\underline{\mathcal{C}}(t)$ is a multi-dimensional column vector with network defined dimension P such that each element $\mathcal{C}_i(t) \in \underline{\mathcal{C}}(t), \forall i = 1, 2, \ldots, P$, can represent any one of the following parameters (not exclusively though): the network-wide maximum cardinality of the task set $\mathcal{Q}$, the maximum queue length for each node, the maximum information accuracy, the smallest information gathering delay, etc.

With reference to our black box view of WSN, we set its output $y(t) \triangleq \mathrm{I}(t) =$

$f(\underline{x}(t))$. Assuming $f(\cdot)$ is (at least) doubly differentiable, we write:

$$\tilde{y} = f\big(\underline{x} + \Delta\underline{x}\big) \approx f(\underline{x}) + \sum_{i=1}^{M} f'_{x_i}\Delta x_i + \frac{1}{2}\Big(\sum_{i=1}^{M} f''_{x_i}\Delta x_i^2 + \sum_{i=1}^{M}\sum_{q\neq i} f''_{x_q x_i}\Delta x_i \Delta x_q\Big), \quad (4.7)$$

where the time index t is implied and $f'_{x_i} = \partial f/\partial x_i$, $f''_{x_i} = \partial f^2/\partial x_i^2$, $f''_{x_j x_i} = \partial f^2/\partial x_q \partial x_i$.

Given more stringent QoI requirements for the input variables, a lower QoI satisfaction index is expected. At the same time, Lemma 4.6.1 indicates that the shape of curve will reach a lowest satisfaction level when QoI satisfaction index $\mathrm{I}(t) = 0$, at which level the QoI network capacity is also defined. This lowest point is estimated based on the curve for $f(\cdot)$ derived along each dimension of the mapping, see Figure 4.6(b) and (c). The procedure is to *project* a "large" task with stringent enough QoI requirement into the network, so that it pushes the system to the capacity bound: the minimum supportable QoI satisfaction index $\mathrm{I}(t) = 0$.

To illustrate this, 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 the WSN can provide to its tasks, $\underline{\mathcal{C}}(t) \triangleq \alpha_{\max}(t)$. We assume that a new task arrives at $t = 0$ when the WSN's state was: $\underline{x}(0) = \big(\underline{x}^1(0), \underline{x}^2(0)\big) = \big(n(0), \alpha(0)\big) \in \mathbb{R}^2$, where $n(0)$ denotes the number of existing tasks as the system parameter, and $\alpha(0)$ denotes the worst-case guaranteed detection probability as the QoI parameter. Then our black box is represented by mapping,

$$y(0) \triangleq \mathrm{I}(0) = f\big(n(0), \alpha(0)\big). \quad (4.8)$$

A large new task admission corresponds to an input change $\Delta\underline{x}(0) = \big(\Delta\underline{x}^1(0), \Delta\underline{x}^2(0)\big) = \big(n(0), \alpha(0)\big) = \big(1, \alpha_{\max}(t) - \alpha(0)\big)$, and for the expected output change,

$$\tilde{y}(0) = f\big(n^{\max}(t), \alpha_{\max}(t)\big) = 0. \quad (4.9)$$

For brevity we show the time index only when necessary; and therefore, we rewrite (4.7)

as,

$$\mathrm{I} + \Delta n f'_n + \Delta\alpha f'_\alpha + \frac{\Delta n^2}{2} f''_n + \frac{\Delta\alpha^2}{2} f''_\alpha + \Delta n \Delta\alpha f''_{n\alpha} = 0, \tag{4.10}$$

or,

$$f'_n + (\alpha_{\max} - \alpha)\,(f'_\alpha + f''_{n\alpha}) + \frac{1}{2} f''_n + \frac{(\alpha_{\max} - \alpha)^2}{2} f''_\alpha = -\mathrm{I}, \tag{4.11}$$

where all partial derivatives are computed at the current system state $\underline{x}(0) = \big(n(0), \alpha(0)\big)$ at time $t = 0$. It is not difficult to observe that (4.11) is a quadratic function with the only decision variable $\alpha_{\max}$. Therefore, we derive its closed-form expression as:

$$\underline{\mathcal{C}} \triangleq \alpha_{\max} = \alpha - \frac{f''_{n\alpha} + f'_\alpha - \sqrt{(f''_{n\alpha} + f'_\alpha)^2 - 2f''_\alpha\,(2f'_n + f''_n - 2\mathrm{I})}}{f''_\alpha}. \tag{4.12}$$

Furthermore, if the shape of curve produced by the mapping f is smooth enough around current system operating point $\underline{x}(0) = \big(n(0), \alpha(0)\big)$ so that the second order derivatives are negligible, we simplify (4.12) as:

$$\underline{\mathcal{C}} \triangleq \alpha_{\max} = \alpha - \frac{\mathrm{I} + f'_n}{f'_\alpha}. \tag{4.13}$$

Figure 4.5(a) illustrates of how this methodology is used, and Figure 4.5(b) provides illustrates real-time statistics (from a system simulation) of QoI satisfaction indexes collected and interpolated to estimate the current shape of the $f(\cdot)$ curve.

4.6.3 Negotiation-based Admission Control

As shown in Figure 4.2, following the estimation of the QoI network capacity, suppose a new task q' with priority $u_{q'}$ and QoI requirements $\{z^r_{q'}\}$, arrives at the sink for the admission decision at time t; the z's scan the elements of vector $\underline{x}^2(t)$ in the figure, in abuse of notation, we will right $z = \underline{x}^2$ for it. Before assigning the task to any sensor(s),

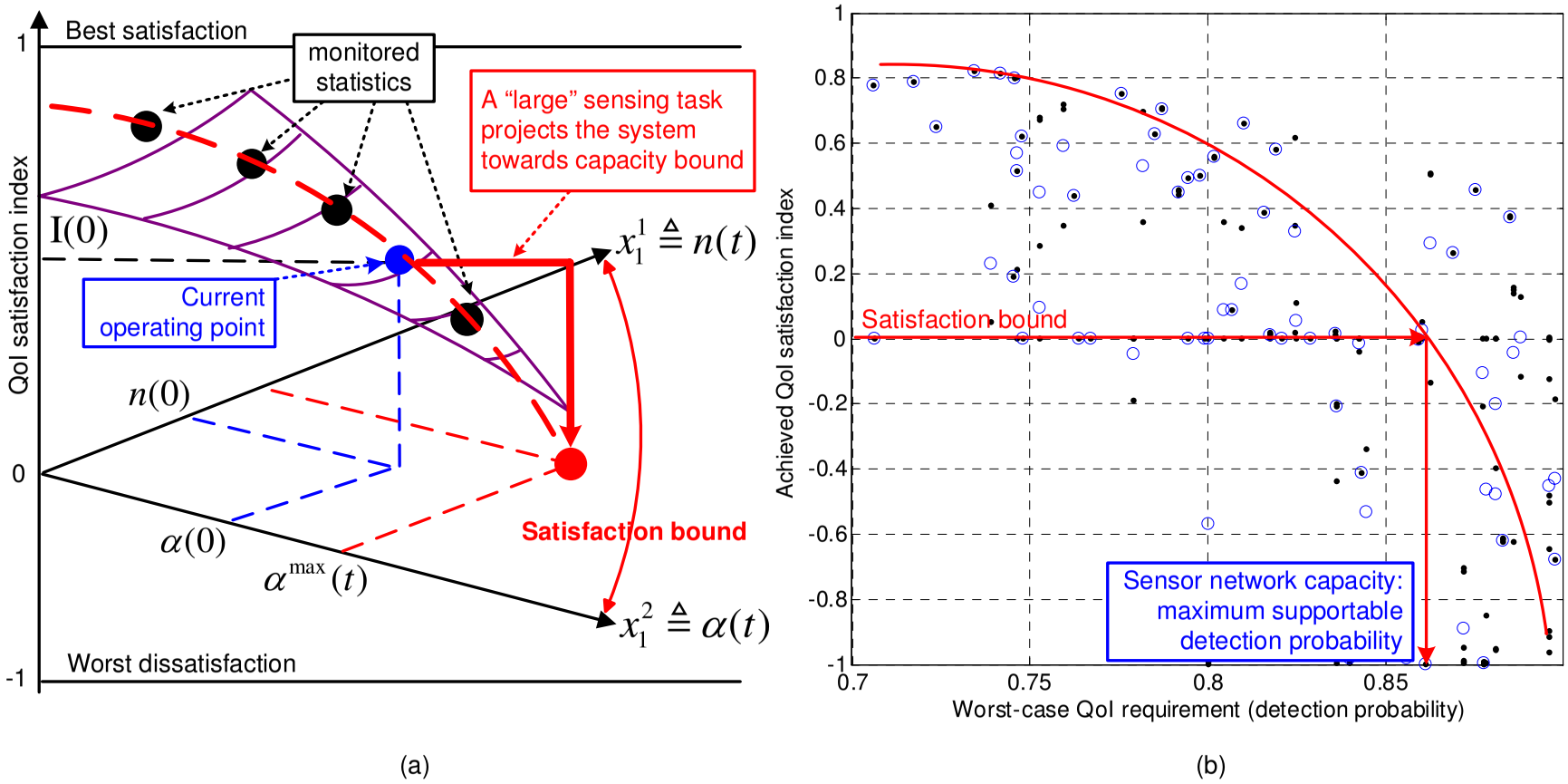

Figure 4.5: (a) An example of the shape of curve produced by the mapping f to show how to obtain the QoI network capacity in term of the maximum probability of detection $\alpha_{\max}(t)$. (b) An real-time statistics example for the QoI network capacity estimation.

admission control decision is made according to the following conditions (see Figure 4.2),

$$\underline{\mathcal{C}}(t) \succeq \{z_{q'}^r\} \begin{cases} \text{Admission, if true,} \\ \text{Negotiation, otherwise,} \end{cases}$$

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, and the admission control functionality, 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 control allocation, sensor selection, integration of data compression, etc.

Under the guidance of the resource optimization, ongoing tasks may internally reconfigure and reallocate network resource usages 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 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:

$$\left\{\underline{\xi}_q^*(t)\right\}_{\forall q\in\mathcal{Q}} = \arg\max \mathcal{F}\left(\left\{z_q^r\right\}_{\forall u_q<u_{q'}}^{z\in\underline{x}^2(t)}, \underline{\xi}_q(t)\big|_{\forall q\in\mathcal{Q}}\right) \tag{4.14}$$

$$\text{subject to: } \begin{cases} z_q^a \geq z_q^r, \forall q \in \mathcal{Q}, z \in \underline{x}^2(t) \\ \underline{\eta}(t) \triangleq \sum_{\forall q \in \mathcal{Q}} \underline{\xi}_q(t) \preceq \underline{\mathcal{R}}(t), \end{cases}$$

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 $\left\{z_q^r\right\}_{\forall u_q < u_{q'}}^{z \in \underline{x}^2(t)}$ of those tasks with lower priority classes, and resource occupancy vector $\underline{\xi}_q(t)\big|_{\forall q \in \mathcal{Q}}$. Note that the optimization is further constrained by the need to respect the QoI satisfaction for the task of different priority groups and resource constraints under current network status.

4.6.4 Optimal Resource Allocation

After the admission decision is made for the new task, network resources will be allocated given that all ongoing tasks' QoI levels cannot be violated, which is guaranteed by an optimization problem as shown in Figure 4.2. Suppose a generic mathematical function $\mathcal{O}(\cdot)$ is used to represent the network design objective, where inputs are resource allocation vector $\underline{\xi}_q(t)$ for all ongoing and new task $q \in \mathcal{Q}$. The optimization problem is constrained with QoI satisfaction for all tasks and resource availabilities under limited resource bound, as:

$$\left\{\underline{\xi}_q^*(t)\right\}_{\forall q \in \mathcal{Q}} = \arg\max \mathcal{O}\left(\underline{\xi}_q(t)\big|_{\forall q \in \mathcal{Q}}\right) \tag{4.15}$$

$$\text{subject to: } \begin{cases} z_q^a \geq z_q^r, \forall q \in \mathcal{Q}, z \in \underline{x}^2(t) \\ \underline{\eta}(t) \triangleq \sum_{\forall q \in \mathcal{Q}} \underline{\xi}_q(t) \preceq \underline{\mathcal{R}}(t), \end{cases}$$

It is worth noting that improper choice of objection function $\mathcal{O}$ and misrepresentation of network resources may highly increase the complexity for the underlying optimization problem. A specific example of the objective functions $\mathcal{F}$ (for the negotiation) and $\mathcal{O}$ (for the resource allocation) will be used in the numerical example later on.

4.7 Numerical Results

4.7.1 The Scenario

We access the proposed scheme under an intruder detection user scenario [93], where multiple 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 that is 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 equal reserve of energy at level $\mathcal{E}$ is assumed for each sensor, and tasks arrive according to the Poisson process with rate λ and last for a random exponential time interval l_q with average duration $1/\mu$. 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. Sensors are equipped with smart 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 [94] using physical properties of the sensors, where the detection probability p_{iq}^d for task q from sensor i is achieved assuming using normalized full power $\gamma_q^*(t) = 1$, *i.e.,*

$$p_{iq}^d = \begin{cases} 1, & \text{if } r_{iq} < d_t^1, \\ e^{-\beta_1 (r_{iq} - d_t^1)^{\beta_2}}, & \text{if } d_t^1 < r_{iq} < d_t^2, \\ 0, & \text{elseif } r_{iq} > d_t^2 > d_t^1, \end{cases} \tag{4.16}$$

$\forall i \in \mathcal{S}_q$, where $\beta_1 = 0.12, \beta_2 = 0.8$, and $d_t^1 = 28$m, $d_t^2 = 58$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 1-D scalar as the power in this use case, *i.e.,* $\underline{\xi}_q^*(t) \triangleq \gamma_q^*(t)$, and the *attained*

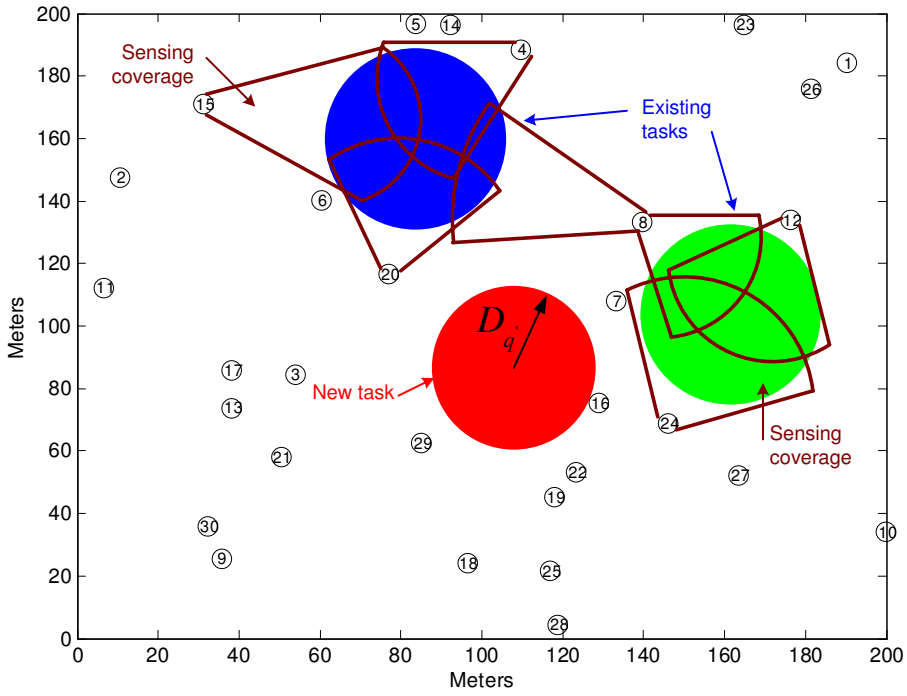

Figure 4.6: Simulation scenario for the considered intruder detection application. Two existing intruder detection tasks exist in the network (marked as the blue and green regions), while a new task (marked as red region) arrives for admission. Several sensors are selected per task as data sources (sensor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

QoI satisfaction index I_q can be explicitly expressed in the following form,

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

where sensor measurement $\alpha_q^a = \gamma_q^*(t) \min_{\forall i \in \mathcal{S}_q} p_{iq}^d$. This is because that for task $q \in \mathcal{Q}$, the actual attained probability of detection α_q^a is attenuated with the power allocation $\gamma_q^*(t)$, and as a product of the minimum detection probability attained by multiple sensor sources, *i.e.,* $\min_{\forall i \in \mathcal{S}_q} p_{iq}^d$ *i.e.,*

$$\alpha_q^a = \gamma_q^*(t) \min_{\forall i \in \mathcal{S}_q} p_{iq}^d, \quad \forall q \in \mathcal{Q}. \tag{4.18}$$

The Lower Bound QoI Parameter k:

Interestingly, under the intruder detection user scenario, the maximum achieved detection probability is 1, while the required detection probability is pre-determined by applications.

Therefore, the selection of k parameter should enforce that the highest QoI satisfaction index, *i.e.,* $\mathrm{I}_q^{\max}=1$, is achieved. In other words, $\mathrm{I}_q^{\max}=\tanh\left(k\ln\frac{1}{\alpha_q^r}\right)=1, \forall q\in\mathcal{Q}_1\bigcup\mathcal{Q}_2$, from where we could derive the lower bounds of parameter k for high and low priority users as:

$$\begin{cases} k_h \geq \tanh^{-1}(\approx 1)\ln\alpha_q^{r,h}, \forall q\in\mathcal{Q}_1, \\ k_l \geq \tanh^{-1}(\approx 1)\ln\alpha_q^{r,l}, \forall q\in\mathcal{Q}_2. \end{cases} \tag{4.19}$$

For tasks with different QoI requirements α_q^r, the lower bounds k_h, k_l will change accordingly, *e.g.,* in a WSN supporting two-priority users, where the high priority task average detection probability $\alpha_q^{r,h}=0.8$ and the low priority task average detection probability $\alpha_q^{r,l}=0.5$, we are able to compute QoI parameter $k_h\geq 17, k_l\geq 5.5$. In other words, these lower bound values k_h, k_l will guarantee that when optimal detection is achieved, the maximum QoI satisfaction index $\mathrm{I}_q^{\max}=1$ is actually obtained.

The Optimal Power Allocation:

As discussed in Section 4.6.4, 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. Customize the optimization problem in (4.15) such that the fairness among all tasks are provided, given QoI satisfaction achieved for all high priority and low priority users, we have:

$$\begin{aligned} \left\{\gamma_q^*(t)\right\}_{\forall q\in\mathcal{Q}} &= \arg\max \mathcal{O}\left(\gamma_q(t)\big|_{\forall q\in\mathcal{Q}}\right) \\ &\triangleq \arg\max \min_{\forall q\in\mathcal{Q}} \mathrm{I}_q \end{aligned} \tag{4.20}$$

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

where the design objective $\mathcal{O}(\cdot)$ in (4.15) is chosen to balance the achieved QoI satisfaction indexes among all ongoing and new tasks. I_q 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 constraint for each sensor. Assuming equal power is allocated for every sensor source 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 user scenario) supporting the new task, existing lower priority ones have to adapt/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, as:

$$\begin{aligned} \left\{\gamma_q^*(t)\right\}_{\forall q \in \mathcal{Q}} &= \arg\max \mathcal{F}\Big(\alpha_q^r|_{\forall q \in \mathcal{Q}_2}, \gamma_q(t)|_{\forall q \in \mathcal{Q}}\Big) \\ &\triangleq \arg\min \max_{\forall q \in \mathcal{Q}_2} \frac{\widetilde{\mathrm{I}}_q - \mathrm{I}_q}{\widetilde{\mathrm{I}}_q} \end{aligned} \tag{4.21}$$

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

where $\widetilde{\mathrm{I}}_q$ denotes the attained QoI level *before* 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.

4.7.2 The Optimal Network Design Analysis:

Given the proposed QoI-aware framework, we would like to explore the system limits under the conditions of constrained network resources and varying QoI requirements for differ-

ent tasks, aiming at higher QoI network capacity, longer system lifetime, and increased admission rate, while satisfying the required QoI of admitted tasks. Particularly, for the considered intruder detection use case, 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 so that the amount of remaining energy reserves can always guarantee a minimum probability of detection $\alpha_{\min}$ for any task appearing at this time, located anywhere within the sensing field.

For this, we view the entire WSN system 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 and, of course, relates to network resource availability, energy consumption rate, etc. 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 $T_{\max}$ provided that all tasks accepted experience satisfactory QoI levels, *i.e.,* $\mathrm{I}_q \geq 0$? Or,

(2) Given minimum WSN lifetime $T_{\min}$ and satisfactory QoI levels for all tasks, what is the region of admissible rates $\lambda \leq \lambda_{\max}$ that the system can sustain as a function of μ?

In the following Lemma we broadly derive 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 1-D scalar as power, $\underline{\xi}_q^* = \gamma_q^*$, and thus the relationship between γ_q^* and QoI satisfaction index I_q can be analytically represented by (4.17), or see at the RHS of Figure 4.2.

Lemma 4.7.1. *The task arrival rate λ vs. WSN lifetime T trade off is of the form $\frac{\lambda T}{\mu} \leq \frac{\mathcal{E}}{\beta \alpha_{\min}}$, where $\beta \triangleq \min_{\forall i \in \mathcal{S}_1} p_{i1}^d$ denotes a constant given geographic locations of*

sensor sources and tasks. Furthermore, the maximum WSN lifetime and the maximum admissible rate can be expressed as $T_{\max} = \beta \frac{\mathcal{E}}{\alpha_{\min}\rho}$, *and* $\lambda_{\max} = \beta \frac{\mathcal{E}_\mu}{\alpha_{\min} T_{\min}}$, *respectively.*

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

$$\gamma_q^*(t) = \alpha_q^r \frac{\exp\left(\frac{1}{k}\tan \mathrm{I}_q\right)}{\min_{\forall i \in \mathcal{S}_q} p_{iq}^d}. \tag{4.22}$$

According to Lemma 4.6.1, the the lower bound resource condition for satisfactory QoI is taken $\mathrm{I}_q = 0$ as the input that produces $\gamma_{q,\min}^*(t) = \gamma_q^*(t)|_{\mathrm{I}_q=0}$, or,

$$\gamma_q^*(t) \geq \gamma_{q,\min}^*(t) = \frac{\alpha_q^r}{\min_{\forall i \in \mathcal{S}_q} p_{iq}^d} \geq \frac{\alpha_{\min}}{\min_{\forall i \in \mathcal{S}_q} p_{iq}^d}, \tag{4.23}$$

where the last equality condition uses the notation $\alpha_q^r \geq \alpha_{\min}, \forall q \in \mathcal{Q}$.

At the same time though, resource constraints enforce the total amount of allocated network resource no more than total energy reserve level $\mathcal{E}$, *i.e.,*

$$\sum_{\forall q \in \mathcal{Q}^T} \gamma_q^*(t) l_q \leq \mathcal{E}, \tag{4.24}$$

where $\mathcal{Q}^T$ denotes the task set has been serviced during WSN lifetime T, 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 arrivals and departures, we use the conditions of expectation to approximate the LHS random variables, as:

$$\begin{aligned}
\mathcal{E} &\geq \mathbb{E}\Big(\sum_{\forall q \in \mathcal{Q}^T} \gamma_q^*(t) l_q\Big) = \mathbb{E}\Big(\mathbb{E}\Big(\sum_{\forall q \in \mathcal{Q}^T} \gamma_q^*(t) l_q \Big| \mathcal{Q}^T\Big)\Big) \\
&= \mathbb{E}\Big(\sum_{\forall q \in \mathcal{Q}^T} \mathbb{E}\Big(\gamma_q^*(t) l_q\Big)\Big) = \mathbb{E}\Big(\mathcal{Q}^T \mathbb{E}\Big(\gamma_1^*(t) l_1\Big)\Big) \\
&= \mathbb{E}\Big(\mathcal{Q}^T\Big)\mathbb{E}\Big(\gamma_1^*(t) l_1\Big) = \lambda T \mathbb{E}\Big(\gamma_1^*(t)\Big)\mathbb{E}\Big(l_1\Big) \\
&= \frac{\lambda T}{\mu}\mathbb{E}\Big(\gamma_1^*(t)\Big),
\end{aligned} \tag{4.25}$$

where we use the fact that the task's arrival process, departure process, and task optimal resource occupancies $\gamma_q^*(t), \forall q \in \mathcal{Q}^T$ 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 Little's theorem 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 (4.25) by using condition (4.23), as:

$$\begin{aligned} \mathcal{E} &\geq \frac{\lambda T}{\mu}\mathbb{E}\Big(\gamma_1^*(t)\Big) \geq \frac{\lambda T}{\mu}\mathbb{E}\Big(\gamma_{1,\min}^*(t)\Big) \\ &\geq \frac{\lambda T}{\mu}\mathbb{E}\left(\frac{\alpha_{\min}}{\min_{\forall i\in\mathcal{S}_1} p_{i1}^d}\right) \\ &= \frac{\alpha_{\min}\lambda T}{\beta\mu}, \end{aligned} \tag{4.26}$$

where $\beta \triangleq \min_{\forall i\in\mathcal{S}_1} p_{i1}^d$ denotes a constant given geographic locations of sensor sources and task. Hence, we rewrite (4.26) as,

$$\frac{\lambda T}{\mu} \leq \frac{\mathcal{E}}{\beta\alpha_{\min}} \tag{4.27}$$

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

$$T_{\max} = \beta\frac{\mathcal{E}}{\alpha_{\min}\rho}, \quad \lambda_{\max} = \beta\frac{\mathcal{E}\mu}{\alpha_{\min}T_{\min}}. \tag{4.28}$$

□

Lemma 4.7.1 proves that Eqn. (4.27) serves as the principle worst-case (in terms of guaranteeing the minimum QoI satisfaction) system design criterion for this scenario, however it shows the powerful and fundamental trade-offs among the maximum network lifetime, the task departure rate, the task arrival rate, and the QoI requirement. For instance, higher QoI requirement ($\alpha_{\min}$) would constrain the energy usage for multiple tasks which in turn has impacts on the admissible arrival rate and the WSN lifetime.

4.7.3 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".

Traditional WSN research is an one-off deployment configuration assuming "static" behaviors of system parameters, where sensors are positioned on 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, independent of application needs. In contrast, with the proposed QoI-aware management, system parameters will be adjusted judiciously, so that WSN lifetime will be longer given satisfactory QoI requirements. In this simulation, for both "static" and the "dynamic" scenarios, we assume that tasks arrive and last stochastically with the same statistics, and we choose that the probability of detection for which the system is designed to operate in the static case is the average of the probability of detections the various tasks request in the dynamic case.

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 lifetime $1/\mu$. QoI outage is defined as the portion of all completed tasks' QoI requirement to fail, where for any given task q, failure is defined as during its lifetime, there was at least one instance when $\mathrm{I}_q < 0$. For fixed average task lifetime, it is interesting to observe the saturation feature of QoI outage probability for all three schemes when we increase the arrival rate since rejections to new tasks help maintain ongoing ones' QoI satisfaction. However, the saturation bounds for three schemes vary significantly: the proposed algorithm can even guarantee 81% of QoI satisfaction for any underlying application, as compared to 74% for "AC only" scheme, and 40% for "Traditional" approach. This is because the impact of newly admitted tasks on existing ones has been estimated and accurately reflected in the parameter of QoI network capacity in terms of the maximum detection probability which controls the QoI-aware network status, and the negotiation process helps optimize resource utilization to release some resources for higher priority users. On the other hand, when average task lifetime is increased, QoI outage increases by 20%. This is because the increasing network load $\rho = \frac{\lambda}{\mu}$ at any time in the network may jeopardize ongoing tasks' QoI satisfaction, since finite network resources are shared by more tasks than before, which in turn may violate the QoI network capacity bound.

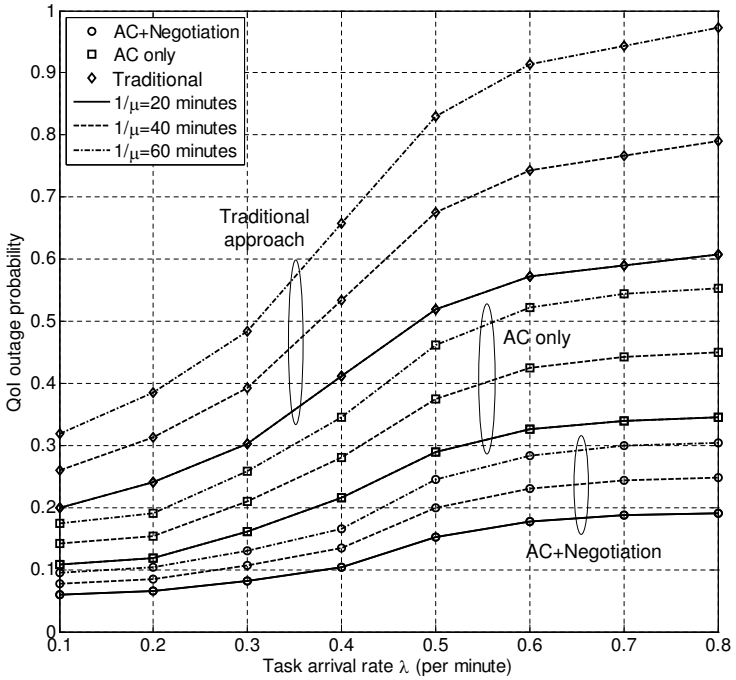

Figure 4.7: Simulation result of the average QoI outage probability among all completed tasks, w.r.t different task arrival rates λ and the average task lifetime $1/\mu$.

The behavior of average QoI outage probability for different priority user groups is shown in Figure 4.8, where only the "AC+Negotiation" scheme is plotted with fixed average task lifetime $1/\mu = 40$ mins. Interestingly, although similar behaviors for high and low priority user groups can be seen, the saturation speed of their QoI outage probability differs significantly. This is primarily because our proposed negotiation process successfully guarantees non-negotiable QoI levels for high priority tasks, however, 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 behavior of the average task blocking probability w.r.t. both task arrival rate and lifetime. While the "Traditional" approach is not plotted in this figure since no rejections are made, task blocking probability increases significantly when more tasks are offered (higher λ). However, these successful task rejections help maintain low QoI outage probability and high QoI satisfaction for existing ones in the network, as shown in Figure 4.7. On the other hand, when network load ρ is increased by enlarging task lifetime, resource availability decreases as being occupied by higher number of con-

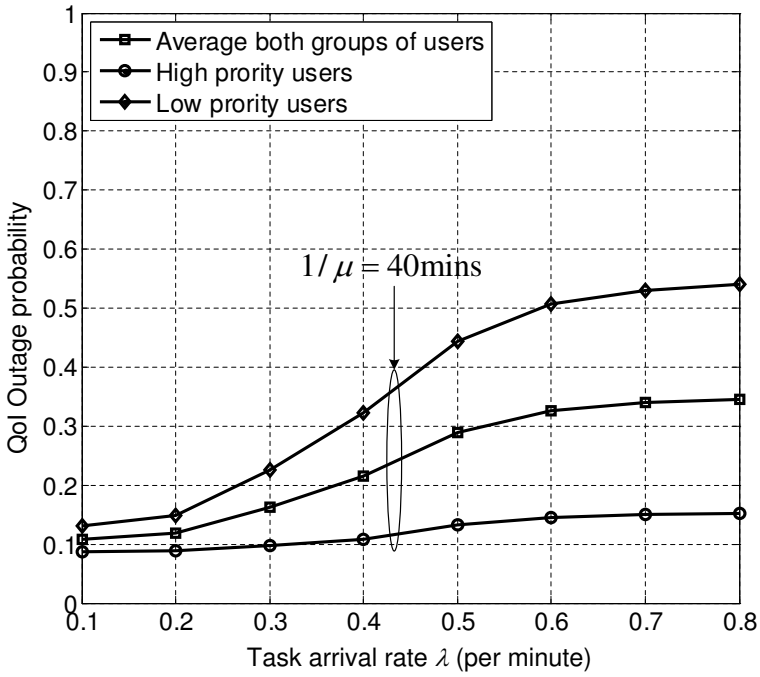

Figure 4.8: Simulation result of the average QoI outage probability among all completed tasks of two different priority user groups, w.r.t different task arrival rates λ and the average task lifetime $1/\mu$.

current tasks serviced. Last, for reasonably loaded system, our scheme "AC+Negotiation" can successfully guarantee as low as 5% blocking probability as compared with 8% when negotiation process is not used.

Table. 4.1 demonstrates the average jitter of QoI satisfaction index among completed and satisfactory tasks, which is defined as the variance of satisfaction indexes, *i.e.,* $\sigma_{\forall q \in \mathcal{Q}}(\mathrm{I}_q)$. Unlike QoI outage and blocking probability, this performance metric directly reflects the human aspect of experiences when interfacing the system that indicates the performance stability (or fairness) for the proposed O&M framework to provide QoI experiences for all tasks. For fixed average task lifetime $1/\mu$, a 31% jitter increase can been seen if full scheme is compared with the other two schemes.

Figure 4.10 shows the normalized WSN lifetime w.r.t. different task arrival rate and departure rates. It can be seen a significant WSN lifetime improvement compared with the traditional settings, and this improvement increases when tasks arrive more frequently (due to more efficient resource allocation among all tasks). Furthermore, the

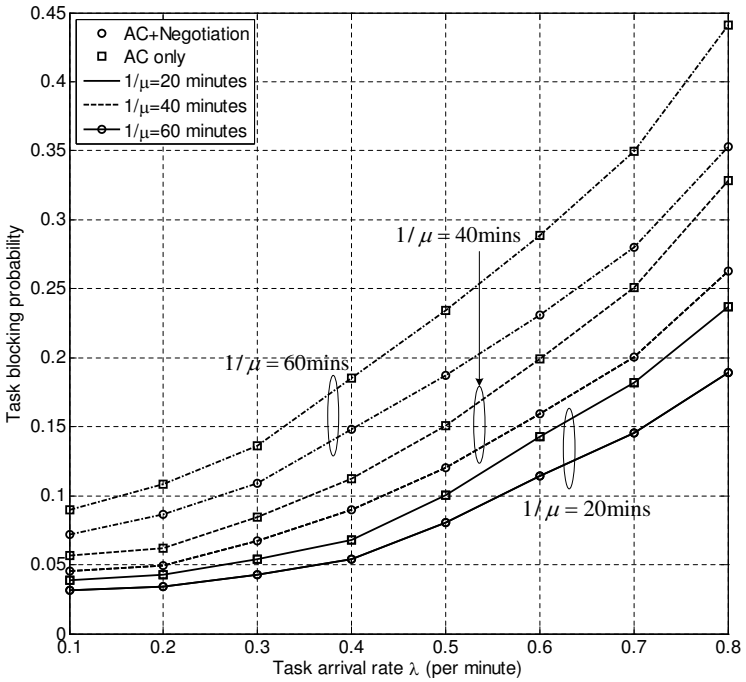

Figure 4.9: Simulation result of the average task blocking probability, w.r.t different task arrival rates λ and the average task lifetime $1/\mu$.

Table 4.1: Average jitter values of the received QoI satisfaction indexes, 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

proposed approach successfully approximate the analytical results given in (4.28) while traditional settings perform far away behind. Meanwhile, given the desired WSN lifetime, this figure also shows the way to obtain the maximum admissible rate $\lambda_{\max}$ the network can support given minimum probability of detection $\alpha_{\min}$.

4.7.4 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, as key design elements for such O&M framework. Figure 4.11(a) illus-

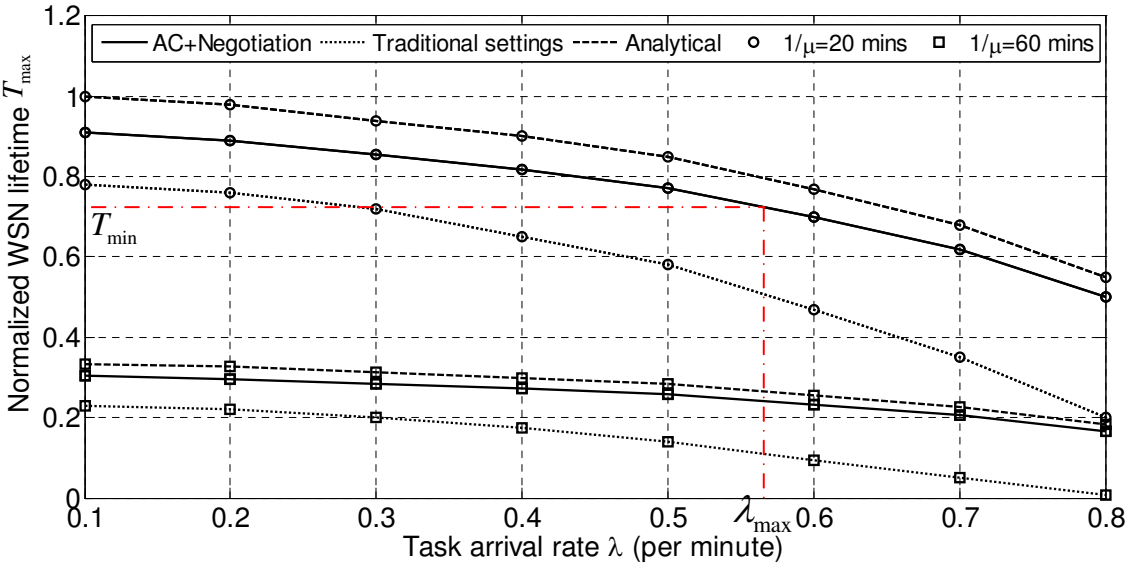

Figure 4.10: Simulation result of the normalized WSN lifetime w.r.t. the different task arrival rate λ and the task departure rate μ.

trates the simulated traffic pattern (*i.e.,* the number of tasks, task arrival and departure processes, QoI requirements), and Figure 4.11(b) and (c) shows dynamic QoI changes experienced by 70 tasks, w.r.t. two different QoI satisfaction index parameter k_h, k_l.

For fixed QoI parameters k_h, k_l, abrupt QoI changes can be seen under the relatively high traffic load conditions. When new task arrives, the negotiation process will attempt to accommodate it while reasonably degrading existing tasks' level of QoI satisfactions, but still maintaining the minimum required levels for them. Meanwhile, when completed tasks are removed, pre-allocated network resources are released by the resource optimizer 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 2500mins to 3000mins), some tasks would experience QoI failures as their QoI satisfaction levels cannot be satisfied in any meaningful anyway; but nevertheless there are still portions of tasks successfully maintain the minimum level, *i.e.,* $\mathrm{I}_j \geq 0$, to utilize the limited network resources[4].

[4]This is more like a game, where tasks compete for limited network resources according to the relative compatibility of their priority and requested QoI requirements with dynamic network status. In other

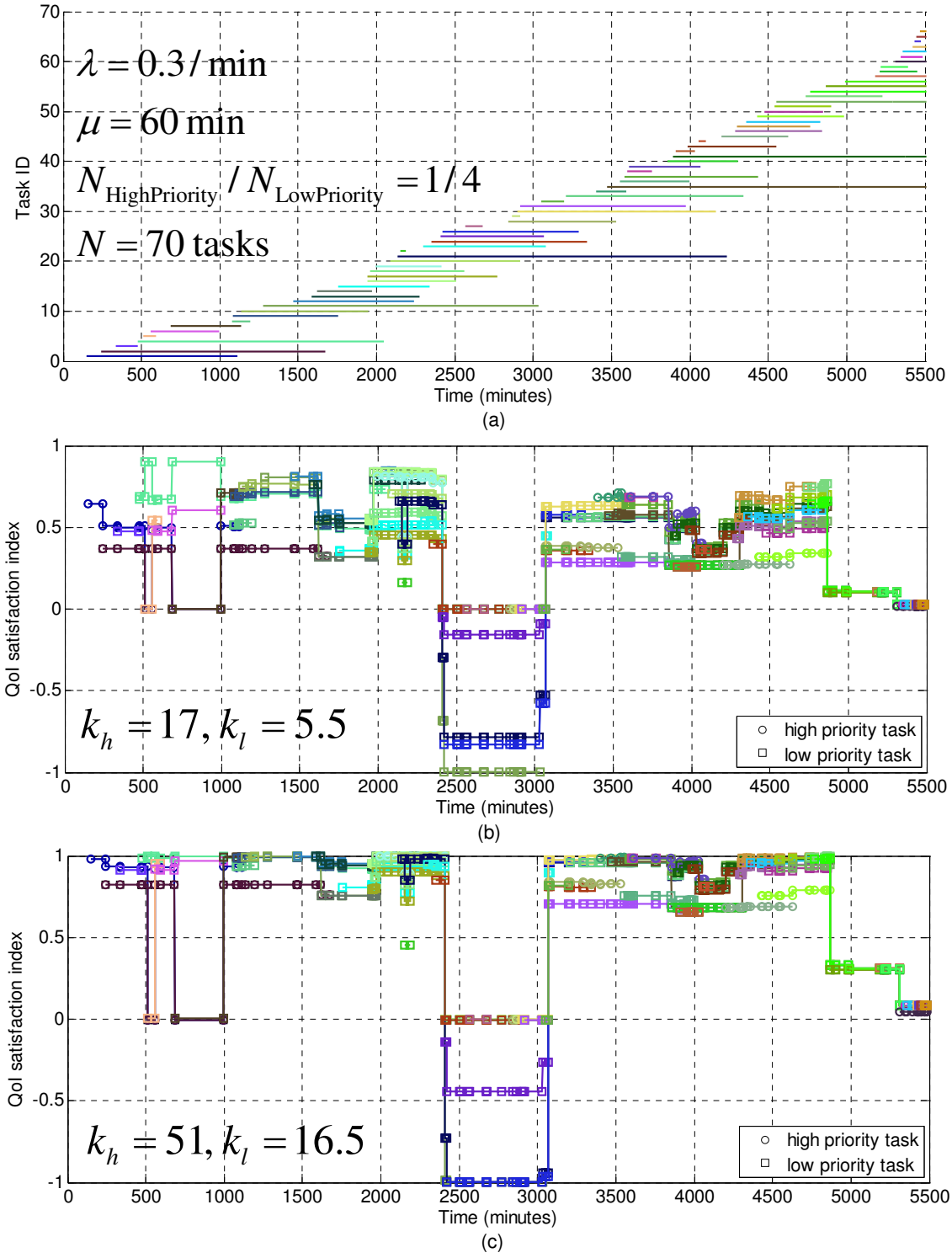

Figure 4.11: Simulation result for the system behavior as a result of resource optimizations and negotiations, where (a) shows the task arrival and departure time line, (b) shows the real-time QoI satisfaction index change with the chosen parameter $k_h = 17, k_l = 5.5$, and (c) shows the real-time QoI satisfaction index change with the chosen parameter $k_h = 51, k_l = 16.5$. Both figure (b) and (c) are plotted with the same set of traffic and their QoI requirements as shows in figure (a).

On the other hand, when we increase QoI parameters k_h, k_l proportionally, which means the improved QoI satisfaction level even with the same attained detection probability, it helps the system ease the resource competition among tasks and increase their satisfaction level (due to higher estimated QoI network capacity).

4.8 Discussions

4.8.1 The Middleware Approach

QoI-aware sensor network operations and management represents a broader area of research challenges that this Chapter only begins to address. In the course of this work, we have identified several important directions for ongoing research activities, mainly motivated by the requirements of deploying an O&M solution in a broader space of application scenarios.

In an effort to make the O&M framework easily *reusable* in real-world sensor network applications, we investigate how to embody the framework in a formalized middleware instantiation. Based on developed a conceptual sensor network middleware framework as described in [91], which, however, never fully developed the logic for factoring QoI into the design and operation of the middleware components, in our proposed architecture as shown in Figure 4.12, several components include: the *resource manager* is responsible for facilitating missions admission and network resource optimization; the *data manager* calculates and provides QoI satisfaction index values for the resource manager to aid the task admission process; and the *interface manager* brokers all data exchange between external tasks and the middleware. While this sample architecture shows how an initial middleware solution might be crafted, we expect that other directions of future research will require alterations to this configuration.

Other future research directions include extending the overall O&M solution to a distributed configuration for large-scale *ad hoc* networked environments as well as investigating extensions to the definitions of capacity and negotiation. The first item will require

words, not necessarily in the extreme case all tasks give up execution, but some low priority tasks with low QoI requirements may successfully survive.

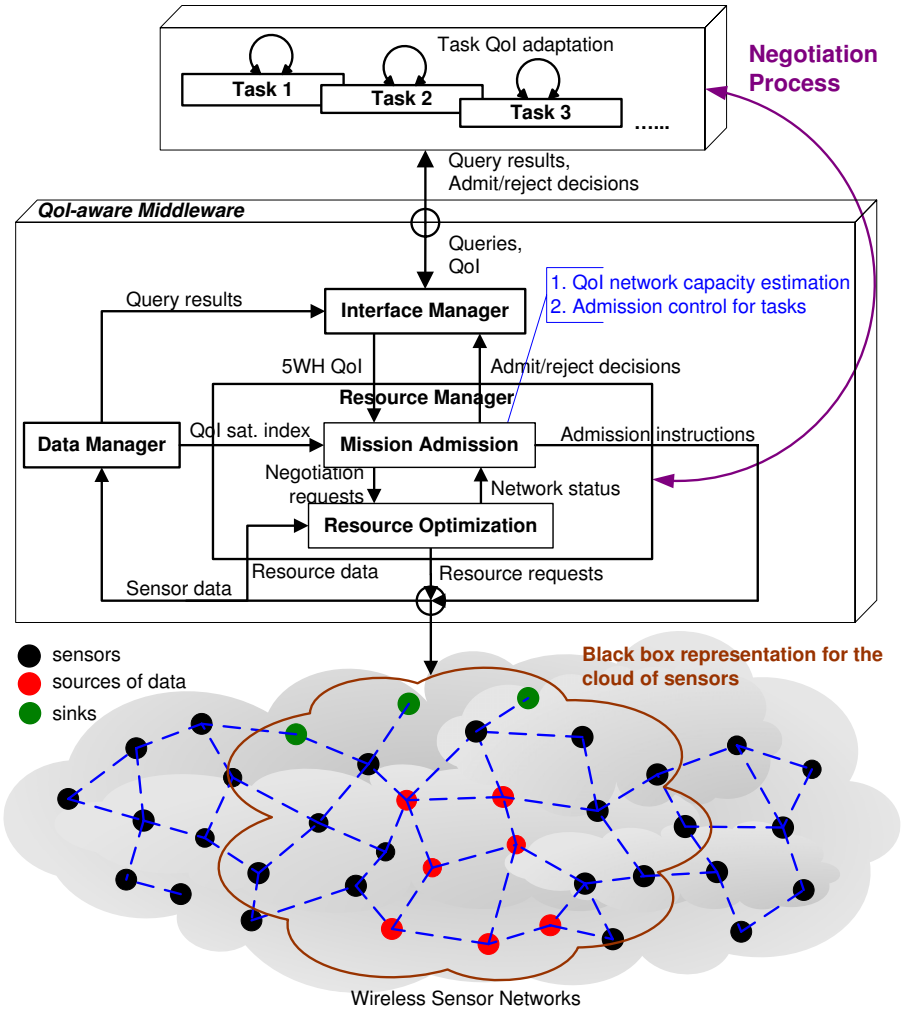

Figure 4.12: The QoI-aware middleware architecture for the proposed network O&M framework.

distributing the functionality for performing negotiation and calculating QoI network capacity and will affect the design of the middleware architecture. The second item will involve researching how additional network and application behaviors will affect capacity and negotiation. Examples include sensor network duty-cycling algorithms as well as the inclusion of networked actuators, which would most likely change the nature of tasks admitted to the framework.

We also plan to investigate the challenges of developing a QoI-aware O&M solution for sensor and *actuator*-enabled applications. This is motivated by the growing class of applications that facilitate environmental *control*, as opposed to solely monitoring. For

instance, we consider that the equivalent of sensor network resources exist for actuators as well. These could range from straightforward resources such as device energy or treatment supplies (*e.g.,* liquids, ammunitions, etc.) to more complex resources such as incentives to perform actuation (*e.g.,* fatigued soldiers need longer UAV support for mission success). Considering such resources in addition to the sensor-oriented resources addressed in the current work would require new solutions for defining things such as network capacity and negotiation algorithms since sensor and actuator resources may not have any relationships. Overall, the new challenge involves how to manage QoI-aware end-to-end control tasks as opposed to tasks. This will require new optimization techniques as well as sophisticated middleware components for managing actuators in a QoI-aware manner.

4.8.2 The Applicability

It is worth to highlight that the applicability of the proposed negotiation-based network O&M approach will not at all be restricted in multi-hop WSN settings, but rather it has wide applicability to any multi-hop wireless networks, including backhaul wireless mesh networks and wireless ad hoc networks, where the notion of quality is well referred to QoS, including delay, throughput, PER etc. The proposed negotiation-based admission control and optimization methods do not require specific protocols like scheduling or routing, nor PHY layer communications technologies. It provides a fundamental view for network O&M that application requirements can be negotiable with detailed network operations such that certain quality are maximized for all existing tasks, and blocking probability is minimized.

4.9 Summary

In this Chapter, a new approach to QoI-aware network O&M design for task-oriented applications in WSNs is proposed. Different from other research work focusing on network utility maximization problem with predefined utility functions, this Chapter employs a unique and runtime design perspective where the WSN learns and optimizes the network utility by probing the satisfaction levels of completed tasks. Four key design elements are

proposed, including a novel concept of QoI satisfaction index, a QoI network capacity, a negotiation-based admission control process, and an optimal resource allocation scheme. Next, the optimal system design parameters are analyzed. Extensions and discussions on the associated middleware architecture and thoughts for pursuing future work on managing actuator-related decisions were presented. 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 Networks

PREVIOUS Chapters mainly focus on the design aspects of a single multi-hop wireless network, from network protocol designs like QoS routing, distributed scheduling, and admission control algorithms, to the negotiation-based network O&M issues. Furthermore, the research so far has proved that the proposed algorithms, models, and protocols could successfully support some notion of quality for a single multi-hop wireless network. However, it is usually necessary to maintain communications to support certain service quality among *multiple*, *disconnected* network components, each of which could be the one designed in previous Chapters; and this is the primary motivation for the research presented in this Chapter, where we shall focus on how to maintain communications among multiple disconnected networks.

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 network components, we assume data ferries are equipped with wireless sensors, which could collect the data from one network domain (served as the source) and deliver the data to the other network domain (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 scope of any one of the entire network domain. While existing work

has explored various trajectory designs for the data ferry by assuming either static nodes or full observations at the data ferry, the problem still remains open when the nodes are mobile in each network domain, 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.

5.1 Introduction

Continuing advances in sensor technologies and pervasive computing brings in new perspectives to solving challenging communication problems. Consider a partitioned network of mobile nodes, each representing a group of physical nodes performing group mobility, 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, the nodes each operate in different regions and do not have direct contacts. 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 [36] 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 [36, 95, 96]. In practice, however, complete sensing coverage

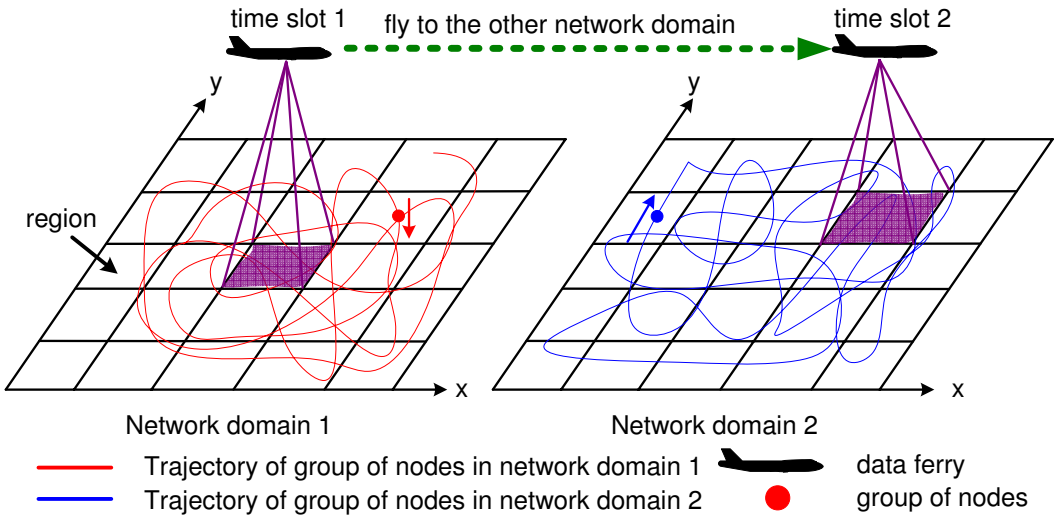

Figure 5.1: An example of how to bridge the communications between two disconnected mobile network domains using a unmanned, sensor-mounted data ferry, where two groups of nodes move on disjoint trajectories and the data ferry has only limited (square as shown in this illustrative example) sensing range.

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 region, 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 network domain into regions as shown in Figure 5.1 (in 1-D case, a region refers to a segment). Furthermore, one sensing point is selected per cell such that the data ferry can sense and communicate with a node anywhere within the cell. 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 sensing points. Periodically, the data ferry senses the presence of nodes and uploads/downloads messages 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 cells of their local region 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, where instead of designing fixed trajectories, we design control policies that dynamically map available information to navigation actions at run time. Our specific contributions are three-fold:

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, modeled by Markov chains on the partitioned space, and the design criteria, modeled by a payoff function and a reward structure. For concrete analysis, we aim to maximize the total number of effective contacts with exponential discounts, although other criteria can also be used.

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 random walks. 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 to benchmarks. 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. Section 5.7 summeriness this Chapter.

5.2 Related Work

Recently, the idea of using designated mobile nodes to support communications in poorly connected networks is emerging [95–98], where a mobile backbone is constructed to cover all the task nodes if sufficient helper nodes are available [97, 98], 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 observation, first proposed in [99]. Although extensively studied in operation research and robotics, to our best knowledge its application on mobility control in commu-

nication networks has not been explored before. Recent work [100] claims to use 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 disconnected network partitions, where each partition contains a group of nodes moving in a disjoint local region according to certain group mobility patterns, 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 messages while awaiting contacts with the data ferry. In the sequel, all references to "node" mean such group heads.

Each data ferry is assigned to serve multiple domains, and the assignment is disjoint for different ferries[1], which makes it suffice to focus on one ferry. The ferry periodically senses node presence within a certain range, exchanges messages with nodes upon contacts, and buffers messages in between. Moreover, it has a controller which can dynamically navigate the ferry among sensing points.

The rest of the subsections specify the framework of control based on POMDP framework.

5.3.2 State Space and Mobility Model

To facilitate control, we partition the network field of each node into cells, labeled $0, \ldots, n_i$ for node i, so that the data ferry is able to sense a node and exchange messages with it once they are in the same cell. In this Chapter, we keep a minimum state space where the controller represents the network state only by the cell index of the node it is trying to contact in the current slot to focus on the issue of stochastic movements and partial observations, *i.e.,* if the current node in slot t is node i, the state is $s_t \in \mathcal{S}_i = \{0, \ldots, n_i\}$.

[1]Coordinated service by multiple ferries per domain will be explored in future work.

Other network characteristics, such as traffic demands, quality of service requirements, and buffer size constraints, are also information of interest and will be explored in future work.

We model the mobility of each node by a Markov chain on the quantized space derived from the above partition. Let $P^i = (P^i_{jk})_{j,k\in\mathcal{S}_i}$ be the transition matrix for node 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 nodes and drop the superscript i in the sequel, but the framework can be easily amended for heterogeneous cases.

5.3.3 Action Space

The action u_t specifies the cell the data ferry will move to in the coming slot, 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 cells nodes may visit, *i.e.,* $u_t \in \bigcup_{\forall i} \mathcal{S}_i$, which may grow linearly with the number of nodes. To keep the policy scalable, we consider a hierarchical design where the action only specifies where to move in the region of the node the ferry is trying to contact, and the order of nodes is controlled by an upper layer policy. For example, one can use existing ferry route design algorithms [95] to obtain a node sequence that optimizes certain performance metrics.

Under such an hierarchical design, the action space is divided into two subsets: "follow" actions $\mathcal{U}_f = \{$"follow" 0, "follow" $1, \ldots,$ "follow" $n\}$ and "switch" actions $\mathcal{U}_s = \{$"switch" 0, "switch" $1, \ldots,$ "switch" $n\}$, where "follow" j means to navigate to cell j of the current region to follow the current node, and "switch" j means to switch to cell j of a new region specified by the upper layer policy. The reason to allow for switching is that occasionally, nodes may wander away from their usual cells, which will cause consecutive misses, and it may be more efficient to move on to other regions first and revisit this node later. In the case where skipping a node is undesirable (*e.g.,* messages 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 on the latter. The

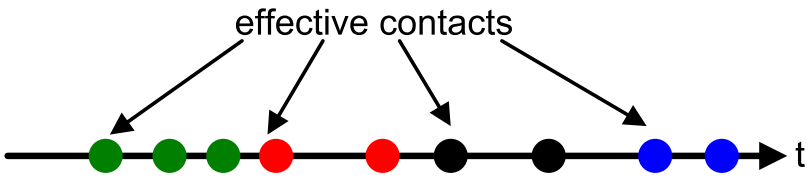

Figure 5.2: An example of the defined effective contacts. Marks of the same color represent consecutive contacts with the same group of nodes in one network domain.

overall action space is given by $\mathcal{U} = \mathcal{U}_f \cup \mathcal{U}_s$.

5.3.4 Observation Model

The onboard sensor produces a binary observation per slot. Let $z_t \in \mathcal{Z} = \{0, 1\}$ denote the observation in slot t, where $z_t = 1$ means "contact" and $z_t = 0$ means "miss". Then under perfect sensing, $z_t = 1$ if and only if $u_{t-1} = s_t$, *i.e.,* the cell the ferry decides to navigate to (one slot earlier) coincides with the cell the node moves 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 $z_t = 1$ with certain probability if $u_{t-1} = s_t$.

5.3.5 Payoff Function

The payoff function represents the goal of control. Since the job of a data ferry is to ferry messages between different nodes, an intuitive goal is to maximize the total number of *effective contacts*, which is defined as the first contact after switching the targeted network domain. Note that this effectively prevents the trivial case of being stuck with the same node since only the first of each run of consecutive contacts with the same node 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 node or not, but we also need to know whether the next contact is effective or not. Let $y_t \in \{0, 1\}$ denote such an indicator, where $y_t = 1$ means effective contact and $y_t = 0$ otherwise. The payoff function $r(z_t, y_t)$ is given by:

$$r(z_t, y_t) = z_t \cdot y_t, \tag{5.1}$$

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* T, which is a period of time the controller is optimized upon. In this Chapter, we consider the discounted rewards:

$$R_T \triangleq \mathbb{E}\Big[\sum_{t=1}^{T} \gamma^t r(z_t,\, y_t)\Big], \tag{5.2}$$

where $\gamma \in (0,\, 1]$ is a discount factor specified by the application. Note that this payoff function has not taken into account the cost. 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 and observations, to a new action in each slot, *i.e.,* $\pi = (\pi_t)_{t=1}^{T}$, and

$$\pi_t\big(\underline{z}_{1:t-1},\, \underline{u}_{1:t-1}\big) = u_t, \tag{5.3}$$

where we use $\underline{x}_{t_1:t_2}$ to denote the vector $(x_{t_1}, \ldots,\, x_{t_2})$ $(t_1 \leq t_2)$[2]. Denoting the expected cumulative payoff under a policy π by

$$R_T^{\pi} \stackrel{\Delta}{=} \mathbb{E}\Big[\sum_{t=1}^{T} \gamma^t r(z_t,\, y_t) \big| u_t = \pi_t\big(\underline{z}_{1:t-1},\, \underline{u}_{1:t-1}\big)\Big], \tag{5.4}$$

where the expectation is taken over all possible node movements and observations, the problem is to find a policy π that maximizes R_T^{π} over all feasible policies for a predetermined horizon T. We are particularly interested in stationary policies, *i.e.,* $\pi_t \equiv \pi$ for all t, which maximizes the long-term payoff as $T \to \infty$.

[2]The underlined notation signifies a vector quantity in this Chapter.

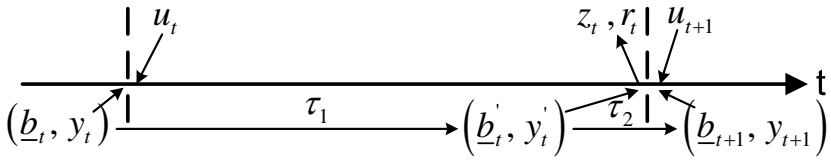

Figure 5.3: The order of action, state transition, and observation during one time slot (*i.e.,* one sensing period).

5.4.1 Belief Updates

It is known that the sufficient statistics for the past information is the *belief vector* (also called belief state) $\underline{b}_t$, which is the posterior distribution of the (current) node given all past observations, *i.e.,* $\underline{b}_t = \big(\Pr\{s_t = j|\underline{z}_{1:t-1}, \underline{u}_{1:t-1}\}\big)_{j\in\mathcal{S}}$. Enriched with the indicator y_t, we obtain a new state $(\underline{b}_t, y_t)$ which is the state the controller will act upon.

To understand the control process, let us zoom in to one slot starting with some initial state $(\underline{b}_t, y_t)$, as illustrated in Figure 5.3. At the beginning of the t^{th} slot, the data ferry chooses the next action based on its policy $u_t = \pi_t(\underline{b}_t, y_t)$ and moves accordingly. At the same time, the node also moves according to its own mobility pattern represented by state transition matrix P, and reaches a new belief state $P^T\underline{b}_t$ by the end of the slot. However, if the data ferry chooses to switch to a new node, then the "current node" changes, and its belief of the new node is reset to the limiting distribution $\underline{b}_0$ (assume it exists) while the indicator y_t is also reset to 1. Combining these two cases give the state transition specified by $(\underline{b}'_t, y'_t) = \tau_1(\underline{b}_t, y_t|u_t)$ as:

$$\tau_1(\underline{b}_t, y_t|u_t) = \begin{cases} (P^T\underline{b}_t, y_t) & \text{if } u_t \in \mathcal{U}_f, \\ (\underline{b}_0, 1) & \text{if } u_t \in \mathcal{U}_s. \end{cases} \tag{5.5}$$

At the end of the slot, the data ferry takes an observation z_t and earns a unit reward if effective contact occurs. The expected payoff is given by

$$r_t = r(\underline{b}'_t, y'_t, u_t) = y'_t b'_t(u_t). \tag{5.6}$$

In addition, the data ferry will update its belief vector according to the Bayesian rule. Let

$\mathbf{e}_u$ denote the unit vector with 1 at the u^{th} element and 0 elsewhere, and $[\underline{b}'_t]_{\setminus u}$ the belief vector derived by setting the u^{th} element of $\underline{b}'_t$ to 0 followed by normalization. Then the Bayesian update $(\underline{b}''_t,\ y''_t) = \tau_2(\underline{b}'_t,\ y'_t|u_t,\ z_t)$ is given by:

$$\tau_2(\underline{b}'_t,\ y'_t|u_t,\ z_t) = \begin{cases} (\mathbf{e}_{u_t},\ 0) & \text{if } z_t = 1, \\ \left([\underline{b}'_t]_{\setminus u_t},\ y'_t\right) & \text{if } z_t = 0. \end{cases} \tag{5.7}$$

The updated state is then used as the new state for the next slot $(\underline{b}_{t+1}, y_{t+1}) = (\underline{b}''_t,\ y''_t)$, and the 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 update reflecting state transition (τ_1) depends on the next action and thus has to be deferred to the next slot.

5.4.2 Optimal Policy and Value Iteration

Let the *value function* $V_T(\underline{b},\ y)$ denote the cumulative payoff over horizon T starting from state $(\underline{b},\ y)$. Then the optimal value function must be the solution of the *value iteration* (VI) in the following form [101]:

$$V_T(\underline{b},\ y) = \gamma \max_{u\in\mathcal{U}} \left[r\left(\underline{b}',\ y',\ u\right) + \sum_z p(z|\underline{b}',\ u) V_{T-1}\left(\underline{b}'', y''\right) \right], \tag{5.8}$$

where $p(z = 1|\underline{b}',\ u) = b'(u)$, and $p(z = 0|\underline{b}',\ u) = 1 - b'(u)$, and the optimal policy must be the one achieving the optimal value function, *i.e.,*

$$\pi_T(\underline{b},\ y) = \arg\max_{u\in\mathcal{U}} \left[r\left(\underline{b}',\ y',\ u\right) + \sum_z p(z|\underline{b}',\ u) V_{T-1}\left(\underline{b}'', y''\right) \right]. \tag{5.9}$$

For infinite horizon $T = \infty$, it is known that the VI will converge as long as $\gamma < 1$, and the limit V_∞ gives the optimal stationary policy that maximizes the long-term total (discounted) payoff. As stationary policies are easy to implement, and the lifetime of a data ferry is typically long relative to its sensing period, we are particularly interested in designing stationary policies with good payoff over large horizons.

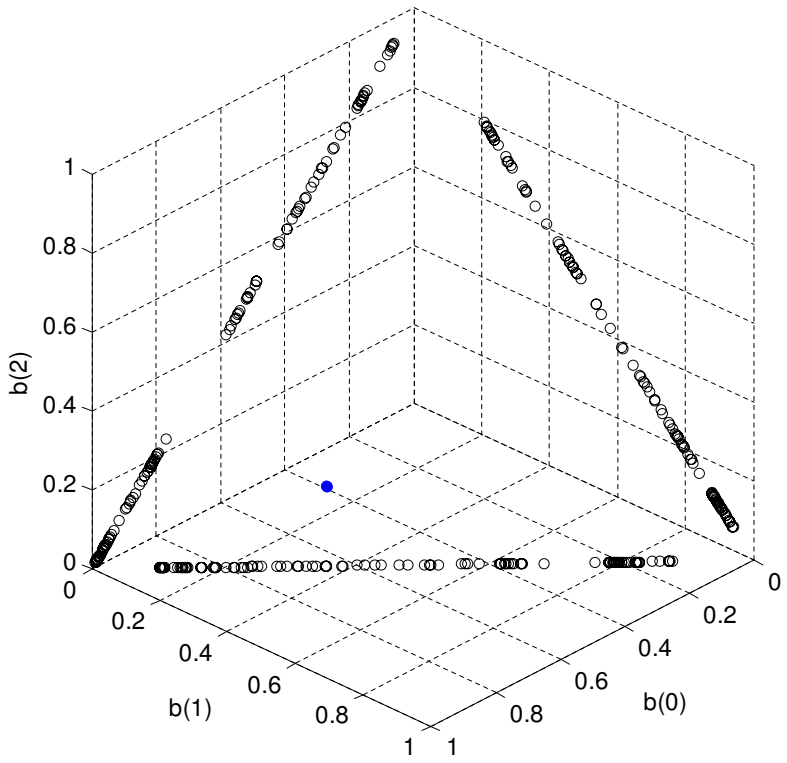

Figure 5.4: The set of reachable belief vectors in 3-D belief simplex where considered parameters are $n = 2$, $p = 0.1, q = 0.2$, and 6 iterations for VI. Legend $\circ$ represents a reachable belief vector.

5.5 Hardness Result and Efficient Heuristic Policies

It is known that to compute π_T for arbitrary initial state is PSPACE-hard [102]. 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 node movement), then it can be shown that the complexity is reduced to $O(|\mathcal{U}|^T)$, where $|\mathcal{U}|$ denotes the number of actions. This is primarily because the number of reachable states grows as $|\mathcal{U}|^{T-1}$ when value function (5.8) iterates at each step, and we need to optimize value functions over $|\mathcal{U}|$ 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 T. As an illustrative example, Figure 5.4 shows $p = 0.1, q = 0.2, n = 2$ case, where observation z_t can limit the belief points to subspaces one dimension smaller than the original simplex and significantly improve the performance.

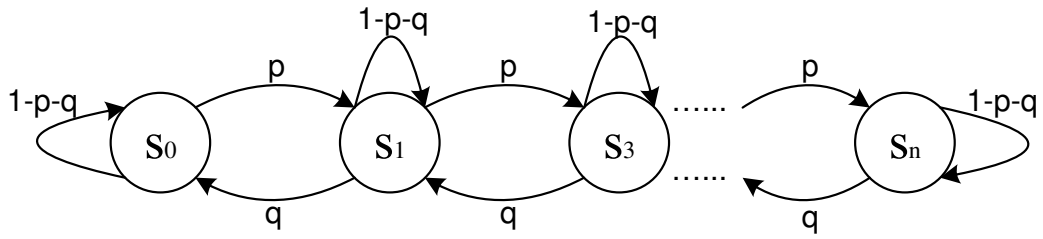

Figure 5.5: The state transition diagram for the considered random walk mobility model.

Now, 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 vector. Ideally, $\mathcal{B}$ should represent all the reachable belief vectors during offline policy computation and online data ferry navigation. Nevertheless, since the number of reachable belief vectors grows exponentially over time, certain approximation method has to be used by selecting *representative* beliefs. Grid-based algorithms [103] have been proposed which limit VI to only a set of belief points sampled from the belief simplex. We use a similar idea but adopt a simple nearest-neighbor quantization approach which approximates the values of un-sampled beliefs in (5.8) by the values of the nearest belief samples, *i.e.,*

$$\hat{\underline{b}}''_{k,z} \leftarrow \arg \min_{j=1,2,\ldots,|\mathcal{B}|-1} \|\underline{b}_j - \underline{b}''_{k,z}\|, \forall \underline{b}_j \in \mathcal{B}, \tag{5.10}$$

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 2 shows the flow of computing suboptimal policy π_T, where ϵ is the application defined parameter for convergence condition.

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 $N = 3$ moving nodes on three disjoint routes

Algorithm 2 : Approximate VI based on belief sampling

Initialize: $T = 1, \epsilon > 0, \gamma > 0$, $\hat{V}_T \leftarrow 0$
for all $\underline{b}_k \in \mathcal{B}$, where $k = 0, 1, \ldots, |\mathcal{B}| - 1$ **do**
  for all $y \in \{0, 1\}$ **do**
    for all $u \in \mathcal{U}$ **do**
      compute $\underline{b}''_{k,z}$ in (5.7) for $z = 0$ and $z = 1$
      if $\nexists \underline{b}''_{k,z} \in \mathcal{B}$ **then**
        $\hat{\underline{b}}''_{k,z} \leftarrow \arg\min_j \|\underline{b}_j - \underline{b}''_{k,z}\|, \forall \underline{b}_j \in \mathcal{B}$.
      end if
      update:

$$Q_T(\underline{b}_k,\, y, u) = r\left(\underline{b}'_k,\, y',\, u\right) + \sum_z p(z|\underline{b}'_k, y')\hat{V}_{T-1}\left(\underline{b}''_{k,z},\, y''\right)$$

    end for
    $\hat{V}_T(\underline{b}_k, y) \leftarrow \max_{u\in\mathcal{U}} \gamma Q_T(\underline{b}_k, y, u)$
    if $\|\hat{V}_T(\underline{b}_k,\, y) - \hat{V}_{T-1}(\underline{b}_k,\, y)\| < \epsilon$ **then**
      continue in Step 3;
    else
      $T \leftarrow T + 1$
    end if
  end for
end for
Return: policy π_T.

(range 50 miles), each of which is partitioned by five (*i.e.,* $n = 4$, which will be tuned later) states. 1-D random walk [104] mobility model is used with parameters p, q, as shown in Fig 5.5. Other design parameters include discount factor $\gamma = 0.7$ and horizon $T = 20$.

Two illustrative examples to demonstrate the real-time data ferry navigation with different mobility patterns are shown in Figure 5.6(a) and Figure 5.6(b), where mobility parameters are set differently to distinguish two special cases, (a) localized mobility pattern, and (b) relatively random mobility pattern. The more localized 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 random mobility, leads to more ineffective navigations/misses. Furthermore, the plot exhibits positive correlation between the randomness in node movements and that in the movement of the ferry, suggesting that the proposed controller is indeed able to adapt to node mobility patterns.

We compare our proposed algorithm with predetermined switching policy as a benchmark, *i.e.,* the data ferry keeps switching among the most likely spots of different

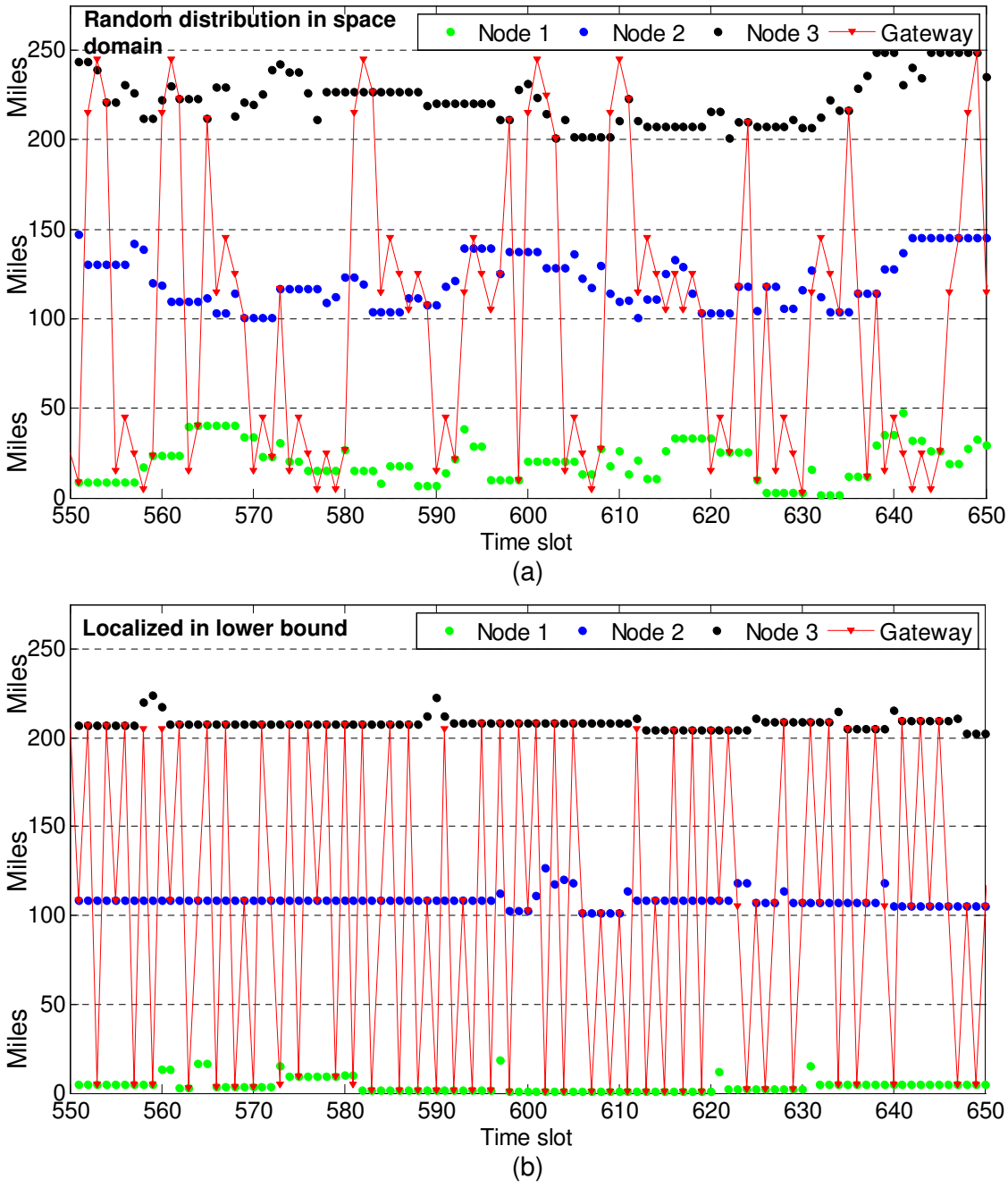

Figure 5.6: Two simulated trajectories of the data ferry, where three groups of nodes move within three disjoint 1-D network domain with length 50 miles each. Conspired mobility patterns include (a) $p = q = 0.3$, and (b) $p = 0.1, q = 0.8$.

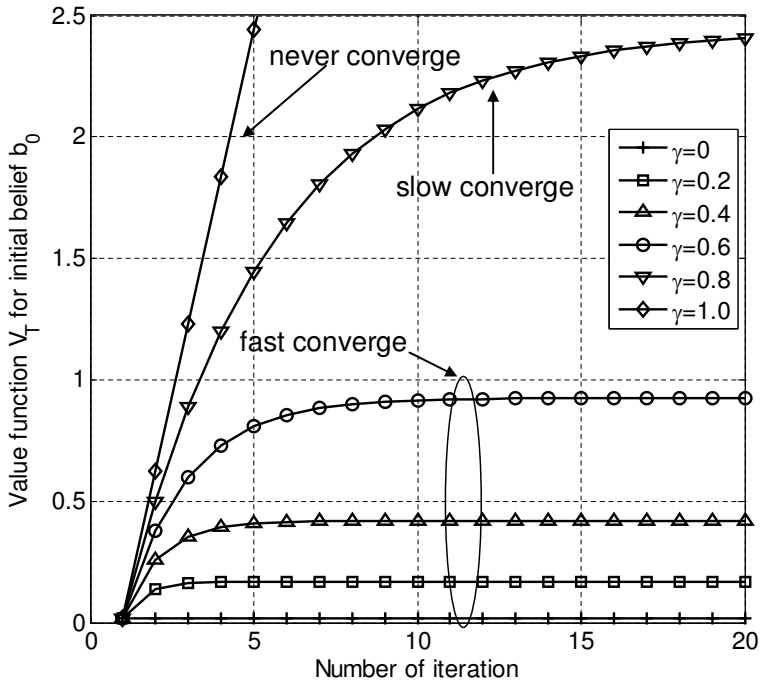

Figure 5.7: Simulation result of the impact discount factor γ on the speed of convergence of VI.

nodes, regardless of the observation. Figure 5.7 demonstrates the impact of discount factor γ on the speed of convergence for VI. It is interesting to observe that for relatively small $\gamma < 0.5$, design horizon $T = 10$ successfully guarantees relatively fast convergence, compared with slow convergence $T = 20$ if γ is increased. Case $\gamma = 1$ yields non-convergence of the VI because the total discounted reward becomes unbounded as design horizon increases.

Figure 5.8 illustrates the impact of different belief sampling techniques on the policy calculation, by comparing the proposed approximate VI based on sub-simplex belief sampling and that based on entire simplex sampling, together with switching policy, w.r.t. the number of samples 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. Meanwhile, for a fixed scheme, a larger number of samples results in higher reward due to less error incurred in quantization, but this increase is slight which suggests the proposed policy computation algorithm is robust to the selection of representative beliefs.

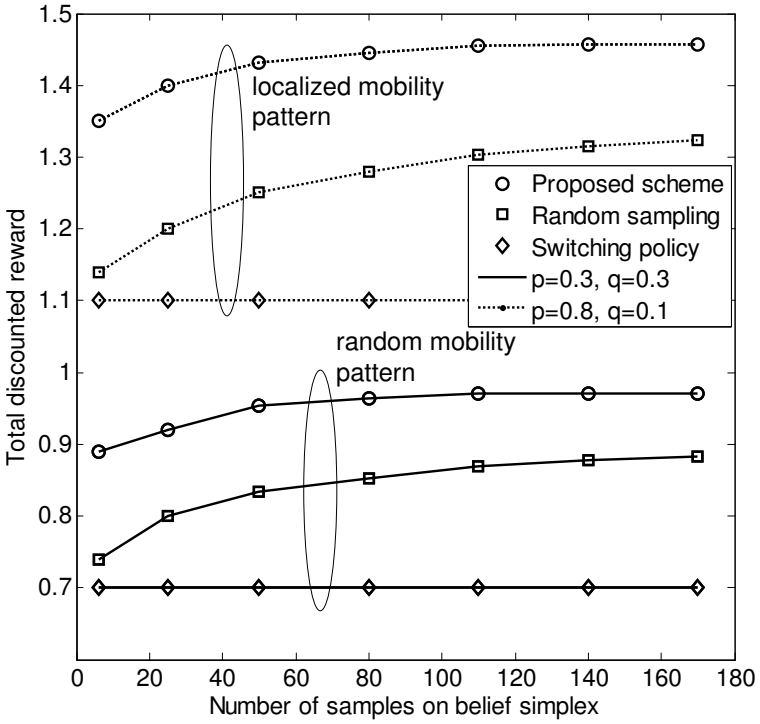

Figure 5.8: Simulation result of the impact of different sampling techniques on the belief simplex w.r.t. different mobility models and number of samples.

Figure 5.9 shows the impact of state partitioning on total discounted reward, w.r.t. different mobility models, while changing mobility parameters p, q. Two cases are compared, localized mobility pattern ($p = 0.8, q = 0.1$), and relatively randomized location distribution ($p = 0.3, q = 0.3$). For fixed geographical area, a larger n represents a smaller sensing range of the data ferry; however, for fixed sensing range, it represents a larger network field. 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 node, 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 localized by using more divergent values for p, q, the decay with n greatly slows down. This is because under more localized mobility, the controller can predict node position relatively accurately within a small neighborhood, and the size of the entire field no longer matters as much. Again, switching policy performs the worst in both cases compared with our proposed scheme.

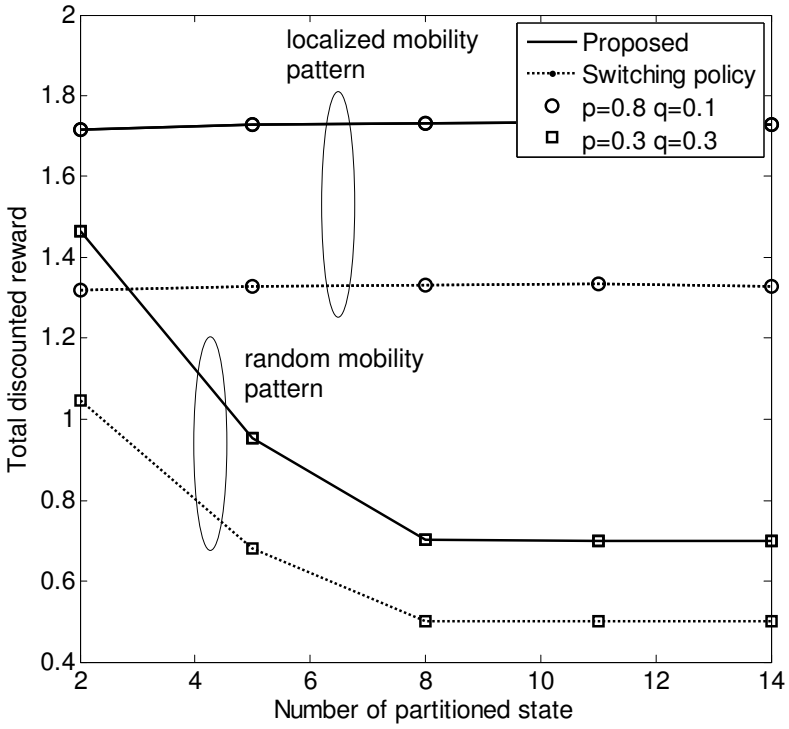

Figure 5.9: Simulation result of the impact of state partitioning w.r.t. different mobility models and the number of state partitions.

5.7 Summary

In this Chapter, the problem of dynamic control of data ferries under partial observations are investigated with the goal of bridging communications between disconnected mobile nodes in a delay-tolerant manner. 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. Initial simulation results show that the proposed scheme achieves significantly more contacts when compared with the predetermined counterpart. Future work includes improving and analyzing the heuristic policies, extending the model to capture the actual scenarios, and evaluating the proposed policies on real mobility traces.

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 operations and management (O&M). 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 separate Chapters.

Chapter 2 proposed a novel cross-layer design solution integrating the distributed

scheduling, QoS routing, and connection admission control (CAC) 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 successful guarantees QoS supports, and at the same time it opened up another dimension of research to understand the network capacity in any multi-hop wireless networks, from which Chapter 3 is motivated. Chapter 3 proposed a generic capacity estimation and QoS control methodology for the purpose of admission control in any packet network, where the well-known difficulties of estimating the network capacity are tackled by modeling the network as a black box and using a runtime feedback control analysis to quantitatively estimate. Next, this Ph.D. research is further built on top of the proposed complete cross-layer solutions to improve the design efficiency, where Chapter 4 is motivated. Chapter 4 proposed a negotiation-based network O&M framework, bridging applications' service quality demands and the network resource management, while using WSNs as illustrative examples. As for so far this dissertation focused on how to maintain service quality in a single multiple-hop wireless network, the questions still remain if multiple multi-hop wireless networks are disconnected but service quality requirements may extend beyond a single multi-hop wireless network. Therefore, the issue of *inter*-domain communications for *multiple*, *disconnected*, *mobile* multi-hop wireless networks were addressed in Chapter 5 to maintain the service quality through communications, where controlled, unmanned data ferries are used 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 like: what is the analytical model to capture the impacts of statistics feedback delay and statistics collection time on network performance, *i.e.,* borrowing ideas from control research domain, can feedback delay be quantified in a mathematical way?

2. For the network O&M research, questions still remain like to study the impacts of distributed duty-cycling policies on defining sensor 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 caused by the number of tasks and their respective QoI requirements. Since defined QoI network capacity has now a temporal component, can we design the policy in an intelligent manner? Can we negotiate based on the predicted capacity at a given point in time? Can we tune the duty-cycling algorithm as a part of performing negotiation? Can we tune the duty-cycling behavior on a spatiotemporal (stressing the "spatio-" part) basis given the QoI required by the waiting task?

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

   (a) How to improve and analyze the proposed heuristic policies?

   (b) Although the optimal solution has been proven PSPACE hard, how far away the proposed heuristic policies can achieve compared with the optimal policy?

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

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

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

Bibliography

[1] N. H. Vaidya, "Tutorial on mobile ad hoc networks: routing, MAC, and transport issue," in *ACM Mobicom*, 2001.

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

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

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

[10] 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.

[11] Y. Yan, H. Cai, and S.-W. Seo, "Performance analysis of ieee802.11 wireless mesh networks," in *IEEE ICC'08*, May 2008, pp. 2547–2551.

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

[13] 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.

[14] 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.

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

[16] 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.

[17] "Medium access control protocols for ad hoc wireless networks: A survey," *Ad Hoc Networks*, vol. 4, no. 3, pp. 326 – 358, 2006.

[18] 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.

[19] 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.

[20] 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.

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

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

[23] 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.

[24] 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.

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

[26] 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.

[27] 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.

[28] ——, "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.

[29] 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.

[30] 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.

[31] 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.

[32] 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.

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

[34] 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.

[35] 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*.

[36] 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.

[37] 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.

[38] 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.

[39] 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.

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

[41] 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.

[42] 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.

[43] 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.

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

[45] 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.

[46] 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.

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

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

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

[50] 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.

[51] 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.

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

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

[54] 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.

[55] 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.

[56] L. Chen, S. H. Low, J. C. Doyle, and M. Chiang, "Cross-layer congestion control, routing and scheduling design in ad hoc wireless networks," in *IEEE INFOCOM 2006*, Barcelona, Spain, 2006, pp. 1–13.

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

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

[65] 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.

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

[67] 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.

[68] 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.

[69] 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.

[70] 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.

[71] 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 2008 Spring*, Singapore, 2008, pp. 2193–2197.

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

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

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

[75] 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.

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

[77] T. Apostol, *Calculus*. Jon Wiley & Sons, Inc., 1967.

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

[79] 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.

[80] 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–.

[81] 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.

[82] 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.

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

[84] 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.

[85] 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.

[86] 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.

[87] 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.

[88] 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.

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

[90] 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.

[91] 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.

[92] 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.

[93] 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.

[94] 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.

[95] 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.

[96] 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.

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

[98] 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.

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

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

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

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

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

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