University of Alberta

An Innovative MultiVariable Control Framework for Effective Wireless Resource Management

by

Chen Liu

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Department of Computing Science

Fall 2012
Edmonton, Alberta

To my dear supervisors Dr. Janelle Harms and Dr. Mike MacGregor, my beloved parents Honglin Liu and Xingxing Ma, and my love Saul Rodriguez.

Abstract

Over the past two decades, wireless networking has become an enabling technology for ubiquitous computing and inexpensive Internet access. The demand for wireless networking has also increased steadily, including a wider range of applications, larger user populations, and larger network scale. Challenges come from the increasing variety of traffic (such as data, voice and video), service criteria (e.g. QoS/QoE, reliability, security etc.), as well as system requirements (e.g. revenue, fairness, aggregate network performance etc.). Furthermore, wireless networks are less stable, tractable and predictable compared with wired environments, due to physical and management realities. From the physical perspective, the time-varying soft capacity of wireless channels, as well as co-channel and adjacent-channel interference degrades transmission quality and reduces their effective capacity. From the management perspective, uncontrolled resource competition and uneven resource distribution degrades network performance. Although hardware advances are critical to satisfy ever-growing user demands, the efficiency of resource management plays an equally (if not more) important role to push wireless networking to its full potential in terms of satisfying diverse user and system requirements. In this thesis, we propose an effective framework of resource management to reduce the gap between diverse user/system demands and limited delivery capability of wireless networks. Our simulation results demonstrate that this proposed framework of resource management achieves the lowest packet loss rate, best end-to-end delay and fairness without compromising network throughput, compared with the state-of-the-art methods. Additionally, our approach is simple in computation and light in overhead.

Acknowledgements

At the end of my Ph.D. journey, there are so many people I would like to thank. Without your support, it is impossible to finish this thesis.

Foremost, I would like to thank my supervisors Dr. Janelle Harms and Dr. Mike MacGregor for their invaluable guidance, wise advice, encouragement, trust and tremendous support. Words cannot express my gratitude and appreciation to them. When I had troubles, no matter in life or research, it was them who guided me to find the right direction. When I got frustrated, it was them who encouraged me and helped me to regain confidence. When I made mistakes, it was also them who patiently gave me wise suggestions, and continuously have faith in me. When I succeeded, they gave all credits to me. During my entire Ph.D. program, their wisdom, knowledge, expertise and experience helped me to grow from a naive student into a mature researcher. They have been my inspiration and role models of enthusiastic and excellent researchers. Their influence will benefit me for my life time.

I would also like to express many thanks to my committee members, Dr. Ioanis Nikolaidis, Dr. Hossam Hassanein and Dr. Armann Ingolfsson. I very much appreciate their valuable suggestions on my thesis. I would also like to express my appreciation for Dr. Hossam Hassanein being the external member of the committee.

I want to thank Dr. Martha Streenstrup for being such a good tutor and friend. She has greatly influenced my research direction, attitude and methodologies. Over the past few year, Martha gave me constructive advice and genuine comments on my research ideas. I benefited a lot from her challenging questions and intriguing discussions. From her, I learned the adventurous spirit of a scientist, critical and honest attitude of a researcher. Furthermore, I appreciate her tremendous help during my job hunting. I would also like to thank my good friend Moslem Noori. His friendship and support helped me to go through my personal crisis during my doctoral program. My appreciation goes to Dr. Damla Turgut for introducing me to the LCN committee, and numerous review opportunities. Here, I would also like to thank Dr. Israat Tanzeena for giving me the opportunity of being a volunteer for the IEEE Northern Canada Section. I would also like to thank the networking group, for their valuable questions and comments.

My sincere and warm thanks to my dad (Honglin Liu) and my mom (Xingxing Ma).

Their unconditional love and support made it possible for me to achieve today's success. I thank them for never giving up on me. My dad has been a great tutor and role model throughout my life. He is the one who warns me potential pitfalls when I succeed, and encourages me when I fail. My mom has been my best friend who I share everything with. She has the magic to always make me laugh and feel relaxed. At every important step of my life, and upon every important decision, my parents have always been there for me by giving me valuable suggestions and support. I also thank them for their sacrifice. They have enduring loneliness without my accompany for more than eight years. I also thank them for not letting me worrying about anything at home. I can never thank enough of their love and support.

Last but not the least, I want to thank the love of my life, Saul Rodriguez. He has been there with me for my entire Ph.D. journey, and witnessed all my success and failures. It was his unconditional love and support that helped me through my ups and downs. With him, I am not afraid of any challenges and difficulties. Because I know no matter what he will always be there to comfort me and help me. He made feel that everything will be okay. He is the one that I can always count on. With him, I do not care where I am or what I do. With him, I know love, happiness, and meaning of life. With him, I know what home is. I would also like to thank Saul's family (Dr. Angel Rodriguez, Eva Vazquez and Liz Vazquez) for their love and support.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Over the past two decades, wireless networking has become an enabling technology for ubiquitous computing and inexpensive Internet access. With more application diversity, larger user populations, and wider network coverage, the demand for wireless access has increased quickly and steadily. According to Cisco's report in [17], traffic from wireless devices will exceed traffic from wired devices by 2015. Furthermore, the user demand for finite resources (e.g. bandwidth, frequency and energy etc.) and service criteria (e.g. quality of service/experience, reliability, security etc.) are diverse, due to the variety of traffic (e.g. data, voice and video etc.). Besides user demand, different systems also have different requirements such as maximizing aggregate performance, network utility, system revenue or fairness.

Against this trend, the limited delivery capability of wireless networking poses unique challenges to satisfy such diverse user/system demand. Some of these capability limits are:

- Limited Wireless Bandwidth
  Bandwidth is scarce in wireless networks. For example, IEEE 802.11a/b/g provides up to 54 Mbps (megabits per second) data rate [36]. The newest standard, IEEE 802.11n, boosts the maximum data rate to 600 Mbps [39]. Besides IEEE 802.11, WiMAX offers higher data rate. IEEE 802.16m promises to support data rate up to 1 Gbps (gigabits per second) [40]. IEEE 802.15.4 provides up to 250 Kbps (kilobits per second) data rate for low-rate wireless personal area networks[37]. Furthermore, when wireless interference is taken into account, the effective bandwidth available for use is even less. In comparison, 10 Gigabit Ethernet is standardized in IEEE 802.3ae [38], and 100 Gbps Ethernet is expected to be on market soon.

- Lossy Wireless Transmission
  Wireless networks are less reliable, tractable and predictable compared with wired environments. Due to interference, transmissions over wireless medium cause collisions and lead to data loss. Without effective control, dynamic interference scenarios can

significantly degrade transmission quality and reduce effective capacity.

- Poor Multi-Hop Delivery
  Wireless multi-hop forwarding offers a large coverage area and reduces costs compared with installing cables. However, network performance is compromised in terms of throughput and delay. The co-existence of intra-path and inter-path interference obstructs data forwarding by causing collisions within the same path and between different paths. Furthermore, wireless interference contributes to network congestion; while congestion returns the favor by aggravating collisions. Consequently, the effective capacity is even smaller in a multi-hop wireless network.

It requires both hardware breakthroughs and management efforts to reduce the gap between diverse user/system demand and limited wireless delivery capability. On one hand, hardware advances help to increase bandwidth, improve resistance to interference, and relieve the resource shortage due to fast-growing user demand. On the other hand, effective resource management plays an equally important role to push a wireless network to its full potential in terms of satisfying diverse user and system requirements. In this work, we focus on effective resource management and aim to solve the following problems in both single-hop and multi-hop wireless networks.

- How to allocate resources despite diverse user/system requirements;
- How to utilize allocated resources efficiently despite the co-existence of conflicts and waste;
- How to effectively control network behavior despite dynamic interference;
- How to solve these three problems with very low control overhead and computational complexity.

1.1 Background and Context

In all networks, resources are finite and consumed by multiple users to accomplish various computation and communication tasks. As the number of users increases, available resources may not be sufficient to satisfy all user requests. Therefore, resource competition among multiple users is inevitable. Without appropriate control, aggressive competition may cause starvation and resource waste. Consequently, the network becomes unstable and users may experience long delay, significant packet loss and poor throughput. How to manage wireless resource consumption efficiently is the main goal of this thesis. Before diving into the solution domain, this section analyzes wireless resources, users and different scenarios of resource competition.

1.1.1 Wireless Resources

Wireless networks have different types of resources including transmission bandwidth, channel frequency, energy, storage, and processing cycles etc. In this work, we focus wireless bandwidth for two reasons. First, as described earlier on, bandwidth is scarce in wireless networks compared to that in wired environments. Uncontrolled competition for such limited resources among multiple users can significantly degrade network performance. Furthermore, wireless bandwidth possesses distinct properties in time and space, which we will elaborate in the rest of this section. In the conclusion chapter, we also discuss how to manage other resources including frequency and energy via the proposed framework.

Definitions

Before analyzing the properties of wireless resources, a number of key definitions are clarified first.

- **Wireless Medium**

  Wireless communication via radio frequencies relies on radio waves (with a certain frequency range and wavelength) to transfer information. Due to the shared nature of this wireless medium, two radio devices may interfere with each other when transmitting at the same frequency or adjacent frequencies. Depending on the strength of interference signals and tolerance capabilities of radio devices, collisions may or may not happen.

- **Wireless RF Channels**

  A wireless RF channel is the spectrum block of radio frequencies, and is uniquely specified by a center frequency and frequency bandwidth [99].

  - **Frequency bandwidth**

    In wireless communication, information is transmitted by radio waves over a certain range of frequencies. Therefore, a transmitted signal often consists of multiple frequencies. This range of frequency is called the bandwidth of signal [99]. To accommodate the spectrum of transmitted signals, a wireless channel has to be wide enough. Otherwise, detection distortion will happen [99]. The frequency width of a wireless channel is called frequency bandwidth.

  - **Transmission bandwidth**

    Unlike frequency bandwidth, transmission bandwidth measures how fast data is transmitted over a wireless channel during a certain period of time (e.g. 1 second). The unit of transmission bandwidth is bits per second (bps). However, besides the differences, there is also a close relation between transmission and frequency bandwidth. Specifically, a higher transmission rate requires a wider frequency bandwidth [99]. In the rest of this thesis, we use transmission bandwidth and bandwidth as interchangeable terms.

Properties

The properties of wireless resources can be classified as the finite-and-shared nature, time-varying, mutual exclusiveness, and geographical distribution. These properties are the major causes of competition among multiple users, and are of great importance to how resource sharing is done.

- **The finite-and-shared nature**

  Wireless resources are finite and shared by multiple users. First, there is a finite range of radio spectrum. This fundamentally limits the total number of non-overlapping channels that can be divided within the spectrum. For instance, IEEE 802.11b offers 14 channels and only a few of them are not overlapping (e.g. 1, 6 and 11)[36]. Although transmitting through non-overlapping channels avoids interference, the growing demand of wireless access causes the scenario of multiple users sharing a single channel to be inevitable. Second, there is finite bandwidth of a wireless channel. This also restricts the rate of data transmission. If a traffic load exceeds the capability of a channel, congestion will happen.

- **The time-varying nature (soft capacity)**

  There are two major differences between wired and wireless links. First, a wired link exists physically by connecting two machines via a cable. In comparison, a wireless link is defined as an abstraction of the transmissions between a pair of sender and receiver. Second, compared with a wired link, the capacity of a wireless link is time-varying due to two factors. For one, the Shannon-Hartley theorem characterizes the channel capacity C of a specified bandwidth B in the presence of noise N as $C = B \cdot \log(1 + \frac{S}{N})$, where S is the average received signal strength [84]. According to this theorem, the smaller the signal to noise ratio (SNR) is, the lower the channel capacity is. For another, a wireless channel may be shared by multiple links. The more links are transmitting over the same channel, the smaller portion of the channel capacity each link can use. Furthermore, more links also mean stronger interference. Consequently, more capacity may be wasted on collisions. We call this time-varying capacity of a wireless link, soft capacity.

- **The mutual exclusiveness of channel access**

  Co-channel interference is the major cause of the competition for channel access in wireless networks. For each transmission pair (sender and receiver), any simultaneous transmissions within a certain region may cause interference and even collisions. Depending on the factors including 1) the level of interference tolerance of radio receivers; and 2) traffic schedules of neighbor nodes, this region may vary over time. However, within the contention region of a transmission pair, mutual exclusive access is required.

- **The geographical distribution of wireless nodes**

  Another property of wireless resources is geographical distribution. This property separates wireless nodes in space and enables parallel transmissions among spatially separated nodes. Therefore, local temporal competition among wireless transmissions can be compensated for via spatial separation.

1.1.2 Wireless Users

Users in wireless networks can be defined in various ways. Typically, different layers of the network protocol stack have different definitions of users. Furthermore, users at different levels have different concerns and requirements.

- Application-level users can be end-users in the real world (e.g. families, organizations and companies).

  Real-world users may be ranked based on the price of paid services. Users with the same rank expect the services with equal quality. Users with different ranks expect service differentiation.

- Network-level users can be end-to-end information sessions (e.g. multi-hop data flows). End-to-end flows may or may not correspond to real-world users. Different flows may carry different types of traffic, and each type of traffic has a certain performance requirement. In addition, depending on the forwarding paths, different flows may experience different performance in terms of packet loss, delay and throughput.

- Link-level users can be individual nodes or transmission pairs (e.g. between a sender and receiver).

  We define a wireless link as an abstraction of the transmissions between a pair of sender and receiver. Link-level users compete for resources locally. Local resource competition varies according to the number of users and traffic demand. Depending on medium access control policies, aggressive users may monopolize channel access and bandwidth, which causes starvation and unbalanced consumption of resources.

Despite these differences, users at different levels belong to the same network. Besides handling their own unique problems, different levels should cooperate to achieve a coherent objective. In this work, users are identified as end-to-end flows at the network layer. Each flow can be mapped to an application level client and decomposed to flow segments at the link layer. A special case is the single-hop networks where a network-layer flow directly corresponds to the flow at the link layer.

1.1.3 Resource Competition

Resource competition is common when finite resources are shared by multiple users. Especially, in wireless networks, interference, limited bandwidth and imperfect scheduling contribute to conflicting resource consumption.

- **Wireless Interference**
  Adjacent transmissions that use the same frequency channel cause co-channel interference. Consequently, multiple transmissions compete for channel access when a single channel is shared. Uncontrolled competition for channel access may lead to conflicts, which needs scheduling to remedy.

- **Insufficient resources**
  Conflicts occur when available resources are insufficient to satisfy user demand. Without control, aggressive users may monopolize limited resources and cause other users to starve. Such unfair competition results in significant performance degradation, and may lead to system instability.

- **Imperfect scheduling**
  Conflicts may also be caused by imperfect scheduling regardless of the sufficiency of available resources. This may cause inefficient resource utilization and even waste.

These three causes usually co-exist, resulting in correlated competition for wireless channels and dynamic interference scenarios in time, space, strength and frequency.

Temporal Interference Dynamics

Temporal interference occurs when interfering signals start at different phases over the lifespan of a transmission. These phases can be summarized as: before, simultaneous with, and after the start of a transmission. We call these three scenarios earlier-transmission-caused, simultaneous-transmission-caused and future-transmission-caused interference.

- **Earlier-Transmission-Caused Interference**
  The first type of interference occurs when two adjacent transmissions A and B start at different but overlapping times. As shown in Fig. 1.1, transmission A starts earlier. From transmission B's perspective, the interference is imposed by the earlier transmission of A.

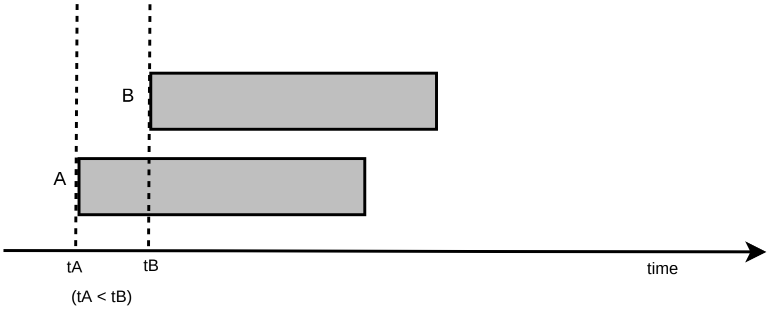

Figure 1.1: Earlier-Transmission-Caused Interference

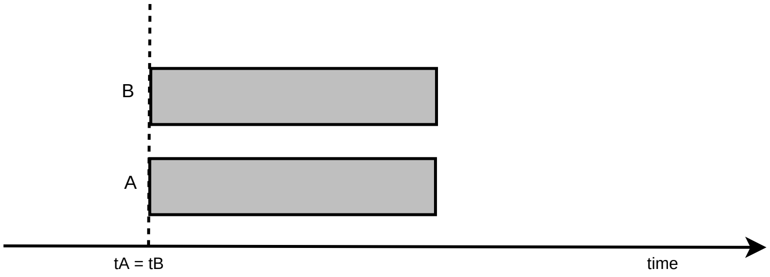

Figure 1.2: Simultaneous-Transmission-Caused Interference

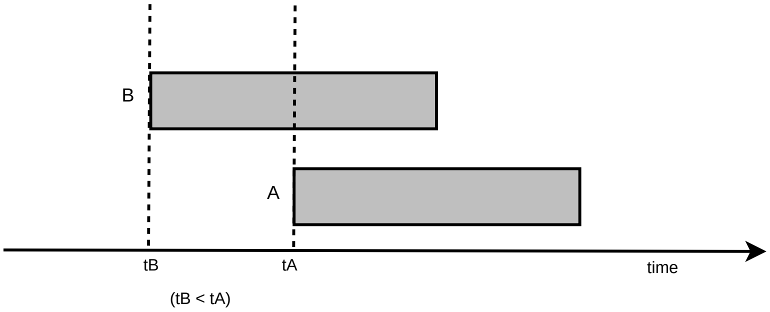

Figure 1.3: Future-Transmission-Caused Interference

- **Simultaneous-Transmission-Caused Interference**

  The second interference scenario is caused by simultaneous transmissions. Fig. 1.2 illustrates that two transmissions A and B that start at the same time and cause interference with each other.

- **Future-Transmission-Caused Interference**

  The third interference scenario occurs when transmission B starts early, and transmission A starts later in time (Fig. 1.3). From B's perspective, the interference is imposed by the later transmission of A. This scenario may occur when A fails to detect B's transmission.

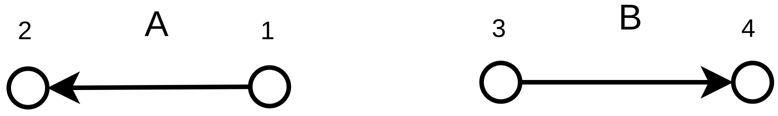

Figure 1.4: Exposed Terminals

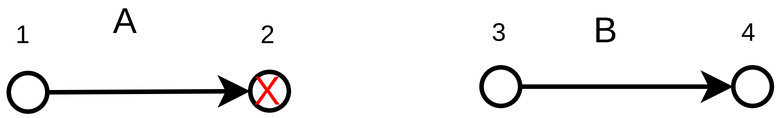

Figure 1.5: Hidden Terminals

Spatial Interference Dynamics

Spatial interference occurs when we consider interference independently of exact timing. Depending on the distance between the two pairs of transmitter and receiver, there are two spatial scenarios: exposed and hidden terminal.

- **Exposed Terminals**

  The first spatial scenario occurs when two transmissions do not proceed at the same time, because they mistake each other as interference. For example, as shown in Fig. 1.4, node 1 starts a transmission to node 2 first. During the transmission period of A, node 3 also needs to transmit to node 4. Although the simultaneous transmission of A and B would not cause a collision, node 3 does not start its transmission until node 1 finishes following CSMA. This is because transmission A is treated as interference.

- **Hidden Terminals**

  The second scenario occurs when two transmissions fail due to undetected mutual interference. For example, as shown in Fig. 1.5, node 1 and 3 cannot hear each other. When they transmit to node 2 and 4 in an overlapping time period, a collision will occur at node 2.

Interference Dynamics in Strength and Frequency

Besides temporal and spatial interference dynamics, wireless interference also varies in signal strength and frequency of occurrence. More specifically, we differentiate signal strength according to whether destructive network behavior (e.g. collisions) is caused. Depending on the patterns of adjacent transmissions, some nodes may experience destructive signals more frequently than others. Depending on the frequency of destructive interference, we differentiate frequent-destructive interference from infrequent-destructive interference.

- **Frequent and destructive interference**

  This scenario can be identified if the transmission between a source-sink pair experiences destructive interference more often than a threshold $dstr\_freq\_thresh$. This scenario has two indications. First, the channel quality is too low to satisfy minimum transmission expectations. Second, fewer transmission opportunities are given to the source-sink pairs which experience frequent-destructive interference. We can remedy this situation in two ways: 1) seeking extra resources, e.g. switch to a different frequency channel; and 2) re-adjusting resource share among frequent-destructive and infrequent-destructive scenarios. For example, we can give frequent-destructive scenarios higher priorities to consume resources. Because multiple channels are not considered in this work, we choose the second option.

- **Infrequent and destructive interference**
  The second scenario describes the case where a source-sink pair experiences destructive interference less often than $dstr\_freq\_thresh$.

Besides the diversity of interference scenarios, their random combinations may also appear and further complicate the situation. Consequently, collisions are hard to eliminate and lead to significant resource waste in wireless networks. Reducing collisions is critical to wireless resource management.

1.2 Scope, Assumptions and Limitations

This section defines the scope of our work, including application scenarios, underlying protocols, traffic model, and hardware limitations etc.

1.2.1 Network Model

We apply the proposed framework of resource management to wireless ad hoc networks. This type of network does not rely on pre-established infrastructure, such as routers or access points [33]. Because all nodes are peers in wireless ad hoc networks, they can communicate with each other directly and participate in routing by forwarding data for other nodes. Our method takes advantage of this feature to create more generalized network topologies, which can generate different interference levels and cover various interference scenarios. Furthermore, our resource management is a generic framework, and can also be applied to infrastructure-based networks, such as wireless mesh networks. Certain modifications might be required. For example, the roles of access points and user devices need to be differentiated. As a first step, we focus on static networks and leave mobility for future work. Both single-hop and multi-hop forwarding are included.

At the physical layer, we assume that all transmissions share a single channel. Although transmitting through channels with non-overlapping channels avoids interference, the growing demand of wireless access causes the scenario of multiple users sharing a single

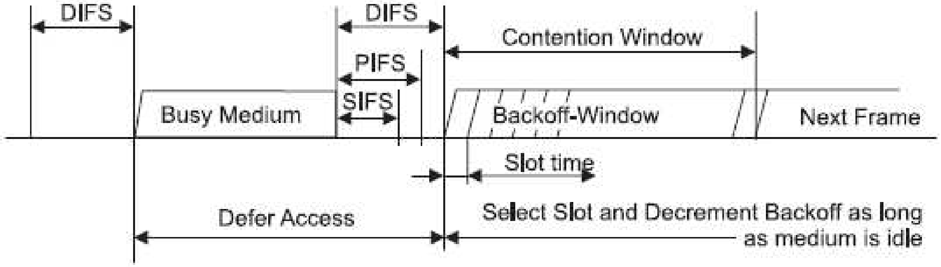

Figure 1.6: Inter-Frame Space Relationship [36]

channel to be inevitable. This research can be easily extended to multi-channel networks. We provide more details at the end of this thesis. Additionally, the antenna in our model is omni-directional, and cannot transmit and receive at the same time, which is consistent with IEEE 802.11 physical layer specifications [36].

1.2.2 Medium Access Control (MAC)

This research focuses on the MAC layer, because medium access is the major performance bottleneck in wireless networks [63]. Poor medium access control results in severe packet collisions, and significantly reduces effective network capacity. We apply G-Local resource management on top of IEEE 802.11 for two reasons. First, IEEE 802.11 is an industrial standard widely adopted in practice. Second, this standard offers carrier sense multiple access with collision avoidance (CSMA/CA) as the basic access mechanism to locally handle various interference scenarios. In comparison to time division multiple access (TDMA), CSMA/CA is simple in computation, requires low control overhead, and is adaptable to dynamic network conditions. IEEE 802.11 also provides an optional mechanism: request to send / clear to send (RTS/CTS) to improve CSMA/CA. Before diving into protocol details, we introduce a number of notations.

- Packet types of interest include RTS (request to send), CTS (clear to send), DATA (data) and ACK (acknowledgment). RTS and CTS packets are used for medium reservation. Besides carrying the address information of sender and receiver, these packets also include the duration of an upcoming transmission. A DATA packet carries application level information. An ACK packet informs a sender that its transmission was successfully delivered.

- Interframe space (IFS) is the time interval between frames. IEEE 802.11 defines four IFSs to provide priority levels: short IFS (SIFS), point coordinated function function IFS (PIFS), distributed control function IFS (DIFS) and extended IFS (EIFS) [36]. We do not discuss PIFS here, because point coordinated function is a centralized method and is not considered in this work. SIFS is the shortest among all four IFSs, and gives

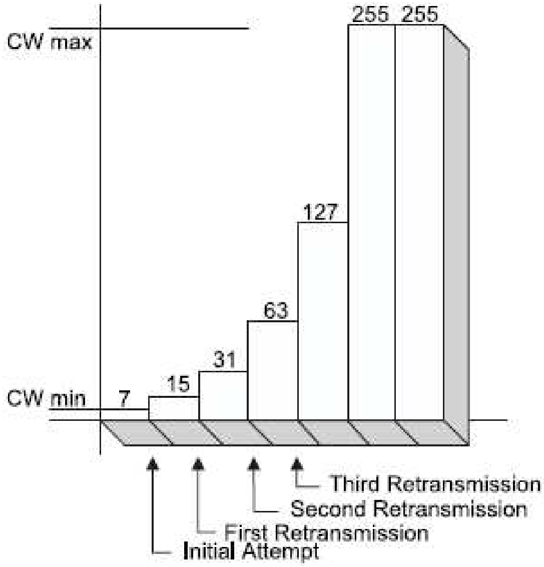

Figure 1.7: Binary Exponential Backoff [36]

the highest priority to the nodes that have seized the medium and need to keep it for the duration of the frame exchange sequence [36]. DIFS is longer than SIFS and used by wireless nodes to contend for medium access. To prevent the packet exchange sequence in progress from being interrupted, IEEE 802.11 forces a wireless node to wait for the medium to remain idle for the duration of a DIFS period. EIFS is the longest interval and used after a failed transmission. According to IEEE 802.11b, SIFS equals 10μs; DIFS equals the duration of SIFS plus two slots ($aSlotTime = 20\mu$s); EIFS equals the sum of SIFS, DIFS and the transmission time of eight ACK packets.

- Retry limit is the maximum number of attempts, for which a packet can be retransmitted. IEEE 802.11 provides two types of retry limit: short and long. The short retry limit is used when RTS/CTS transmission fails. The long retry limit is applied when DATA/ACK transmission fails.

- Contention window ($cwin$) specifies a range of time slots that a node can randomly select to back off ($CWIN\_MIN \leq cwin \leq CWIN\_MAX$). According to IEEE 802.11b, $CWIN\_MIN = 31 \times aSlotTime$ and $CWIN\_MAX = 1023 \times aSlotTime$. $cwin$ is initialized with $CWIN\_MIN$, and increased by twofold plus one during each retransmission as shown in Fig. 1.7 [36].

- Physical carrier sensing range specifies the distance within which a transmission can be detected, but may not be successfully decoded.

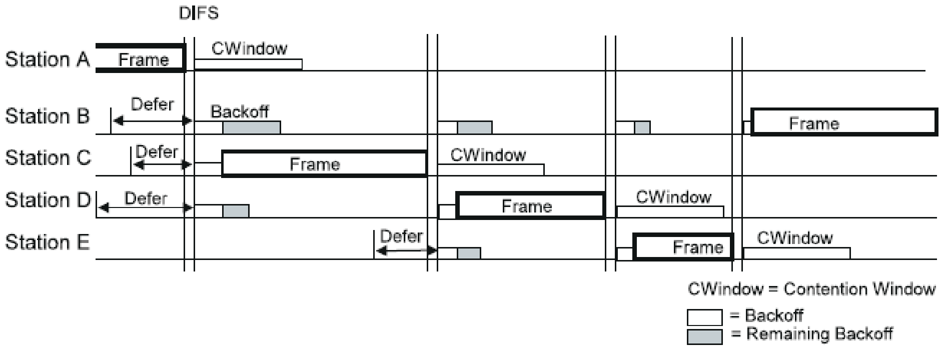

Figure 1.8: Backoff Procedure [36]

- Transmission range is the distance within which a transmission can be successfully decoded.

CSMA/CA

CSMA/CA is a contention-based method that allows multiple wireless devices to share medium access. We divide its access control into two phases: collision avoidance and collision resolution.

First, CSMA/CA uses physical carrier sensing and random access to avoid collisions. Physical carrier sensing is a hardware solution that a wireless node applies to determine the state of the medium. If a sender node detects an idle medium via physical carrier sensing, it transmits immediately. Otherwise, the sender node defers its transmission for the duration of busy medium plus a DIFS period, as shown in Fig. 1.6. While the medium is busy, multiple transmissions from different senders may be deferred and wait to start as soon as the medium becomes idle again. If all of them start at the same time, collisions will occur. To avoid collisions, random access is invoked after the deferral procedure. Fig. 1.8 shows the backoff procedure via an example of five competing transmissions. In this example, B, C and D defer their transmissions because A is transmitting. After A's transmission ends and the medium remains idle for a DIFS period, B, C and D select a random time slot within the contention window *cwin* and start backing off. As Fig. 1.8 illustrates, C has the shortest backoff period and becomes the first to transmit. Consequently, B and C suspend their backoff until C's transmission ends for a DIFS period. A recently deferred transmission from E also starts backing off at the same time when B and C resume their backoff. Because the remaining backoff period of D is shorter than that of B and E, D successfully seizes the medium to transmit. Similarly, B and E contend for the medium after D's transmission based on their remaining backoff periods. E transmits before B, and B finishes the last.

Second, CSMA/CA depends on acknowledgments and retransmissions to resolve a failed transmission. Although physical carrier sensing and random backoff helps to avoid collisions,

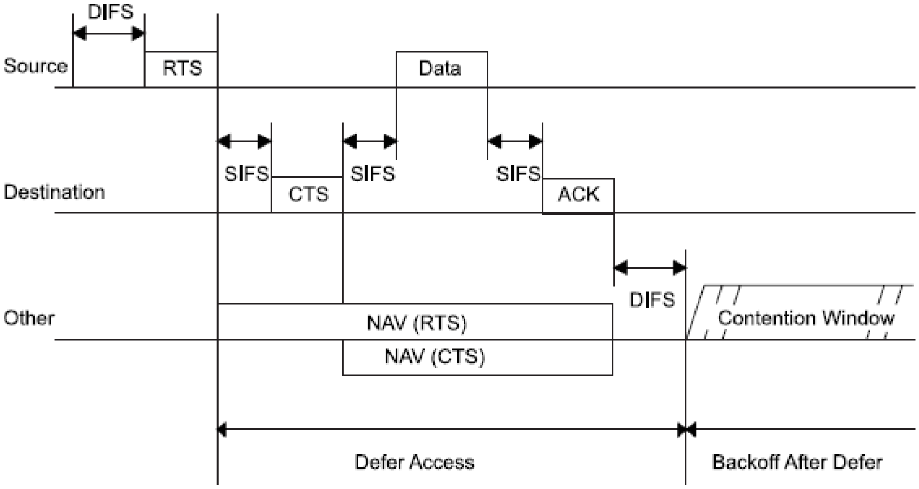

Figure 1.9: Network Allocation Vector Updates [36]

transmission failures may still occur. This is because interference varies in time, space, strength and frequency, and it is hard to accurately detect all interference scenarios all the time (e.g. hidden terminals). Therefore, CSMA/CA requires a receiver to acknowledge every successful transmission by sending an ACK packet to its sender. If no ACK is received after the duration of data transmission plus a SIFS period and propagation delay, the sender interprets it as a failed transmission. To recover from the failure, the sender increases the contention window according to $2 \times cwin + 1$, and invokes the backoff procedure. The same packet is retransmitted as soon as the backoff period ends and the medium keeps being idle without interruption for the duration of EIFS. A sender keeps retransmitting the same packet until it receives an expected ACK or the long retry limit is reached.

RTS/CTS

RTS/CTS is an optional mechanism that can be combined with CSMA/CA to alleviate hidden terminals. We describe the phases of collision avoidance and resolution, when RTS/CTS is in use.

First, RTS/CTS uses virtual carrier sensing, a software solution, to reserve the medium for an upcoming transmission. After physical carrier sensing determines that the medium is idle and this state remains for the duration of a DIFS period, a sender broadcasts an RTS packet as shown in Fig. 1.9. The RTS packet serves two purposes. For one, RTS silences the neighbors of its sender within the transmission range. If a neighbor node receives an RTS packet, it retrieves the length of the upcoming transmission from RTS's duration field (denoted *rts_duration*), and updates the local network allocation vector (NAV), if *rts_duration* is greater than the current NAV value. *rts_duration* includes the time to transmit a CTS, a DATA and an ACK packet plus three SIFSs as shown in Fig. 1.9. For

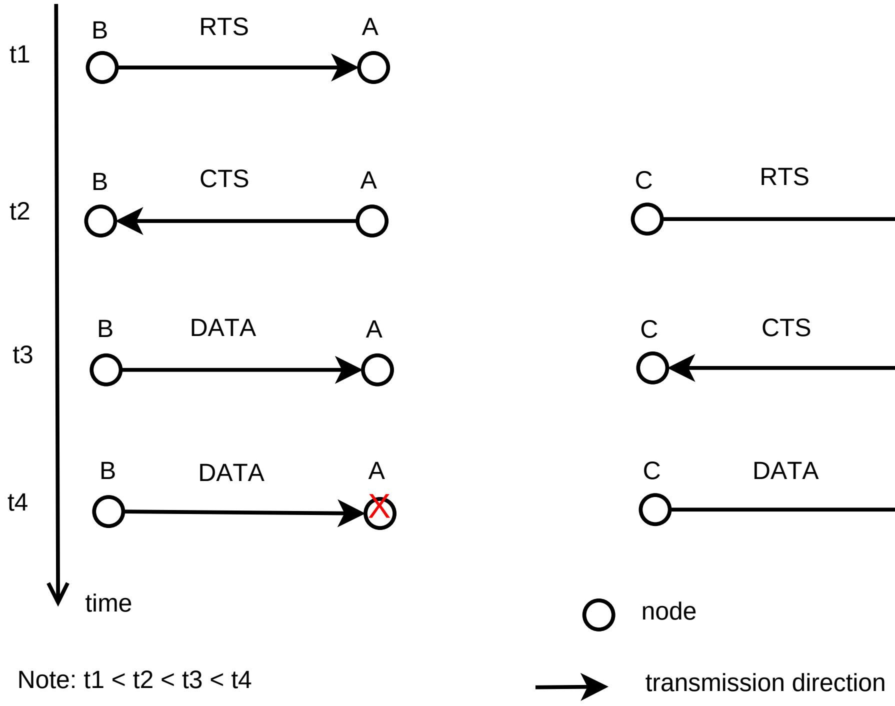

Figure 1.10: Extended Hidden Terminals

the second purpose, the RTS packet initiates a conversation with the receiver. If the receiver correctly decodes the RTS packet, it broadcasts a CTS packet after the period of a SIFS. Similar to RTS, CTS silences the neighbors of its receiver to reserve the medium at the receiver's side. The neighbor nodes update their NAV settings according to the duration *cts_duration* specified in the CTS packet, if $cts\_duration > NAV$. *cts_duration* includes the transmission period of a DATA and an ACK packet as well as inter-frame spaces, and therefore is shorter than *rts_duration*. After successfully receiving a CTS packet, the sender concludes that the medium is available and starts the data transmission.

Second, although RTS/CTS aims to alleviate hidden terminals, collisions may still happen. For example, as shown in Fig. 1.10, C is sending an RTS to D at time $t2$, while A is transmitting CTS to B. Because C is in the transmission mode, it cannot detect A's CTS. Likewise, A cannot detect C's RTS. Consequently, at time $t3$ A starts its data transmission to B, and D sends a CTS to C. These two transmissions overlap in time, which results in a collision at A at time $t4$. If a collision occurs during the RTS/CTS handshake, the same RTS packet is retransmitted until a CTS is successfully received, or the short retry limit is reached. Long retransmissions are invoked if DATA/ACK fails. For example, in the case that a sender did not receive an ACK packet within an expected period, the sender increases its long retry limit and re-initiates the RTS/CTS handshake. This procedure is repeated until an ACK packet is received, or the long retry limit is reached.

We use CSMA/CA instead of RTS/CTS for two reasons. First RTS/CTS wastes bandwidth on frequently exchanging control packets. Second, this method also causes more complicated interference scenarios due to collisions among control and data messages [88][65].

1.2.3 Traffic, Routing and Performance Evaluation

We evaluate the proposed frameworrk of resource management in both single-hop and multi-hop wireless ad hoc networks. To generate various network topologies in terms of the size and interference level, we carry out experiments via simulation in Network Simulator version 2 (ns2) [26]. To improve the simulation accuracy, we added a SINR-based collision model with aggregate interference [20] and a preamble detection function [92] to the IEEE 802.11 implementation in ns2.

In this work, we assume elastic traffic, which has loose time requirements and can adapt to the available resources [30]. All traffic flows are assumed to have the same bandwidth demand. Our resource management can be modified to support dynamic traffic demands. We give more details in the conclusion chapter. For multi-hop networks, we assume traffic is forwarded via the shortest paths that are pre-determined. We will explore how to jointly control routing and medium access in future study.

1.3 Thesis Contributions

Uncontrolled competition for finite resources significantly degrades network performance, and leads to poor user experience. Effective resource management makes wireless networks more tractable so that limited resources can be allocated to satisfy various user/system demand, and efficiently utilized to provide better network service/experience. In this thesis, we propose a novel framework: globalized-local (G-Local) resource management to effectively manage resources in wireless networks. More specifically, our contributions are:

- G-Local resource management offers G-Local optimization, a novel method that 1) supports different fairness criteria to allocate resource, 2) improves resource utilization efficiency by reducing both conflicts and waste; 3) minimizes the gap between the allocated and utilized resources caused by dynamic interference and imperfect scheduling; and 4) requires zero control message passing for single-hop applications.

- G-Local resource management provides adaptive multivariable control to improve control effectiveness. This method differentiates competition scenarios in time, space, frequency and intensity, and selects control variables that have major impact on these scenarios. By tuning the selected control variables, the adaptive multivariable control enables us to manipulate network behavior including: transmissions, collisions and idling. Guided by the G-Local optimization, adaptive multivariable control can steer a network to approach a desired state. Furthermore, the adaptive multivariable control can be applied to different scheduling algorithms. In this thesis, we demonstrate its effectiveness on controlling IEEE 802.11b.

- G-Local resource management supports multi-hop forwarding. We propose two unique methods to handle 1) the co-existence of intra-path and inter-path interference; and 2) correlated congestion and collisions in multi-hop wireless networks. These methods significantly improve multi-hop forwarding performance by coordinating the internal transmissions within a path, guiding different paths to compete for their fair share, and explicitly handling the mutual impact of congestion and collisions. Furthermore, the proposed algorithms are adaptable to time-varying traffic patterns; require simple computation; and minimize the requirement for message passing.

1.4 Chapter Summary

Competition for limited wireless resources among multiple users causes severe performance degradation. Effective mechanisms of resource management are needed to remedy the situation. In the rest of this thesis, Chapter 2 overviews the main tasks of wireless resource management, and reviews related work. Chapter 3 outlines the proposed G-Local resource management framework and formulates its theoretical foundation. In the following chapters, we present the G-Local resource management for wireless single-hop networks in Chapter 4, and describe the mechanisms proposed for wireless multi-hop networks in Chapter 5. Finally, Chapter 6 concludes this thesis by summarizing our contributions, describing potential applications, and outlining future directions.

Chapter 2

Wireless Resource Management

In a shared-resource system, resource competition without appropriate control leads to chaotic network performance. Wireless networks, in particular, present very challenging competition scenarios including the scarcity of bandwidth, temporal and spatial competition, and soft capacity. Without carefully managing the scarce bandwidth resource and handling complicated competition scenarios, it is impossible to provide any level of performance guarantee. An effective resource management mechanism aims to allocate wireless resources to satisfy a certain metric, and efficiently utilizes the allocated resources. In this chapter, we analyze these two important tasks of wireless resource management: resource allocation and utilization; and review existing methods in the literature.

2.1 Resource Allocation

Resource allocation determines the resources that should be assigned to each competing user. These decisions are made according to a certain allocation criteria. The major advantage of global allocation is coordinating users to achieve a certain system-wide optimality. However, due to the complex nature of wireless networks, existing methods adopt various simplified assumptions to derive mathematically tractable models, which leads to significant resource waste. Generally, resource allocation can be formulated as an optimization problem to either maximize network throughput, maximize network utility, or satisfy some fairness requirement. Please note that we present previous work by using their original symbols and notations.

2.1.1 Resource Allocation as Network Throughput Maximization

Aggregate network throughput is an important performance metric to evaluate the effectiveness of bandwidth consumption. There has been tremendous effort for modeling network throughput. Recently, new methods were proposed to maximize throughput in a distributed manner without message passing. We review previous work for both wireless local area networks (WLANs) and wireless ad hoc networks. The selected research in WLANs is related to our work in terms of using local CSMA/CA control to improve global network performance.

Throughput Maximization for Wireless Local Area Networks

In a wireless local area network, multiple wireless devices (e.g. laptops and various personal digital devices etc.) share the Internet access via an access point under the CSMA/CA mechanism of IEEE 802.11 standard [36].

Most existing work aims to maximize the throughput of a WLAN by adaptively tuning CSMA/CA parameters according to two models. The first model was proposed by Cali et al. in 1998 to characterize channel utilization. Their method relies on an estimation of the average contention window, which is derived by assuming each node attempts to use a backoff interval sampled from a geometric distribution [10]. In 2000, Bianchi proposed a multi-stage Markov Chain model to characterize the binary exponential backoff mechanism of CSMA/CA. Both models assume perfect channel conditions and no hidden/exposed terminals.

Cali's Capacity Model Cali et al.[10] aim to maximize the throughput of IEEE 802.11 WLANs. They derive a closed-form expression of network throughput in terms of the channel utilization ρ based on three assumptions: 1) the backoff window is geometrically distributed; 2) every node transmits with the same probability p; and 3) the channel condition is perfect and there are no hidden terminals. To achieve the theoretical capacity limit, they proposed an iterative method to adjust the contention window, CW.

The channel utilization is modeled as $\rho = \frac{\bar{m}}{t_v}$, where $\bar{m}$ is the average transmission time of a packet, and t_v is the virtual transmission time (see Eq. 2.1). The virtual transmission period is the average time that a channel is occupied in transmitting a message. t_v consists of the average contention window (denoted $E[CW]$), and the actual transmission time (denoted $E[S]$). According to the basic access method of IEEE 802.11 CSMA/CA, $E[S]$ (expressed in Eq. 2.2) is the sum of propagation time (denoted τ), data transmission time (denoted m), acknowledgment transmission time (denoted ACK), and short/long inter-frame space (denoted $SIFS/DIFS$). When multiple stations are active during a virtual transmission period, $E[CW]$ consists of collision intervals ($E[N_c] \cdot E[Coll]$), idle intervals ($(E[N_c]+1) \cdot E[Idle]$), and one successful transmission ($E[S]$). $E[N_c]$, $E[Coll]$ and $E[Idle]$ are given in Eq. 2.3, where M is the network size.

$$\begin{aligned} t_v =& E[CW] + E[S] \\ =& E[N_c] \cdot E[Coll] + \tau + DIFS + E[Idle] \cdot (E[N_c] + 1) + E[S] \end{aligned} \tag{2.1}$$

$$E[S] = 2 \cdot \tau + m + SIFS + ACK + DIFS \tag{2.2}$$

$$\begin{aligned} E\left[N_c\right] =& \frac{1-(1-p)^M}{Mp\cdot(1-p)^{M-1}}-1 \\ E\left[Coll\right] =& \frac{t_{slot}}{1-(1-p)^M-M\cdot p\cdot(1-p)^{M-1}} \\ E\left[Idle\right] =& \frac{(1-p)^M}{1-(1-p)^M}\cdot t_{slot} \end{aligned} \tag{2.3}$$

Adaptive Contention Window Adjustment By solving Eq.2.1, Cali et al.[10] derive the optimal value of the transmission probability (denoted p_{min}). Based on the relationship between p_{min} and the contention window (denoted CW) given in Eq. 2.4, CW is updated according to $curr\_cw = \alpha_2 \cdot curr\_cw + (1-\alpha_2)\cdot(2/p_{min}-1)$. The authors conclude via simulation that the enhanced protocol is very close to the maximum theoretical efficiency.

$$E\left[CW\right] = \frac{2}{p_{min}}-1 \tag{2.4}$$

In spite of their good simulation results, the assumption of perfect channel condition and no hidden terminals significantly limits the practical usage of this method. Further, they assume all nodes transmit with the same probability, which hinders its application when traffic differentiation is required.

Asymptotically Optimal Backoff Mechanism Based on the same capacity model that Cali et al. developed [10], Bianchi et al. [5] proposed a simple method to achieve the theoretical limits on channel utilization via asymptotically adjusting the backoff mechanism in CSMA.

The proposed method uses a simple metric S_U to evaluate the current contention level in a network via observing the number of busy and available slots periodically. S_U is estimated as $S\_U = \frac{Num\_Busy\_Slots}{Num\_Available\_Slots}$. This metric indicates that the higher S_U is, the more likely collisions will happen. To adapt the transmission attempt of a node to the contention level, this work introduces a new parameter P_T, the probability of transmission. P_T depends on S_U according to a heuristic formula $P\_T(S\_U, N\_A) = 1 - S\_U^{N\_A}$, where N_A is the number of attempts already performed by the station for the transmission of the current frame. When traffic is intense, this simple formula gives a node with multiple transmission failures a higher priority.

To maximize channel utilization, the optimal contention level (denoted opt_S_U) is approximated according to $opt\_S\_U = M \cdot p_{min}$, where M is the network size, and p_{min} is the optimal p-persistence probability. Correspondingly, P_T is tuned to achieve opt_S_U via $P\_T(opt\_S\_U, S\_U, N\_A) = 1 - min(1, \frac{S\_U}{opt\_S\_U})^{N\_A}$. The authors also provide a customized transmission policy to support traffic differentiation: $P\_T(opt\_S\_U, S\_U, N\_A) = 1 - min(1, \frac{S\_U}{opt\_S\_U})^{N\_A \cdot pr_{level}}$, where pr_{level} is the priority level.

The advantages of this method include: 1) no requirement of estimating the network size M; 2) asymptotically achieving the theoretical capacity limit; 3) and supporting traffic differentiation. However, it shares the same disadvantages of Cali's work [10] in terms of not considering channel condition and hidden terminals. Hidden terminals directly mislead the estimation of S_U.

Bianchi's Model The second capacity model was proposed by Bianchi [5] to characterize the exponential backoff mechanism in IEEE 802.11 CSMA/CA. A multi-stage Markov Chain model was proposed for backoff window size with up to m retransmissions. Based on this model, the authors obtain the stationary probability τ that a station transmits a packet in a randomly chosen slot time as shown in Eq. 2.5. This probability depends on the collision probability p, contention window W, and the number of retransmissions m. When $m = 0$, Eq. 2.5 becomes $\tau = \frac{2}{W+1}$, which is independent of the conditional collision probability p.

$$\tau = \frac{2}{1 + W + pW \cdot (1 - (2p)^m)} \tag{2.5}$$

The throughput model is expressed in Eq. 2.6. S is the normalized system throughput, defined as the fraction of time the channel is used to successfully transmit payload bits. In this equation, T_s is the average transmission time; T_c is the average time of a collision; $E[P]$ is the average payload size; and δ is the size of an empty slot. Furthermore, their values are constant. P_{tr} (expressed in Eq. 2.7) is the transmission probability, while P_s (Eq. 2.8) denotes the probability of a successful transmission.

$$S = \frac{E[P]}{T_s - T_c + \frac{\delta(1-P_{tr})/P_{tr} + T_c}{P_s}} \tag{2.6}$$

$$P_{tr} = 1 - (1 - \tau)^n \tag{2.7}$$

$$P_s = \frac{n \cdot \tau \cdot (1 - \tau)^{n-1}}{1 - (1 - \tau)^n} \tag{2.8}$$

After some mathematical transformation, the author derives the solution to maximize S of Eq. 2.6 in Eq. 2.9, where $T_c^\star = \frac{T_c}{\delta}$. According to this equation, τ is only dependent on the network size n, because $T_c^\star$ is a constant determined by the physical layer and access mechanism.

$$\tau = \frac{1}{n\sqrt{T_c^\star/2}} \tag{2.9}$$

Adaptive Contention Window Following the second model, Bianchi et al. proposed an adaptive algorithm that adjusts the contention window to fulfill Eq. 2.9. By setting $m = 0$, the optimal contention window depends only on the number of competing transmissions according to $W_{opt} = n\sqrt(2T_c^\star)$. To estimate n, this work uses Eq. 2.10, where $c(B)$ is the number of slots observed busy due to other stations transmitting over a period of B slots. For a smooth control, the contention window is adjusted according to Eq. 2.11.

$$n \approx 1 + \frac{E\left[c(B)\right] \cdot (W+1)}{2B} \tag{2.10}$$

$$\begin{aligned} W =& s(n) \cdot \sqrt{2T^{\star}} \cdot n, \ where \\ s(n) =& 1 + \frac{h}{\sqrt{n}} \end{aligned} \tag{2.11}$$

Although their performance evaluation claims close-to-optimal performance, hidden terminals are once again not considered. Hidden terminals and complex channel conditions make accurate estimation of n hard to achieve.

Idle Sense Heusse et. al proposed an optimal access method for high throughput and fairness, called Idle Sense [35]. This method aims to find the optimal attempt probability, P_e^{out} that maximizes the throughput (expressed in Eq. 2.12). This objective can be achieved by reducing the time wasted on collisions and in contention, and increasing the time spent in transmissions. Therefore, maximizing throughput $X(P_e)$ is the same as minimizing the cost $Cost(P_e)$.

$$X(P_e) = \frac{P_t \cdot s_d}{P_t \cdot T_t + P_c \cdot T_c + P_i \cdot T_{SLOT}} \tag{2.12}$$

$$Cost(P_e) = \frac{\frac{T_c}{T_{SLOT}} \cdot P_c + P_i}{P_t} \tag{2.13}$$

The optimal attempt probability (denoted P_e^{opt}) is derived by setting the first derivative of Eq. 2.13 to zero. The optimal value of consecutive slots is $\bar{n}_{i\infty}^{opt} = \frac{e^{-\varsigma}}{1-e^{-\varsigma}}$, where $\varsigma = N \cdot P_e^{out}$. To approach the theoretical capacity limit, their method lets each host estimate the number of consecutive idle slots between two transmission attempts ($\bar{n}_i$), and uses it to compute its contention window CW. By adjusting CW, a host makes $\bar{n}_i$ converge to a common value $\bar{n}_i^{target}$. To avoid estimating the number of competing hosts, they fix $\bar{n}_i^{target}$ at a value that is close to optimal value $\bar{n}_{i\,\inf}^{opt}$. Besides achieving the maximum throughput, Idle Sense also improves short-term fairness by enforcing all nodes to converge to the same $\bar{n}_i^{target}$ via the principle of additive increase and multiplicative decrease. The performance study shows that Idle Sense achieves similar throughput performance as the asymptotically optimal backoff method with better short-term fairness.

Maximizing Throughput-Fairness Luis et al. [67] proposed a new method to maximize network throughput by finding the optimal access probability $\tau^{\star}$, which is a function of the network size n. To improve short-term fairness, all nodes use the same $\tau^{\star}$.

$$\hat{n} = argmin_{n \in N} \sqrt{(\tilde{p_a} - (1-\tau_i)^{n-1})^2} \tag{2.14}$$

n is estimated according to Eq. 2.14, where $\tilde{p_a}$ is the probability of finding an idle slot by observing channel states during the contention period. To obtain a slowly-varying measurement of $\tilde{p_a}$ over B slots, an autoregressive moving average filter is used in Eq. 2.15. If slot i is sensed idle, $Slot_i = 0$; otherwise, $Slot_i = 1$. Furthermore, this work uses a proportional-integral controller to adapt the individual medium access probability τ_{k+1} according to the reference $\tilde{\tau}$ and the filter dynamics.

$$\tilde{p_a}(k+1) = \alpha \cdot \tilde{p_a}(k) + \frac{1-\alpha}{B} \cdot \sum_{i=1}^{B} Slot_i \tag{2.15}$$

Control-Theoretical Method Patras et al. [77] proposed a control-theory method that tunes all nodes in a network to achieve the same desired collision probability p_{col}. They developed a distributed algorithm of adaptive control to adjust CW_{min} of each station with the goal of driving the WLAN to the optimal point of operation [5]. Without explicitly modeling $CW_{min,i}$ as a function of e_i, each station runs an independent proportional-integral controller, which takes as input the error signal e_i (see Eq. 2.16) and gives as output the $CW_{min,i}$ of the system.

$$\begin{aligned} e_i &= e_{collision,i} + e_{fairness,i} \\ e_{collision,i} &= p_{others,i} - p_{col} \\ e_{fairness,i} &= p_{others,i} - p_{own,i} \end{aligned} \tag{2.16}$$

In summary, all the methods above ignore the impact of hidden terminals in wireless local area networks. As demonstrated by A. Zahedi et al. and F. Liu et al., hidden terminals can degrade network throughput up to 50% in WLANs [66][107]. Furthermore, these methods assume all nodes transmit with the same transmission probability, which is not suitable for traffic differentiation. Thirdly, the assumption of independent and constant collision probabilities is also unrealistic. If a node does not wait long enough upon collisions, the probability of repeated collisions increases. Additionally, the dynamic occurrence of hidden terminals causes collision probabilities to change over time and space. Therefore, these methods can not achieve optimal throughput in practice. At last, these methods are designed for single-hop wireless local area networks, and do not address special issues of multi-hop forwarding, such as co-existing intra-path and inter-path interference.

Throughput Maximization for Wireless Ad Hoc Networks

Compared with WLANs, wireless ad hoc networks do not need a shared access point. Wireless nodes can talk to each other either via single-hop communication or multi-hop forwarding. Two throughput models are widely used to maximize the throughput of a wireless ad hoc network. The first model was proposed by Tassiulas and Ephremides in 1992, and proves that queue-length-based maximum weight scheduling is throughput optimal [95]. The second model was proposed by Boorstyn et al. to calculate the throughput of random access in multi-hop wireless networks based on CSMA [6].

Tassiulas's Model Tassiulas et al.[95] model a wireless network where links are dynamically activated at each time slot according to the queue length. The optimality of throughput is defined in the sense that any arrival rate within the capacity region of a network can be accommodated without causing the network queues to grow infinitely. The authors prove that a scheduling algorithm is throughput-optimal if it can find a set of maximum weight independent sets where the weight is a function of the total queue length within an independent set. An independent set contains a number of non-conflicting links. However, the problem of finding maximum weight schedules is NP-hard for general networks [95]. To reduce computational complexity, many heuristic algorithms have been proposed to achieve maximal weight scheduling [13] [103]. These methods can only achieve a fraction of the maximum throughput [49].

Boorstyn's Capacity Model Instead of using scheduling-based methods, another approach aims to achieve throughput-optimality via random access. Boorstyn et al.[6] developed a continuous time Markov Chain to characterize state transitions among different independent sets, and model network throughput (or service rate) in a product form. Their model assumes perfect capture and zero propagation delay that theoretically eliminates hidden terminals. We summarize recent efforts on deriving distributed solutions without message passing to maximize network throughput based on this model.

Adaptive CSMA Without Collisions Li. et al.proposed a method that adapts CSMA to achieve optimal throughput in wireless networks [48]. Based on Boorstyn's CSMA model and the product-form throughput (or service rate), they formulate the problem of maximizing throughput as a maximum likelihood function Eq. 2.17. In Eq. 2.17, p_i (also denoted $p(x^i; r)$) is the stationary distribution of any feasible state x^i of index i in the Markov Chain. The expression for p_i is given in Eq. 2.18. $\bar{p_i}$ is the measured value of p_i obtained from observations. A back-pressure method Eq. 2.19 adjusts the transmission aggressiveness r_i and solves Eq. 2.17. Specifically, link k adjusts the rate r_k based on local information: average arrival rate λ'_k and service rate s'_k. Li proved that this method can theoretically achieve the maximal throughput via slow adaptation by using a small step size α.

$$F(r) := \sum_i \bar{p_i} \cdot \log(p_i(r)) \tag{2.17}$$

$$p(x^i; r) = \frac{exp(\sum_{k=1}^{K} x_k^i \cdot r_k)}{\sum_i exp(\sum_{k=1}^{K} x_k^i \cdot r_k)} \tag{2.18}$$

$$r_k(t+1) = [r_k(t) + \alpha \cdot (\lambda_k'(t) - s_k'(t))]_+ \tag{2.19}$$

Adaptive CSMA with Collisions In a consecutive work, Li. et al.remove the assumption of perfect CSMA (zero collision). By introducing probe messages to enable channel reservation, they assume collisions only happen between probe messages and last for a short period [47]. A new throughput model is derived after re-designing the Markov Chain model. Throughput maximization is also formulated as a maximum likelihood function to accommodate all feasible arrival rates. Instead of adjusting the transmission aggressiveness r_k via the contention window, Li chose to adjust the mean payload length according to $T_k^p := T_0 \cdot exp(r_k)$.

Although this method takes collisions into account, the assumption that collisions only occur among probe messages is impractical. As analyzed in [88], when RTS/CTS mode is used, collisions may happen among RTS/CTS/DATA/ACK in a dynamic manner due to hidden terminals. In addition, exposed terminals also occur and lead to inefficient channel utilization. Furthermore, adopting dynamic transmission lengths sacrifices short-term fairness and increases delay. Our performance evaluation in Chapter 5 shows that this method leads to high collision rates and ultimately degrades network throughput.

Q-CSMA Srikant et al. proposed another method to achieve the same objective as Li's adaptive CSMA [48]. They developed a synchronized CSMA scheduling that allows multiple links to update their states in a single time slot [76]. Based on this basic scheduling algorithm, the authors model CSMA as a discrete time Markov Chain, and find the stationary distribution of any feasible state. Q-CSMA is throughput-optimal by choosing an activation probability $p_i = \frac{e^{w_i(t)}}{e^{w_i(t)}+1}$. The weight w_i is a function of link i's queue length. To reduce hidden terminals, Q-CSMA uses a synchronized RTD/CTD (request to deliver/clear to deliver) exchange. The algorithm is as follows.

Algorithm: Q-CSMA (at Link i in Time Slot t)

- Link i selects a random backoff time T_i uniformly in $[0, W-1]$ and waits for T_i control mini-slots

- If link i hears an RTD or CTD message from a link in $C(i)$ before the (T_{i+1})-th control mini-slot, i will not be included in $m(t)$ and will not transmit an RTD/CTD message anymore. Link i will set $x_i(t) = x_i(t-1)$

- If link i does not hear an RTD/CTD message from any link in $C(i)$ before the (T_i+1)-th control mini-slot.
    - If there is a collision, link i will not be included in $m(t)$ and will set $x_i(t) = x_i(t-1)$.
    - If there is no collision, link i will be included in $m(t)$ and decide its state as follows:
        - $*$ if no links in $C(i)$ were active in the previous data slot
          $x_i(t) = 1$ with probability p_i, $0 \leq p_i \leq 1$;
          $x_i(t) = 0$ with probability $\tilde{p_i} = 1 - p_i$.
        - $*$ else
          $x_i(t) = 0$
- If $x_i(t) = 1$, link i will transmit a packet in the data slot

Compared with Li's adaptive CSMA [48][47], Q-CSMA explicitly handles exposed/hidden terminals via a synchronized RTD/CTD mechanism. However, this method requires perfect synchronization, and incurs constant overhead. A more extensive performance study is needed to evaluate how well the proposed method can reduce hidden/exposed terminals.

2.1.2 Resource Allocation as Network Utility Maximization

Another trend is to formulate the resource allocation problem as network utility maximization [54]. The basic form of network utility maximization is given in Eq. 2.20. U is a utility function that can be used to characterize user perception behavior models, application traffic elasticity, efficiency of resource allocation and fairness among competing users [15].

$$\begin{gathered} \sum_s U_s(x_s) \\ s.t.\ Rx \geq c \end{gathered} \tag{2.20}$$

We are interested in solving Eq. 2.20 without message passing, and therefore describe two representative methods: utility-optimal random access and utility-optimal CSMA.

Utility-Optimal Random Access Mohsenian et al.designed a distributed contention-based MAC algorithm to maximize network utility at the link layer without explicit message passing among users [62]. Their method considers a single-hop wireless ad hoc network with N sets of wireless links, where a receiver node can overhear all other transmissions. As a result, a transmission is successful if it is the only transmission in the current time slot.

The network utility maximization problem is formulated in Eq. 2.21, where U_i is a utility function, r_i is the average data rate of link i as shown in Eq. 2.22, and p is the vector of all users' transmission probabilities. The task is to find the value p that solves Eq. 2.21.

$$\max_{p \in P} \sum_{i \in N} U_i(r_i(p)) \tag{2.21}$$

$$r_i(p) = \gamma_i p_i \prod_{j \in N-i} 1 - p_j \tag{2.22}$$

To solve Eq. 2.21 locally, the authors transform this global optimization problem to a local network utility maximization problem Eq. 2.23, where $P_{-i} = p_j \ , \forall j \in N - i$. Please not that the authors use $-i$ to denote the set of nodes except i. The optimal p_i is derived by estimating two values: 1) $n_{i,j}^{decd}$ the number of time slots between any two consecutive successful transmissions of another user j in the network; and 2) n_i^{idle} the number of non-idle time slots that user i observes between any two consecutive idle time slots. They estimate both locally.

$$\max_{p_i \in P_i} \sum_{j \in N} U_j(p, P_{-i}) \tag{2.23}$$

As a first step toward utility optimal random access without message passing, this algorithm shows good theoretical performance. However, its assumption that a node can estimate the network topology via signaling is unrealistic. Especially in ad hoc wireless networks, interference scenarios are dynamic over time and space. Second, this method requires a user i to estimate $n_{i,j}^{decd}$ of every other user j in the network iteratively. Third, this method ignores dynamic channel quality and hidden/exposed terminals, and therefore cannot achieve throughput optimality in reality.

Utility-Optimal CSMA Li's previous work of adaptive CSMA [48] is proved to achieve throughput optimality theoretically with a perfect CSMA. By perfect CSMA, they assume CSMA eliminates collisions. Furthermore, they also assumed that arrival rates are within the capacity region and can be accommodated. In practice, this is hard to determine because the capacity region of a wireless ad hoc network is often complex and non-convex [22]. Furthermore, Li's adaptive CSMA cannot handle saturated traffic. To overcome these limitations, Li proposed a joint rate and flow control by combining throughput maximization with network utility maximization. This new problem is formulated in Eq. 2.24, where $-\sum_i u_i \cdot \log(u_i)$ is the dual problem of maximizing throughput, and $\sum\limits_{m=1}^{M} v_m(f_m)$ is the total utility. Specifically, u_i is the stationary probability of state i in the CSMA Markov Chain given the transmission aggressiveness of all links; f_m is the rate of flow m; and v_m is the utility function of flow m.

$$
\begin{aligned}
\max_{u,s,f} \; & -\sum_i u_i \cdot \log(u_i) + \sum_{m=1}^{M} v_m(f_m) \\
s.t. \;\; & s_{km} \geq 0, \; \forall m, k : a_{mk} = 1 \\
& s_{km} \geq s_{up(k,m)}, \forall m, k : a_{mk} = 1, k \neq \delta(m) \\
& \sum_i u_i \cdot x_k^i = \sum_{m:a_{mk}=1} s_{km}, \forall k \\
& u_i \geq 0, \sum_i u_i = 1.
\end{aligned}
\tag{2.24}
$$

After applying the Lagrange transformation and some mathematical manipulation, Li decomposed the problem of joint throughput and utility maximization into two sub-problems: 1) source rate control, and 2) local scheduling. The algorithm is described below.

Joint Scheduling and Rate Control initially assume that all queues are empty, and set $q_{km} = 0$, $\forall k, m$. Then iterate:

- Link k transmits the head-of-line packet from a flow with the maximal back-pressure $z_k = \max_{m:a_{mk}=1}(q_{km} - q_{down(k,m),m})$ when it gets the opportunity to transmit.

- Link k lets $r_k = z_k$ in the CSMA operation.

- Rate control: For all flow m, if link k is its source link, then the transmitter of link k let $f_m = argmax_{f'_m} \beta \cdot v_m(f'_m) - q_{km} \cdot f'_m$.

- The dual variables q_{km} (maintained by the transmitter of each link) are updated by a sub-gradient algorithm: $q_{km} \leftarrow \left[q_{km} + \alpha(s_{up(k,m),m} - s_{km})\right]_+$ if $k \neq \delta(m)$; and $q_{km} \leftarrow [q_{km} + \alpha(f_m - s_{km})]_+$ if $k = \delta(m)$.

In this algorithm, q_{km} is the virtual queue of link k belonging to flow m; s_{km} is the service rate of flow m at link k; and $s_{up(k,m),m}$ is the service rate from the upstream link $k-1$, as well as the receiving data rate at link k of flow m.

Lee et al.[62] extended Li's method by providing a generalized update policy of the virtual queue q in Eq. 2.25, where U is a utility function, W is the weight function, V controls the accuracy of the algorithm, and b ensures convergence. With this method, CSMA is utility optimal, as long as the product of λ_l (the transmission intensities) and μ_l (the transmission durations) is set to $exp(W(q_l\,[t+1]))$.

$$
q_l\,[t] = \left[q_l\,[t] + \frac{b\,[t]}{W'(q_l\,[t])} \cdot (U'^{-1}(\frac{W(q_l\,[t])}{V}) - S_l\,[t]) \right]_{q^{min}}^{q^{max}}
\tag{2.25}
$$

2.1.3 Resource Allocation and Fairness

Fairness is another criteria of resource allocation besides optimizing a certain network metric. Although every system should guarantee a certain level of fairness, it is a complicated issue due to the variety of user/system requirements, the co-existence of elastic and inelastic, short-term and long-term traffic, as well as dynamic interference and channel qualities. This section analyzes fairness from two viewpoints: end-to-end and local.

End-to-End Fairness

At the network level, a user is defined as a data flow between a pair of source and destination nodes via multiple hops. Compared with wired networks, users in wireless networks have a number of unique properties worth being considered to define fairness criteria.

User Properties Network-layer users can be different in many ways. These differences impose challenges on fairness criteria definition.

- **Differences of traffic load**
  Different flows may have different requirements of traffic load. These requirements can be specified by relative concepts: high, medium and low, or real numbers. If a traffic load is too high, it may over-inject the network and cause significant congestion. How much traffic can be admitted for each user? Should the load difference be considered for channel access? These questions need to be considered when defining fairness criteria

- **Differences of multi-hop path properties**
  Different flows may go through different paths in terms of length and route. Depending on the distance between the source and destination, network topology, connectivity and routing strategies, path properties may vary a lot. Flows over long paths tend to experience poor performance (e.g. longer delay, lower throughput and higher packet loss).

- **Differences of intra- and inter-flow contention**
  Unlike data flows in wired networks, wireless flows experience interference within flows, as well as between flows. Intra-flow interference describes the interference between nodes that are upstream and downstream. Inter-flow interference happens when parts of two flows share common nodes or contention regions. In addition, contention is dependent on location [75], the levels of intra- and inter-flow contention varies with different flows.

Fairness Criteria With the aforementioned differences, how should fairness be defined in wireless multi-hop networks? There are a few options for reference.

- **Throughput fairness vs. temporal fairness**
  Equal share is a basic type of fairness. There are two forms of fairness based on the

equal-share criteria. The first criterion is throughput fairness that requires that all flows achieve the same throughput [8]. This approach may not be appropriate for two reasons: 1) it is not necessary that all flows have the same throughput, due to the differences of traffic load and type; 2) throughput fairness is achieved by sacrificing well-performing flows, simply because flows through channels with poor transmission quality experience low throughput. The second form of equal share demands that all users have equal share of channel time [8]. This form is called temporal fairness. Compared with throughput fairness, temporal fairness achieves more efficient resource utilization. Flows with better performance will not be punished for flows with poor performance.

- **Proportional fairness vs. max-min fairness**

  Due to user differences, fairness does not necessarily mean that all users should be treated equally. Max-min fairness offers an optimal solution when bottlenecks exist. Instead of equally sharing resources, max-min fairness [96] guarantees that users with lower requirements (called weaker users) will not be degraded because of higher requirements of other users (called stronger users). Stronger users can only increase their share of resources without influencing weaker users. This criteria acknowledges user differences as well as avoiding the starvation of weaker users. However, the disadvantage of max-min fairness is that it gives absolute priority to weaker users [54], even if decreasing a very small amount of weaker users' resource share can lead to significant gain of stronger users. In comparison, proportional fairness [54] allows weaker users to be sacrificed if it benefits the whole network. Specifically, a vector of rates $x^\star$ is proportionally fair if it is within the capacity region, and if for any other feasible vector x, the aggregate of proportional change $\sum_i \frac{x_i - x_i^\star}{x_i^\star} \leq 0$ is negative [54]. This is achieved by assigning each user a utility function $\log(x_i)$ and solving the corresponding problem of network utility maximization. With proportional fairness, the more resources a user is assigned, the higher utility the user achieves. But the speed of utility gain decreases with the increase of resource share.

- **α-fairness**

  α-fairness generalizes the max-min fairness and proportional fairness [71]. A vector of rates $x^\star$ is α-fair if it is within the capacity region, and for any other feasible vector x, $\sum_i \frac{x_i - x_i^\star}{{x_i^\star}^\alpha} \leq 0$. Correspondingly, the utility function for α-fairness is $(1-\alpha)^{-1} \cdot x^{1-\alpha}$. When α equals 1, α-fairness is proportionally fair. When $\alpha \to \infty$, α-fairness is max-min fair.

In summary, the fairness requirements of network-level users vary from system to system, and application to application. Therefore, our work supports different fairness criteria, instead of specifying a certain type.

MAC-Layer Fairness

Situated beneath the network layer, the MAC layer manages when a wireless node is allowed to access wireless medium. At the MAC layer, users are defined locally. Depending on the designing objectives, a user at the MAC layer can be defined as a node, a link or a flow segment. A flow segment is the part of traffic that a flow imposes on each link along its route.

User Properties MAC-layer users may also differ from each other. These differences are reflected from the following properties.

- **Difference in traffic loads**
  Because flows may have different traffic loads, carried loads of individual nodes or of flow segments between a transmission pair may differ.
- **Difference in contention levels**
  Depending on the location of a node or a transmission pair and the number of competing pairs around it, different nodes or flow segments may experience different levels of contention. When competing users experience different contention at the MAC layer, one user may gain more chances of channel access.
- **Difference in distance**
  The third difference is the distance between a transmission pair. Because the power of received signals is related to distance, a transmission pair with closely-located sender and receiver may bias neighbor transmission pairs that are more separated in distance.
- **Difference in parameter values** Due to hardware differences or parameter tuning, different nodes may have different physical layer configurations. For example, parameters including the transmit power, carrier sensing range and contention window can be adjusted in IEEE 802.11 devices.

Fairness Criteria Besides considering the differences listed above, the time-varying property of channel condition and contention scenarios should also be taken into account.

- **Per-node fairness vs. per-flow fairness**
  Because users can be viewed from different angles (such as nodes, links or flow segments), the same fairness criteria may result in different allocation policies. By per-node fairness, we mean the competition among contending nodes. Per-node fairness can be defined to support different concerns, such as equal share, traffic load and

number of flows. Per-flow fairness is defined at the MAC layer where a flow segment is viewed as a user. It requires each flow to have an individual queue at the node and fair queueing is required. In wired networks, nodes do not interfere with each other, so only per-flow fairness is required. This is not the case in wireless networks.

- **Short-term fairness vs. long-term fairness**
  Depending on the lifespan of a network flow, both short-term and long-term fairness may be required. Compared with long-term fairness, short-term fairness requires smaller granularity of control, e.g. packet-based or time instant [57].

In summary, fairness requirements at the MAC layer also exhibit huge variety. Besides supporting such variety, user differences, time-varying quality of wireless channels as well as the balance between resource utilization and fairness should be considered.

2.2 Resource Utilization

Another mission of resource management is to efficiently utilize resources. Especially when resources are not sufficient to satisfy user demand, it is important to increase the effective consumption of available resources and decrease the waste of limited resources. To specify this requirement in wireless networks, available channel access right and transmission bandwidth can also be viewed as transmission opportunities. Transmission opportunities are consumed by successful transmissions and wasted by failed transmissions. Therefore, the requirement of efficient resource utilization in wireless networks is to increase the number of successful transmissions and decrease the number of transmission failures.

Because most of existing resource utilization methods aim to derive local methods that are effective in reducing resource waste caused by certain types of interference. These methods are distributed in nature, simple in computation and fast in response. However, due to the lack of global picture, these methods may lead to user conflicts and jeopardize system-level objectives.

Due to the properties of mutual exclusiveness and geographical distribution, wireless resources can be divided in time and space, and shared among multiple users. Seeking transmission opportunities over time and space helps to improve resource consumption.

2.2.1 Temporal Reuse

The mutual exclusiveness property of wireless-channel access restricts each channel to allowing one transmission at a time, and the nature of the shared medium requires multiple users to transmit over a single channel in turn. The problem of improving temporal reuse of a wireless channel is to increase the number of transmissions, as well as to reduce collisions caused by temporally-correlated interference. Temporally-correlated interference is caused by multiple transmissions overlapping in time and contention regions. Depending on how competing transmissions overlap in time, interference can be caused by earlier, simultaneous and future transmissions [68]. Previous solutions aim to improve the temporal reuse of wireless channels from two perspectives: collision avoidance and resolution.

Collision Avoidance

Collisions are one of the major hurdles for efficient utilization of wireless channels. Due to its close relation with interference, we differentiate collisions caused by interference from earlier, simultaneous and future transmissions. Corresponding avoidance strategies are described in the following.

Avoiding Collisions Caused By Earlier Transmissions As defined in Section 2.2.1, earlier-transmission-caused interference occurs when a node initiates a transmission and there is at least one earlier transmission within its contention region. To avoid collisions caused by interference from earlier transmissions, a node needs to find a way to detect interfering transmissions that have started earlier. This procedure is called interference detection. Based on the detection result, a node may either start or defer its transmission according to certain strategies, which are classified as deferral policies of channel access.

Interference Detection

Earlier transmissions can be detected by both physical and virtual carrier sensing. Physical carrier sensing is implemented at the physical layer and supported in hardware. It helps a node to determine whether the wireless medium is busy. In comparison, virtual carrier sensing is different from physical carrier sensing in two ways. First, it is a soft approach, which requires message exchanges between a sender and receiver. Second, virtual carrier sensing considers both the medium state at the sender and receiver side. It is only when both sender and receiver sides are idle, that a transmission can be initiated. A typical example is the RTS/CTS protocol applied in IEEE 802.11 MAC [36]. Compared with physical carrier sensing, virtual carrier sensing improves the accuracy of interference detection. However, the exchange of RTS and CTS message causes higher overhead. In addition, collisions may happen among control packets and waste transmission opportunities.

Channel Access Deferral

The policies of deferring channel access control when a node starts transmission and how long to defer before its next retry, after interference detection. Existing policies include 1-persistent, non-persistent, p-persistent CSMA and access control based on the network allocation vector.

1. **1-persistent CSMA** [56] is a greedy protocol. When a node needs to send a packet, it first senses the channel. If the channel is idle, the packet is transmitted immedi-

ately. Otherwise, the node keeps listening until the channel becomes idle again, and then transmits immediately. The problem with 1-persistent CSMA is that it reduces collisions caused by interference from earlier transmissions, but results in more collisions caused by interference from simultaneous transmissions. This is because when the channel is idle, all deferred nodes transmit at the same time.

2. **Non-persistent CSMA** [56] reduces the greediness of 1-persistent CSMA. When the channel is busy, instead of listening until the channel becomes idle again, a node waits for a random period of time, and senses the channel again. If the channel is idle, the node starts a transmission immediately. Randomization helps to reduce the number of simultaneous transmissions.

3. **p-persistent CSMA** [56] compromises between 1-persistent and non-persistent CSMA. A node monitors the channel state before transmission. If the channel is idle, the node transmits with a probability of p, or delays its transmission with the probability of $(1-p)$ for a random time period t. Otherwise, it keeps listening to the channel until it becomes idle again, and repeats the same procedure. The value of p has to be selected carefully, because it influences the level of interference, as well as the length of idle time.

4. **Network Allocation Vector** [36] differs from the above persistence-based policies by introducing a network allocation vector. This approach has been applied with RTS/CTS on top of CSMA/CA in IEEE 802.11. Each node uses this vector to record the upcoming transmission periods of its neighbor nodes, and remains silent when its neighbor is transmitting. In comparison with persistence-based approaches, deferral policies based on the network allocation vector can be more efficient because these approaches explicitly notify the schedules of interfering transmissions. However, message exchanges also increase overhead. In addition, because RTS/CTS suffers from the collisions of control packets, as well as exposed and hidden terminals, the accuracy of deferral policies based on network allocation vector is compromised. Another approach to avoid earlier-transmission-caused interference can be achieved by scheduling. Approaches based on scheduling are described in Section 2.2.2.

Avoiding Collisions Caused By Simultaneous Transmissions Although interference detection and access deferral are effective at alleviating interference from earlier transmissions, these approaches are not suitable to avoid collisions caused by interference from simultaneous transmissions. Because of simultaneous transmissions, this type of interference is difficult to detect. Instead of detection, previous work uses randomization to reduce the chance of simultaneous transmissions. Existing randomization strategies fall into two categories: uniform and non-uniform randomization.

Uniform randomization

The first type of randomization is based on generating of random numbers with a uniform distribution.

1. **SEEDEX: A MAC protocol for ad hoc networks** [83] observes that random-number generation is deterministic with an initial seed that is fixed. The state of each node at a certain slot is determined by the recurrence method used to generate random numbers. By exchanging the initial seed between two-hop neighbors, each node can compute the possible state of its neighbors at a certain slot. These states are specified as "possible transmit" or "listen". Therefore, a node can deduce the number of "possible transmit" nodes and "listen" nodes at each slot and adjust its probability of transmission accordingly. Furthermore, nodes also exchange their current states of random number generation periodically. Because a node at the "possible transmit" state may not have data to send, other nodes can adjust the parameter of access aggressiveness a (according to the traffic load) to achieve optimal access.

2. **Adjusting access probability based on collisions** [60] is inspired by four observations: 1) event-driven sensor nodes experience traffic bursts; 2) not all packets need to be successfully received; 3) energy consumption is important in wireless sensor networks; and 4) the chance of collisions increases when the number of active nodes is larger. Accordingly, in order to reduce the number of simultaneous transmissions, the probability of channel access of sensor nodes is decreased, when a collision happens. This is achieved by reducing the wake-up probability of a sensor node.

Non-uniform randomization

The second trend of randomization strategies is to apply a non-uniform distribution to reduce simultaneous transmissions.

1. **Sift: A MAC Protocol for Event-Driven Wireless Sensor Networks** [44] is also proposed for wireless sensor networks based on the same observation that the chance of simultaneous transmissions increases with the number of active nodes. Instead of adjusting the channel access probability of a node according to the measurement of collisions, Sift uses a non-uniform distribution to determine the probability of channel access by each node. The basic idea is to let a node access the channel with a smaller probability during the first few slots, because the number of competing nodes is often large at the beginning of a contention period. With the progression of time, this access probability increases so that it approaches to 1 at the end.

2. **Hashing backoff** [91] observes that collision avoidance is similar to the insertion problem of hash tables in operating systems. The goal is to generate a hash function

that is orthogonal for each node in the form of $b_i^j = a_i^j + m \times ran(n)$, where b_i^j is the slot number for node i during the contention cycle j, $m \times n = CW$ and $0 \leq a_i^j \leq m-1$. Upon a collision, a_i^j is reselected. The authors prove that this procedure guarantees asymptotic convergence to a collision-free state, given that the number of active nodes is less than or equal to the modulus m.

In summary, approaches based on randomization are simple and efficient in terms of reducing overhead. However, they may cause idle slots and cannot eliminate collisions.

Avoiding Collisions Caused By Future Transmissions Future-transmission-caused interference is defined in Section 2.2.1. If a node is currently transmitting, another transmission that arrived later within its contention range causes interference from future transmissions. Counter-solutions can be summarized from the viewpoints of the early and future transmission.

Interference Detection

Because earlier-transmission-caused and future-transmission-caused interference describe the same scenario from perspectives of the future and earlier transmission respectively, collisions caused by interference from future transmissions can also be reduced by avoiding collisions caused by interference from earlier transmissions. Specifically, if a node which prepares to transmit can detect earlier transmissions, future-transmission-caused interference can be avoided by using the solutions described in Section 2.2.1. However, depending on the ability of interference detection by a wireless node, not all earlier transmissions can be detected (e.g. if RTS packets collide or the physical carrier sensing range is not large enough, collisions caused by hidden terminals will still happen).

Collision Detection

From the viewpoint of an ongoing transmission, it is hard to predict the occurrence of a future transmission. In wired networks, collision detection has been successfully applied to stop collisions caused by interference from future transmissions. For example, carrier sense multiple access with collision detection (CSMA/CD) enables a node to terminate its transmission when another signal is detected [94]. This method helps to reduce collisions and shorten the time before a retransmission can be made. However, due to hardware limitation [78], CSMA/CD cannot be used in wireless networks. Despite that, soft collision detection can still be used.

1. **Floor Acquisition Multiple Access With Pauses and Jamming (FAMA-PJ)** [27] is proposed to reduce hidden terminals by floor acquisition, active and passive jamming in wireless networks with shared channels that are half-duplex. A node with packets to transmit has to acquire the floor via RTS/CTS. Because a node may send one or multiple packets, other nodes may have to wait for a long time [27].

Active jamming is used when a node hears carriers after sending an RTS to notify its neighbors about the collisions. Passive jamming is used to avoid hidden terminals if RTS packets collide. This jamming signal is sent by a node in the passive state that overhears carriers, but cannot hear a complete RTS packet. The detection of RTS collisions helps to avoid data collisions.

2. **Out-of-band control** [78] can be used for collision avoidance and detection. The basic idea is to divide the wireless channel into two bands. A separate narrow band is used to transmit control pulses (single-frequency waves) and a broader band is used to transmit data packets. A contention (RTS) pulse has a fixed-active phase and a random-pause phase. Busy-tone waves are transmitted during the active phase of a pulse to signal a bursty data channel [78]. Data packets are transmitted on the data channel along with a contention pause. The pause phase is used for collision detection. A CTS pulse is used as a clear channel signal and it is relayed to the neighbors of the receiver for medium reservation. Because the length of a CTS pulse is determined by the random value specified in the previously received data header, the original sender can detect collision if a shorter or longer CTS pulse is received. As soon as a collision is detected, the ongoing data transmission is aborted.

Collision Resolution

Up until now, we have classified the approaches of collision avoidance. The co-existence of all three types of temporally correlated interference causes very complicated and dynamic interference scenarios. This makes collision elimination extremely difficult. Even though we can eliminate all possible collisions, the price will be very high in terms of the amount of information required for accurate interference detection. Therefore, rather than eliminating collisions, collisions can be resolved with a lower price. A common approach is to retransmit a collided packet. However, because retransmissions are also subject to all types of interference, the goal is to reduce repeated collisions of the same packet as much as possible. The basic problems are 1) the waiting duration before the next retransmission; 2) the fairness issue; and 3) the sequence of re-scheduling all transmissions that have collided.

Fast recovery and fairness improvement Most approaches belonging to this category are backoff mechanisms based on the contention window. The basic idea of these backoff mechanisms is to increase the contention window upon collision, and to decrease the contention window upon success. Therefore, it is also called the X increase Y decrease approach [53]. X indicates the aggressiveness of increasing the contention window and determines the recovery speed. Y represents the speed of decreasing the contention window and influences the fairness of channel access by different nodes.

A commonly-used approach of collision resolution is the binary exponential backoff mechanism. The binary exponential backoff mechanism [36] doubles the contention window of a node when its transmission collides. The doubling of contention window continues if retransmissions keep colliding. If the ith retransmission succeeds and i is less than the retry limit, the contention window is reduced to a minimum pre-determined value. However, this approach suffers from long delay and severe unfairness. Most previous work focuses on modifying the binary exponential backoff mechanism by changing the speed of increasing and decreasing the contention window to either improve fairness or to enable faster adaptation to external traffic that is frequently changing. For example, linear increase linear decrease (LILD) [100] helps to reduce collisions when the traffic is light. Multiplicative increase linear decrease (MILD) [4] enables fast responses to collisions when the traffic is heavy. Both approaches alleviate the monopolization of channel access by slowly decreasing the contention window. Exponential increase exponential decrease (EIED) [45] achieves further improvement by allowing faster responses to collisions and faster recoveries from collisions.

Splitting tree algorithm Besides backoff schemes based on the contention window, the splitting tree algorithm [11] is another approach to resolve collisions. It sorts the sequence in which to retransmit packets that have collided in a random and distributed way. The key idea is to divide transmissions that have collided into two sets: transmit and non-transmit. This splitting is achieved by letting each node with colliding transmissions randomly select 0 (not to send) or 1 (send). At the end of a slot, some of these retransmissions may succeed and others may collide again. At the next slot, those new transmissions that have collided and the set that did not transmit previously merges and regroups into a new transmit and non-transmit set. The transmit set is retransmitted. This procedure continues until all nodes successfully re-transmit their colliding packets. Thus, delay can be reduced because a node does not increase the contention window. However, because the last few nodes have to experience more than one collision to achieve successful retransmissions, time, energy and bandwidth are wasted. Various improvements have been proposed [69] [70] to reduce delay and increase throughput. Theoretical analysis shows that the best solution can achieve close to 50% of the network capacity.

In summary, accurate and prompt detection of transmission opportunities is required to improve temporal reuse of wireless channels, as well as the avoidance of all three types of temporally-correlated interference. However, this is difficult to achieve because 1) accurate detection requires extra resources and causes higher overhead; 2) temporal interference may change dynamically and compromises detection accuracy.

2.2.2 Spatially-Parallel Transmissions

Temporal reuse is one dimension to improve resource utilization under the mutual exclusive transmission constraint. Geographical distribution provides another opportunity to increase resource consumption through parallel transmissions.

The basic constraint of parallel transmissions is the contention region. Spatially-parallel transmission can occur only if simultaneous transmissions do not share an overlapping contention region. However, this is difficult to guarantee due to the following two factors.

- **Dynamic contention region**
  The contention region of a transmission pair (including a sender and receiver) has neither a fixed range nor a regular shape, because of unstable and asymmetric channel conditions, capability of radio reception and environmental influences. Accumulated interference and time-varying transmit power also influence the range of the contention region.

- **Dynamic traffic scenarios**
  The roles of sender and receiver of a transmission pair may also change due to handshaking procedures (e.g. RTS/CTS, DATA/ACK). The switching of transmission roles may cause two interference-free transmissions to interfere, and dramatically changes the patterns of spatial contention.

We summarize previous work to improve the spatial reuse of wireless channels. This section first reviews existing collision models that are commonly used to characterize the successful reception criteria of a radio receiver, including the protocol and physical collision models. Secondly, we describe medium access control mechanisms according to channel access predictability and traffic load awareness.

The Protocol Collision Model

The protocol collision model is based on the assumption that interfering nodes are within a certain range of a receiver. Depending on whether it considers the dynamic nature of spatially correlated interference, this model can be further classified as the fixed and adaptive interference range models.

The Fixed Interference Range Model Protocols based on the fixed interference range model can be expressed as $|x_k - x_j| \geq (1 + \Delta) \times |x_i - x_j|$ [33]. In this equation, x is the location of a node, i and j are the sender and receiver, k is any simultaneously transmitting node, and Δ is the guarding zone. By checking whether all simultaneously transmitting nodes satisfy the equation, a protocol can determine whether the transmission between x_i

and x_j is successful. Existing approaches for interference detection include sender-based, receiver-based, and sender-receiver-based.

Sender-Based Interference Detection

The sender-based interference detection approach requires a sender node to check whether there is any ongoing transmission within its interference range, before starting a transmission. A typical example is CSMA. In CSMA, a sender uses physical carrier sensing to determine the state of the wireless medium within the pre-set carrier sensing range. If the medium is idle, the sender starts transmitting after a random backoff period. Another example is the graph-based TDMA scheduling. Coloring algorithms are commonly used in node-based scheduling to assign different colors to interfering nodes (each color corresponds to a slot), and to assign non-interfering nodes the same color [25] [81]. Link-based scheduling [34] [80] [23] usually converts the network topology graph into a contention graph, in which a vertex represents a link, and an edge means interference. If two links interfere with each other, there is an edge between their representative vertices. Maximal matching can be used to find a collision-free schedule to guarantee all links are scheduled at least once within every cycle.

The receiver-side interference level is crucial to successful transmissions. However, sender-based interference detection solutions only consider sender-side interfering nodes. This approach may cause both hidden and exposed terminal problems.

Receiver-Based Interference Detection

Instead of making a sender detect interference for a receiver, receiver-based approaches initiate transmissions based on the receiver-side interference level. In Multiple access with collision avoidance by invitation (MACA-BI) [93], a new transmission starts upon a receiver's invitation. A receiver sends a request-to-receive packet to request data from a sender. After receiving the request-to-receive packet, the target sender starts transmitting data. In addition, the sender also includes its backlog information in the packet header, so that the receiver can adjust its polling rate accordingly. However, the drawback of MACA-BI is the possibility of the request-to-receive packet collisions. To avoid such collisions, receiver initiated multiple access with simple polling (RIMA-SP) [29] introduces an no-request-to-receive control packet. An no-request-to-receive packet is sent after a receiver transmits a request-to-receive packet and detects interference afterward. Receiver initiated multiple access with dual-use polling (RIMA-DP) [29] supports the same procedure, except it also allows a receiver to transmit data to a sender.

With the receiver-based interference detection approach, hidden terminals caused by simultaneous data transmissions are avoided. However, request-to-receive packets may collide and exposed terminals still exist because the backoff procedure of a node only depends on physical carrier sensing.

Sender-Receiver-Based Interference Detection

The sender-receiver-based interference detection solution is designed to enable sender-initiated transmissions, and receiver-based interference detection. A data transmission is initiated by an RTS packet sent by a sender, and established by a CTS packet from the intended receiver. RTS and CTS packets can also serve as notifications to interfering nodes of both the sender and receiver to reserve the medium and reduce hidden terminals.

LocalTalk CSMA/CA [52] uses RTS and CTS control packets to establish a connection between a sender and receiver. The other nodes do not take advantage of these packets to avoid collisions. In multiple access with collision avoidance (MACA) [52], a sender broadcasts an RTS to its 1-hop neighbors to defer potential transmissions. After the receiver receives the RTS packet, it broadcasts a CTS packet to its 1-hop neighbors to defer transmissions for collision avoidance. If the CTS packet is successfully received by the sender, it starts a data transmission right away. However, because MACA does not apply physical carrier sensing, collisions of RTS and CTS packets can be significant. IEEE 802.11 [36] provides an optional mechanism of RTS/CTS as the virtual carrier sensing on top of the basic access scheme based on physical carrier sensing. Floor acquisition multiple access (FAMA) [28] also addresses the shortcomings of MACA, and combines RTS/CTS with a non-persistent CSMA. Although physical carrier sensing helps to reduce control packet collisions, this approach is subject to exposed terminals. Therefore, MACAW (a medium access protocol for wireless LAN's)[4] introduces two extra control packets: the data-to-send and retransmit-RTS packets, to further improve performance. The data-to-send packet lets exposed terminals with regard to a sender know when the forthcoming data transmission will start, so that they can decide when to start their own transmissions. Medium access via collision avoidance with enhanced parallelism (MACA-P) [2] considers switching transmission and reception roles between a sender and receiver due to the exchanges of RTS, CTS, DATA and ACK control packets. This role switching causes different hidden and exposed terminal scenarios. In order to reduce exposed terminals, MACA-P introduces control gaps before a data transmission and ACK transmission to allow exposed terminals to align their transmissions with the ongoing RTS-CTS-DATA-ACK sequence. The control gap allows exposed terminals to exchange RTS and CTS packets after the CTS transmission of an earlier-started node and before its data transmission. Another gap between the end of data transmission and the start of ACK transmission is used to align the start of transmitting time of ACK packets from exposed terminals.

The introduction of RTS/CTS seems to further avoid hidden terminals, but it complicates the exposed-terminal scenarios, which also reduces channel utilization efficiency. In addition, the RTS-CTS-DATA-ACK sequence complicates interference scenarios and may cause more collisions among RTS, CTS, DATA and ACK packets.

The Adaptive Interference Range Model As mentioned before, the model with fixed interference range suffers from significant hidden and exposed terminal problems, because it lacks the ability to adapt to dynamic spatially correlated interference. Specifically, an interference range that is too small causes significant collisions, while an over-sized range makes wireless nodes too conservative and wastes transmission opportunities. Therefore, adaptive adjustment of interference range has been proposed to reduce collisions and improve channel utilization efficiency.

Jiang and Lew [46] describe the extended hidden terminal problem caused by ACK packets required in CSMA/CA. The authors derive safe distance constraints to enable successful concurrent transmissions under different circumstances. However, they do not suggest how to adjust the carrier sensing range to both reduce collisions and avoid sacrificing possible concurrent transmissions. Deng et al. [21] tune the carrier sensing range to achieve optimal network throughput. Ma et al. differentiate collision causes and tune carrier sensing range and transmit power accordingly [68]. They adjust the contention window solely for fairness tuning, rather than for reducing temporal contention.

The Physical Collision Model

Compared with protocol collision models, physical collision models consider lower level metrics to determine when collisions occur. Typical metrics include the received signal strength (RSS), signal to interference ratio (SIR) and signal to interference plus noise ratio (SINR). Because SINR also considers environment noise and is more accurate, we do not list the approaches based on SIR measurement here.

Self-learning collision avoidance (SELECT) [14] demonstrates the correlation between RSS at both the sender and receiver sides. Based on this observation, SELECT uses the sender-side RSS versus transmission success ratio mapping as a criteria for transmission. If a sender senses an idle channel and its RSS value maps to a high transmission success ratio, it starts a transmission. However, the sender-side RSS does not have a monotonic relation with the transmission success ratio. Even though a self-learning algorithm helps to improve performance, RSS from the sender side is not a good metric for collision modeling. In comparison, SINR is a more accurate metric, because of the direct correspondence between SINR and the packet error rate (PER). The interference aware MAC protocol(IA-MAC) [12] proposes to add the receiver-side SINR to the CTS packet. By overhearing a CTS packet, a potential sender can estimate whether its transmission may collide with the ongoing transmission. However, this approach does not consider the scenario of multiple senders. If a number of nodes decide to transmit simultaneously based on their SINR estimation, collisions will occur. The SINR-based collision model has also been used as a constraint to form the contention graph in scheduling algorithms [7] [101].

Predictability of Channel Access

According to the degree of control of channel access, existing approaches can be differentiated as random access, schedule-based and reservation-based. Random access allows each node to initiate transmissions based on its own traffic schedule. Whether or not a transmission succeeds depends on the interference imposed in time and space. In comparison, schedule-based access specifies the access time of each transmission based on spatial transmission constraints to control how interference correlate in time and space in order to reduce collisions. Reservation-based access schemes combine the strength of both random and scheduled approaches. They allow each node to initiate its channel access request in a random way, but schedule transmissions according to spatial constraints.

Random Access The spirit of random access is to allow a node to transmit whenever it wants. Control protocols of random access can be further classified as the unslotted and slotted versions.

Unslotted Random Access Unslotted random access is the simplest version of medium access. A node can start a transmission whenever it is ready. No resources are consumed for scheduling.

1. **ALOHA** [1] is the first unslotted random access developed in 1970. There is no collision model considered; a node can transmit at any time. Collisions are handled by retransmissions. Due to significant collisions, the theoretical maximum throughput achieved by ALOHA is only 18.4%.

2. **CSMA** [36] is proposed to improve ALOHA by using physical carrier sensing to avoid collision. Various modifications of CSMA are proposed to further reduce collisions. As we described earlier, p-persistent CSMA adopts probabilistic transmissions after the medium becomes idle again; CSMA/CA uses random backoff to reduce the possibility of simultaneous transmissions; and CSMA/CD enables a transmitting node to detect interfering signals and stop transmitting. However, CSMA uses the fixed-range collision model (Section 2.2.2) and suffers from hidden and exposed terminal problems.

3. **RTS/CTS** [36] aims to further improves the performance of CSMA/CA by exchanging RTS and CTS packets to reserve wireless medium. Although this method is supposed to mitigate the hidden terminal problem, RTS/CTS may achieve worse performance than CSMA in certain scenarios [88], due to control packet collisions, complex interference relation, and higher overhead.

With light traffic, unslotted random access can achieve good performance with very low overhead. However, when the traffic is heavy, a larger portion of resources is consumed for collision avoidance and resolution, which leads to lower capacity.

Slotted Random Access

Instead of letting a node start a transmission at any time, slotted random access divides time into slots, and only allows nodes to compete for medium access at the beginning of each slot. If a node misses or fails a transmission opportunity in the current slot, it has to wait until the next slot to retry. The benefit of slotting is the elimination of earlier-transmission-caused and future-transmission-caused interference.

1. **Slotted Aloha** [82] divides channel time into slots with length equal to a single packet's transmission time. Transmitters are permitted to transmit only at the beginning of each time slot to avoid partial packet overlapping. This results in an improvement in channel utilization efficiency. However, it also increases the complexity of implementation, because all transmitters have to be synchronized. Under this protocol, network capacity can reach 36.8%.

2. **Slotted CSMA** also partitions time into fixed-length slots. At the beginning of each cycle, a transmitter accesses the channel based on a certain persistence policy, if no carriers are sensed. Otherwise, it waits for the next slots. Slotted CSMA/CA [104] is used in the beacon mode of IEEE 802.15.4. A coordinator of a personal area network broadcasts beacon messages periodically to synchronize all nodes within the personal area network. A node transfers data during a superframe between the intervals marked by two beacons. A superframe includes an active and inactive period. The active period is composed of a contention access period, and an optional contention free period. Slotted CSMA/CA is used during the contention access period for data transmissions. A contention access period consists of a number of backoff periods. Channel access can only occur at the boundaries between backoff periods. Before sensing the channel state, a node has to wait for a few backoff periods to avoid collisions. Then it starts iterative physical carrier sensing, if the channel is idle. Upon detecting any carrier, the node starts backing off again.

Scheduled Access Although slotted random access mitigates earlier-transmission-caused and future-transmission-caused temporal interference, collisions caused by simultaneous transmissions can still be significant, and hidden/exposed terminals are still major threats to efficient channel utilization. Approaches based on scheduled-access set the time when a transmission starts, to alleviate interference from simultaneous transmissions. Most of these approaches are centralized and require global information to make scheduling decisions. Existing scheduling schemes can be organized according to how they respond to dynamic interference, different traffic demand and topology changes.

Response to dynamic interference

To determine a set of simultaneous conflict-free transmissions, the basic constraint is the spatial interference relation among wireless nodes. As described in Section 2.2.2, collision models can be protocol-based and physical. In scheduling, these models are in the form of node-based, link-based and SINR-based.

1. **Node-based scheduling** is also called broadcast scheduling. It represents a network as a graph, with nodes as vertices and conflicts as edges. Vertex coloring can be used to assign conflicting nodes different colors (time slots). Conflicting nodes are defined in terms of the constraints concerned. First, the primary conflict occurs when a node transmits and receives at the same time. The primary conflict can be avoided by using the 1-hop model, which considers neighbors within 1-hop distance (transmission range) as conflicting nodes. The secondary conflict is that a node cannot receive two transmissions at the same time. By applying the 2-hop model, we can avoid secondary conflicts because neighbors within two-hop distance cannot transmit at the same time. Finally, the k-hop model [32] considers interference within k hops.

2. **Link-based scheduling** treats transmission pairs (the link between a sender and receiver) as scheduling objects. The graph described in node-based scheduling can be used here, by representing transmission links with arcs. Edge coloring can be applied to find conflict-free schedules. Two edges (a,b) and (c,d) are assigned the same color if they do not share common nodes (primary conflict), and there are no edges between a and d or c and b (secondary conflict). Contention/conflict graph [108] is another approach that represents links as vertices and conflicts as edges. The k-hop model can also be added to control interference from nodes farther away.

3. **SINR-based scheduling** allows a link to transmit only if the SINR at the receiver is greater than a threshold. One trivial approach is to select a schedule satisfying the SINR constraint from all possible candidates. However, due to the large number of candidates, [72] uses a column generation algorithm to simplify the problem. Another approach is based on a two-level-graph model [79]. It explicitly adds interference edges to the second level graph. The first level graph includes the nodes and those links whose signal to noise ratio (SNR) is greater than a threshold. Those links whose SNR lower than the threshold, but greater than an interference threshold, are added to the second level graph as interference edges.

Previous work has made great efforts to develop centralized or distributed polynomial solutions to approximate optimal solutions. However, these approaches offer fixed assignment without considering traffic demand.

Response to idle slots

Due to fixed time slot allocation, schedule-based access schemes do not adapt to the offered load at each node. Therefore, slots are assigned to a node even when it does not have data to transmit. These idle slots are wasted while other nodes wait for transmissions. The idle slot reuse scheme [58] [85] helps to solve this problem. It is composed of idle slot detection, cancellation and reuse. If a node has transmitted all its data and the following slots will not be used, it sends a cancellation request to a network coordinator. The network coordinator broadcast a cancellation message. After receiving the cancellation message, all nodes with data to send compete for reusing the idle slot by random access, such as CSMA. However, the exchange of cancellation messages may cause high overhead.

Response to topology changes

Another problem of scheduled access is how to reduce the overhead of adapting to network topology changes. Although we can re-schedule every time the topology changes, if changes occur frequently, this trivial solution may cause very high overhead and long delay. Here, we describe existing adaptive approaches.

1. **Unifying Dynamic Distributed Multichannel TDMA Slot Assignment Protocol (USAP)** [105] is a unified slot assignment protocol, which divides time into cycles. Each cycle has N frames, and each frame consists of M slots. The maximum number of nodes allowed in the network cannot exceed N. By exchanging the net-manager-operational packet among nodes, each node maintains the slot assignments of its neighbors. If a new node joins the network, it first collects these net-manager-operational packets and occupies an unassigned slot, if there are any available. However, this approach has to make sure that N and M are large enough for all possible network sizes. Therefore, USAP has very low channel utilization.

2. **USAP Multiple Access: Dynamic Resource Allocation for Mobile Multihop Multichannel Wireless Networking (USAP-MP)** [106] extends USAP by introducing the adaptive broadcast cycle. Aided by the adaptive broadcast cycle, USAP-MP does not need to fix the length (N) of each cycle, and N can adapt to the change of topology and node movement. If a node discovers an unassigned slot with smaller number than its current assigned slot number, its current assigned slot is replaced by the empty slot. As a result, unassigned slots are pushed to the latter half of a frame cycle, and frame length can be reduced.

3. **Dynamic TDMA Slot Assignment for Ad Hoc Networks (ASAP)** [51] is an adaptive time slot assignment approach. When a new node joins the network, it first collects the slot assignment information from its neighbors. If there are unassigned slots, the new node selects one for itself. Otherwise, the node that owns the largest

number of slots releases one to the new node. In the worst case, where no unassigned slots are available and every node only has one slot, this frame is doubled by appending a new frame at the end. Because the first slot of each frame is reserved without assignment, it is allocated to the new node. All previously assigned slots are copied to the remaining part of the frame. The adaptive TDMA slot assignment protocol in ad-hoc sensor networks (E-ASAP) [50] improves ASAP by exchanging the frame length information, which enables more efficient utilization of wireless channel.

4. **Topology-independent scheduling** [9] is a robust solution to frequent topology change. Instead of guaranteeing the conflict-free property in each slot, this approach satisfies a weaker condition, which guarantees a successful transmission of every node and their neighbor nodes at least one slot in a frame. This requirement is achieved by applying the mathematical properties of finite (Galois) fields. Therefore, as long as a bound on the maximum number of nodes and neighbors is known, this approach guarantees conflict-free operation under any frequency of topology changes.

Reservation-based Access Besides random and scheduled access schemes, reservation-based access is another method.

Synchronized reservation

Similar to scheduled access, synchronized reservation divides time into slots. A cycle is composed of multiple slots. Each cycle is divided into a contention and contention-free access period. During the contention period, ready nodes either send reservation requests through a handshake procedure [86], or exchange schedules among neighbors [16]. Those nodes with assigned slots are guaranteed with collision-free transmissions during the contention-free access period.

Unsynchronized reservation

Instead of synchronizing all nodes to align their transmissions, GAMA-PS [74] uses an optional mechanism of RTS/CTS and packet sensing for reservation. Channel time is divided into contention and group transmission periods. A transmission group includes all the nodes that are allowed for transmission (successful RTS/CTS exchanges). Before a node starts a data transmission, it first sends a begin-to-transmit packet to notify the other group members to backoff. By overhearing the RTS/CTS and begin-to-transmit messages, every node in a transmission group figure out its position in the group and transmits data in turn. The end of a group transmission is marked by a transmission-request control packet sent by the last member in the group. The next contention and group transmissions can then start.

Traffic Demand Awareness

So far, we have described channel access mechanisms with different predictability. Another transmission constraint is traffic demand. Existing scheduling approaches can be differentiated according to traffic assumptions.

Saturated Traffic Model The medium access control schemes based on fixed assignment assume all nodes are saturated and derive conflict-free schedules based on a spatial interference relation. TDMA falls into this category. Scheduling schemes based on saturated traffic do not consider the offered load imposed on each node, and lead to very low channel utilization because of resource waste.

Differentiated Traffic Demand Unlike those scheduling schemes based on fixed assignment, traffic-aware scheduling methods consider the traffic demand of each user. Traffic differences can be reflected by the data rate [31], backlog size [29] or the number of flows going through a common node [87].

Depending on the traffic load of each user, there are two approaches to divide the channel bandwidth through scheduling. The first is to divide the channel time into fractions, and assign each user the amount of time requested. An alternative is to adjust the transmission rate of each user. These approaches can be implemented in random-access, schedule-based or reservation-based access schemes. For example, p-persistent CSMA can adjust the probability of transmissions based on traffic demand. The concept of transmission opportunity introduced in IEEE 802.11e can be used in random access to adjust the channel occupancy time of each user [55]. Schedule-based and reservation-based schemes can assign a consecutive number of slots to users with higher traffic load [98].

Intermittent Traffic Demand Although traffic-aware scheduling schemes consider demand differences among users, these approaches either assume long-term static demand or require frequent control messages to determine varying traffic features. In contrast, random access schemes are suitable for intermittent traffic by allowing each user to initiate transmissions when there are packets to send.

2.3 A Synthesized View

Resource allocation and utilization are two important sub-problems of wireless resource management. On one hand, resource allocation partitions resources to achieve a certain global optimality. On the other hand, resource utilization controls local competition and reduces resource waste. Ideally, solutions to both problems should support each other, so that the allocated resources can be fully utilized to satisfy user/system requirements.

However, existing efforts are devoted to solving these problems separately. In Section 2.1 and 2.2, we categorize previous methods into two classes: global resource allocation and local resource utilization. This section compares the strength and weaknesses of both classes, and describes a new direction that we pursue to achieve effective resource management.

2.3.1 Global Resource Allocation Analysis

Existing methods of resource allocation share three common features: 1) global objectives; 2) simplified assumptions; and 3) theoretical upper-bounds. Typically, these methods form an optimization problem to maximize network throughput/utility or achieve fairness. Due to the complex nature of wireless networks, simplified assumptions are applied to derive mathematically tractable models. For example, Cali and Bianchi assume that 1) all nodes can hear each other; 2) all nodes transmit or collide with the same probability; and 3) hidden/exposed terminals do not exist in wireless local area networks. Similar assumptions are also applied to wireless ad-hoc networks. Most derivative works based on Tassiulas's model (e.g. Jiang's adaptive CSMA/CA) assume that the underlying scheduling eliminates collisions. There has been little effort to characterize the stability region when hidden/exposed terminals exist.

Consequently, allocation policies derived from these theoretical models are over-positive and difficult to achieve in practice. Furthermore, directly applying these policies to guide user resource consumption can lead to significant resource waste in terms of aggravated collisions and congestion. As our simulation results demonstrate in Chapter 5, Jiang's adaptive CSMA/CA experiences severe packet loss that is caused by hidden/exposed terminals. Despite these weakness, global resource allocation enables different users to coordinate resource consumption and achieve a common goal. Although the theoretical upper-bound is hard to achieve, it can serve as a benchmark to encourage new hardware technologies and management methods to approach.

2.3.2 Local Resource Utilization Analysis

Resource utilization is often handled locally by medium access control. Typical techniques include 1) local interference detection; 2) scenario differentiation; and 3) collision avoidance/resolution. For example, existing methods detect interference either based on some assumption that interference happens within a certain range, or based on physical measurements such as the received signal strength (RSS), signal to interference ratio (SIR) and signal to interference plus noise ratio (SINR). Different methods are also used to differentiate dynamic interference in time and space. When interference is detected, a certain method is applied to avoid collisions (e.g. random access or scheduled access). Resolution methods are also applied to reduce repetitive collisions.

Due to the dynamic nature of wireless interference, it is hard to achieve accurate detection of all interference scenarios. As a consequence, collision avoidance/resolution methods make false-negative and false-positive decisions, which lead to collisions and unnecessary idling. Furthermore, local resource utilization can easily cause conflicts among users and compromise system-level performance without a global view. Despite the limited control effectiveness and selfish nature of local resource utilization, it is widely adopted because of the advantages of fast response, distributed control and low control overhead.

In summary, there is a discrepancy between global resource allocation and local resource utilization. Global resource allocation cannot effectively handle local resource competition, due to the requirement of frequent message passing and delayed response. In comparison, local utilization policies are more effective to handle dynamic contention and time-varying channel conditions. Without the support from local control, global resource allocation leads to severe resource waste. Conversely, local utilization methods do not have a global view of the whole network. Without the guidance of global resource allocation, local utilization policies may jeopardize system-wide objectives. Therefore, effective resource management should combine the strength of both global resource allocation and local utilization, and mediate the discrepancy between. In this thesis, we propose a novel method to minimize the gap between the amount of allocated and utilized resources by enabling global resource allocation to consider local competition, and maximizing local utilization to approach the benchmark set by global resource allocation.

Chapter 3

Globalized-Local Optimal Resource Management

Uncontrolled competition for limited resources causes unfair resource consumption and degrades performance significantly in shared-resource systems. Allocating resources fairly to users, and using these limited resources efficiently is crucial for improving system performance and satisfying user requirements. Particularly, in wireless networks, competition for finite resources causes severe congestion and collisions. Without effective management, users may experience very long delay, significant packet loss and poor throughput. In this chapter, we first present G-Local resource management: a framework for effective resource management in wireless networks. Compared with existing methods, G-Local resource management offers novel mechanisms that fairly allocate resources, efficiently utilize resources, and effectively control network behavior with very low computational and control overhead.

In the rest of this chapter, we present the layered structure of G-Local resource management, and overview its major modules in Section 3.1. Section 3.2 presents the formulation of G-Local optimization, the theoretical foundation of this framework. Finally, we summarize this chapter in Section 3.3.

3.1 Framework Overview

The overall structure of G-Local resource management is shown in Fig. 3.1. We present this framework in a layered architecture to ease the separation of different concerns, and to demonstrate inter-layer interaction. In this framework, we lay the theoretical foundation of G-Local optimization, and reify this theory with novel control methods at the second layer. On top of these two layers, in the following chapters we augment and customize G-Local theory and control to tackle unique challenges in two types of wireless networks: single-hop and multi-hop.

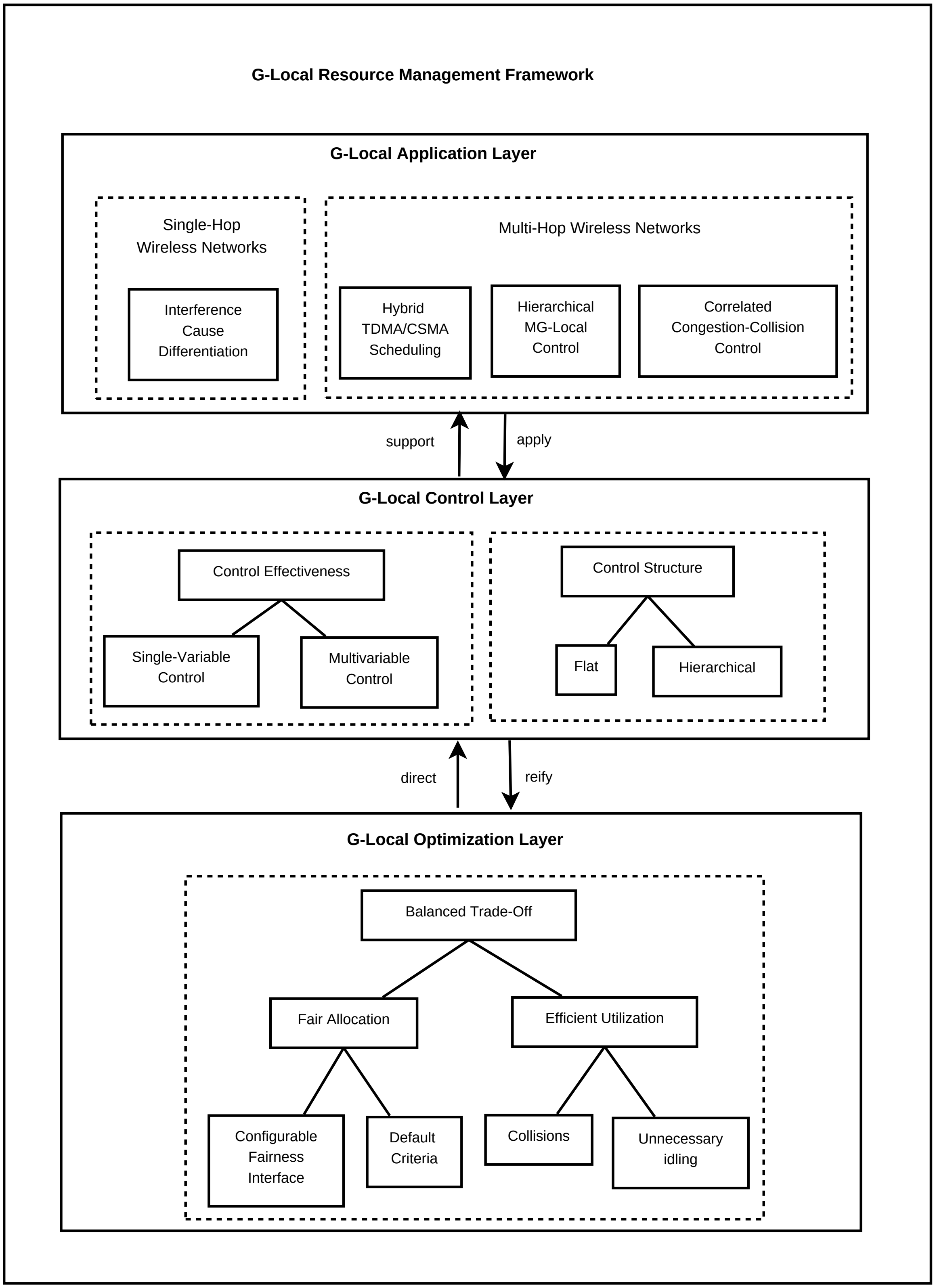
G-Local Resource Management Framework
G-Local Application Layer
Single-Hop Wireless Networks
Interference Cause Differentiation
Multi-Hop Wireless Networks
Hybrid TDMA/CSMA Scheduling
Hierarchical MG-Local Control
Correlated Congestion-Collision Control
support
apply
G-Local Control Layer
Control Effectiveness
Single-Variable Control
Multivariable Control
Control Structure
Flat
Hierarchical
direct
reify
G-Local Optimization Layer
Balanced Trade-Off
Fair Allocation
Efficient Utilization
Configurable Fairness Interface
Default Criteria
Collisions
Unnecessary idling

Figure 3.1: Framework Architecture

3.1.1 Layer One: G-Local Optimization

In this chapter, we introduce G-Local optimization as the fundamental layer of the proposed framework. It offers a new theoretical foundation that 1) supports configurable and adaptive fairness, 2) suppresses the co-existence of conflicts and waste, 3) balances the trade-off between fairness, efficiency and effectiveness, and 4) approaches desired network states via a local method with zero control-message passing.

1. **Adaptive and configurable fairness**

   Instead of restricting the framework to a pre-defined type of fairness, G-Local optimization can be configured with different fairness criteria. This is because no single fairness criteria can be generalized to meet diverse user requirements and system differences. We assume that network managers/operators select a desired fairness criteria, derive bandwidth fair share for each user, and enter derived fair share to G-Local via a fairness interface. This assumption is based on the following fact. First, fairness criteria essentially prioritize users and indicate which group can be compromised when resources are insufficient. Second, it requires authority, specific knowledge regarding the types and features of applications that are running in the network, as well as the demands and classification of users to define these rules. Therefore, protocol designers should support and fulfill, rather than making these subjective and judgmental decisions. Accordingly, G-Local optimization supports different fairness criteria by separating the fairness definition from its fulfillment, and simplifies the derivation of fair share. In addition, conventional fairness criteria are designed for wired networks where no interference exists and network conditions are more stable. Directly applying these criteria in wireless systems leads to a gap between resource allocation and utilization due to the lossy nature of wireless transmissions, imperfect medium access control (MAC), and time-varying quality of wireless channels. In the rest of this thesis, we call the difference between a configured fair share and utilized resource, the *allocation-utilization gap*. G-Local optimization explicitly characterizes the impact of wireless interference and dynamic network conditions on fairness, and re-adjusts the fair share.

2. **Two-way reduction of resource waste**

   G-Local optimization handles the co-existence of collisions and unnecessary idling, which are caused by two factors. First, wireless transmissions are lossy due to dynamic interference and network conditions. Second, MAC-scheduling algorithms such as TDMA and CSMA are imperfect. Although scheduling is supposed to fully utilize transmission opportunities and reduce collisions, existing protocols (e.g. CSMA) often make wrong decisions due to aggressive/conservative policies. Consequently, wireless bandwidth is wasted on either collisions or nodes being idle. We call the second situation *unnecessary idling*. Both collisions and unnecessary idling add up to the allocation-utilization gap in wireless networks. G-Local optimization accounts for these two factors to reduce resource waste.

3. **Balanced trade-off among fairness, efficiency and effectiveness**
   G-Local optimization balances between maximizing network utility by driving the user consumption toward the configured fair share, and minimizing resource waste caused by collisions and unnecessary idling. On one hand, due to wireless interference, it may not be possible to achieve the configured fair share. Instead of continuously pushing the system toward the configured fair share, G-Local optimization re-evaluates the fair share to reflect the impact of wireless interference. On the other hand, G-Local optimization characterizes the impact of wireless interference, dynamic network conditions and imperfect scheduling by calculating a waste cost, so that control methods can be applied to minimize collisions and unnecessary idling. In addition, the overhead incurred by reducing collisions and unnecessary idling is also taken into account. By balancing these two effects, G-Local optimization uses fairness to drive resource consumption, and re-adjusts the fair share when the selected control method cannot reduce the waste cost any more.

4. **Zero control-message passing**
   G-Local optimization minimizes control overhead by intelligently exploiting local information. We enable each node to infer the level of competition, and jointly maximize resource utilization and minimize resource waste caused by collisions and unnecessary idling. In the next chapter, our simulation results demonstrate that compared with a classic global optimization algorithm, G-Local optimization achieves competitive results with zero message passing.

3.1.2 Layer Two: G-Local Control

Chapter 4 introduces G-Local control, the layer that reifies G-Local optimization. While G-Local optimization balances the trade-off between allocation fairness and utilization efficiency, control methods help to reduce the allocation-utilization gap defined in Section 3.1.1. In this layer, we offer two alternative control methods: single-variable and multivariable control, and provide two ways: flat or hierarchical, to implement G-Local control.

1. **Single-Variable Control vs. Multivariable Control**
   The single-variable method models network behavior: transmissions, collisions and idleness as functions of a single control variable. With this method, G-Local control adjusts the single control variable to reduce collisions and unnecessary idling, and

re-adjusts the configured fair share by taking into account resource waste. The multivariable control method improves control effectiveness and minimizes the gap between the configured fair share and utilized resource. In this method, we categorize wireless interference dynamics into four dimensions: time, space, strength and frequency, and differentiate their causes. Correspondingly, we select a number of control variables that have strong impacts on each dimension, and model network behavior as multivariable functions of these control variables. Aided by the multivariable method, G-Local control drives the system to reach a desired network state via adjusting the selected control variables.

2. **Flat vs. Hierarchical Control**

   G-Local control can be implemented as two structures: flat and hierarchical. The basic idea of flat control is to find a desired network state via adjusting the selected control variables. In contrast, the hierarchical control uses a desired network state to drive the underlying control method to adjust the selected variables. It constructs a two-tier hierarchy between G-Local optimization and a selected control method, and provides a clearer direction for making control decisions.

3.1.3 Layer Three: G-Local Application

Chapter 5 demonstrates the application of G-Local resource management in multi-hop wireless networks. We propose three mechanisms to handle the 1) co-existence of intra-path and inter-path interference, 2) chaotic and unfair competition, and 3) correlated congestion and collisions.

1. **Hybrid TDMA/CSMA**

   This first mechanism is proposed to mitigate the co-existence of intra-path and inter-path interference. Our hybrid scheduling scheme eliminates intra-path interference via time-division-based scheduling, and handles inter-path interference via random-access-based scheduling. Compared with existing work, which relies on heavy message passing and complex computation, this method achieves comparable performance with very simple computation, and fast adaptation to handle dynamic traffic patterns with minimal message passing. These advantages are achieved by enabling each path to derive its own schedule locally, promptly adjusting existing schedules upon traffic pattern changes.

2. **Hierarchical MG-Local Control**

   The second mechanism aims to achieve fair allocation and efficient utilization of wireless bandwidth in multi-hop networks. We apply a multivariable version of G-Local control to guide different paths to compete for their fair share of wireless bandwidth,

Table 3.1: G-Local Symbols

Parameter	Value
U_i	network utility
C_i	consumption cost
W_i	waste cost
I_i	non-conflict-caused waste cost
F_i	conflict-caused waste cost
x_i	bandwidth consumed for transmission
x_i^f	bandwidth fair share
$coll_i$	bandwidth wasted on collisions
$idle_i$	bandwidth wasted on idling
B	channel bandwidth
k	scaler

and effectively control chaotic inter-path competition in wireless networks. This mechanism enforces fairness-driven resource allocation, achieves efficient utilization of allocated bandwidth, and effective control of network behavior.

3. **Correlated Congestion-Collision Control**

   Congestion in multi-hop wireless networks is caused by both traffic overload and wireless interference. Correlated congestion-collision control lets each forwarding-node compute a congestion cost based on its incoming/outgoing traffic, and local interference level. This congestion cost is passed back to the upstream node to adjust its outgoing traffic in order to alleviate congestion at the downstream node. Compared with conventional congestion control, this method effectively handles the mutual impact of congestion and collisions in wireless multi-hop networks.

3.2 G-Local Optimization

We design G-Local optimization to guide each user to consume a configured fair share of network bandwidth by increasing the amount of resources being used for successful transmissions, and reducing the amount of resources being wasted by collisions and unnecessary idling. If the configured fair share is not achievable due to dynamic interference, time-varying network conditions, or imperfect scheduling, G-Local optimization re-adjusts the fair share to reflect the impact of these negative factors. This section first presents the formulation of G-Local optimization, and then analyzes how it achieves fair allocation, efficient utilization as well as a balanced trade-off between fairness and efficiency.

Before diving into the details, we introduce a number of symbols in Table 3.1.

3.2.1 Formulation

G-Local optimization is structured as in Eq. 3.1 with three components: utility (U_i), consumption cost (C_i) and waste cost (W_i). i is a local user. A user can be specified as a node, an abstract wireless link (see definition in Chapter 1), or a network flow. Because we present the generic formulation of G-Local optimization in this chapter, the generic concept of user i is used here. In the next two chapters, we specify a user as a wireless link and reify G-Local optimization with specific control methods. G-Local maximizes utility and minimizes costs. We elaborate each component of G-Local Optimization as follows:

$$\max\ k \cdot U_i \ - \ (1-k) \cdot (C_i \ + \ W_i) \tag{3.1}$$

- U_i represents the benefit or satisfaction of a local user i when its packet delivery ratio is x_i. We specify $U_i = \log(x_i)$ so that user satisfaction increases with x_i, and so that the marginal benefit decreases as x_i increases.

- C_i is the consumption cost and plays two important roles. First, it controls the greedy behavior of a local user, by imposing a limit on the resource consumption x_i. Thus, the more a user consumes, the higher the cost it has to pay. Second, C_i drives x_i toward a configured fair allocation x_i^f, in the form of $C_i = \frac{1}{x_i^f} \cdot x_i$.

- W_i is the cost of resource waste and characterizes the impact of network dynamics including interference and network conditions on the allocation-utilization gap. We specify $W_i = \frac{coll_i}{B} + \frac{idle_i}{B}$, where, $coll_i$ and $idle_i$ are the amount of bandwidth wasted on collisions and unnecessary idling, and B is the channel capacity.

Eq. 3.2 is the concrete form of G-Local optimization, and is derived by substituting $\log(x_i)$, $\frac{1}{x_i^f} \cdot x_i$ and $\frac{coll_i}{B} + \frac{idle_i}{B}$ for U_i, C_i and W_i respectively in Eq. 3.1. We solve Eq. 3.2 in the next chapter using two specific control methods.

$$\begin{aligned} &\max\ V(x_i, coll_i, idle_i) \\ V\ &= k \cdot \log x_i - (1-k) \cdot (\frac{1}{x_i^f} \cdot x_i + \frac{coll_i}{B} + \frac{idle_i}{B}) \end{aligned} \tag{3.2}$$

3.2.2 Fair Allocation

G-Local optimization supports different fairness criteria.

- First, we separate the derivation of fair share from its fulfillment. Specifically, we let network managers/operators select a fairness criteria that is suitable for the types and features of applications running in their systems. For example, in Chapter 5 we use max-min fairness for multi-hop forwarding. In case network operators/managers do not have a particular preference, we also provide a simple fairness criteria as the default option. This default criterion specifies that each link shares the channel bandwidth with interfering links equally.

- Second, we simplify the derivation of fair share. Specifically, we allow network managers/operators to assume a simple interference model and stable network conditions. For example, they can assume interference is only from transmission within a certain distance and does not change over time. However, the impact of network dynamics is at lost. It is explicitly handled by adding the waste cost, W_i, to G-Local optimization.

- Third, we provide a configurable fairness interface via x_i^f for network managers/operators to enter the desired fair share. The value entered for x_i^f is called the configured fair share in the rest of this paper. By formulating the consumption cost as $C_i = \frac{1}{x_i^f} \cdot x_i$, we ensure that G-Local optimization converges to $x_i = x_i^f$ if there are no network dynamics. This is derived by ignoring W_i, and transforming Eq.3.2 to Eq. 3.3.

$$\begin{aligned} &\max\ U_i - C_i \\ => &\max\ \log(x_i) - \frac{1}{x_i^f} \cdot x_i \end{aligned} \tag{3.3}$$

We call Eq. 3.3 the *G-Local utilization control*, and it represents the effort to achieve fair allocation.

3.2.3 Efficient Utilization

G-Local optimization achieves efficient utilization by handling the co-existence of collisions and unnecessary idling. In wireless networks, there is always a gap between the amount of resources being allocated, and the amount of resources being used for successful transmissions. This allocation-utilization gap is attributed to two factors: 1) wireless transmissions are lossy due to dynamic interference and network conditions; and 2) MAC-scheduling algorithms are imperfect and sometimes make incorrect decisions due to aggressive/conservative policies. Consequently, wireless bandwidth is wasted on either collisions or on nodes being idle. Therefore, G-Local optimization formulates the non-conflict-caused waste cost I_i and conflict-caused waste cost F_i in Eq. 3.4.

$$\begin{aligned} I_i &= \frac{idle_i}{B} \\ F_i &= \frac{coll_i}{B} \\ W_i &= I_i + F_i \end{aligned} \tag{3.4}$$

We call the second aspect of G-Local optimization the *G-Local waste control.* It represents the effort to minimize resource waste (Eq. 3.5).

$$\begin{aligned} &\max\ U_i - W_i \\ => &\max\ \log(x_i) - (\frac{coll_i}{B} + \frac{idle_i}{B}) \end{aligned} \tag{3.5}$$

3.2.4 Balanced Trade-off

By combining utilization and waste control, G-Local optimization balances the effort to achieve the configured fair share, and minimize resource waste. We categorize the trade-offs into local and global.

1. The local trade-off is characterized as the allocation-utilization gap. The waste cost W_i indicates the price of reducing this gap by pushing utilization toward the configured fair share. As shown in Eq. 3.4, this cost has two properties: 1) it decreases with more efficient resource utilization, because of decreased collisions and reduced idling; 2) its reduction speed is a non-decreasing function, because smaller resource waste requires greater effort from the underlying control method. Furthermore, how much the resource waste can be reduced is determined by the effectiveness of the underlying control method. In other words, W_i reflects the trade-off between control effectiveness and control overhead. Therefore, these two properties of W_i guarantee that G-Local optimization does not keep pushing a user to aggressively consume resources when the cost exceeds the gain.

2. The global trade-off between fairness and efficiency is reflected as the fact that users experience interference at different intensity and frequency over time and space. Users experiencing more frequent destructive interference are heavily punished. We introduce a weight $k = \frac{x_i^f - x_i}{x_i^f}$, and use it as the weight in the G-Local objective function Eq. 3.2. This weight gives a user a lower weight if its x_i is closer to the ideal fair-share, so that some bandwidth goes towards the transmissions that are less fortunate.

3.2.5 Zero Message Passing

Because Eq. 3.2 only requires local information x_i, $coll_i$ and $idle_i$, G-Local optimization requires zero control-message passing. By intelligently exploiting local information, G-Local enables each user to infer the level of competition, and jointly maximizes resource utilization and minimizes resource waste caused by collisions and unnecessary idling.

3.3 Chapter Summary

In this chapter, we provided an overview of the layered structure of G-Local resource management. The first layer of G-Local optimization is the theoretical foundation of this work. We present its formulation and show how it supports fair allocation and efficient utilization with a balanced trade-off and zero message passing. The second layer G-Local control reifies the theory of G-Local optimization with both single-variable and multivariable control. The third layer is the G-Local application in multi-hop wireless networks.

Chapter 4

Resource Management in Single-Hop Wireless Networks

In Chapter 3, we gave an overview of the complete structure of G-Local resource management, and introduced its first layer G-Local optimization. This chapter focuses on the second layer: reified G-Local control, and applies it in single-hop environment. The multi-hop application will be introduced in the next chapter.

Resource competition is inevitable in shared resource systems. In wireless networks, this problem is aggravated due to dynamic interference and network conditions. Without appropriate control, harmful competition causes unfair user consumption of resources (e.g. starvation), and resource waste due to collisions and idleness. In this chapter, we focus on resource management in single-hop wireless networks, where a sender node (also called a source) sends data to its designated receiver (also called a destination) directly through a shared wireless channel. In single-hop wireless networks, external interference changes over time and space; causes lossy transmissions; and significantly degrades performance. To tackle these challenges, we reify G-Local optimization with two control methods: single-variable and multivariable control. The reified G-Local optimization is called G-Local control. We also provide two control structures, flat and hierarchical, to implement G-Local control. The hierarchical control will be introduced in the next chapter.

1. **Single-Variable vs. Multivariable Control**

   The single-variable method treats three network behaviors: transmissions, collisions and idleness as functions of a single control variable. With this method, G-Local control adjusts the control variable to reduce collisions and unnecessary idling, and re-adjusts the configured fair share by taking resource waste into account. Our performance study demonstrates that this method achieves competitive performance comparable to a classic algorithm of global optimization. This is achieved with zero control-message passing.

The multivariable control method improves control effectiveness and minimizes the gap between the configured fair share and utilized resource. In this method, we differentiate the causes of wireless interference in three dimensions: time, space, and frequency-intensity. Correspondingly, we select a number of control variables that have impact on each dimension, and model network behavior as multivariable functions of these control variables. When using the multivariable method, G-Local control drives the system to reach a desired network state via adjusting the selected control variables.

2. **Flat vs. Hierarchical Control**
   G-Local control can be implemented in two structures: flat and hierarchical. The basic idea of flat control is to transform G-Local optimization from a multi-state optimization problem into a multi-variable control problem via direct substitution. The flat control finds a desired network state via adjusting the selected control variables. In contrast, the hierarchical control separates the multi-state optimization and multi-variable control into two problems. It first finds a desired network state and uses it to guide the adjustment of the selected control variables. Therefore, the hierarchical control constructs a two-tier hierarchy between G-Local optimization and control, and provides a clearer direction for making control decisions. This chapter introduces the flat method, and we present the hierarchical method in the next chapter.

In the rest of this chapter, Sections 4.1 and 4.2 introduce the single-variable and multi-variable methods respectively. We conclude this chapter by summarizing our contributions in Section 4.3.

4.1 G-Local Optimization with Single-Variable Control

We introduce a single-variable control as the first control method in the G-Local control layer. This method controls transmissions, collisions, and unnecessary idling by modeling them as functions of a single control variable. The control method steers the network toward desired states by adjusting the selected control variable. This single-variable control is computationally simple and minimizes control overhead. We call the combination of G-Local Optimization and single-variable control, the *SG-Local control*.

Before diving into the details, we list a number of symbols that will be used in the rest of this section in Table 4.1. In this chapter, we specify a user as a wireless link (i, j), which is defined as an abstraction of the transmissions between a source i and destination j.

Table 4.1: SG-Local Symbols

Parameter	**Value**
(i,j)	link (i,j)
$x^s_{(i,j)}$	bandwidth consumed for transmissions of link (i,j)
$x^f_{(i,j)}$	bandwidth fair share
$coll_{(i,j)}$	bandwidth wasted on collisions
$coll^m_{(i,j)}(t)$	measured collisions during the t-th control period
$idle_{(i,j)}$	bandwidth wasted on idling
$p_{(i,j)}$	transmission probability
B	channel bandwidth
k	scaler
h	a small disturbance
λ	Lagrange multiplier or shadow price

4.1.1 Single-Variable Control

Our method models three basic network behaviors as functions of a single control variable. These behaviors include: 1) $x^s_{(i,j)}$, the fraction of bandwidth that is consumed for transmissions over link (i,j) (s denotes sending rate); 2) $coll_{(i,j)}$, the fraction of bandwidth that is wasted on collisions; and 3) $idle_{(i,j)}$, the fraction of bandwidth wasted on unnecessary idling.

Control Variable Selection

The single control variable is $p_{(i,j)}$, the transmission probability. It is used in combination with ALOHA, and defined as the probability to start a transmission when a node has data to send. We use ALOHA instead of CSMA/CA for two reasons. First, ALOHA supports a more direct causality between $p_{(i,j)}$ and the attempted transmissions $x^s_{(i,j)}$. Although CSMA/CA has better performance than ALOHA, its collision avoidance and resolution mechanisms blur the relationship between $p_{(i,j)}$ and $x^s_{(i,j)}$. Second, we use ALOHA to demonstrate the impact of control effectiveness on G-Local optimization: 1) low control effectiveness does not influence the ability of G-Local optimization to balance allocation fairness and minimizing resource waste; and 2) low control effectiveness influences the allocation-utilization gap $W_{(i,j)}$ (e.g. the less effective a control method is, the bigger the gap). For the subsequent multivariable control, we use CSMA/CA as the underlying MAC.

Single-Variable Model

With $p_{(i,j)}$, we model transmission bandwidth, $x^s_{(i,j)}$, as the fraction of bandwidth that is consumed for transmissions (Eq. 4.1). The unnecessary idling $idle_{(i,j)}$ is the amount of bandwidth $x^f_{(i,j)}$ that should be utilized but is not (Eq. 4.2).

$$\begin{aligned} x^s_{(i,j)} =& X(p_{(i,j)}) \\ =& p_{(i,j)} \cdot B \end{aligned} \tag{4.1}$$

$$\begin{aligned} idle_{(i,j)} =& IDLE(p_{(i,j)}) = x^f_{(i,j)} - x^s_{(i,j)} \\ =& x^f_{(i,j)} - p_{(i,j)} \cdot B \end{aligned} \tag{4.2}$$

Compared with $x^s_{(i,j)}$ and $idle_{(i,j)}$, $coll_{(i,j)}$ is caused by the interference from other transmissions. It is hard to find the exact form of $coll_{(i,j)}$ without a model for interfering transmissions. Instead, we approximate the collision model as shown in Eq. 4.4. In this approximation, $coll^m_{(i,j)}(t)$ and $coll_{(i,j)}(t)$ are the measured and estimated collisions of the t-th control period respectively. $p_{(i,j)}(t)$ is the transmission probability used by node i to transmit to j, during the t-th period. $\left(\frac{dCOLL}{dp_{(i,j)}}\right)_{p_{(i,j)}=p_{(i,j)}(t)}$ is estimated via the classic finite difference method as shown in Eq. 4.5 [73]. h is a small disturbance we use to change $p_{(i,j)}$ from $p_{(i,j)}(t)$ to $p_{(i,j)}(t) + h$.

$$coll_{(i,j)} = COLL(p_{(i,j)}) \tag{4.3}$$

$$coll_{(i,j)}(t+1) = coll^m_{(i,j)}(t) + \left(\frac{dCOLL}{dp_{(i,j)}}\right)_{p_{(i,j)}=p_{(i,j)}(t)} \cdot (p_{(i,j)}(t+1) - p_{(i,j)}(t)) \tag{4.4}$$

$$\left(\frac{dCOLL}{dp_{(i,j)}}\right)_{p_{(i,j)}=p_{(i,j)}(t)} \approx \frac{COLL(p_{(i,j)}(t)+h) - COLL(p_{(i,j)}(t))}{h} \tag{4.5}$$

4.1.2 SG-Local Control

G-Local optimization provides a method to drive a system to achieve fair allocation and efficient utilization, which requires a control method to fulfill. In this section, we reify G-Local optimization Eq. 4.6 via the single-variable models Eq. 4.1, 4.2 and 4.3. This reified G-Local optimization is called SG-Local control (see Eq. 4.7). The SG-Local control adopts the default fairness criteria, which specifies that each link shares channel bandwidth equally with interfering links. The fair share $x^f_{(i,j)} = \frac{B}{n^{share}_{(i,j)}}$ can be derived during the system startup procedure, before the SG-Local control starts. $n^{share}_{(i,j)}$ is the number of links that use the same wireless channel as link (i,j), including link (i,j) itself.

$$\begin{aligned} & \max V(x^s_{(i,j)}, coll_{(i,j)}, idle_{(i,j)}),\ where \\ V =& k \cdot \log(x^s_{(i,j)}) - (1-k) \cdot (\frac{1}{x^f_{(i,j)}} \cdot x^s_{(i,j)} + \frac{coll_{(i,j)}}{B} + \frac{idle_{(i,j)}}{B}) \\ =& k \cdot \log(x^s_{(i,j)}) - (1-k) \cdot (\frac{n^{share}_{(i,j)} - 1}{B} \cdot x^s_{(i,j)} + \frac{coll_{(i,j)}}{B} + \frac{1}{n^{share}_{(i,j)}}) \end{aligned} \tag{4.6}$$

$$
\begin{aligned}
&\max\ V_f(p_{(i,j)}),\ where\\
V_f(p_{(i,j)}) =& k \cdot \log(X(p_{(i,j)})) - (1-k) \cdot (\frac{n^{share}_{(i,j)} - 1}{B} \cdot X(p_{(i,j)}) + \frac{COLL(p_{(i,j)})}{B} + \frac{1}{n^{share}_{(i,j)}})\\
&s.t.\ 0 \le p_{(i,j)} \le 1
\end{aligned}
\tag{4.7}
$$

Because $\log(x^s_{(i,j)})$ is a concave function, and $V_f(p_{(i,j)})$ is a non-decreasing function of $p_{(i,j)}$, Eq.4.6 is a strictly concave function, according to the proof (Lemma 3.2) given in [90]. However, whether it has a unique optimal point depends on two conditions: $(i)\mathrm{V}_f(p_{(i,j)}) \to -\infty$ as $p_{(i,j)} \to 0$; and $(ii)\mathrm{V}_f(p_{(i,j)}) \to -\infty$ as $p_{(i,j)} \to \infty$. This requirement has been considered in our formulation by the introduction of k ($0 < k < 1$). We dynamically set k as $1 - \frac{x^s_{(i,j)} \cdot n^{share} B_{(i,j)}}{}$ so that k decreases when $p_{(i,j)}$ increases.

By applying the Lagrangian transformation, Eq.4.7 is transformed to an unconstrained format Eq.4.9, where λ_{p1} and λ_{p2} are shadow prices. The corresponding dual problem is given in Eq. 4.11.

$$
\begin{aligned}
&L_f(p_{(i,j)}, \lambda_{p1}, \lambda_{p2})\\
=&V_f(p_{(i,j)}) + \lambda_{p1} \cdot p_{(i,j)} - \lambda_{p2} \cdot (p_{(i,j)} - 1)
\end{aligned}
\tag{4.8}
$$

$$
\max\ L_f(p_{(i,j)}, \lambda_{p1}, \lambda_{p2}) \tag{4.9}
$$

$$
D_f(p_{(i,j)}, \lambda_{p1}, \lambda_{p2}) = \max\ L_f \tag{4.10}
$$

$$
\min\ D_f(p_{(i,j)}, \lambda_{p1}, \lambda_{p2}) \tag{4.11}
$$

To obtain the shadow prices λ_{p1} and λ_{p2}, we solve the dual problem Eq. 4.11 via $\lambda_{p1} = -h_1 \cdot p_{(i,j)}$ and $\lambda_{p2} = h_2 \cdot (p_{(i,j)} - 1)$. The scalars h_1 and h_2 are determined experimentally to both equal 0.1. Specifically, we choose a few possible ranges for h_1 and h_2, and select the value that leads to the best convergence speed. Then we solve Eq. 4.9 by applying gradient descent. The control policy for $p_{(i,j)}$ is derived in Eq. 4.12. Each user applies this control policy periodically until the algorithm converges.

$$
p_{(i,j)}(t) = p_{(i,j)}(t-1) + k_p \cdot \frac{\partial L_f}{\partial p_{(i,j)}} \tag{4.12}
$$

4.1.3 Performance Evaluation

In this section, we evaluate the performance of SG-Local control (denoted GL in the figures), and compare it with three alternative candidates: G-Local utilization control (denoted

GLU); G-Local waste control (denoted GLW); and Chiang's global optimization [61] (denoted GB). As explained in Chapter 3, GL combines GLU and GLW by maximizing resource utilization and minimizing waste. G-Local utilization control was introduced in Chapter 3. This method aims to achieve the configured fair share $x^f_{(i,j)}$ by increasing $x^s_{(i,j)}$. We specify the single-variable version of G-Local utilization control in Eq.4.13. The GLU control policy 4.15 is derived by applying the Lagrange transformation Eq. 4.14 and gradient-descent method.

$$\begin{aligned} &\max\ V_u(p_{(i,j)}),\ where \\ V_u(p_{(i,j)}) =& \log(X(p_{(i,j)})) - \frac{1}{x^f_{(i,j)}} \cdot R(p_{(i,j)}) \\ &s.t.\ 0 \le p_{(i,j)} \le 1 \end{aligned} \tag{4.13}$$

$$\begin{aligned} &L_u(p_{(i,j)}, \lambda_{p1}, \lambda_{p2}) \\ =& V_u(p_{(i,j)}) + \lambda_{p1} \cdot p_{(i,j)} - \lambda_{p2} \cdot (p_{(i,j)} - 1) \end{aligned} \tag{4.14}$$

$$p_{(i,j)}(t) = p_{(i,j)}(t-1) + k_p \cdot \frac{\partial L_u}{\partial p_{(i,j)}} \tag{4.15}$$

The second alternative, G-Local waste control (Eq. 4.16), aims to minimize resource waste caused by collisions and unnecessary idling. Similar to the first candidate, we derive the policy for waste control in Eq. 4.18 via Lagrange transformation Eq. 4.17 and gradient descent.

$$\begin{aligned} &\max\ V_w(p_{(i,j)}),\ where, \\ V_w(p_{(i,j)}) =& \log(X(p_{(i,j)})) - (\frac{COLL(p_{(i,j)})}{B} + \frac{IDLE(p_{(i,j)})}{B}) \\ &s.t.\ 0 \le p_{(i,j)} \le 1 \end{aligned} \tag{4.16}$$

$$\begin{aligned} &L_w(p_{(i,j)}, \lambda_{p1}, \lambda_{p2}) \\ =& V_w(p_{(i,j)}) + \lambda_{p1} \cdot p_{(i,j)} - \lambda_{p2} \cdot (p_{(i,j)} - 1) \end{aligned} \tag{4.17}$$

$$p_{(i,j)}(t) = p_{(i,j)}(t-1) + k_p \cdot \frac{\partial L_w}{\partial p_{(i,j)}} \tag{4.18}$$

Chiang's global optimization [61] (Eq. 4.19) is the third alternative, which aims to maximize network utility via iteratively adjusting each link (i, j)'s transmission probability $p_{(i,j)}$ with message passing. We compare our GL with Chiang's GB because both work share a similar goal to fairly allocate and efficiently utilize wireless bandwidth. Furthermore, GB represents the typical method of managing wireless resource via network utility maximization. For fair allocation, GB enforces proportional fairness [54]. By maximizing network utility, their method improves resource utilization efficiency. Specifically, the utility function is $U_{(i,j)}(x_{(i,j)}) = log(x_{(i,j)})$ [61]; $x_{(i,j)}$ is the data delivery ratio of link (i, j) and specified as the amount of channel capacity that (i, j) transmits while all nodes that interfere with link

(i,j) (denoted $N_{to}^{I}(i,j)$) remain silent. A link (i,j) transmits with a probability $p_{(i,j)}$, and the transmission probability of a node n (denoted P^n) is the total transmission probability of its outgoing links (denoted $L_{out}(n)$). The more data a link successfully delivers, the higher utility it achieves. But the marginal gain reduces with an increasing $x_{(i,j)}$. Following the Lagrange transformation, dual decomposition and subgradient method, they derive the control policies for $p_{(i,j)}$ and P^n as shown in Eq. 4.20. To enforce these policies, each link (i,j) sends the shadow price ($\lambda_{(i,j)}(t)$) to all of its interfering nodes in $N_{to}^{I}(i,j)$ so that they can compute P^n for the next control period. Similarly, each node n sends P^n to all links that it interferes with (denoted $L_{from}^{I}(n)$) so that they can compute $p_{(i,j)}$ for the next round. The shadow prices are increased if the current data rate of link (i,j) is less than the desired $x'l$. Upon the reception of $\lambda_{(i,j)}(t)$ and $P^n(t)$, node n computes $P^n(t+1)$, $x'_{(i,j)}(t+1)$, $p_{(i,j)}(t+1)$,and $\lambda_{(i,j)}(t+1)$ according to Eq. 4.20. $k_n(t)$ is updated according to $\sum_{l \in L_{out}(n)} \lambda_{(i,j)}(t) + \sum_{k \in L_{from}^{I}} \lambda_k(t)$ and $\beta(t) = 1/t$ at each iteration. More details of this global optimization are available in [61].

$$
\begin{aligned}
\max \sum_{(i,j)} & U_{(i,j)}(x_{(i,j)}) \\
s.t.\ x_{(i,j)} &= W \cdot p_{(i,j)} \cdot \prod_{k \in N_{to}^{I}(i,j)} (1 - P^k),\ \forall (i,j) \\
\sum_{(i,j) \in L_{out}(n)} p_{(i,j)} &= P^n,\ \forall n \\
0 \leq P^n &\leq 1,\ \forall n \\
0 \leq p_{(i,j)} &\leq 1,\ \forall (i,j)
\end{aligned}
\tag{4.19}
$$

$$
\begin{aligned}
&P^n(t+1) = \begin{cases} \frac{\sum_{(i,j) \in L_{out}(n)} \lambda_{(i,j)}(t)}{\sum_{(i,j) \in L_{out}(n)} \lambda_{(i,j)}(t) + \sum_{k \in I_{from}(n)} \lambda_k(t)}, if\ k_n(t) \neq 0 \\ \frac{|L_{out}(n)|}{\left|L_{out}(n) + L_{from}^{I}(n)\right|}, if\ k_n(t) = 0 \end{cases} \\
&p_{(i,j)}(t+1) = \begin{cases} \frac{\lambda_{(i,j)}(t)}{\sum_{(i,j) \in L_{out}(n)} \lambda_{(i,j)}(t) + \sum_{k \in L_{from}^{I}(n)} \lambda_k(t)}, if\ k_n(t) \neq 0 \\ \frac{1}{\left|L_{out}(n) + L_{from}^{I}(n)\right|}, if\ k_n(t) = 0 \end{cases} \\
&{x'}_{(i,j)}(t+1) = argmax\ U'({x'}_{(i,j)}) - \lambda_{(i,j)}(t) \cdot {x'}_{(i,j)} \\
&and \\
&\lambda_{(i,j)}(t+1) = \left[\lambda_{(i,j)}(t) - \beta(t) \left(c'_l + log(p_{(i,j)}(t)) + \sum_{k \in N_{to}^{I}(i,j)} log(1 - P^k(t)) - {x'}_{(i,j)}(t) \right) \right]
\end{aligned}
\tag{4.20}
$$

Implementation

We implemented all four control methods in Network Simulator version 2 (ns2) [26]. To improve the simulation accuracy, we added a SINR-based collision model with aggregate interference [20] to the IEEE 802.11 implementation in ns2. The aggregate interference experienced by a node at a particular time is computed as the total signal strength it receives from all active transmissions at that time. Furthermore, a preamble detection function [92] is also added in ns2. A p-persistent ALOHA is used as the medium access control. In this protocol, if a node has a packet to send, it transmits with a probability of p and delays for the propagation delay of one packet with probability $(1-p)$. Each control method updates p according to its own policy. In this section, we describe the implementation of GL.

GL updates its medium access probability dynamically according to Eq. 4.12. The update only requires three values: $x^m_{(i,j)}$, $coll^m_{(i,j)}$ and $n^{share}_{(i,j)}$. The first two values can be retrieved locally, and $n^{share}_{(i,j)}$ can be obtained when the network first starts up. $x^m_{(i,j)}$ is calculated by dividing the number of packets transmitted (not necessarily delivered) by the length of a control period. $coll^m_{(i,j)}$ is calculated as the average number of unacknowledged packets during a control period. All these values can be easily obtained in real time and locally.

Experiments

Experiments are carried out for two configurations:a 3-link network and larger random networks. We use a Poisson model to generate traffic. All traffic flows are assumed to have the same bandwidth demands. In these experiments, parameters are categorized as fixed or varied. The fixed parameters are listed in Table 4.2. We vary the offered load and interference level to study their impact on the control methods according to four metrics. Furthermore, each experiment is replicated sixteen times and a 95% confidence interval is computed for each metric.

- Jain's fairness index is calculated as $f(x) = \frac{(\sum x)^2}{n \cdot \sum x^2}$, and ranks the fairness of a resource allocation strategy using values between 0 and 1 [43]. Higher values indicate fairer allocation.

- The throughput is the total rate at which packets are successfully delivered to the destinations in the network.

- The packet loss rate is the percentage of packets that are lost due to collisions.

- The end-to-end delay is the average time from a packet being generated at a source node to its successful reception at a destination node.

Table 4.2: System Parameters

Value	**Parameter**
Transmission range	200 meters
Transmit power	0.2818 W
Packet size	512 bytes
Channel capacity	1 Mbps
running time	30 minutes

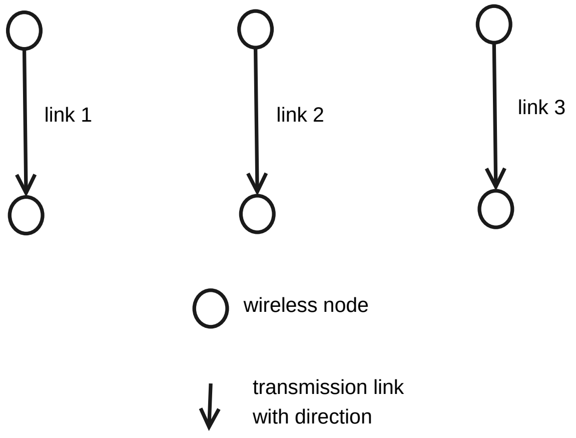

Figure 4.1: 3-Link Network Topology

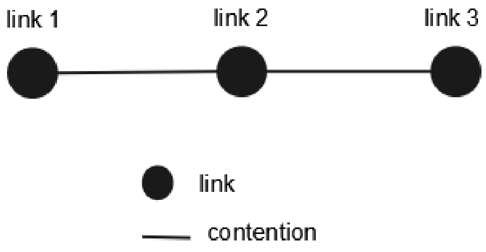

Figure 4.2: 3-Link Network Contention Graph

Three-Link Network

The three-link topology is shown in Fig. 4.1. Fig. 4.2 gives the corresponding contention graph, in which a vertex represents a link, and a line between two vertices indicates that two links interfere with each other. As illustrated in Fig. 4.2, link 2 interferes with link 1 and 3, and conversely, link 1 and link 3 interfere with link 2. Based on these contention relationships, the ideal allocations (denoted ID) for link 1, 2, and 3 are $\frac{B}{2}$, $\frac{B}{3}$ and $\frac{B}{2}$, according to our default fairness criteria (see Chapter 3). The experimental results of per-link allocation and aggregate network performance are shown in Table 4.3 and Fig. 4.3.

Table 4.3 shows the bandwidth utilized by each link in a saturated network. By saturated, we mean the offered load on each link exceeds its ideal allocation which is derived according to our default fairness criteria (see Chapter 3). For example, in this experiment,

Table 4.3: Per-Link Allocation

	link 1(pps)	**link 2(pps)**	**link 3(pps)**
ID	122.07	81.38	122.07
GB	67.23	44.45	67.35
GL	68.93	42.65	69.21
GLU	59.18	56.45	72.23
GLW	141.65	17.43	140.63

the source node of each link sends data at 200 pps, and the ideal allocation for all three links are approximately 122 pps, 81 pps and 122 pps.

First, compared with the ideal allocation (ID in the table), the throughput of the global optimization (GB in the table) is slightly greater than half of the ideal allocation, at 67 pps, 44 pps and 67 pps. Our SG-Local control (GL in the table) achieves very similar results of 68 pps, 42 pps and 68 pps, and this is without any message passing in contrast to GB.

Second, G-Local utilization control (GLU in the table) achieves similar throughput and fairness to GL and GB. However, GLU causes more packet loss (see Fig. 4.3). This is because G-Local utilization control drives the user consumption toward the configured fair-share without considering the waste cost. In contrast, with G-Local waste control (GLW in Table 4.3), links 1 and 3 utilize more bandwidth because these links experience fewer collisions, and therefore cause lower waste-cost. Link 2, on the other hand, is subject to more collisions. Instead of compensating for the bandwidth waste of link 2 to improve fairness, GLW gives more resources to link 1 and 3 causing link 2 to have very low throughput. By combining GLU and GLW, the SG-Local control balances the efforts of driving the system toward configured fair shares and minimizing resource waste caused by collisions and unnecessary idling.

Fig. 4.3 shows the aggregate performance of the three-link network. First, the aggregate performance of GL is very close to GB. Both algorithms achieve more than half of the ideal allocation. Second, GLU achieves higher throughput at the cost of more loss (caused by collisions). Third, GLW results in the highest throughput and lowest loss, but sacrifices fairness. Table 4.3 shows that link 2 is nearly starved when GLW is in use.

The Impact of Offered Load The previous experiment uses a fixed offered load of 200 pps. We now examine results from using different levels of traffic load, from very light (50 pps), unsaturated (100 pps), to saturated (200 pps) and over-saturated (400 pps). The results are shown in Fig. 4.4.

First, when the network is not saturated, the offered load has very little impact on all four methods. This is simply because there is enough bandwidth for each user. In comparison to GB and GL, the performance of GLU and GLW is more sensitive to the offered load in saturated situations. Both GB and GL tend to keep the partitioning of resources the same

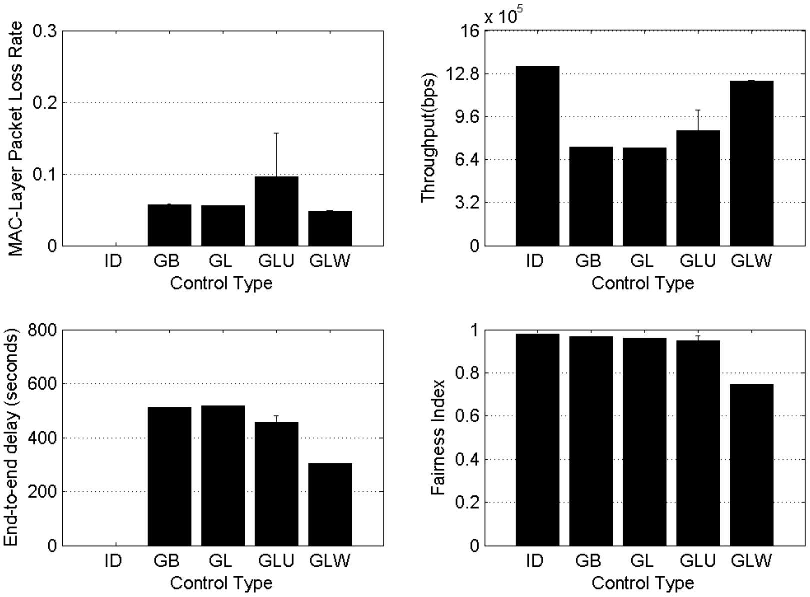

Figure 4.3: Aggregate Performance Comparison

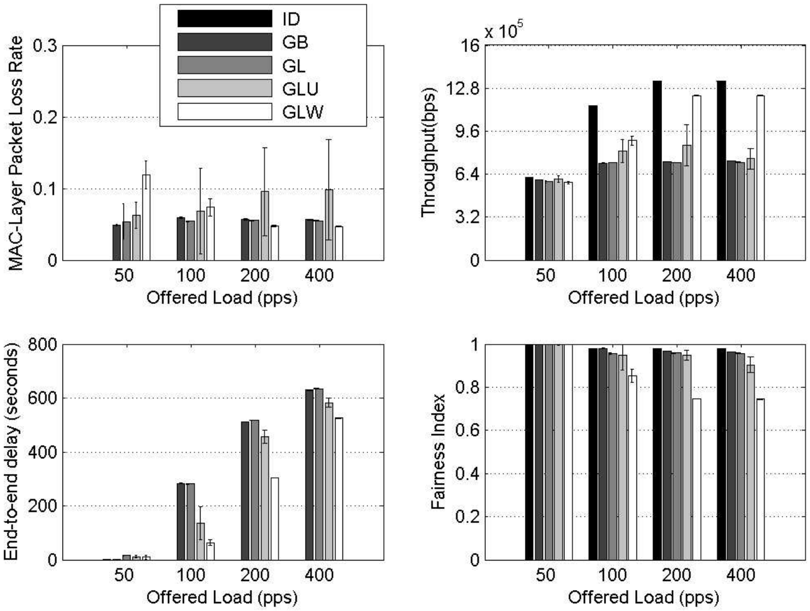

Figure 4.4: Network Aggregate Performance

with an increasing offered load. This is because both methods are driven by fairness, instead of offered load.

Second, Fig. 4.4 shows that the aggregate performance is consistent with our analysis of per-link allocation. The packet loss rate and network throughput of GB and GL do not change with increasing offered load when the network is saturated. The end-to-end delay increases significantly because the higher the offered load, the more packets are buffered. Therefore, the queueing delay increases exponentially. Furthermore, GLU and GLW exhibit opposite performance. Due to the greedy nature of GLU, the slight increase of network throughput leads to significant packet loss. On the other hand, GLW increases the throughput of the links experiencing less competition and therefore reduces packet loss and increases network throughput by sacrificing fairness. Furthermore, the fairness indices of GL, GB and GLU change slightly with increased offered load. GLW, on the other hand, achieves fairer allocation when the offered load is lighter. But with heavier traffic, the fairness index of GLW decreases much faster than GL, GB and GLU.

In summary, these experiments show that the SG-Local control achieves performance competitive with Chiang's global optimization. This performance is achieved by GL in a "purely" local manner without any message passing. In addition, this very encouraging result demonstrates that the SG-Local control achieves a desired trade-off between fairness and efficiency by combining the efforts of driving the system toward a configured fair share and minimizing resource waste caused by collisions and unnecessary idling.

Larger Random Networks

A key observation from the previous experiments is that GL achieves performance very close to GB without the cost of message passing. To further test this observation, we carry out a set of experiments that compare the performance of GL and GB in larger, randomly generated single-hop networks. We generate different network topologies by deploying 200 nodes in a 800-by-800 m^2 field and selecting source-destination pairs randomly.

The Impact of Interference Levels First, we test how interference level influences these two methods. The interference level is varied by changing the number of active source-destination pairs (also called links). We test networks with 10, 20 and then 30 links. Each experiment is replicated 16 times; each replication uses a distinct topology by selecting the required number of links randomly; each run spans 30 minutes, and all performance metrics are averaged after transients are removed. All other parameters were fixed, including the offered load (50 pps) and packet size (512 bytes). Fig. 4.5 shows the average performance and 95% confidence intervals for each control algorithm over all 16 replications.

We observe that the performance of both GL and GB degrades as the number of links increases. This is because the interference level is higher when the number of active links is larger. We note that GL is competitive with GB at all network sizes, and in terms of all four metrics of aggregate network performance: packet loss rate, throughput, end-to-end delay and fairness.

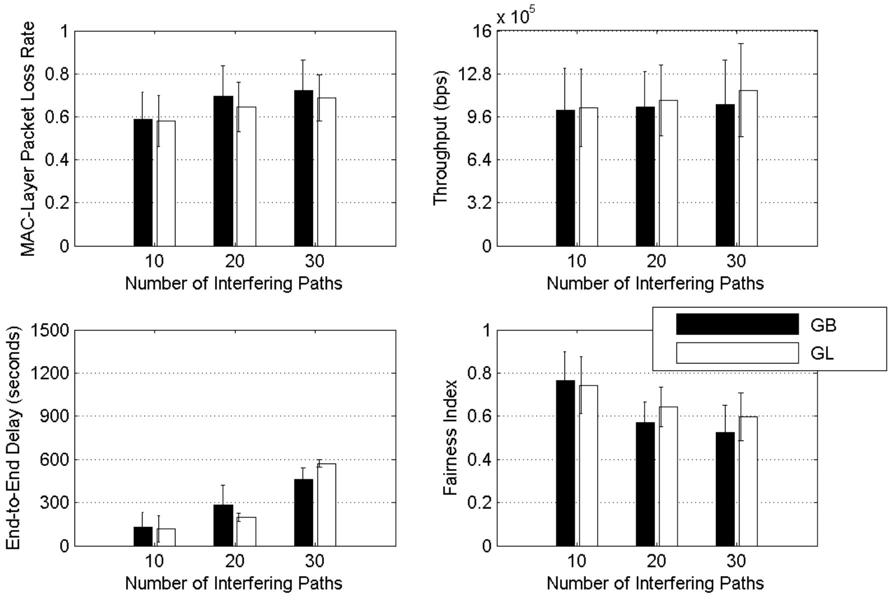

Figure 4.5: The Impact of Interference Level (single-hop)

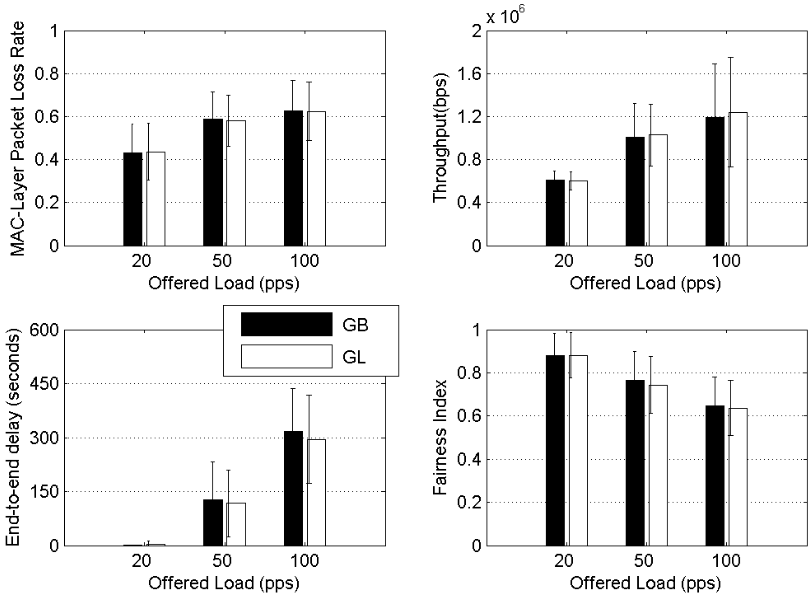

Figure 4.6: The Impact of Offered Load (single-hop)

The Impact of Offered Load Second, we test GL and GB under the traffic load of 50 pps, 100 pps and 200 pps. The results show that GL is again competitive with GB (see Fig 4.6).

Although SG-Local is tailored for single-hop wireless networks, we also measure its performance in random multi-hop networks. In these experiments, we varied offered load and

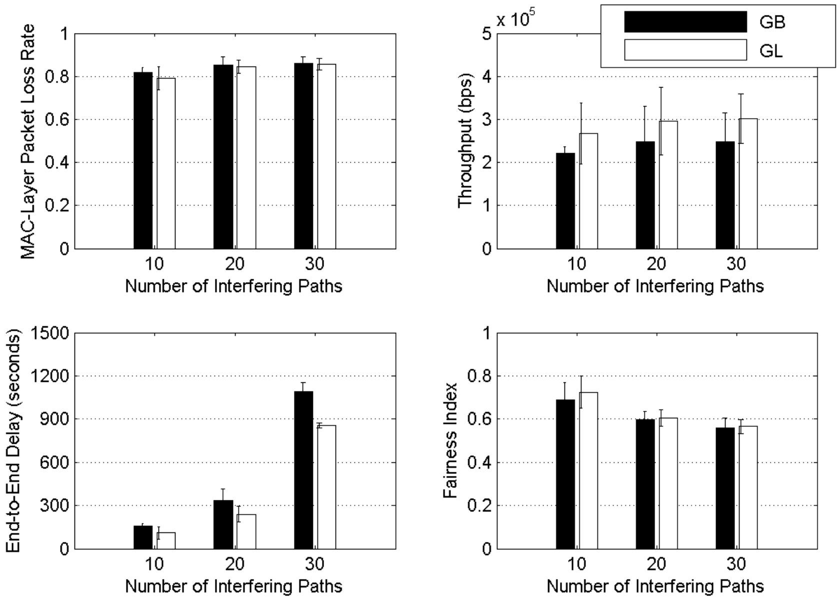

Figure 4.7: The Impact of Interference Level (multi-hop)

interference levels. Our results show that GL is once more competitive with GB. These results are shown in Fig. 4.8 and 4.7. In some cases, it appears that GL has higher throughput and lower delay, however, it is not clear how significant this is due to the variability caused by different topologies as evidenced by the confidence intervals. In any case, our results show that GL is once more competitive with Chiang's method. In Chapter 5, we explicitly customize G-Local optimization to handle multi-hop forwarding.

In summary, this second set of experiments further demonstrates the effectiveness of the SG-Local control. We find that GL is competitive with GB, while doing away with the need for any message passing.

4.2 G-Local Optimization with Multivariable Control

To further improve control effectiveness and minimize the allocation-utilization gap, we propose adaptive multivariable control. This method differentiates the causes of dynamic interference scenarios in time, space, and frequency-intensity; selects control variables that have major impact on each dimension of interference dynamics; and guides network transmissions, collisions and idling toward desired states by adjusting the selected control variables. To adapt to network condition changes, the multivariable model is updated periodically. We derive the MG-Local control by reifying G-Local optimization via this adaptive multivariable control.

Before presenting the details, we introduce a few symbols for the rest of this section in Table 4.4.

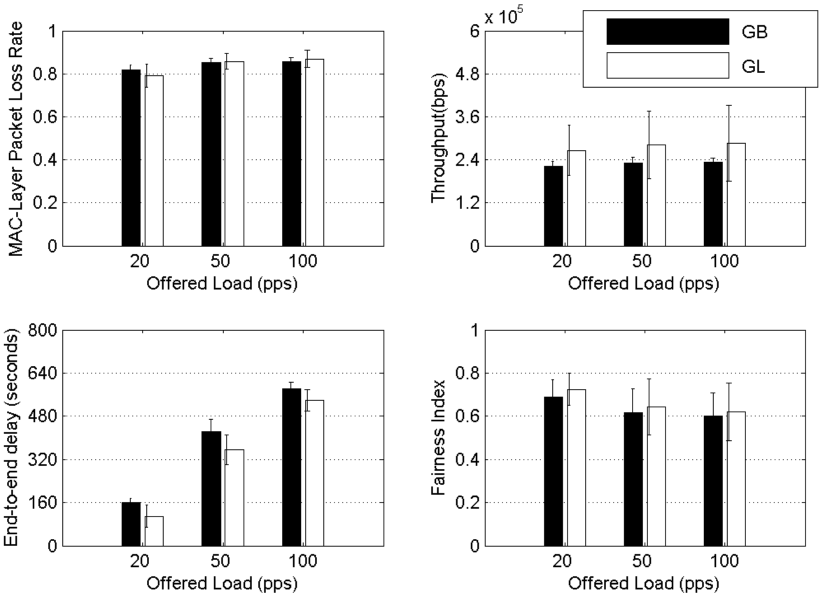

Figure 4.8: The Impact of Offered Load (multi-hop)

Table 4.4: MG-Local Symbols

Parameter	**Value**
(i, j)	link (i, j)
$x^s_{(i,j)}$	bandwidth consumed for transmissions over link (i, j)
$x^m_{(i,j)}(t)$	measured transmissions during the t-th control period
$x^f_{(i,j)}$	bandwidth fair share
$coll_{(i,j)}$	bandwidth wasted on collisions
$coll^m_{(i,j)}(t)$	measured collisions during the t-th control period
$idle_{(i,j)}$	bandwidth wasted on idling
$idle^m_{(i,j)}(t)$	measured idling during the t-th control period
$p_{(i,j)}$	transmission probability
$awin_{(i,j)}$	avoidance window
$rwin_{(i,j)}$	resolution window
B	channel bandwidth
k	scaler
λ	Lagrange multiplier or shadow price

4.2.1 Adaptive Multivariable Control

Our method effectively controls the occurrence of different interference scenarios in time, space, and frequency-intensity as analyzed in Chapter 1. To characterize the three basic network behaviors: transmissions, collisions and unnecessary idling, we measure the amount of bandwidth used for successful data transmissions (denoted $x_{(i,j)}$), wasted for packet collisions (denoted $coll_{(i,j)}$), and unnecessary idling (denoted $idle_{(i,j)}$) of user i. We model $x_{(i,j)}$, $coll_{(i,j)}$ and $idle_{(i,j)}$ as functions of three selected control variables, which are explained next.

4.2.2 Control Variable Selection

We modify CSMA/CA by introducing three control variables: transmission probability (p), collision-avoidance window ($awin$), and collision-resolution window ($rwin$). We chose CSMA/CA because it provides the basic structure to handle most interference scenarios. We avoided time-division-based scheduling because it requires accurate information regarding how nodes interfere over time and space, complex computation, and lacks flexibility to adapt to network changes.

1. **Transmission Probability (p)**
   The first variable controls the probability that a node transmits when physical carrier sensing detects a busy medium. p is manipulated to reduce the occurrence of both hidden and exposed terminals. These two scenarios coexist, and appear dynamically in wireless networks. It is very challenging to achieve accurate detection of both hidden and exposed terminals. For example, physical carrier sensing can detect hidden terminals by increasing its sensing range. However, at the same time, the increased carrier sensing range aggravates exposed terminals by treating all overheard signals as interference. To enable a node to control both hidden and exposed terminals, we fix the carrier sensing range to twice the transmission range to reduce hidden terminals, and allow node i to transmit with a probability $p_{(i,j)}$ to j when physical carrier sensing detects a busy medium to reduce exposed terminals. The transmission probability used by our method is different from the persistence probability of p-persistent CSMA, which is designed to reduce hidden terminals, but does not address exposed terminals. This is because after carrier sensing detects an idle medium, p-persistent CSMA allows a node to further backoff, instead of transmitting.

2. **Contention Avoidance Window ($awin$)**
   The second variable specifies the maximum number of slots that a node can randomly decide to wait before starting its transmission. We use $awin$ to avoid potential collisions caused by simultaneous transmissions. It is similar to the contention window $cwin_{(i,j)}$ of CSMA, except that $awin_{(i,j)}$ is only used when a packet is transmitted for the first time (the collision avoidance phase). During this phase, $awin_{(i,j)}$ should be set to allow potentially simultaneous transmissions to start at different times, as well as to avoid unnecessary waiting time. In contrast, CSMA/CA sets $cwin_{(i,j)}$ to its minimum value $CWIN\_MIN$ for collision avoidance, and exponentially increases $cwin_{(i,j)}$ upon each collision until either $cwin_{(i,j)}$ is greater than or equal to its maximum $CWIN\_MAX$, or the retransmission limit is reached. Using the single variable $cwin_{(i,j)}$ for both collision avoidance and resolution forces an unnecessary trade-off between collisions and delay. For example, if $cwin_{(i,j)}$ is set small because of the lim-

ited number of simultaneous transmissions, it may not be increased quickly enough to avoid repeated collisions. Conversely, if $cwin_{(i,j)}$ is set large to reduce the number of retransmissions, unnecessary waiting time may be incurred for collision avoidance. Because collision avoidance and resolution are two different phases, we introduce different control variables to avoid unnecessary trade-off between delay and collisions.

3. **Contention Resolution Window ($rwin$)**
   The third variable, $rwin$, is the contention window used to avoid repeated collisions of a packet. Besides separating collision resolution from avoidance, this variable also controls collisions caused by future transmissions. Although predicting interference from future transmissions is difficult, adjusting $rwin_{(i,j)}$ helps to prevent repeated collisions of the same packet.

Regression Model

To reify G-Local optimization via multivariable control, we model the state variables of its original form Eq. 4.34: $x_{(i,j)}$, $coll_{(i,j)}$ and $idle_{(i,j)}$, as functions of the selected control variables: $p_{(i,j)}$, $awin_{(i,j)}$ and $rwin_{(i,j)}$ via an approximation as shown in Eqs. 4.21, 4.22 and 4.23. This model considers the individual impact of $p_{(i,j)}$, $awin_{(i,j)}$ and $rwin_{(i,j)}$, as well as their correlation. We denote the coefficients of these three models as $E = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7\}$, $F = \{f_1, f_2, f_3, f_4, f_5, f_6, f_7\}$ and $L = \{l_1, l_2, l_3, l_4, l_5, l_6, l_7\}$.

$$\begin{aligned} x_{(i,j)} &= X(p_{(i,j)}, awin_{(i,j)}, rwin_{(i,j)}) \\ &= e_1 \cdot p_{(i,j)} + e_2 \cdot awin_{(i,j)} + e_3 \cdot rwin_{(i,j)} \\ &+ e_4 \cdot p_{(i,j)} \cdot awin_{(i,j)} + e_5 \cdot p_{(i,j)} \cdot rwin_{(i,j)} \\ &+ e_6 \cdot awin_{(i,j)} \cdot rwin_{(i,j)} \ + e_7; \end{aligned} \tag{4.21}$$

$$\begin{aligned} coll_{(i,j)} &= COLL(p_{(i,j)}, awin_{(i,j)}, rwin_{(i,j)}) \\ &= f_1 \cdot p_{(i,j)} + f_2 \cdot awin_{(i,j)} + f_3 \cdot rwin_{(i,j)} \\ &+ f_4 \cdot p_{(i,j)} \cdot awin_{(i,j)} + f_5 \cdot p_{(i,j)} \cdot rwin_{(i,j)} \\ &+ f_6 \cdot awin_{(i,j)} \cdot rwin_{(i,j)} \ + f_7; \end{aligned} \tag{4.22}$$

$$\begin{aligned} idle_{(i,j)} &= IDLE(p_{(i,j)}, awin_{(i,j)}, rwin_{(i,j)}) \\ &= l_1 \cdot p_{(i,j)} + l_2 \cdot awin_{(i,j)} + l_3 \cdot rwin_{(i,j)} \\ &+ l_4 \cdot p_{(i,j)} \cdot awin_{(i,j)} + l_5 \cdot p_{(i,j)} \cdot rwin_{(i,j)} \\ &+ l_6 \cdot awin_{(i,j)} \cdot rwin_{(i,j)} \ + l_7; \end{aligned} \tag{4.23}$$

Least Square Fitting

To derive the coefficients E, F and L of Eq. 4.21, Eq. 4.22 and Eq. 4.23, we apply the method of least square fitting [59]. Specifically, given N samples of measured responses

$\{x^m_{(i,j)}(t), coll^m_{(i,j)}(t), idle^m_{(i,j)}(t)\}$ and N corresponding sets of $\{p_{(i,j)}(t), awin_{(i,j)}(t), rwin_{(i,j)}(t)\}$, we formulate three optimization problems to find the best fitting models that have the least square errors as shown in Eq. 4.24, 4.25 and 4.26. In these equations, $p_{(i,j)}(t)$, $awin_{(i,j)}(t)$ and $rwin_{(i,j)}(t)$ are the t-th values of $p_{(i,j)}$, $awin_{(i,j)}$ and $rwin_{(i,j)}$; $x^m_{(i,j)}(t)$, $coll^m_{(i,j)}(t)$ and $idle^m_{(i,j)}(t)$ are the t-th sample of measured values of $x_{(i,j)}$, $coll_{(i,j)}$ and $idle_{(i,j)}$.

$$\begin{aligned} &\min \; J_x \;, \; where \\ J_x \; &= \frac{1}{N} \cdot \sum_{t=1}^{N} \left[(X(p_{(i,j)}(t), awin_{(i,j)}(t), rwin_{(i,j)}(t)) - x^m_{(i,j)}(t))^2 \right] \end{aligned} \tag{4.24}$$

$$\begin{aligned} &\min \; J_{coll} \;, \; where \\ J_{coll} \; &= \frac{1}{N} \cdot \sum_{t=1}^{N} \left[(COLL(p_{(i,j)}(t), awin_{(i,j)}(t), rwin_{(i,j)}(t)) - coll^m_{(i,j)}(t))^2 \right] \end{aligned} \tag{4.25}$$

$$\begin{aligned} &\min \; J_{idle} \;, \; where \\ J_{idle} \; &= \frac{1}{N} \cdot \sum_{t=1}^{N} \left[(IDLE(p_{(i,j)}(t), awin_{(i,j)}(t), rwin_{(i,j)}(t)) - idle^m_{(i,j)}(t))^2 \right] \end{aligned} \tag{4.26}$$

Please note that E, F and L are coefficients in our models. $X(p_{(i,j)}, awin_{(i,j)}, rwin_{(i,j)})$, $COLL(p_{(i,j)}, awin_{(i,j)}, rwin_{(i,j)})$ and $IDLE(p_{(i,j)}, awin_{(i,j)}, rwin_{(i,j)})$, and $x^m_{(i,j)}$, $coll^m_{(i,j)}$, $idle^m_{(i,j)}$ are known measurements. To obtain the least squared error, the coefficients must yield zero first derivatives as shown in Eq. 4.27. By solving this matrix using inversion, we derive the coefficients E, F and L.

$$\begin{gathered} \tfrac{\partial J_x}{\partial e_1} = 2 \cdot \sum_{t=1}^{n} p_{(i,j)}(t) \cdot \left[x^m_{(i,j)}(t) - X(p_{(i,j)}(t), awin_{(i,j)}(t), rwin_{(i,j)}(t)) \right] = 0; \\ \tfrac{\partial J_x}{\partial e_2} = 2 \cdot \sum_{t=1}^{n} awin_{(i,j)}(t) \cdot \left[x^m_{(i,j)}(t) - X(p_{(i,j)}(t), awin_{(i,j)}(t), rwin_{(i,j)}(t)) \right] = 0; \\ \tfrac{\partial J_x}{\partial e_3} = 2 \cdot \sum_{t=1}^{n} rwin_{(i,j)}(t) \cdot \left[x^m_{(i,j)}(t) - X(p_{(i,j)}(t), awin_{(i,j)}(t), rwin_{(i,j)}(t)) \right] = 0; \\ \tfrac{\partial J_x}{\partial e_4} = 2 \cdot \sum_{t=1}^{n} p_{(i,j)}(t) \cdot awin_{(i,j)}(t) \cdot \left[x^m_{(i,j)}(t) - X(p_{(i,j)}(t), awin_{(i,j)}(t), rwin_{(i,j)}(t)) \right] = 0; \\ \tfrac{\partial J_x}{\partial e_5} = 2 \cdot \sum_{t=1}^{n} p_{(i,j)}(t) \cdot rwin_{(i,j)}(t) \cdot \left[x^m_{(i,j)}(t) - X(p_{(i,j)}(t), awin_{(i,j)}(t), rwin_{(i,j)}(t)) \right] = 0; \\ \tfrac{\partial J_x}{\partial e_6} = 2 \cdot \sum_{t=1}^{n} awin_{(i,j)}(t) \cdot rwin_{(i,j)}(t) \cdot \left[x^m_{(i,j)}(t) - X(p_{(i,j)}(t), awin_{(i,j)}(t), rwin_{(i,j)}(t)) \right] = 0; \\ \tfrac{\partial J_x}{\partial e_7} = 2 \cdot \sum_{t=1}^{n} \left[x^m_{(i,j)}(t) - X(p_{(i,j)}(t), awin_{(i,j)}(t), rwin_{(i,j)}(t)) \right] = 0; \end{gathered} \tag{4.27}$$

Periodic Updating and Noise Processing

The coefficients in Eq. 4.21, 4.22 and 4.23 are periodically updated by replacing old measurements of $x_{(i,j)}$, $coll_{(i,j)}$, and $idle_{(i,j)}$ with fresh ones. We denote these measurements $x^m_{(i,j)}$, $coll^m_{(i,j)}$ and $idle^m_{(i,j)}$. These periodic updates help the method to capture nonlinear system behavior, and adapt to dynamic system conditions. We also apply an exponential filter to alleviate the negative impact of noise in the measurements [19]. According to Eq. 4.28, a new measurement of $x^m_{(i,j)}$ at the t-th control period (i.e. $x^m_{(i,j)}(t)$) is linearly combined with the previous measurement at time $t-1$ (i.e. $x^m_{(i,j)}(t-1)$). w is set to be 0.6 (an empirical value used in practice). Similarly, $coll^m_{(i,j)}$ and $idle^m_{(i,j)}$ are updated according to Eq. 4.29 and Eq. 4.30.

$$x^m_{(i,j)}(t) = w \cdot x^m_{(i,j)}(t) + (1-w) \cdot x^m_{(i,j)}(t-1); \tag{4.28}$$

$$coll^m_{(i,j)}(t) = w \cdot coll^m_{(i,j)}(t) + (1-w) \cdot coll^m_{(i,j)}(t-1); \tag{4.29}$$

$$idle^m_{(i,j)}(t) = w \cdot idle^m_{(i,j)}(t) + (1-w) \cdot idle^m_{(i,j)}(t-1); \tag{4.30}$$

We give the pseudo code of least square fitting with periodical updates in Alg. 1. This algorithm is applied to keep updating Eqs. 4.21,4.22 and 4.23 when it converges. For example, the mean square error of X varies within a threshold $\epsilon = 0.01$, which is experimentally determined. The reason we choose seven measurements for each iteration is because there are seven unknow coefficients.

Algorithm 1 Least Square Fitting With Periodical Updates

```
// This algorithm is executed for each link $(i,j)$

Initialize E, F and L for X(), COLL() and IDLE ()
with the full-factorial experiment results

for each control iteration $t$ do
  Collect the new measurements $x^m_{(i,j)}(t)$, $coll^m_{(i,j)}(t)$ and $idle^m_{(i,j)}(t)$
  Smooth $x^m_{(i,j)}(t)$ according to $w \cdot x^m_{(i,j)}(t) + (1-w) \cdot x^m_{(i,j)}(t-1)$
  Smooth $coll^m_{(i,j)}(t)$ according to $w \cdot coll^m_{(i,j)}(t) + (1-w) \cdot coll^m_{(i,j)}(t-1)$
  Smooth $coll^m_{(i,j)}(t)$ according to $w \cdot coll^m_{(i,j)}(t) + (1-w) \cdot coll^m_{(i,j)}(t-1)$
  if there are seven new measurements then
    Compute and update the coefficients E, F and L
    Compute the current mean square error $J_x$ according to Eq. 4.24
    if the different between the current and previous mean square errors is less than $\epsilon$
    then
      break;
    else
      continue;
    end if
  end if
end for
```

4.2.3 SPSA-Based Multivariable Control

We also introduce an estimation-based multivariable control. Instead of modeling network behavior explicitly, this method estimates the gradients of Eq. 4.21, 4.22 and 4.23 via the simultaneous perturbation stochastic optimization (SPSA) proposed by Spall [89]. With this method, all control variables are simultaneously perturbed twice in a random manner during each control iteration. This yields two measured responses, the difference between them (denoted δy), and the distance between the two perturbations of each variable (denoted $\delta\theta$). The gradient of each variable is estimated as $\frac{\delta y}{\delta\theta}$ [89]. Specifically, we estimate the gradients of X, $COLL$ and $IDLE$ according to Eq. 4.31, 4.32 and 4.33. We use $h1$ and $h2$ to denote the first and second perturbations.

The pseudo code of this algorithm is given in Alg. 2. During each control iteration t, random delta values are generated for $p_{(i,j)}$, $awin_{(i,j)}$ and $rwin_{(i,j)}$, denoted $\delta^{p}_{(i,j)}$, $\delta^{awin}_{(i,j)}$ and $\delta^{rwin}_{(i,j)}$ respectively. These perturbations are used to update control variables in two steps. First, at the beginning of the first half iteration, control variables $p_{(i,j)}$, $awin_{(i,j)}$ and $rwin_{(i,j)}$ are perturbed for the first time. a_t and c_t are the gain sequences, and their values are derived according to $a_t = \frac{a}{(t+A)^{\alpha}}$ and $c_t = \frac{c}{t^{\gamma}}$. a, c, A, α and γ are non-negative coefficients; Practically effective values for α and γ are 0.602 and 0.101 [89]. We set $a = 0.16$, $c = 0.1$ and $A = 100$ according to the suggestions given in the same paper. At the end of the first half iteration, we collect the measurements of $x^{m}_{(i,j)}(t,h1)$, $coll^{m}_{(i,j)}(t,h1)$ and $idle^{m}_{(i,j)}(t,h1)$. Similarly, the algorithm updates control variables with the second perturbations at the beginning of the second half iteration, and collects new measurements of $x^{m}_{(i,j)}(t,h2)$, $coll^{m}_{(i,j)}(t,h2)$ and $idle^{m}_{(i,j)}(t,h2)$ at the end of the second half iteration. The gradients of X, COLL, and IDLE are computed according to Eqs. 4.31, 4.32 and 4.33. The algorithm converges until $x^{m}_{(i,j)}$ varies within a threshold $\epsilon = 0.01$, which is also experimentally determined.

$$\begin{aligned}
\left(\frac{dX}{dp_{(i,j)}}\right)_{p_{(i,j)}=p_{(i,j)}(t)} &\approx \frac{x^{m}_{(i,j)}(t,h1) - x^{m}_{(i,j)}(t,h2)}{2 \cdot c_t \cdot \delta^{p}_{(i,j)}} \\
\left(\frac{dX}{dawin_{(i,j)}}\right)_{awin_{(i,j)}=awin_{(i,j)}(t)} &\approx \frac{x^{m}_{(i,j)}(t,h1) - x^{m}_{(i,j)}(t,h2)}{2 \cdot c_t \cdot \delta^{awin}_{(i,j)}} \\
\left(\frac{dX}{drwin_{(i,j)}}\right)_{rwin_{(i,j)}=rwin_{(i,j)}(t)} &\approx \frac{x^{m}_{(i,j)}(t,h1) - x^{m}_{(i,j)}(t,h2)}{2 \cdot c_t \cdot \delta^{awin}_{(i,j)}}
\end{aligned} \qquad (4.31)$$

$$
\begin{aligned}
\left(\frac{dCOLL}{dp_{(i,j)}}\right)_{p_{(i,j)}=p_{(i,j)}(t)} &\approx \frac{coll^m_{(i,j)}(t,h1) - coll^m_{(i,j)}(t,h2)}{2 \cdot c_t \cdot \delta^p_{(i,j)}} \\
\left(\frac{dCOLL}{dawin_{(i,j)}}\right)_{awin_{(i,j)}=awin_{(i,j)}(t)} &\approx \frac{coll^m_{(i,j)}(t,h1) - coll^m_{(i,j)}(t,h2)}{2 \cdot c_t \cdot \delta^{awin}_{(i,j)}} \\
\left(\frac{dCOLL}{drwin_{(i,j)}}\right)_{rwin_{(i,j)}=rwin_{(i,j)}(t)} &\approx \frac{coll^m_{(i,j)}(t,h1) - coll^m_{(i,j)}(t,h2)}{2 \cdot c_t \cdot \delta^{awin}_{(i,j)}}
\end{aligned}
\tag{4.32}
$$

$$
\begin{aligned}
\left(\frac{dIDLE}{dp_{(i,j)}}\right)_{p_{(i,j)}=p_{(i,j)}(t)} &\approx \frac{idle^m_{(i,j)}(t,h1) - idle^m_{(i,j)}(t,h2))}{2 \cdot c_t \cdot \delta^p_{(i,j)}} \\
\left(\frac{dIDLE}{dawin_{(i,j)}}\right)_{awin_{(i,j)}=awin_{(i,j)}(t)} &\approx \frac{idle^m_{(i,j)}(t,h1) - idle^m_{(i,j)}(t,h2))}{2 \cdot c_t \cdot \delta^{awin}_{(i,j)}} \\
\left(\frac{dIDLE}{drwin_{(i,j)}}\right)_{rwin_{(i,j)}=rwin_{(i,j)}(t)} &\approx \frac{idle^m_{(i,j)}(t,h1) - idle^m_{(i,j)}(t,h2))}{2 \cdot c_t \cdot \delta^{rwin}_{(i,j)}}
\end{aligned}
\tag{4.33}
$$

Algorithm 2 SPSA-based Multivariable Control

```
// This algorithm is executed for each link $(i, j)$

for each control iteration $t$ do
  Compute the gain sequences $a_t$ and $c_t$;
  Generate random $\delta values :\delta^p_{(i,j)}$, $\delta^{awin}_{(i,j)}$ and $\delta^{rwin}_{(i,j)}$;
  Compute the perturbations: $h1_p$ as $c_t \cdot \delta^p_{(i,j)}$; $h2_p$ as $-c_t \cdot \delta^p_{(i,j)}$;
                             $h1_{awin}$ as $c_t \cdot \delta^{awin}_{(i,j)}$; $h2_{awin}$ as $-c_t \cdot \delta^{awin}_{(i,j)}$
                             $h1_{rwin}$ as $c_t \cdot \delta^{rwin}_{(i,j)}$; $h2_{rwin}$ as $-c_t \cdot \delta^{rwin}_{(i,j)}$

  // At the beginning of the first half iteration
  Perturb $p_{(i,j)}$, $awin_{(i,j)}$ and $rwin_{(i,j)}$ from $p_{(i,j)}(t)$, $awin_{(i,j)}(t)$ and $rwin_{(i,j)}(t)$ with $h1^p_{(i,j)}$, $h1^{awin}_{(i,j)}$ and $h1^{rwin}_{(i,j)}$

  // At the end of the first half iteration
  Collect measurements: $x^m_{(i,j)}(t,h1)$, $coll^m_{(i,j)}(t,h1)$ and $idle^m_{(i,j)}(t,h1)$

  // At the beginning of the second half iteration
  Perturb $p_{(i,j)}$, $awin_{(i,j)}$ and $rwin_{(i,j)}$ from $p_{(i,j)}(t)$, $awin_{(i,j)}(t)$ and $rwin_{(i,j)}(t)$ with $h2^p_{(i,j)}$, $h2^{awin}_{(i,j)}$ and $h2^{rwin}_{(i,j)}$

  // At the end of the second half iteration
  Collect measurements: $x^m_{(i,j)}(t,h2)$, $coll^m_{(i,j)}(t,h2)$ and $idle^m_{(i,j)}(t,h2)$

  Compute the gradients according to Eqs. 4.31, 4.32 and 4.33.

  if the difference between $x^m_{(i,j)}(t,h1)$ and $x^m_{(i,j)}(t,h2)$ is less than $\epsilon$ then
    break;
  else
    continue;
  end if
end for
```

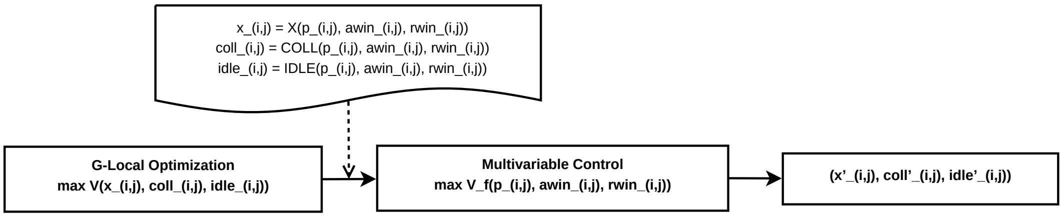
x_(i,j) = X(p_(i,j), awin_(i,j), rwin_(i,j))
coll_(i,j) = COLL(p_(i,j), awin_(i,j), rwin_(i,j))
idle_(i,j) = IDLE(p_(i,j), awin_(i,j), rwin_(i,j))
G-Local Optimization
max V(x_(i,j), coll_(i,j), idle_(i,j))
Multivariable Control
max V_f(p_(i,j), awin_(i,j), rwin_(i,j))
(x'_(i,j), coll'_(i,j), idle'_(i,j))

Figure 4.9: Flat MG-Local Control

4.2.4 Flat MG-Local Control

In this section, we describe how to implement the MG-Local control in a flat structure (flat MG-Local control). The basic idea of flat control is to find a desired network state via adjusting the selected control variables. As shown in Fig. 4.9, there is no intermediate state adjustment for the flat control. Specifically, this method transforms the original form of G-Local optimization Eq. 4.34 from a multi-state function of $x_{(i,j)}$, $coll_{(i,j)}$ and $idle_{(i,j)}$ to a multivariable function of $p_{(i,j)}$, $awin_{(i,j)}$ and $rwin_{(i,j)}$ as shown in Eq. 4.35. This transformation is derived by substituting $x_{(i,j)}$, $coll_{(i,j)}$ and $idle_{(i,j)}$ with Eq. 4.21, Eq. 4.22 and Eq. 4.23. We call Eq. 4.35 the flat MG-Local control.

$$\begin{aligned} &\max\ V(x_{(i,j)}, coll_{(i,j)}, idle_{(i,j)})\ ,\ where \\ V\ &=\ k \cdot \log x_{(i,j)} - (1-k) \cdot (\frac{1}{x^f_{(i,j)}} \cdot x_{(i,j)} + \frac{coll_{(i,j)}}{B} + \frac{idle_{(i,j)}}{B}) \end{aligned} \tag{4.34}$$

$$\begin{aligned} &\max\ V_f(p_{(i,j)}, awin_{(i,j)}, rwin_{(i,j)})\ ,\ where \\ V_f\ &=\ k \cdot \log(X(p_{(i,j)}, awin_{(i,j)}, rwin_{(i,j)})) \\ &- (1-k) \cdot (\frac{1}{x^f_{(i,j)}} \cdot X(p_{(i,j)}, awin_{(i,j)}, rwin_{(i,j)}) \\ &+ \frac{COLL(p_{(i,j)}, awin_{(i,j)}, rwin_{(i,j)})}{B} + \frac{IDLE(p_{(i,j)}, awin_{(i,j)}, rwin_{(i,j)})}{B}) \\ s.t.\ &0 < p_{(i,j)} \leq 1; \\ &AMIN \leq awin_{(i,j)} \leq AMAX; \\ &RMIN \leq rwin_{(i,j)} \leq RMAX; \end{aligned} \tag{4.35}$$

By applying the Lagrange transformation, the constrained optimization problem Eq. 4.35 is converted to an unconstrained format Eq. 4.36, where λ_{p1}, λ_{p2}, λ_{a1}, λ_{a2}, λ_{r1} and λ_{r2} are shadow prices. The corresponding dual problem is given in Eq. 4.39.

$$\begin{aligned} &L_f(p_{(i,j)}, awin_{(i,j)}, rwin_{(i,j)}, \lambda_{p1}, \lambda_{p2}, \lambda_{a1}, \lambda_{a2}, \lambda_{r1}, \lambda_{r2}) \\ &= V_f(p_{(i,j)}, awin_{(i,j)}, rwin_{(i,j)}) + \lambda_{p1} \cdot p_{(i,j)} - \lambda_{p2} \cdot (p_{(i,j)} - 1) \\ &\quad - \lambda_{a1} \cdot (AMIN - awin_{(i,j)}) - \lambda_{a2} \cdot (awin_{(i,j)} - AMAX) \\ &\quad - \lambda_{r1} \cdot (RMIN - rwin_{(i,j)}) - \lambda_{r2} \cdot (rwin_{(i,j)} - RMAX) \end{aligned} \tag{4.36}$$

$$\max L_f(p_{(i,j)}, awin_{(i,j)}, rwin_{(i,j)}, \lambda_{p1}, \lambda_{p2}, \lambda_{a1}, \lambda_{a2}, \lambda_{r1}, \lambda_{r2}) \tag{4.37}$$

$$D_f(p_{(i,j)}, awin_{(i,j)}, rwin_{(i,j)}, \{\lambda\}) = \max\ L_f \tag{4.38}$$

$$\min\ D_f(p_{(i,j)}, awin_{(i,j)}, rwin_{(i,j)}, \{\lambda\}) \tag{4.39}$$

We solve Eq. 4.37 by applying the method of gradient descent. The control policies for $p_{(i,j)}$, $awin_{(i,j)}$ and $rwin_{(i,j)}$ are derived in Eq. 4.40, Eq. 4.41 and Eq. 4.42. Each user applies this control policy periodically until the algorithm converges.

$$p_{(i,j)}(t) = p_{(i,j)}(t-1) + k_p \cdot \frac{\partial L_f}{\partial p_{(i,j)}} \tag{4.40}$$

$$awin_{(i,j)}(t) = awin_{(i,j)}(t-1) + k_a \cdot \frac{\partial L_f}{\partial awin_{(i,j)}} \tag{4.41}$$

$$rwin_{(i,j)}(t) = rwin_{(i,j)}(t-1) + k_r \cdot \frac{\partial L_f}{\partial rwin_{(i,j)}} \tag{4.42}$$

Finally, we solve the dual problem (Eq.4.39) to obtain the shadow prices. The solutions are listed in Eq. 4.43. The variables $\{h\}$ are scalars and their values are determined experimentally. We currently use 0.1 for all h scalars.

$$\begin{aligned}
\lambda_{p1} &= -h_1 \cdot p_{(i,j)} \\
\lambda_{p2} &= h_2 \cdot (p_{(i,j)} - 1) \\
\lambda_{a1} &= h_3 \cdot (AMIN - awin_{(i,j)}) \\
\lambda_{a2} &= h_2 \cdot (awin_{(i,j)} - AMAX) \\
\lambda_{r1} &= h_4 \cdot (RMIN - rwin_{(i,j)}) \\
\lambda_{r2} &= h_5 \cdot (rwin_{(i,j)} - RMAX)
\end{aligned} \tag{4.43}$$

4.2.5 Performance Evaluation

In this section, we evaluate the performance of MG-Local control (denoted AM in the figures) and compare it with four alternatives: 1) the SPSA-based multivariable control (denoted SP) [89]; 2) SG-Local control (denoted SVC); 3) Chiang's global optimization (denoted GB) [61]; and 4) CSMA/CA.

The SPSA-based multivariable control was introduced in Section 4.2.3. It derives unknown gradients via simultaneously perturbing all control variables.

Our SG-Local control of p-persistent ALOHA is the same method we introduced in Section 4.1 to reify G-Local optimization via single-variable control. This comparison helps to demonstrate the impact of control effectiveness. We did not apply the single-variable control method to CSMA/CA because its performance is influenced by multiple factors (e.g. contention window and physical carrier sensing range), as our previous full-factorial analysis illustrates [64]. It is obviously incorrect to attribute CSMA/CA performance to a single factor.

Chiang's global optimization is the same method used in the previous section, which aims to maximize network utility by adjusting the transmission probability with message passing. Each link adjusts local persistence probability based on the shadow price and the transmission probability of interfering nodes. This information is obtained via message passing among neighbor nodes.

CSMA/CA is used as a reference to examine whether the selected control variables behave as expected. Specifically, the separation of avoidance window ($awin_{(i,j)}$) and resolution window ($rwin_{(i,j)}$) is compared with the single contention window of CSMA/CA in terms of collision and delay. We further differentiate two CSMA/CA configurations. The first reduces hidden terminals by setting the physical carrier sensing range to twice the transmission range, and is denoted CSMA/CA-HT in Table 4.5.

Implementation

We apply MG-Local to IEEE 802.11b in ns2 by modifying its basic access method CSMA/CA. First, we add three control variables: per-link transmission probability (p), per-link avoidance window ($awin$) and per-link resolution window ($rwin$) to CSMA/CA. Second, in this modified CSMA/CA if a node i has a packet to send to j, it starts backing off with a probability $p_{(i,j)}$ for a random period, and waits for the propagation delay of one packet with a probability $(1 - p_{(i,j)})$. If it is the first time to transmit a packet, the back-off period is selected within the avoidance window $awin_{(i,j)}$. Upon a retransmission, resolution window $rwin_{(i,j)}$ is used to generate back-off period. Third, $p_{(i,j)}$, $awin_{(i,j)}$ and $rwin_{(i,j)}$ are updated periodically according to Eq. 4.40, 4.41 and 4.42 until MG-Local converges. This update requires each node to periodically measure the data sending rate ($x^m_{(i,j)}$), data collision rate ($coll^m_{(i,j)}$), and unnecessary idling ($idle^m_{(i,j)}$). $x^m_{(i,j)}$ is calculated via dividing the number of transmitted packets by the length of a control period. $coll^m_{(i,j)}$ is calculated via dividing the product of the number of unacknowledged packets and retry limit by the length of a control period. $idle^m_{(i,j)}$ is calculated as the difference between data sending rate $x^m_{(i,j)}$ and bandwidth fair share $x^f_{(i,j)}$, representing the portion of bandwidth that should be utilized, but is not. These measurements are used to derive the coefficients (E, F and L) of our multi-variable functions $X(p_{(i,j)}, awin_{(i,j)}, rwin_{(i,j)})$ (Eq. 4.21), $COLL(p_{(i,j)}, awin_{(i,j)}, rwin_{(i,j)})$ (Eq. 4.22)and $IDLE(p_{(i,j)}, awin_{(i,j)}, rwin_{(i,j)})$ (Eq. 4.23) via least square fitting (Alg. 1).

MG-Local can support dynamic traffic with a simple modification, by requiring the iterative least square fitting to keep tracking the difference between estimated and actual performance after its convergence. If the gap is greater than a certain threshold which needs to be experimentally determined, the fitting algorithm notifies the system about this change and starts updating the models (X, COLL and IDLE) until they converge again. The system invokes MG-Local to adapt to this traffic change. We leave the performance study of how MG-Local behaves under dynamic traffic to the future work.

Experiments

MG-Local control is implemented on top of IEEE 802.11b with the CSMA/CA mode in ns2. Similar to our previous experiments for single-variable control, we improve simulation accuracy by adding a SINR-based collision model with aggregate interference, and a preamble detection function to the IEEE 802.11 implementation in ns2. Experiments are evaluated according to fairness, efficiency, effectiveness and delay.

1. Fairness: We measure the fairness of resource allocation via Jain's fairness index. This is calculated as $f(x) = \frac{(\sum x_{(i,j)})^2}{n \cdot \sum x_{(i,j)}^2}$, and ranks the fairness of a resource allocation strategy between 0 and 1. Higher values indicate fairer allocation.

2. Efficiency: We measure the efficiency of resource utilization using the aggregate throughput. Throughput is the total rate at which packets are successfully delivered to destinations in the network.

3. Effectiveness: We measure the control effectiveness via the packet loss rate. Packet loss rate is the percentage of packets that are lost due to collisions.

4. Delay: We measure the average time from packet generation to successful reception, for packets that are not lost due to collision.

Similar to the single-variable experiments, we use two different network topologies. The first topology is a 3-link network, and is designed to compare all five methods. The second set of experiments uses larger, randomly-generated networks to evaluate MG-Local control under the impact of different traffic load and interference levels. By varying the offered load and number of source-destination pairs in each network. We replicate each experiment 16 times, and show the average performance and 95% confidence intervals. A Poisson model is used to generate traffic. All traffic flows are assumed to have the same bandwidth demands.

System parameters are categorized into fixed, changing, and control variables. Fixed parameters are: a packet size of 512 bytes, transmit power of 0.2818 Watt, transmission range of 200 meters, physical carrier sensing range of 400 meters, and running time of 30 minutes. Changing parameters are offered load and interference level. Control variables are the transmission probability $p_{(i,j)}$, collision avoidance window $awin_{(i,j)}$ and resolution window $rwin_{(i,j)}$.

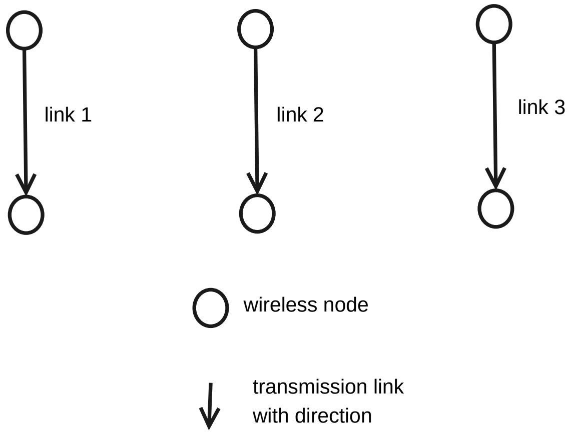

Figure 4.10: 3-Link Network Topology

Table 4.5: Per-Link Allocation

	link 1(pps)	**link 2(pps)**	**link 3(pps)**
ID	122.26	81.38	122.26
MG-AMVC	116.32	61.68	114.54
MG-SPSA	120.71	25.41	121.02
MG-SVC	67.23	44.45	67.35
GB	68.93	42.65	69.21
CSMA/CA-ET	182.10	1.02	182.09
CSMA/CA-HT	65.37	55.55	65.30

Three-Link Networks

With the 3-link topology shown in Fig 4.10, we derive the configured fair share according to the default fairness criteria. Similar to the single-variable experiments in Section 4.1.2, the ideal fair share (denoted ID) for link 1, 2 and 3 are $\frac{B}{2}$, $\frac{B}{3}$ and $\frac{B}{2}$.

Comparative Study The first experiment fixes the offered load for each transmission pair. The selected offered load is 200 pps, and saturates the whole network. That is, the total offered load on each link exceeds its ideal fair share. Our comparative results for the per-link allocation, and aggregate network performance are shown in Table 4.5, and Figure 4.11, respectively.

First, our MG-Local control (AM in Fig. 4.11) achieves a close-to-ideal allocation. As shown in Table 4.5, the ideal allocation for links 1, 2, and 3 are approximately 122 pps, 81 pps and 122 pps. Our method achieves 116 pps for link 1, 61 pps for link 2, and 114 pps for link 3. Fig. 4.11 shows that the fairness index of MG-Local control (denoted AM) is very close to the ideal case. In addition, compared to the other four alternatives, our MG-Local control utilizes resources more efficiently. This is demonstrated in Fig 4.11, where the gap between our method and the ideal case is the smallest, for both aggregate throughput and packet loss rate.

Second, compared to the SPSA-based multivariable control (SP in Fig. 4.11), MG-Local control achieves a fairer allocation and higher utilization of resources. As shown in Table 4.5, the per-link allocation of the estimation-based control is 120 pps for link 1, 25 pps for link 2, and 121 pps for link 3. Its low throughput on link 2 has a negative impact on both aggregate throughput and fairness. The simultaneous perturbation has an inferior performance to our method because of its slow convergence.

Third, compared to our SG-Local control (SVC in Fig. 4.11), our MG-Local control achieves a similar level of fairness as reflected by Jain's fairness index, and a much higher aggregate throughput. Moreover, both the SG-Local control, and SPSA-based control have a higher packet loss rate than our multivariable method. The low resource utilization of SG-Local control shows that it has a limited ability to deal with the complicated network behavior, and demonstrates the need for multivariable control.

Fourth, Chiang's distributed global-optimization (GB in Fig. 4.11) achieves similar results to the SG-Local control (SVC in the figure), but at the cost of message passing. It is also a single-variable control, and results in lower throughput and higher packet loss than our MG-Local control.

Fifth, CSMA/CA-ET aims to prevent links 1 and 3 from becoming exposed terminals by using a smaller carrier sensing range. However, due to the inferior location of link 2, and the unfair nature of CSMA/CA [102], the excellent performance of link 1 and 3 comes at the expense of starving link 2. In contrast, CSMA/CA-HT sets the carrier sensing range to twice the transmission range to avoid hidden terminals. Although the fairness is improved over CSMA/CA-HT, links 1 and 2 waste resources because they are exposed terminals. In comparison, our MG-Local control succeeds in handling both hidden and exposed terminals.

In summary, for the 3-link case, MG-Local control outperforms the other candidates, and achieves effective resource management, improving both allocation fairness and utilization efficiency. It does this without message-passing.

The Impact of Offered Load In the previous experiment, we used a fixed offered load of 200 pps. We now examine results from using different levels of traffic load, from light (50 pps) and unsaturated (100 pps), to saturated (200 pps), and over-saturated (400 pps). The result of aggregate performance is shown in Fig.4.12. These results show that MG-Local control maintains its performance advantage at different levels of offered load. Our previous observation with an offered load of 200 pps holds, even when the network is over-saturated with an offered load of 400 pps.

When the traffic load is low (50 pps for each link), all methods achieve similar results because user demands for resources can be satisfied easily. However, at higher load, the advantage of MG-Local control becomes obvious. Our method achieves lower packet loss

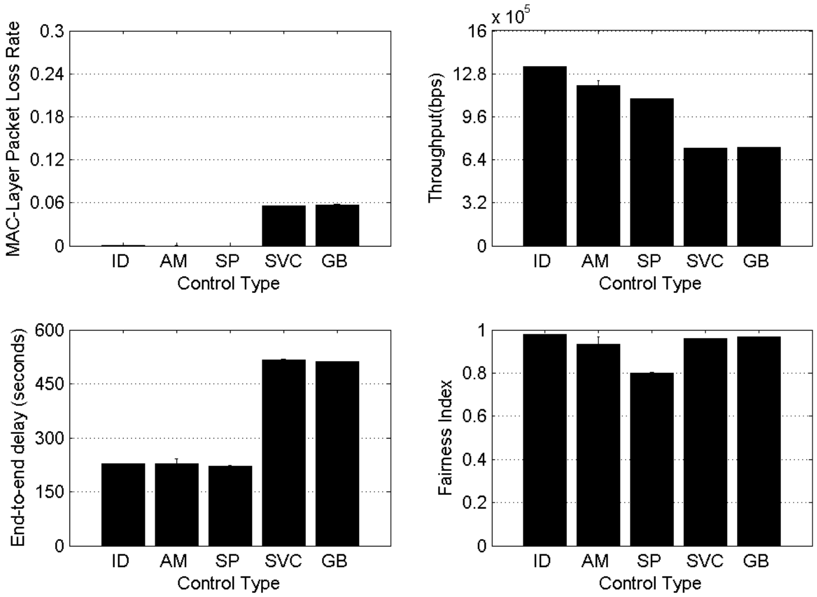

Figure 4.11: Aggregate Performance Comparison

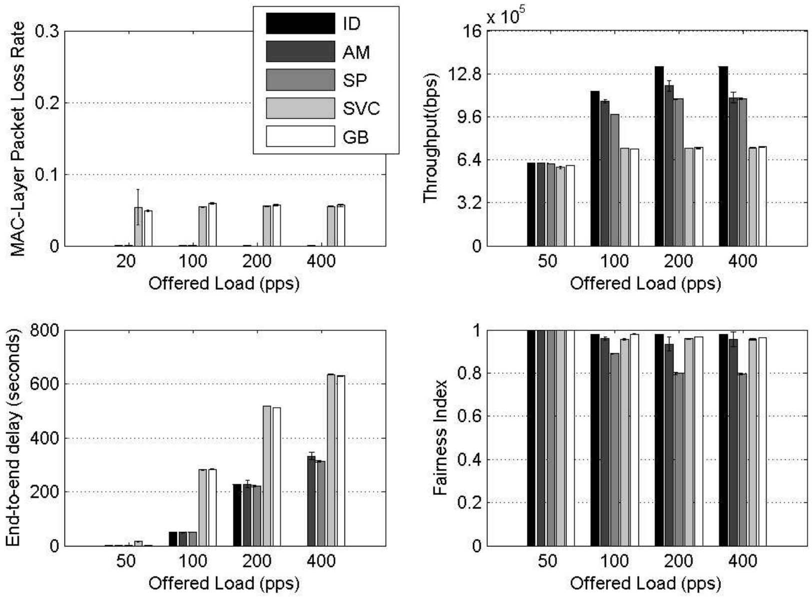

Figure 4.12: Impact of Offered Load (3-Link Network)

rate and end-to-end delay, and higher aggregate throughput and fairness. This trend holds even when the offered load increases to over-saturate the network.

Large Random Networks

The second set of experiments evaluates MG-Local control in larger, randomly generated single-hop networks. We create different network topologies by deploying 200 nodes in a 800-by-800 m^2 field and selecting source-destination pairs randomly.

The Impact of Interference Levels The first experiment tests the influence of interference levels. To generate different interference levels, we vary the network size by creating 10, 20 and 30 source-destination pairs. The fixed parameters include the offered load (50 pps) and packet size (512 bytes). Fig. 4.13 shows the average performance and 95% confidence intervals for each control algorithm, over all replications.

Fig. 4.13 shows that MG-Local control (AM in Fig. 4.13) achieves excellent performance consistent with the three-link experiment. At each interference level, our method achieves the best trade-off between throughput and fairness, without compromising delay. Even when the interference level is the highest, at 30 links, MG-Local control yields much higher throughput and shorter delay than the other candidates, while achieving a similar fairness index to the others.

Our MG-Local control is a significant improvement over CSMA/CA, due to the transmission probability $p_{(i,j)}$ alleviating the exposed-terminal problem. We attribute the reduction of packet loss rate to the separate adjustment of the avoidance and resolution windows $awin_{(i,j)}$ and $rwin_{(i,j)}$. Also, the removal of exponentially increasing $cwin_{(i,j)}$ helps to reduce the end-to-end delay. In contrast, the SG-Local control (SVC in Fig. 4.13) and Chiang's distributed global-optimization (GB in Fig. 4.13) only adjust the transmission probability. Although exposed terminals can be alleviated, without appropriate adjustment of other factors (e.g. the contention window), these methods increase the occurrence of hidden terminals. As a result, they suffer from significant packet loss.

The Impact of Offered Load The second experiment tests performance at offered load of 50 pps, 100 pps, and 200 pps. The results show that our MG-Local control again outperforms other alternatives. It achieves more efficient utilization, and fairer allocation (see Fig 4.14).

With an increasing offered load, the gap between the throughput of MG-Local control (AM in Fig. 4.14) and that of other candidates is more distinct, without trading off delay. In terms of fairness, the worst fairness index achieved by our method is as good as the SG-Local control (SVC in the figure) and Chiang's distributed global-optimization (GB in the figure). MG-Local achieves higher throughput, lower packet loss rate, and shorter delay.

In summary, the performance study demonstrates that MG-Local control improves control effectiveness by reducing the allocation-utilization gap, while balancing the efforts between driving consumption toward the configured fair share and minimizing resource waste

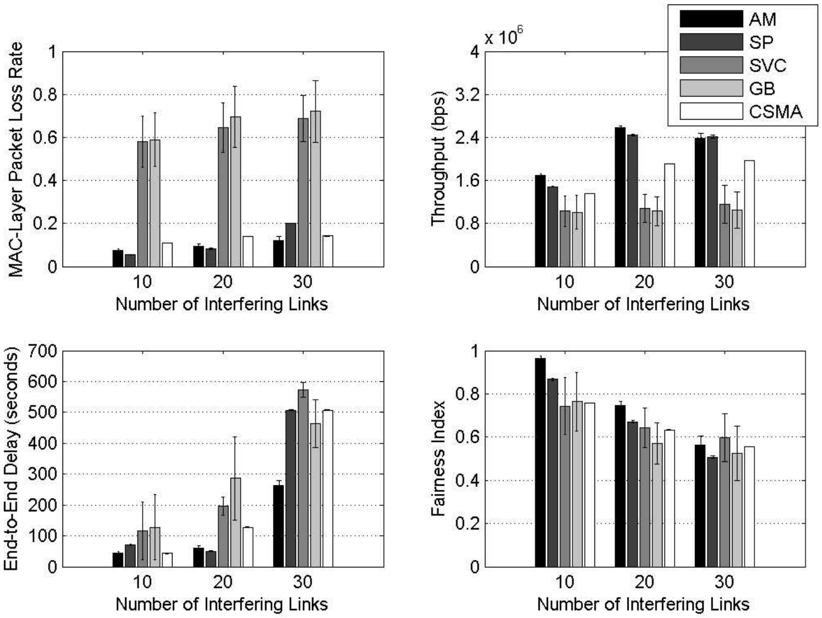

Figure 4.13: Impact of Interference Level (Random 1-Hop Network)

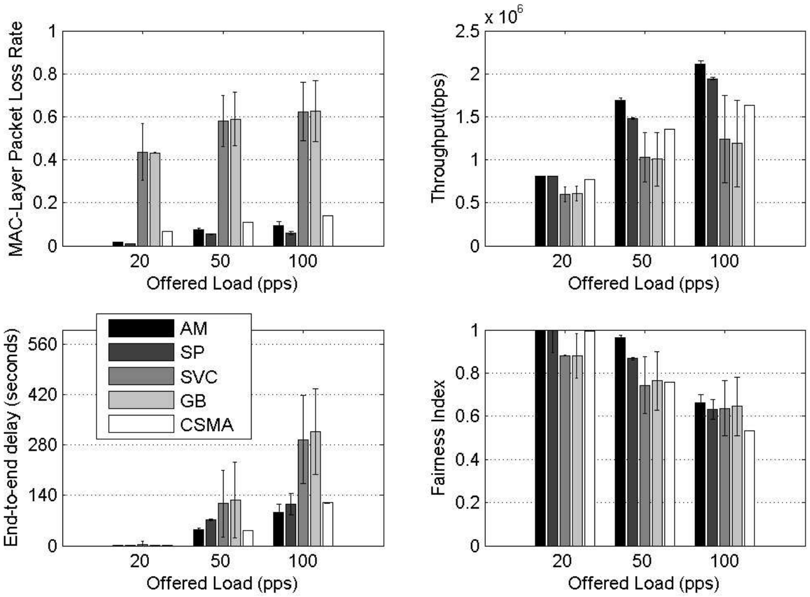

Figure 4.14: Impact of Offered Load (Random 1-Hop Network)

caused by collisions and unnecessary idling. Similar to the SG-Local control, MG-Local does not require any message passing.

4.3 Chapter Summary

In this chapter, we reify G-Local optimization via two control methods: single-variable and multivariable control. The SG-Local control balances the efforts of achieving the configured fair share and minimizing collisions via tuning a single control variable. It achieves very competitive performance in comparison to a classic method of global optimization. Additionally, this competitive performance is gained with zero message passing.

To improve control effectiveness, we introduce a multivariable control. This method differentiates the causes of interference and controlling network behavior via tuning multiple control variables. Furthermore, the adaptive multivariable control improves CSMA/CAs capability to handle both hidden and exposed terminals. Consequently, MG-Local control significantly reduces resource waste and achieves a much smaller allocation-utilization gap than the SG-Local control, besides supporting different fairness criteria. These benefits are achieved without message passing. We compare MG-Local with four other methods, and the experiment results show that MG-Local significantly outperforms all four alternatives.

Chapter 5

Resource Management in Multi-Hop Wireless Networks

So far, we have discussed how to apply G-Local optimization in single-hop wireless networks with both single-variable and multivariable control. Although G-Local was evaluated in multi-hop networks in the experiments of the previous chapter, we have not optimized its design for multi-hop forwarding yet. Compared to single-hop forwarding, wireless multi-hop forwarding offers ubiquitous and inexpensive Internet access within a wider coverage area. This method propagates traffic over a long distance via a path composed of multiple geographically distributed nodes. A multi-hop network is therefore abstracted as a collection of paths that forward traffic for a number of source and destination pairs. However, multi-hop forwarding significantly degrades network performance due to 1) co-existing intra-path and inter-path interference; 2) uncontrolled bandwidth competition; and 3) correlated congestion and collisions.

Challenges The first challenge of multi-hop forwarding is that it aggravates the impact of wireless interference on network performance. As elaborated in Chapter 1, wireless interference is dynamic in time, space, signal strength and frequency. With multi-hop forwarding, these dynamic interference scenarios have pronounced impact on transmission within the same path and between different paths, and are abstracted as intra-path and inter-path interference.

The second challenge is that multi-hop forwarding significantly reduces effective channel bandwidth that can be used for successful transmissions because of aggravated interference, and leads to chaotic resource competition between different paths, and within the same path. For example, depending on routing topologies and traffic patterns, a multi-hop path may compete with different paths along its trajectory. Additionally, intra-path interference transforms expected coordination among nodes within the same path into unwanted competition. Such uncontrolled competition can lead to unfair and inefficient resource consumption. Without carefully managing the scarce bandwidth resource and handling complicated competition scenarios, it is impossible to provide any level of performance guarantee.

The third challenge is that multi-hop forwarding leads to correlated congestion and collisions. Interfering transmission causes network congestion, while congestion returns the favor by aggravating collisions. On one hand, when collisions occur, the effective channel capacity is reduced because of the bandwidth wasted on failed transmissions and retransmissions. Consequently, the reduced capacity leads to slower service rates and aggravated congestion. On the other hand, when congestion occurs, a node attempts to keep pushing the backlogged traffic and causes more frequent interference. As a result, collisions occur more often.

In this chapter, we develop a novel solution by customizing G-Local optimization to tackle chaotic multi-hop forwarding.

- First, we offer hybrid TDMA/CSMA (time division medium access / carrier sense medium access) for medium access control that effectively handles co-existing intra-path and inter-path interference. This method differentiates the causes of various interference scenarios, and uses appropriate techniques to design counter solutions. Our results demonstrate that hybrid TDMA/CSMA significantly reduces collisions. Furthermore, this algorithm is simple in computation, light in control and adaptable to time-varying traffic patterns.

- Second, we provide hierarchical MG-Local control to effectively control chaotic competition for limited bandwidth within the same path and between different paths. This mechanism enforces fairness-driven resource allocation, achieves efficient utilization of allocated bandwidth, and effectively controls network behavior on top of hybrid TDMA/CSMA. Our results show improved fairness and significantly reduced collisions without compromising network throughput. Furthermore, this mechanism makes local decisions with zero message passing.

- Third, we develop a novel method of correlated congestion-collision control that explicitly handles the mutual impact of congestion and collisions in wireless multi-hop networks. This method computes a congestion cost based on its incoming/outgoing traffic, and local interference level. This congestion cost is used by a hop-based control method to reduce congestion. Our results demonstrate that this method significantly reduces end-to-end delay and packet loss.

In the rest of this chapter, we first present hybrid TDMA/CSMA in Section 5.1, then describe hierarchical MG-Local control in Section 5.2. Section 5.3 introduces correlated congestion-collision control. Performance evaluation of the complete solution is given in Section 5.4. Finally, we conclude this chapter by summarizing our contributions in Section 5.5.

5.1 Hybrid TDMA/CSMA

Hybrid TDMA/CSMA serves as the medium access control layer, and aims to effectively manage co-existing intra-path and inter-path interference in multi-hop wireless networks. Our method uses a novel TDMA-like scheduling to take advantage of the predictable scenarios of intra-path interference, and coordinates transmissions within the same path to avoid collisions. This scheduling algorithm is combined with a slotted CSMA-like contention control to handle various types of inter-path contention.

Hybrid TDMA/CSMA consists of two parts: 1) intra-path scheduling; and 2) inter-path contention control. Before presenting the details, we introduce a number of symbols, terms and prerequisites.

- $p_{(s,d)}$ represents a path, where s and d are the identifications of the source and destination node.

- $hop\_count^i_{(s,d)}$ is the number of nodes to forward traffic from source s to i for $p_{s,d}$.

- $prev\_hop^i_{(s,d)}$ is the previous hop of node i on $p_{(s,d)}$.

- $next\_hop^i_{(s,d)}$ is the next hop of node i on $p_{(s,d)}$.

- $seed_{(s,d)}$ is a unique seed used to generate random numbers. It is shared by all nodes on the same path $p_{(s,d)}$ to resolve schedule conflicts.

- num_paths_i is a local variable at a common node i, which keeps the number of paths that share node i. A common node receives data from and/or forwards data for multiple paths.

- $num\_share^i_{(s,d,j)}$ is a local record at node i of the number of paths that share a common node j $(i \neq j)$. Both node i and j are neighbors within two hops, and forward data for path $p_{(s,d)}$.

- $num\_share\_map_i$ is a local map at node i that records all $num\_share^i_{(s,d,j)}$. We use (s, d, j) as the key, because a path may share different common nodes with different paths.

- $max\_num\_share_i$ is the current largest value that is recorded in $num\_share\_map_i$.

We assume that the above information is initialized before the scheduling algorithm starts via the following method. First, $hop\_count^i_{(s,d)}$, $prev\_hop^i_{(s,d)}$, $next\_hop^i_{(s,d)}$ and $seed_{(s,d)}$ can be easily obtained by sending an initialization message along each path $p_{(s,d)}$ after the routing phase. Node i keeps monitoring the number of paths it forwards/receives data for/from, and updates num_path_i when a change occurs. The common node map $num\_share\_map_i$

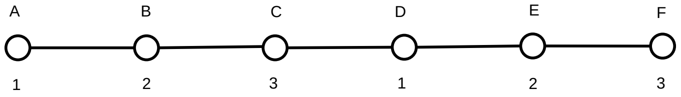

Figure 5.1: Intra-Path Interference

is initially empty. We describe how to update $num\_share^{i}_{(s,d,j)}$, $num\_share\_map_i$ and $max\_num\_share_i$ for both fixed and dynamic traffic in Section 5.1.2.

5.1.1 Intra-Path Scheduling

When traffic is forwarded in a multi-hop manner, the transmissions between different nodes along the same path may interfere. For instance, if node A and C transmit at the same time for $p_{A,F}$, collisions will happen at B (see Fig. 5.1). Intra-path scheduling aims to minimize collisions caused by such self interference. This algorithm takes advantage of the predictable self-obstructive conflicts within a scope of three hops, and enables a node i to infer its path-wise transmission schedule $sched^{i}_{(s,d)}$ locally. Specifically, if $hop\_count^{i}_{(s,d)}$ can be divided by 3, then $sched^{i}_{(s,d)}$ equals 3; otherwise, $sched^{i}_{(s,d)}$ equals $hop\_count^{i}_{(s,d)}\%3$. For example, $sched^{i}_{(s,d)}$ for A, B, C, D, E, and F are 1, 2, 3, 1, 2 ,3 (Fig. 5.1). These schedules eliminate collisions caused by intra-path interference, if the routing algorithm guarantees that nodes beyond two hops on the same path do not interfere with each other. Otherwise, our algorithm reduces intra-path collisions by guaranteeing that no self-obstructive transmissions occur within every three hops, and leaves interference beyond two hops away to be handled by a slotted CSMA/CA-like algorithm (see Section 5.1.2). Compared with graph-based TDMA scheduling [25] [81], our intra-path scheduling requires no message passing to infer local schedules. Furthermore, this algorithm transforms unwanted competition within the same path to coordinated transmissions.

5.1.2 Inter-Path Contention Control

Compared to intra-path interference, transmission conflicts between different paths is less predictable in terms of when they occur and where the interference comes from. Depending on routing topology and traffic pattern, the nodes along a multi-hop path may experience different levels and scenarios of interference. We categorize inter-path interference into general and special types. By general interference we mean that adjacent transmissions between different paths interfere with each other. General interference is further differentiated into various temporal and spatial scenarios. The Special interference scenario occurs when different paths share a common node. In this section, we first present special conflict resolution, and then describe our method for general contention control.

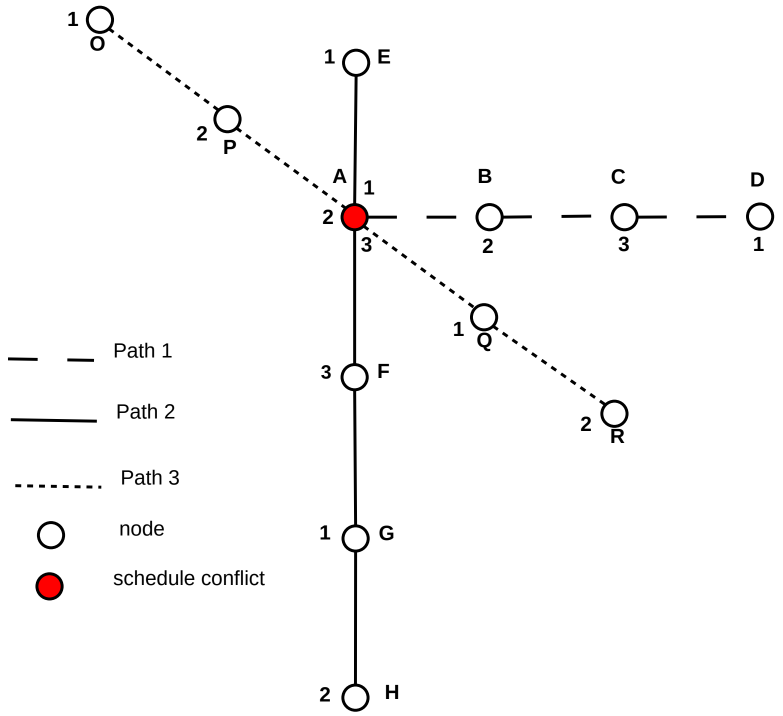

Figure 5.2: Common Node Scheduling with Fixed Traffic

Special Conflict Resolution

When multiple paths share the same common node, transmission conflicts are unavoidable. With the intra-path scheduling, these transmission conflicts become predictable schedule conflicts. For example, as shown in Fig. 5.2, node A has three scheduled slots: slot 1 for path $p_{(A,D)}$, slot 2 for $p_{(E,H)}$, and slot 3 for $p_{(O,R)}$. If node A always transmits according to these schedules, severe collisions are inevitable within its two hop range. Our method keeps existing schedules derived from intra-path scheduling without modification, and resolves schedule conflicts by coordinating paths to equally share their conflicting schedules via randomization. This resolution algorithm includes two parts: controlled broadcasting and randomized schedule sharing.

Controlled Broadcasting First, controlled broadcasting aims to update the largest number of paths that node i has conflicting schedules with, $max\_num\_share_i$. Because a node may be within the two hop range of more than one common node, we keep a map $num\_share\_map_i$ at node i to record these common nodes and the number of paths they forward data for and/or receive data from. Because schedule conflicts only occur within the two hop range of each common node according to intra-path scheduling, we use a two hop broadcast to reduce overhead.

In the case of fixed traffic, this controlled broadcasting is invoked during system initialization after routing. Specifically, a common node j broadcasts a message that includes num_paths_j and the identifications of all the paths that share j. To achieve controlled broadcast, a counter field is included in the broadcast message. This counter is initially set to two by a common node, and is updated by subtracting one each time it is received. After receiving this broadcast, a neighbor i first checks whether it belongs to a path $p_{(s,d)}$ that is included in the message, and either $prev\_hop^i_{(s,d)} = j$ or $next\_hop^i_{(s,d)} = j$ is satisfied. If these conditions are satisfied, i creates a record of $num\_share^i_{(s,d,j)} = num\_paths_j$ for j; inserts it to $num\_share\_map_i$; subtracts the value of the counter field by one. If the updated counter value equals one, i broadcasts this message. Otherwise, i discards the message.

For dynamic traffic, the same algorithm is invoked when a new/old path joins/leaves the network. By monitoring local traffic, an affected common node i detects these changes, updates num_path_i, and starts controlled broadcasting. An affected neighbor j updates the common node map $num\_share\_map_j$ and $max\_num\_share_j$ upon receiving such a message.

Randomized Schedule Sharing The second part of special conflict resolution aims to share conflicting schedules via randomization. Specifically, we assign each node $p\_share^i_{(s,d)}$, the schedule probability that node i transmits during its scheduled slot for path $p_{(s,d)}$. If $max\_num\_share_i \geq 1$, the transmission schedule of node i for path $p_{(s,d)}$ is conflicting with $(max\_num\_share_i - 1)$ other paths, and should share this conflicting schedule with a probability of $p\_share^i_{(s,d)} = 1/max\_num\_share_i$. Otherwise, node i has no conflicting schedules with any path, and always transmits during its scheduled slot. Therefore, $max\_num\_share_i$ allows randomization among multiple paths that share a common node to reduce conflict. Furthermore, we use a unique $seed_{(s,d)}$ for each path that allows intra-path scheduling to keep track of when to transmit, and therefore guarantees that during each round of slot 1, 2 and 3, the common node and its two hop intra-path nodes make non-conflicting decisions regarding whether to transmit during their designated slots. For instance, for path 1 in round n, if A does not transmit in slot 1, B, C and D do not transmit in slot 2, 3, and 1 respectively.

General Contention Control

Besides transmission conflicts caused by sharing a common node, general interference between adjacent transmissions is the fundamental contributor to inter-path competition. We categorize general interference scenarios into temporal and spatial types. Temporal scenarios include interference caused by earlier, simultaneous and future transmissions. Spatial scenarios cover both hidden and exposed terminals. Due to the dynamic nature of general interference, we propose a CSMA/CA-like contention control that is simple in computation, light in control overhead, and adaptable to dynamic network conditions.

Temporal Contention Control First, we use a slotted CSMA/CA-like method to avoid collisions caused by earlier, simultaneous and future transmissions. Specifically, time is slotted and each slot equals the transmission period of a pair of data and acknowledgment packets, as well as contention overhead. The contention overhead includes a maximum contention window, interframe spaces and propagation delay. Before starting a transmission, a node first determines whether the medium is occupied by earlier transmissions via physical carrier sensing. If yes, the node waits until the occupation is over. Upon an idle medium, a node avoids simultaneous transmissions by selecting a random period within the contention window $cwin_i$ to back off at the beginning of this scheduled slot. $cwin_i$ will be dynamically adjusted via hierarchical MG-Local (see Section 5.2). Random backoff reduces the possibility that multiple transmissions start at the same time. Furthermore, no transmission can start after the maximum contention window expires during each slot. This limits the period when future-transmission-caused interference may occur. If a collision occurs, retransmissions are adopted to improve reliability. Unlike CSMA/CA, our method does not exponentially increase the contention window after a failed transmission. This is because every retransmission only starts in the next scheduled slot, and the waiting time is longer than the transmission time of a data packet.

Spatial Contention Control In space, interference scenarios include hidden or exposed terminals. Hidden terminals occur when two transmissions fail due to undetected mutual interference. In contrast, exposed terminals occur when two transmissions do not proceed in parallel due to mistaking each other as interference. Our spatial contention control handles both scenarios. First, we avoid exposed terminals by setting the carrier sensing range equal to the transmission range. Upon overhearing a packet *pkt_ovrd* within the transmission range, node *i* checks whether the transmission of *pkt_ovrd* and *i*'s pending transmission *pkt_nxt* (e.g. the first packet from *i*'s queue) will form exposed terminals. We assume each packet carries the location information of its sender and receiver. Specifically, *i* retrieves the locations of itself and three other nodes: *pkt_nxt*'s receiver (denoted *nxt_rcv*) by reading the receiver field of *pkt_nxt*; *pkt_ovrd*'s sender (denoted *snd_ovrd*) and receiver (denoted *rcv_ovrd*) from the corresponding fields. Based on this information, we estimate the signal strength at *nxt_rcv* and *rcv_ovrd*, if *pkt_ovrd* and *pkt_nxt* proceed in parallel. If the signal strength is lower than a commonly used threshold 10 dB, *i* is allowed to contend for transmission right away. Otherwise, *i* waits until the medium becomes idle.

Second, we use a probabilistic contention method to reduce hidden terminals. Because the carrier sensing range is reduced to help detect exposed terminals, physical carrier sensing detects an idle medium, even when hidden terminals occur. Therefore, we introduce the idleness probability (p_i^{idle}) to enable a node to decide not to contend for transmission when physical carrier sensing detects an idle medium. p^{idle} can be adjusted according to the severity of hidden terminals, in terms of the number of collisions over a period. The more often hidden terminals occur, the higher p^{idle} should be. We dynamically adjust p^{idle} via hierarchical MG-Local in Section 5.2.

This method overcomes the problem of only adjusting carrier sensing. Because hidden and exposed terminals co-exist and occur dynamically, physical carrier sensing cannot handle these two cases at the same time. For example, increasing carrier sensing range reduces hidden terminals, but aggravates exposed terminals. Conversely, decreasing this range reduces exposed terminals, but leads to severe hidden terminals. In comparison, our method effectively handles both hidden and exposed terminals.

5.1.3 The Complete Algorithm and Complexity Analysis

Hybrid TDMA/CSMA combines intra-path scheduling and inter-path contention control to provide medium access control for multi-hop wireless networks. During system initialization, the controlled broadcasting is called by each common node. After the system starts, time is slotted and each slot is assigned a number 1, 2 or 3 according to intra-path scheduling. Node i contends for medium access at its scheduled slot by following the special contention control. If i has no schedule conflicts with other nodes or randomized schedule sharing allows i to transmit, it invokes general contention control to avoid collisions. Otherwise, i waits for the next scheduled slot.

This algorithm requires computations at two places. First, intra-path schedules are inferred locally via a modulo operation. Second, special conflict resolution requires that each node computes a schedule probability via a simple division, and determines whether to transmit by comparing a random number with the schedule probability. Message passing is only required for controlled broadcasting. Assuming m common nodes in a network of size n and the neighbor degree of each node is d, the first hop broadcast requires m messages. Because not all neighbors of a common node are on affected paths, we assume k neighbors of each common node are involved in the second hop broadcasting on average ($k \leq d$). Therefore, for average cases, the second hop broadcast requires k messages, and the message complexity is $O(m \cdot k)$. In the worst case scenario, all d neighbors of each common node transmit, and the message complexity is $O(m \cdot d)$.

5.2 Hierarchical MG-Local Control

In the previous section, we presented hybrid TDMA/CSMA, the medium access control layer of our multi-hop solution. It handles co-existing intra-path and inter-path interference, but does not control how different paths compete for limited wireless bandwidth. To tackle this challenge, we propose hierarchical MG-Local control to fairly allocate and efficiently

Table 5.1: Hierarchical MG-Local Symbols

Parameter	**Value**
(i,s,d)	identification for transmissions from node i for path $p_{s,d}$
$x_{(i,s,d)}$	bandwidth consumed for successful transmissions
$x^m_{(i,s,d)}(t)$	measured transmissions during the t-th control period
$x^f_{(i,s,d)}$	bandwidth fair share
$coll_{(i,s,d)}$	bandwidth wasted on collisions
$coll^m_{(i,s,d)}(t)$	measured collisions during the t-th control period
$idle_{(i,s,d)}$	bandwidth wasted on idling
$idle^m_{(i,s,d)}(t)$	measured idling during the t-th control period
$p^{idle}_{(i,s,d)}$	idleness probability
$cwin_{(i,s,d)}$	contention window
B	channel bandwidth
k	scaler
λ	Lagrange multiplier or shadow price

utilize limited bandwidth. Hierarchical MG-Local extends the previous flat MG-Local with 1) a customized adaptive multivariable model; 2) an improved model derivation based on iterative least square fitting; 3) multi-hop fairness support; and 4) hierarchical control.

Before presenting the algorithms, we first introduce a few symbols for the rest of this section in Table 5.1.

5.2.1 Customized Adaptive Multivariable Model

Compared with our previous method which is designed for single-hop wireless networks in Chapter 4, this adaptive multivariable control is different in two ways. First, we select two different control variables from hybrid TDMA/CSMA scheduling to more effectively handling exposed/hidden terminals. Second, we apply iterative least square fitting to improve modeling accuracy.

Control Variable Selection

According to the dynamic nature of interference in time and space, we select two control variables from the algorithms of temporal and spatial contention control presented in Sections 5.1.2 and 5.1.2.

Contention window *cwin* is the key parameter to avoid simultaneous-transmission-caused collisions in our temporal contention control. We control the temporal transmission aggressiveness of each node by adjusting *cwin* to differentiate the access priorities among different transmissions. In the previous chapter, we modify CSMA/CA by using a small collision avoidance window (*awin*) when a new packet is transmitted to reduce delay, and applying a large resolution window (*rwin*) to enforce a retransmission to wait long enough in order to avoid repetitive collisions. In comparison, when hybrid TDMA/CSMA scheduling is in use, a node has to wait for its schedule in the next round to retransmit upon a collision.

The waiting time equals three slots and is much longer than a packet's transmission time. Therefore, we no longer need to differentiate avoidance and resolution windows.

Idleness probability p^{idle} is the major factor to avoid collisions caused by hidden terminals. Because hybrid TDMA/CSMA requires carrier sensing equal to transmission range to reduce exposed terminals, the capability to detect hidden terminals of a node is limited. By enabling a node not to contend for transmission when physical carrier sensing detects an idle medium, p^{idle} allows us to control the spatial transmission aggressiveness of a link according to the severity of hidden terminals measured by the number of collisions over a period.

Regression Model

With the new control variables, we have three customized multivariable models that are compatible with the hybrid TDMA/CSMA scheduling for successful transmissions Eq. 5.1, collisions Eq. 5.2 and unnecessary idling Eq. 5.3. These models characterize the impact of each control variable and their correlations. $E = \{e_1, e_2, e_3, e_4\}$, $F = \{f_1, f_2, f_3, f_4\}$ and $L = \{l_1, l_2, l_3, l_4\}$ are the coefficients to estimate.

$$\begin{aligned} x_{(i,s,d)} &= X(p^{idle}_{(i,s,d)}, cwin_{(i,s,d)}) \\ &= e_1 \cdot p^{idle}_{(i,s,d)} + e_2 \cdot cwin_{(i,s,d)} + e_3 \cdot p^{idle}_{(i,s,d)} \cdot cwin_{(i,s,d)} + e_4; \end{aligned} \tag{5.1}$$

$$\begin{aligned} coll_{(i,s,d)} &= COLL(p^{idle}_{(i,s,d)}, cwin_{(i,s,d)}) \\ &= f_1 \cdot p^{idle}_{(i,s,d)} + f_2 \cdot cwin_{(i,s,d)} + f_3 \cdot p^{idle}_{(i,s,d)} \cdot cwin_{(i,s,d)} + f_4; \end{aligned} \tag{5.2}$$

$$\begin{aligned} idle_{(i,s,d)} &= IDLE(p^{idle}_{(i,s,d)}, cwin_{(i,s,d)}) \\ &= l_1 \cdot p^{idle}_{(i,s,d)} + l_2 \cdot cwin_{(i,s,d)} + l_3 \cdot p^{idle}_{(i,s,d)} \cdot cwin_{(i,s,d)} + l_4; \end{aligned} \tag{5.3}$$

5.2.2 Iterative Least Square Fitting

Previously we used least square fitting to estimate the coefficients E, F and L of Eq. 5.1, 5.2 and 5.3. However, this method may not always produce a solution due to the failure of matrix inversion. To overcome this drawback, we apply iterative least square fitting, which gradually improves the models by iteratively searching for the minimum square error in a gradient-descent manner. The initial models are obtained via a set of 2^2 full-factorial experiments [42].

First, we formulate three optimization problems Eq. 5.4, 5.5 and 5.6 to minimize the mean square errors between the proposed models (Eq. 5.1, 5.2 and 5.3) and measured responses ($x^m_{(i,s,d)}$, $coll^m_{(i,s,d)}$ and $idle^m_{(i,s,d)}$).

$$\begin{aligned} &\min \ J_x \ , \ where \\ J_x &= \frac{1}{N} \cdot \sum_{t=1}^{N} \left[(x^m_{(i,s,d)}(t) - X(p^{idle}_{(i,s,d)}(t), cwin_{(i,s,d)}(t)))^2 \right] \end{aligned} \tag{5.4}$$

$$\begin{aligned} & \min \ J_{coll} \ , \ where \\ J_{coll} \ &= \frac{1}{N} \cdot \sum_{t=1}^{N} \left[(coll^m_{(i,s,d)}(t) - COLL(p^{idle}_{(i,s,d)}(t), cwin_{(i,s,d)}(t)))^2 \right] \end{aligned} \tag{5.5}$$

$$\begin{aligned} & \min \ J_{idle} \ , \ where \\ J_{idle} \ &= \frac{1}{N} \cdot \sum_{t=1}^{N} \left[(idle^m_{(i,s,d)}(t) - IDLE(p^{idle}_{(i,s,d)}(t), cwin_{(i,s,d)}(t)))^2 \right] \end{aligned} \tag{5.6}$$

Second, we derive the update policies Eq. 5.7, 5.8 and 5.9 for E, F and L by applying the gradient-descent method to Eq. 5.4, 5.5, and 5.6. We apply these policies to update E, F and L iteratively, and improve the model accuracy by searching for the minimum least square error in a gradient-descent manner.

$$\begin{aligned} \frac{\partial J_x}{\partial e_h} \ &= \ \frac{1}{N} \cdot \sum_{t=1}^{N} \left[\frac{\partial X}{\partial e_h} \cdot (x^m_{(i,s,d)}(t) - X(p^{idle}_{(i,s,d)}(t), cwin_{(i,s,d)}(t))) \right] \\ e_h \ &= \ e_h + k_e \cdot \frac{\partial J_x}{\partial e_h} \end{aligned} \tag{5.7}$$

$$\begin{aligned} \frac{\partial J_{coll}}{\partial f_h} \ &= \ \frac{1}{N} \cdot \sum_{t=1}^{N} \left[\frac{\partial COLL}{\partial f_h} \cdot (coll^m_{(i,s,d)}(t) - COLL(p^{idle}_{(i,s,d)}(t), cwin_{(i,s,d)}(t))) \right] \\ f_h \ &= \ f_h + k_f \cdot \frac{\partial J_{coll}}{\partial f_h} \end{aligned} \tag{5.8}$$

$$\begin{aligned} \frac{\partial J_{idle}}{\partial l_h} \ &= \ \frac{1}{N} \cdot \sum_{t=1}^{N} \left[\frac{\partial IDLE}{\partial l_h} \cdot (idle^m_{(i,s,d)}(t) - IDLE(p^{idle}_{(i,s,d)}(t), cwin_{(i,s,d)}(t))) \right] \\ l_h \ &= \ l_h + k_l \cdot \frac{\partial J_idle}{\partial l_h} \end{aligned} \tag{5.9}$$

Alg. 3 gives the pseudo-code of iterative least square fitting and model update. In this algorithm, we first initialize the coefficients of Eq. 5.1, 5.2 and 5.3 with the results from a 2^2 full-factorial experiment. Then, after collecting every N measurements of $x^m_{(i,s,d)}, coll^m_{(i,s,d)}, idle^m_{(i,s,d)}$, we compute the coefficient updates by applying Eq. 5.7, 5.8 and 5.9. This procedure continues until the mean square error of X varies within a threshold $\epsilon = 0.05$, which is determined experimentally.

5.2.3 Hierarchical MG-Local Control

By combining adaptive multivariable control and G-Local optimization in a hierarchical manner, we have hierarchical multivariable G-Local optimization. Previously, we introduced flat MG-Local control, which transforms the original form of G-Local optimization from a multi-state function of $(x_{(i,s,d)}, coll_{(i,s,d)}, idle_{(i,s,d)})$ to a multivariable function of $(p_{(i,s,d)}, awin_{(i,s,d)}, rwin_{(i,s,d)})$. In comparison, the hierarchical method separates G-Local

Algorithm 3 Iterative Least Square Fitting

```
// This algorithm is executed at each node

Initialize E, F and L for X(), COLL() and IDLE ()
with the full-factorial experiment results

for every $N$ new measurements of $x^m_{(i,s,d)}$, $coll^m_{(i,s,d)}$ and $idle^m_{(i,s,d)}$ do
  Update E, F and L according to Eqs. 5.7, 5.8 and 5.9
  if the different between the current and previous mean square errors is less than $\epsilon$ then
    break;
  else
    continue;
  end if
end for
```

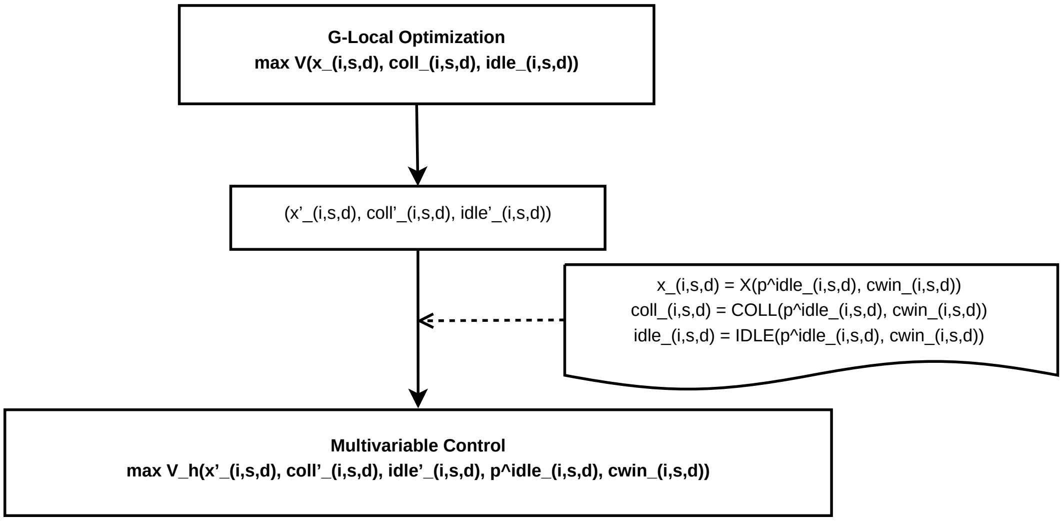
G-Local Optimization
max V(x_(i,s,d), coll_(i,s,d), idle_(i,s,d))
(x'_(i,s,d), coll'_(i,s,d), idle'_(i,s,d))
x_(i,s,d) = X(p^idle_(i,s,d), cwin_(i,s,d))
coll_(i,s,d) = COLL(p^idle_(i,s,d), cwin_(i,s,d))
idle_(i,s,d) = IDLE(p^idle_(i,s,d), cwin_(i,s,d))
Multivariable Control
max V_h(x'_(i,s,d), coll'_(i,s,d), idle'_(i,s,d), p^idle_(i,s,d), cwin_(i,s,d))

Figure 5.3: Hierarchical MG-Local Control

optimization from multivariable control as two problems, and constructs a two-tier hierarchy to provide a clear direction for making control decisions. The first tier determines a set of desired network states via G-Local optimization, and the second tier guides adaptive multivariable control to reach these desired states.

Tier 1: G-Local Target States

The goal of Tier 1 is to determine the desired network states via the G-Local optimization as shown in Eq. 5.10. To support fair multi-hop forwarding, we use maxmin fairness to derive the fair share $x^f_{(i,s,d)}$ [3]. The desired states are expressed as $argmaxV(x_{(i,s,d)}, coll_{(i,s,d)}, idle_{(i,s,d)})$, and denoted $\left\{x'_{(i,s,d)}, coll'_{(i,s,d)}, idle'_{(i,s,d)}\right\}$. Although Eq. 5.10 can be solved via multi-objective optimization, we solve it via a simple method based on the following intuition.

$$\begin{aligned} &\max\ V(x_{(i,s,d)}, coll_{(i,s,d)}, idle_{(i,s,d)})\ ,\ where \\ V\ &=\ k \cdot \log x_{(i,s,d)} - (1-k) \cdot (\frac{1}{x^{f}_{(i,s,d)}} \cdot x_{(i,s,d)} + \frac{coll_{(i,s,d)}}{B} + \frac{idle_{(i,s,d)}}{B}) \end{aligned} \tag{5.10}$$

First, instead of treating Eq. 5.10 as a multi-state function $V(x_{(i,s,d)}, coll_{(i,s,d)}, idle_{(i,s,d)})$, we view it as a single-state function $V_h(x_{(i,s,d)})$, and consider $coll_{(i,s,d)}$ and $idle_{(i,s,d)}$ as prices. The transformed objective function is given in Eq. 5.12, where $x^{m}_{(i,s,d)}(t)$, $coll^{m}_{(i,s,d)}(t)$ and $idle^{m}_{(i,s,d)}(t)$ are the measurements of round t. As shown in Alg. 4, we can easily derive the conditional optimum of $x_{(i,s,d)}$ according to Eq. 5.11. Second, we aim to reduce the price of $coll_{(i,s,d)}$ and $idle_{(i,s,d)}$ by explicitly setting $coll'_{(i,s,d)} = (1-\alpha) \cdot coll^{m}_{(i,s,d)}(t)$ and $idle'_{(i,s,d)} = (1-\beta) \cdot idle^{m}_{(i,s,d)}(t)$. Currently, we set α and β to 0.1 according to experimental observations.

$$x'_{(i,s,d)} = \frac{x^{f}_{(i,s,d)} \cdot x^{m}_{(i,s,d)}(t) \cdot B}{x^{m}_{(i,s,d)}(t) \cdot B + (coll^{m}_{(i,s,d)}(t) + idle^{m}_{(i,s,d)}(t)) \cdot x^{f}_{(i,s,d)}} \tag{5.11}$$

$$\begin{aligned} &\max\ V_h(x_{(i,s,d)})\ ,\ where \\ V_h\ &= k \cdot \log x_{(i,s,d)} - (1-k) \cdot (\frac{1}{x^{f}_{(i,s,d)}} + \frac{coll^{m}_{(i,s,d)}(t)}{x^{m}_{(i,s,d)}(t) \cdot B} + \frac{idle^{m}_{(i,s,d)}(t)}{x^{m}_{(i,s,d)}(t) \cdot B}) \cdot x_{(i,s,d)} \end{aligned} \tag{5.12}$$

Tier 2: Guided Adaptive Multivaraible Control

The goal of Tier 2 is to approach the desired network states $\left\{x'_{(i,s,d)}, coll'_{(i,s,d)}, idle'_{(i,s,d)}\right\}$ by adjusting the control variables $p^{idle}_{(i,s,d)}$ and $cwin_{(i,s,d)}$ in the adaptive multivariable models $X(p^{idle}_{(i,s,d)}, cwin_{(i,s,d)})$, $COLL(p^{idle}_{(i,s,d)}, cwin_{(i,s,d)})$ and $IDLE(p^{idle}_{(i,s,d)}, cwin_{(i,s,d)})$. This is achieved by minimizing the gap between $\{x', coll', idle'\}$ and $\{X, COLL, IDLE\}$. We formulate this problem in Eq. 5.13.

$$\begin{aligned} &\min\ V_h(p^{idle}_{(i,s,d)}, cwin_{(i,s,d)}, x'_{(i,s,d)}, coll'_{(i,s,d)}, idle'_{(i,s,d)})\ ,\ where, \\ V_h\ &= (X(p^{idle}_{(i,s,d)}, cwin_{(i,s,d)}) - x'_{(i,s,d)})^2 + \\ &\quad (COLL(p^{idle}_{(i,s,d)}, cwin_{(i,s,d)}) - coll'_{(i,s,d)})^2 + \\ &\quad (IDLE(p^{idle}_{(i,s,d)}, cwin_{(i,s,d)}) - idle'_{(i,s,d)})^2 \\ &s.t.\ 0 < p^{idle}_{(i,s,d)} \leq 1; \\ &\quad CMIN \leq cwin_{(i,s,d)} \leq CMAX; \end{aligned} \tag{5.13}$$

The hierarchical control problem is solved by applying the Lagrange transformation, which transforms Eq. 5.13 to 5.14. In this equation, λ_{p1}, λ_{p2}, λ_{c1}, and λ_{c2} are shadow prices. The corresponding dual problem is presented in Eq. 5.17.

$$\begin{aligned}
&L_h(p^{idle}_{(i,s,d)}, cwin_{(i,s,d)}, \lambda_{p1}, \lambda_{p2}, \lambda_{c1}, \lambda_{c2})\\
&= V_h(p^{idle}_{(i,s,d)}, cwin_{(i,s,d)}, x'_{(i,s,d)}, coll'_{(i,s,d)}, idle'_{(i,s,d)}) + \lambda_{p1} \cdot p^{idle}_{(i,s,d)} - \lambda_{p2} \cdot (p^{idle}_{(i,s,d)} - 1)\\
&- \lambda_{c1} \cdot (CMIN - awin_{(i,s,d)}) - \lambda_{c2} \cdot (cwin_{(i,s,d)} - CMAX)
\end{aligned} \tag{5.14}$$

$$\min L_h(p^{idle}_{(i,s,d)}, cwin_{(i,s,d)}, \lambda_{p1}, \lambda_{p2}, \lambda_{c1}, \lambda_{c2}) \tag{5.15}$$

$$D_h(p^{idle}_{(i,s,d)}, cwin_{(i,s,d)}, \{\lambda\}) = \min\ L_h \tag{5.16}$$

$$\min\ D_h(p^{idle}_{(i,s,d)}, cwin_{(i,s,d)}, \{\lambda\}) \tag{5.17}$$

The shadow prices are obtained by solving the dual problem (Eq.5.17). The solutions are $\lambda_{p1} = -h_1 \cdot p_i$, $\lambda_{p2} = h_2 \cdot (p_i - 1)$, $\lambda_{c1} = h_3 \cdot (CMIN - cwin_{(i,s,d)})$, $\lambda_{c2} = h_4 \cdot (cwin_{(i,s,d)} - CMAX)$. $\{h\}$ are scalars and their values are determined experimentally. We currently use 0.1 for these $\{h\}$ scalars. The control policies for $p^{idle}_{(i,s,d)}$ and $cwin_{(i,s,d)}$ are Eq.5.18 and 5.19. They are derived by applying the gradient-descent search to Eq. 5.15. As shown in Alg. 4, each user applies this control policy periodically until the algorithm converges. The convergence condition is that the value V from Eq. 5.10 varies within an experimentally determined threshold $\epsilon = 0.1$

$$p^{idle}_{(i,s,d)}(t) = p^{idle}_{(i,s,d)}(t-1) + k_p \cdot \frac{\partial L_h}{\partial p^{idle}_{(i,s,d)}} \tag{5.18}$$

$$cwin_{(i,s,d)}(t) = cwin_{(i,s,d)}(t-1) + k_a \cdot \frac{\partial L_h}{\partial cwin_{(i,s,d)}} \tag{5.19}$$

Algorithm 4 Hierarchial MG-Local Control

```
//This algorithm is executed locally for each outgoing link.

for each control iteration $t$ do
    Collect new measurements $x^m_{(i,s,d)}(t)$, $coll^m_{(i,s,d)}(t)$, $idle^m_{(i,s,d)}(t)$

    if the difference between $V(t-1)$ and $V(t)$ from Eq. 5.10 is less than $\epsilon$ then
        // the algorithm converges
        break;
    end if

    Compute Tier 1 conditional optimum ($x'_{(i,s,d)}$, $coll'_{(i,s,d)}$ and $idle'_{(i,s,d)}$)
    Update Tier 2: Eq. 5.13
    Update $p^{idle}_{(i,s,d)}$ and $cwin_{(i,s,d)}$ according to Eqs. 5.18 and 5.19
end for
```

Table 5.2: Correlated Congestion-Collision Control Symbols

Parameter	**Value**
(i, s, d)	identification for transmissions from node i for path $p_{s,d}$
$x_{(i,s,d)}$	bandwidth consumed for successful transmissions
$x^f_{(i,s,d)}$	bandwidth fair share
$coll_{(i,s,d)}$	bandwidth wasted on collisions
$idle_{(i,s,d)}$	bandwidth wasted on idling
$x^o_{(i,s,d)}$	offered load at node i, $(i = s)$
$x^r_{(i,s,d)}$	received load at node i
$p^{idle}_{(i,s,d)}$	idleness probability
$cwin_{(i,s,d)}$	contention window
B	channel bandwidth
k	scaler
λ	Lagrange multiplier or shadow price

5.3 Correlated Congestion-Collision Control

Wireless interference aggravates congestion by causing collisions and retransmissions. To tackle this problem, we propose correlated congestion-collision control as the third component of our multi-hop solution. This method computes a congestion cost based on a node's incoming/outgoing traffic, and local interference level. This congestion cost is used by a hop-based control method to reduce congestion.

Before presenting the algorithms, we first introduce a few symbols for the rest of this section in Table 5.2.

5.3.1 Adaptive Multivariable Control

We modify the multivariable model of Section 5.2 to explicitly model the correlated impact of traffic load and interference on congestion.

Control Variable Selection

The correlated congestion-collision control uses three control variables. The idleness probability and contention window are kept the same as in Section 5.2. The third variable is the offered/received load. The offered load (denoted $x^o_{(i,s,d)}$) represents the traffic load generated at a source node, and the received load (denoted $x^r_{(i,s,d)}$) is the traffic load received at an intermediate forwarding node. $x^o_{(i,s,d)}/x^r_{(i,s,d)}$ is the major factor that influences congestion.

Regression Model

By characterizing the impact of each factor as well as their interaction on network behaviors in terms of successful transmissions, collisions and unnecessary idleness, we have Eq. 5.20, 5.21 and 5.22 for source nodes. Eq. 5.23, Eq. 5.24 and Eq. 5.25 are used at intermediate nodes. The only difference between the models at source and intermediate nodes is the usage of $x^o_{(i,s,d)}$ and $x^r_{(i,s,d)}$. Parameters in these models are estimated via iterative least square fitting.

$$\begin{aligned} x_{(i,s,d)} &= X(x^o_{(i,s,d)}, p^{idle}_{(i,s,d)}, cwin_{(i,s,d)}) \\ &= e_1 \cdot x^o_{(i,s,d)} + e_2 \cdot p^{idle}_{(i,s,d)} + e_3 \cdot cwin_{(i,s,d)} \\ &+ e_4 \cdot x^o_{(i,s,d)} \cdot p^{idle}_{(i,s,d)} + e_5 \cdot x^o_{(i,s,d)} \cdot cwin_{(i,s,d)} \\ &+ e_6 \cdot p^{idle}_{(i,s,d)} \cdot cwin_{(i,s,d)} + e_7; \end{aligned} \tag{5.20}$$

$$\begin{aligned} coll_{(i,s,d)} &= COLL(x^o_{(i,s,d)}, p^{idle}_{(i,s,d)}, cwin_{(i,s,d)}) \\ &= f_1 \cdot x^o_{(i,s,d)} + f_2 \cdot p^{idle}_{(i,s,d)} + f_3 \cdot cwin_{(i,s,d)} \\ &+ f_4 \cdot x^o_{(i,s,d)} \cdot p^{idle}_{(i,s,d)} + f_5 \cdot x^o_{(i,s,d)} \cdot cwin_{(i,s,d)} \\ &+ f_6 \cdot p^{idle}_{(i,s,d)} \cdot cwin_{(i,s,d)} + f_7; \end{aligned} \tag{5.21}$$

$$\begin{aligned} idle_{(i,s,d)} &= IDLE(x^o_{(i,s,d)}, p^{idle}_{(i,s,d)}, cwin_{(i,s,d)}) \\ &= l_1 \cdot x^o_{(i,s,d)} + l_2 \cdot p^{idle}_{(i,s,d)} + l_3 \cdot cwin_{(i,s,d)} \\ &+ l_4 \cdot x^o_{(i,s,d)} \cdot p^{idle}_{(i,s,d)} + l_5 \cdot x^o_{(i,s,d)} \cdot cwin_{(i,s,d)} \\ &+ l_6 \cdot p^{idle}_{(i,s,d)} \cdot cwin_{(i,s,d)} + l_7; \end{aligned} \tag{5.22}$$

$$\begin{aligned} x_{(i,s,d)} &= X(x^r_{(i,s,d)}, p^{idle}_{(i,s,d)}, cwin_{(i,s,d)}) \\ &= e_1 \cdot x^r_{(i,s,d)} + e_2 \cdot p^{idle}_{(i,s,d)} + e_3 \cdot cwin_{(i,s,d)} \\ &+ e_4 \cdot x^r_{(i,s,d)} \cdot p^{idle}_{(i,s,d)} + e_5 \cdot x^r_{(i,s,d)} \cdot cwin_{(i,s,d)} \\ &+ e_6 \cdot p^{idle}_{(i,s,d)} \cdot cwin_{(i,s,d)} + e_7; \end{aligned} \tag{5.23}$$

$$\begin{aligned} coll_{(i,s,d)} &= COLL(x^r_{(i,s,d)}, p^{idle}_{(i,s,d)}, cwin_{(i,s,d)}) \\ &= f_1 \cdot x^r_{(i,s,d)} + f_2 \cdot p^{idle}_{(i,s,d)} + f_3 \cdot cwin_{(i,s,d)} \\ &+ f_4 \cdot x^r_{(i,s,d)} \cdot p^{idle}_{(i,s,d)} + f_5 \cdot x^r_{(i,s,d)} \cdot cwin_{(i,s,d)} \\ &+ f_6 \cdot p^{idle}_{(i,s,d)} \cdot cwin_{(i,s,d)} + f_7; \end{aligned} \tag{5.24}$$

$$\begin{aligned} idle_{(i,s,d)} &= IDLE(x^r_{(i,s,d)}, p^{idle}_{(i,s,d)}, cwin_{(i,s,d)}) \\ &= l_1 \cdot x^r_{(i,s,d)} + l_2 \cdot p^{idle}_{(i,s,d)} + l_3 \cdot cwin_{(i,s,d)} \\ &+ l_4 \cdot x^r_{(i,s,d)} \cdot p^{idle}_{(i,s,d)} + l_5 \cdot x^r_{(i,s,d)} \cdot cwin_{(i,s,d)} \\ &+ l_6 \cdot p^{idle}_{(i,s,d)} \cdot cwin_{(i,s,d)} + l_7; \end{aligned} \tag{5.25}$$

5.3.2 MG-Local Optimization with Correlated Congestion-Collision Control

By combining the new models and G-Local optimization with the two-tier hierarchical structure, we formulate correlated congestion-collision control. Tier one Eq. 5.26 derives the target states $\{x'_{(i,s,d)}, coll'_{(i,s,d)}, idle'_{(i,s,d)}\}$ via the same method introduced in Section 5.2.

In the second tier, we formulate correlated control for source and intermediate nodes in Eq. 5.27 and 5.28 respectively.

$$\begin{aligned} &\max\ V(x_{(i,s,d)}, coll_{(i,s,d)}, idle_{(i,s,d)})\ ,\ where \\ V\ &=\ k \cdot \log x_{(i,s,d)} - (1-k) \cdot (\frac{1}{x^f_{(i,s,d)}} \cdot x_{(i,s,d)} + \frac{coll_{(i,s,d)}}{B} + \frac{idle_{(i,s,d)}}{B}) \end{aligned} \tag{5.26}$$

$$\begin{aligned} &\min\ V^s_h(x^o_{(i,s,d)}, p^{idle}_{(i,s,d)}, cwin_{(i,s,d)}, x'_{(i,s,d)}, coll'_{(i,s,d)}, idle'_{(i,s,d)})\ ,\ where, \\ V^s_h\ &= (X(x^o_{(i,s,d)}, p^{idle}_{(i,s,d)}, cwin_{(i,s,d)}) - x'_{(i,s,d)})^2 + \\ &(COLL(x^o_{(i,s,d)}, p^{idle}_{(i,s,d)}, cwin_{(i,s,d)}) - coll'_{(i,s,d)})^2 + \\ &(IDLE(x^o_{(i,s,d)}, p^{idle}_{(i,s,d)}, cwin_{(i,s,d)}) - idle'_{(i,s,d)})^2 \\ &s.t. XMIN \leq x^o_{(i,s,d)} \leq XMAX; \\ &0 \leq p^{idle}_{(i,s,d)} \leq 1; \\ &CWIN \leq cwin_{(i,s,d)} \leq CMAX; \end{aligned} \tag{5.27}$$

$$\begin{aligned} &\min\ V^r_h(x^r_{(i,s,d)}, p^i dle_{(i,s,d)}, cwin_{(i,s,d)}, x'_{(i,s,d)}, coll'_{(i,s,d)}, idle'_{(i,s,d)})\ ,\ where, \\ V^r_h\ &= (X(x^r_{(i,s,d)}, p^{idle}_{(i,s,d)}, cwin_{(i,s,d)}) - x'_{(i,s,d)})^2 + \\ &(COLL(x^r_{(i,s,d)}, p^{idle}_{(i,s,d)}, cwin_{(i,s,d)}) - coll'_{(i,s,d)})^2 + \\ &(IDLE(x^r_{(i,s,d)}, p^{idle}_{(i,s,d)}, cwin_{(i,s,d)}) - idle'_{(i,s,d)})^2 \\ &s.t. 0 \leq p^{idle}_{(i,s,d)} \leq 1; \\ &CMIN \leq cwin_{(i,s,d)} \leq CMAX; \end{aligned} \tag{5.28}$$

We first apply the Lagrange method to transform these constrained problems Eq. 5.27 and 5.28 into unconstrained problems Eq. 5.30 and 5.33. The shadow prices $\{\lambda_{x1}, \lambda_{x2}, \lambda_{p1}, \lambda_{p2}, \lambda_{c1}, \lambda_{c2}\}$ are computed by solving the dual problems Eq. 5.32 and 5.35. Corresponding solutions are given in Eq. 5.29.

$$\begin{aligned} \lambda_{x1} &= h_1 \cdot (XMIN - x^o_{(i,s,d)}) \\ \lambda_{x2} &= h_2 \cdot (x^o_{(i,s,d)} - XMAX) \\ \lambda_{p1} &= -h_3 \cdot p_{(i,s,d)} \\ \lambda_{p2} &= h_4 \cdot (p_{(i,s,d)} - 1) \\ \lambda_{c1} &= h_5 \cdot (CMIN - cwin_{(i,s,d)}) \\ \lambda_{c2} &= h_6 \cdot (cwin_{(i,s,d)} - CMAX) \end{aligned} \tag{5.29}$$

$$\begin{aligned} &L^s_h(x^o_{(i,s,d)}, p^{idle}_{(i,s,d)}, cwin_{(i,s,d)}, \lambda_{x1}, \lambda_{x2}, \lambda_{p1}, \lambda_{p2}, \lambda_{c1}, \lambda_{c2}) \\ &= V^s_h(x^o_{(i,s,d)}, p^{idle}_{(i,s,d)}, cwin_{(i,s,d)}) - \lambda_{x1} \cdot (XMIN - x^o_{(i,s,d)}) \\ &- \lambda_{x2} \cdot (x^o_{(i,s,d)} - XMAX) + \lambda_{p1} \cdot p^{idle}_{(i,s,d)} - \lambda_{p2} \cdot (p^{idle}_{(i,s,d)} - 1) \\ &- \lambda_{c1} \cdot (CMIN - awin_{(i,s,d)}) - \lambda_{c2} \cdot (cwin_{(i,s,d)} - CMAX) \end{aligned} \tag{5.30}$$

$$D_h^s(x^o_{(i,s,d)}, p^{idle}_{(i,s,d)}, cwin_{(i,s,d)}, \{\lambda\}) = \min\ L_h^s \tag{5.31}$$

$$\min\ D_h^s(x^o_{(i,s,d)}, p^{idle}_{(i,s,d)}, cwin_{(i,s,d)}, \{\lambda\}) \tag{5.32}$$

$$\begin{aligned} & L_h^r(x^r_{(i,s,d)}, p^{idle}_{(i,s,d)}, cwin_{(i,s,d)}, \lambda_{p1}, \lambda_{p2}, \lambda_{c1}, \lambda_{c2}) \\ & = V_h^r(x^r_{(i,s,d)}, p^{idle}_{(i,s,d)}, cwin_{(i,s,d)}) + \lambda_{p1} \cdot p^{idle}_{(i,s,d)} - \lambda_{p2} \cdot (p^{idle}_{(i,s,d)} - 1) \\ & - \lambda_{c1} \cdot (CMIN - awin_{(i,s,d)}) - \lambda_{c2} \cdot (cwin_{(i,s,d)} - CMAX) \end{aligned} \tag{5.33}$$

$$D_h^r(x^r_{(i,s,d)}, p^{idle}_{(i,s,d)}, cwin_{(i,s,d)}, \{\lambda\}) = \min\ L_h^r \tag{5.34}$$

$$\min\ D_h^r(x^r_{(i,s,d)}, p^{idle}_{(i,s,d)}, cwin_{(i,s,d)}, \{\lambda\}) \tag{5.35}$$

With the gradient-descent method, we derive the control policies Eq. 5.36, 5.37, and 5.38 for the source and Eq. 5.39, 5.37, and 5.38 for intermediate nodes. Before introducing the concept of congestion cost, we have to resolve an issue with current control policies for $x^o_{(i,s,d)}$ and $x^r_{(i,s,d)}$. A source node has direct control of $x^o_{(i,s,d)}$ by increasing or throttling the offered traffic according to Eq. 5.36. However, an intermediate node has no direct control over $x^r_{(i,s,d)}$, the amount of traffic received from its previous hop.

$$x^o_{(i,s,d)}(t) = x^o_{(i,s,d)}(t-1) + k_x \cdot \frac{\partial L_h}{\partial x^o_{(i,s,d)}} \tag{5.36}$$

$$p^{idle}_{(i,s,d)}(t) = p^{idle}_{(i,s,d)}(t-1) + k_p \cdot \frac{\partial L_h}{\partial p^{idle}_{(i,s,d)}} \tag{5.37}$$

$$cwin_{(i,s,d)}(t) = cwin_{(i,s,d)}(t-1) + k_a \cdot \frac{\partial L_h}{\partial cwin_{(i,s,d)}} \tag{5.38}$$

$$x^r_{(i,s,d)}(t) = x^r_{(i,s,d)}(t-1) + k_x \cdot \frac{\partial L_h}{\partial x^r_{(i,s,d)}} \tag{5.39}$$

$$\Delta x^r_{(i,s,d)} = k_x \cdot \frac{\partial L_h}{\partial x^r_{(i,s,d)}} \tag{5.40}$$

In order to enforce Eq. 5.39, we move $\Delta x^r_{(i,s,d)}$ (shown in Eq. 5.40), the adjustment of $x^r_{(i,s,d)}$ from an intermediate node i to its 1-hop upstream node $(i-1)$. This migration is based on the fact that $x^r_{(i,s,d)}$ equals $x_{i-1,s,d}$, which means the amount of received traffic at node i is the amount of traffic that node $(i-1)$ successfully delivered to node i. By passing $\Delta x^r_{(i,s,d)}$ from node i to $(i-1)$, we enable the upstream node to increase or decrease its outgoing traffic according to whether node i can handle more or less traffic. This can be achieved by explicitely sending a message from node i to $(i-1)$, or piggyback this information in the acknowledgement packet. Because $\Delta x^r_{(i,s,d)}$ is computed based on the incoming traffic ($x^r_{(i,s,d)}$), outgoing traffic ($x_{(i,s,d)}$), as well as local collision situation ($coll_{(i,s,d)}$), we call it the congestion cost.

$$x^o_{(i,s,d)}(t) = x^o_{(i,s,d)}(t-1) + k_x \cdot (\frac{\partial L}{\partial x^o_{(i,s,d)}} + \Delta x^r_{i+1,s,d} \cdot \frac{\partial x_{(i,s,d)}}{\partial x^o_{(i,s,d)}}) \tag{5.41}$$

$$p^{idle}_{(i,s,d)}(t) = p^{idle}_{(i,s,d)}(t-1) + k_p \cdot (\frac{\partial L}{\partial p^{idle}_{(i,s,d)}} + \Delta x^r_{i+1,s,d} \cdot \frac{\partial x_{(i,s,d)}}{\partial p^{idle}_{(i,s,d)}}) \tag{5.42}$$

$$cwin_{(i,s,d)}(t) = cwin_{(i,s,d)}(t-1) + k_a \cdot (\frac{\partial L}{\partial cwin_{(i,s,d)}} + \Delta x^r_{i+1,s,d} \cdot \frac{\partial x_{(i,s,d)}}{\partial cwin_{(i,s,d)}}) \tag{5.43}$$

The new policies for correlated congestion-collision control are given in Eq. 5.41, 5.42 and 5.43. If a node j is a source, it takes into account the congestion cost $\Delta x^r_{i+1,s,d}$ when adjusting $x^o_{(i,s,d)}$, $p^{idle}_{(i,s,d)}$ and $cwin_{(i,s,d)}$. Similarly, an intermediate node adjusts $p^{idle}_{(i,s,d)}$ and $cwin_{(i,s,d)}$ by considering the congestion cost.

Algorithm 5 Hierarchial MG-Local Control

```
//This algorithm is executed locally at a node $i$.
for each control iteration $t$ do
  Collect new measurements $x^m_{(i,s,d)}(t)$, $coll^m_{(i,s,d)}(t)$, $idle^m_{(i,s,d)}(t)$
  if the difference between $V(t-1)$ and $V(t)$ from Eq. 5.26 is less than $\epsilon$ then
    // the algorithm converges
    break;
  end if
  Compute Tier 1 conditional optimum ($x'_{(i,s,d)}$, $coll'_{(i,s,d)}$ and $idle'_{(i,s,d)}$)
  if node $i$ is the source for path $p_{s,d}$ then
    Update Tier 2: Eq. 5.27
    Adjust the offered load $x^o_{(i,s,d)}$ according to Eq. 5.41
  else
    Update Tier 2:Eq. 5.28
    Compute the congestion cost $\Delta x^r_{(i,s,d)}$ according to Eq. 5.40)
    Pass the congestion cost to the previous hop $(i-1)$
  end if
  Adjust $p^{idle}_{(i,s,d)}$ and $cwin_{(i,s,d)}$ according to Eqs. 5.42 and 5.43
end for
```

5.4 The Complete Solution

We construct the complete solution of multi-hop resource management by combining hybrid TDMA/CSMA, hierarchical MG-Local control and correlated congestion-collision control. The structure of this solution is illustrated in Fig.5.20. We also show the system flow diagram of the complete solution in Fig. 5.21.

5.4.1 Implementation

We implement the complete solution in ns2 by modifying the basic access method CSMA/CA of IEEE 802.11b. First, we add the selected control variables: per-link offered load ($x^{o}_{(i,s,d)}$), per-link received load ($x^{r}_{(i,s,d)}$), per-link idleness probability ($p^{idle}_{(i,s,d)}$), and per-link contention window ($cwin_{(i,s,d)}$). Second, we divide time into slots. If a node i has a packet for path $p_{(s,d)}$ to send and the current slot is i's designated turn according to hybrid TDMA/CSMA scheduling, node i starts backing off with a probability $1-p^{idle}_{(i,s,d)}$, and waits for the propagation delay of one packet with a probability $p^{idle}_{(i,s,d)}$. The back-off period is randomly selected within the contention window $cwin_{(i,s,d)}$. Third, $x^{o}_{(i,s,d)}$, $x^{r}_{(i,s,d)}$, $p^{idle}_{(i,s,d)}$, and $cwin_{(i,s,d)}$ are updated according to Eq. 5.41, 5.39, 5.42 and 5.43 until the algorithm converges. This update requires each node to periodically measure the packet sending rate ($x^{s}_{(i,s,d)}$), packet delivery rate ($x^{m}_{(i,s,d)}$), packet collision rate ($coll^{m}_{(i,s,d)}$), and unnecessary idling ($idle^{m}_{(i,s,d)}$). All of these are done locally. $x^{s}_{(i,s,d)}$ is calculated via dividing the number of transmitted packets by the length of a control period. Similarly, $x^{m}_{(i,s,d)}$ is the ratio of acknowledged packet to the control period. $coll^{m}_{(i,s,d)}$ is calculated via dividing the product of the number of unacknowledged packets and retry limit by a control period. $idle^{m}_{(i,s,d)}$ is the difference between the packet sending rate ($x^{s}_{(i,s,d)}$) and bandwidth fair share ($x^{f}_{(i,s,d)}$), representing the portion of bandwidth that should be utilized but actually not. These measurements are used to derive the coefficients (E, F and L) of our multivariable models X, $COLL$ and $IDLE$ (Eq. 5.20, 5.23, 5.21, 5.24, 5.22, 5.25) via iterative least square fitting (Alg. 1).

5.4.2 Experiments

We carry out two types of experiments to study the performance of MSC in depth and breadth. First, the in-depth study compares MSC with its three components: hybrid TDMA/CSMA (denoted HS), hierarchical MG-Local control (denoted HM), and correlated congestion-collision control (denoted CC). Second, the in-breadth study compares MSC with three alternative methods: adaptive CSMA [48] (denoted AC), CSMA/CA-HD (denoted HC), and CSMA/CA-ED (denoted EC). Similar to our previous experiments, simulation accuracy is improved by adding a SINR-based collision model with aggregate interference, and a preamble detection function to the IEEE 802.11 implementation in ns2.

We select adaptive CSMA because it is a comprehensive multi-hop solution and shares similar objectives to MSC. To achieve fair allocation and efficient utilization of wireless bandwidth, AC applies utility-based optimization to control the traffic load at the source node of each flow, and formulates throughput maximization as a maximum likelihood function to accommodate all admitted arrival rates. For reducing collisions, AC uses the request-to-send/clear-to-send protocol (RTS/CTS) to reserve the medium for each transmission. AC also allows transmission over a link, where collisions occur more often, to transmit with a longer period once it seizes the medium. For congestion control, AC throttles excessive traffic at each source and gives the nodes with more backlogged data higher priority to transmit, so that queueing delay can be reduced. Additionally, we introduce two configurations of CSMA/CA to represent the scenarios when exposed and hidden terminals dominate respectively. CSMA/CA-ED fixes the physical carrier sensing range to twice the transmission range to reduce hidden terminals. CSMA/CA-HD reduces exposed terminals by setting the carrier sensing range equal to the transmission range.

These experiments are carried out under different network topologies, traffic loads and interference levels. Performance is evaluated according to four metrics: packet loss rate, Jain's fairness index, end-to-end delay and aggregate network throughput. We simulate a 800-by-800 m^2 network with 200 nodes and generate Poisson traffic at each source node. The fixed parameters include: packet size (512 bytes), channel bit rate (1 Mbps), and transmit power (0.2818 Watt). Each experiment lasts 15 minutes.

In-Depth Performance Study

We compare our MSC with its three elements: hybrid TDMA/CSMA (HS), hierarchical MG-Local control (HM) and correlated congestion-collision control (CC). The goal is to show whether each component fulfills its objective, and how they contribute to MSC's overall performance. Because CSMA/CA-HD (HC) has no control over hidden terminals, congestion and unfair resource consumption, we use it as the reference of lower-bound performance.

Experiments With Random Topologies The first set of experiments are carried out in sixteen different routing topologies with a fixed offered load of 50 pps. Each topology is generated by randomly selecting fifteen multi-hop paths via Dijkstra's algorithm [18]. For visual clarity, we sort each sub-plot in Fig. 5.4 according to the performance of CSMA/CA-HD. Experiments are repeated with randomly generated Poisson traffic for ten times over each topology. Because the resulted confidence intervals are small, we did not include them in the already crowded figure, Fig. 5.4. We do present confidence intervals in the experiments of controlled topologies in Section 5.4.2. It needs to be noted that we use the line presentation in Fig. 5.4 only for easier visual comparison. A line, in this case, does indicate any relation between different topologies.

First, hybrid TDMA/CSMA (HS) reduces the packet loss rate significantly in all sixteen topologies as shown in Fig. 5.4. This is because HS minimizes collisions within the same path via intra-path scheduling; and reduces collisions between different paths caused by scheduling conflicts and various temporal and spatial interference. In comparison, hierarchical MG-Local control (HM) reduces collisions slightly because it does not differentiate intra-path and inter-path interference. Without effective handling of intra-path interference (e.g. intra-path scheduling), HM is less effective in terms of collision control. Correlated

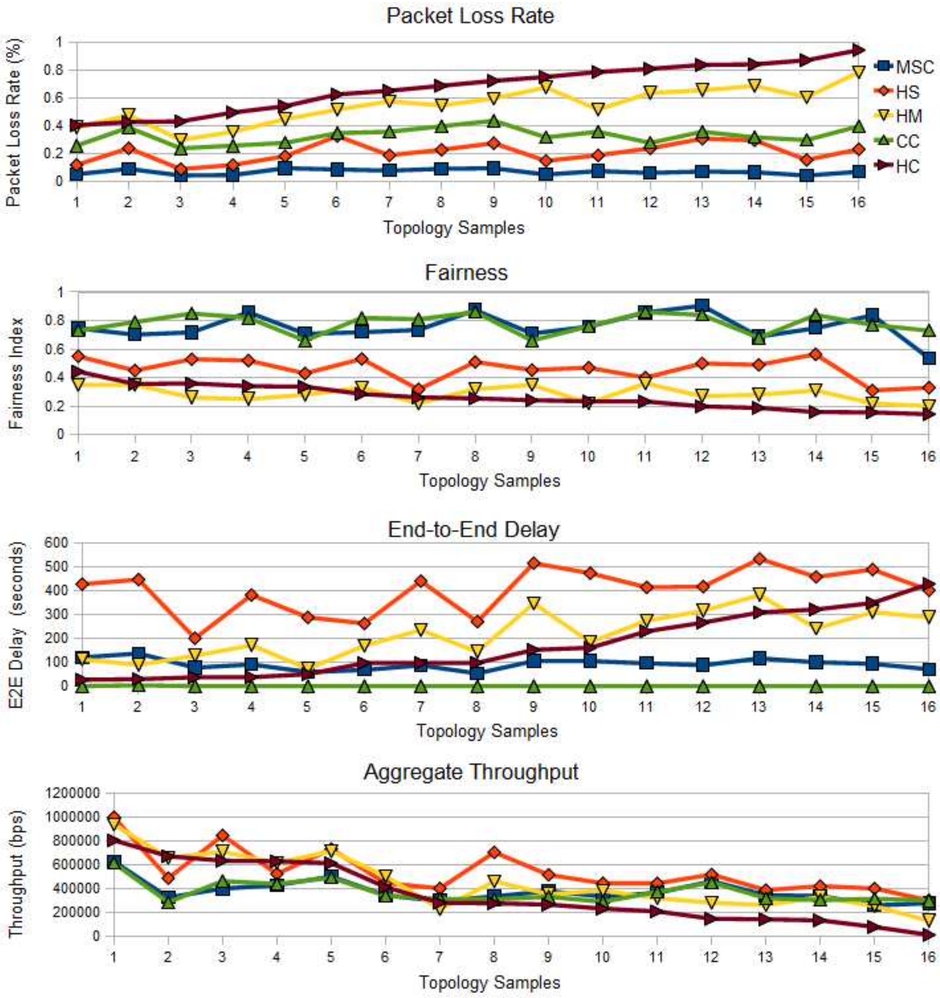

Figure 5.4: Random Topologies - In-Depth Experiment

congestion-collision control (CC) reduces the negative impact of congestion on collisions, and achieves lower packet loss rate than HM. By combining these three components, our MSC achieves the lowest packet loss rate by handling collisions caused by intra-path, inter-path interference as well as congestion in all sixteen topologies. CSMA/CA-HD incurs the highest packet loss rate because its physical carrier sensing range equals transmission range. Consequently, it experiences significant packet loss caused by hidden terminals as well as intra-path interference and congestion.

Second, HS improves fairness by transforming unwanted competition within the same path into coordinated transmission via intra-path scheduling. Furthermore, its randomized schedule sharing allows multiple paths to share a conflicting schedule with equal chances. In comparison, although HM is a fairness-driven method, its fairness is compromised by the high packet loss rate. Besides applying the same fairness-driven allocation as HM, CC also reduces the mutual negative impact of congestion and collision. Therefore, CC achieves higher fairness than HM. Altogether, MSC combines the advantages of HS, HM and CC, and therefore maintains good fairness in all sample topologies. HC has no fairness policy for multi-hop networks, and performs poorly in all topologies.

Third, HS incurs the longest delay because of its scheduling overhead. HM ranks second simply because it has no scheduling component, and therefore incurs shorter delay. CC reduces the end-to-end delay significantly because it reduces congestion by throttling excessive traffic and reducing the negative impact of collisions. This merit is inherited by MSC to overcome the long delay drawback of HS. Furthermore, MSC achieves good end-to-end delay in all sixteen topologies. Although HC achieves short delay in some of the topologies, this gain is at the price of sacrificing fairness. Specifically, HC gives the transmissions which experience less intense interference more chances to access the medium, and starves the others.

At last, throughput varies significantly upon topology changes due to different interference levels/scenarios and transmission bottlenecks. As Fig. 5.4 shows, HS achieves higher throughput than the other methods in most of the topologies due to its lower packet loss rate. This good throughput is gained at the price of longer delay. Although CC trades off throughput for shorter delay by reducing source rate, its throughput is higher than HC in nearly half of the topology samples. As a result, MSC achieves similar throughput as CC, but with much less packet loss.

Experiments in Controlled Topologies Because throughput varies significantly with network topology, we select three topologies from the previous set of experiments to represent the cases where MSC achieves lower, similar and higher throughput than CSMA/CA-HD (HC), and compare MSC, HS, HM, CC and HC according to packet loss rate, fairness index and end-to-end delay. These three topologies are denoted Topologies 1, 2 and 3 respectively. Besides topology, we study the impact of two other factors: offered load and interference levels.

Two sub-sets of experiments are carried out. First, traffic loads of 10, 50 and 100 pps are used to address three situations: partially saturated, saturated and over-saturated traffic. By fixing the interference level at 15 paths, we run MSC, HS, HM, CC and HC with all three traffic loads in all three topologies. Second, we evaluate the impact of interference by generating three interference levels of 10, 15 and 20 paths. With a fixed traffic load of 50 pps, MSC, HS, HM and HC are applied under all three interference levels for all three topologies. Furthermore, each experiment is replicated 10 times with randomly generated Poisson traffic. A 95% confidence interval is computed for each metric. We summarize these experiments by presenting representative results as follows.

First, Figs. 5.5, 5.6, and 5.7 show the impact of offered load in Topologies 1, 2 and 3 respectively.

Packet loss rate: Hybrid TDMA/CSMA (HS) reduces the packet loss rate the most compared to hierarchical MG-Local control (HM) and correlated congestion-collision control (CC) under all three loads. This is because its intra-path scheduling significantly reduces collisions within the same path, and its inter-path contention control offers effective counter-solutions to schedule conflicts, temporal and spatial interference. HS's good performance also holds in all three topologies. When traffic load increases, CC reduces collisions more effectively and achieves lower packet loss rate than HM, due to its correlated congestion-collision control. HM achieves better packet loss rate than HC when the load is 50 pps in all three topologies, but this advantage disappears with a more severe degree of congestion caused by a higher load of 100 pps. By combining all three elements, MSC achieves the lowest packet loss rate in all three topologies, and its performance remains at the same level under all three traffic loads. **Fairness:** Compared to hybrid TDMA/CSMA and hierarchical MG-Local control, correlated congestion-collision control is fairer under all three loads in all three topologies. This is because it applies MG-Local's fairness-driven fair allocation. Although HM applies the same method, its fairness is compromised by the high packet loss rate. HS is fairer than HM because its intra-path scheduling transforms unwanted competition into coordinated transmissions. Additionally, HS's randomized schedule sharing allows multiple paths to have equal chances to utilize a conflicting schedule. Combining the strength of HS and CC, our MSC achieves good fairness despite varying traffic load and topology. **End-to-end delay:** Hybrid TDMA/CSMA incurs longer delay than hierarchical MG-Local control and correlated congestion-collision control because its scheduling overhead increases queue delay. Furthermore, HS's delay increases with a higher load because more severe congestion leads to more backlogged data and longer queueing delay. Without HS's scheduling overhead, HM's delay is shorter than that of HS. Similarly, HM's delay increases when congestion is aggravated by more traffic. CC has the shortest delay because its congestion control reduces both collisions and congestion by throttling excessive traffic. Due to the CC component, MSC consistently achieves good delay performance with varying load and topology. The reason that MSC's delay is longer than CC is because MSC incurs scheduling overhead for the HS scheduling component. The occasional short delay of HC is again at the expense of fairness.

Second, Figs. 5.8, 5.9, and 5.10 demonstrate the impact of interference level on MSC, HS, HM, CC and HC in Topologies 1, 2 and 3 respectively.

Packet loss rate: Under different levels of interference, hybrid TDMA/CSMA reduces collisions more significantly than hierarchical MG-Local control and correlated congestion-collision control. This further demonstrates the effectiveness of HS's intra-path scheduling and inter-path contention control. HM's high packet loss is due to its incapability of han-

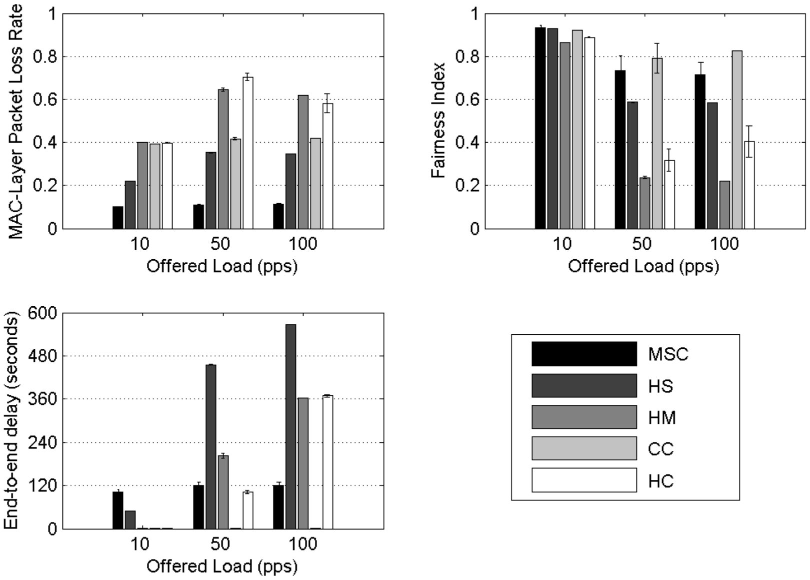

Figure 5.5: The Impact of Offered Load (Scenario 1)

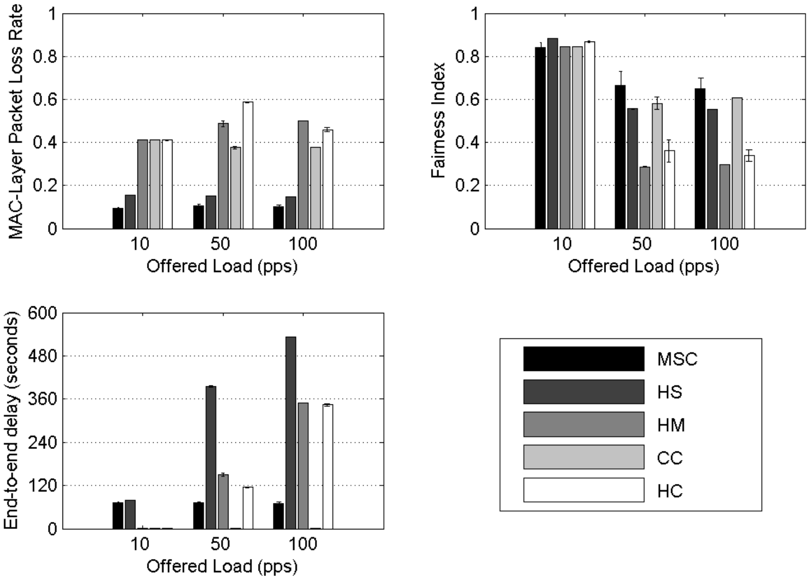

Figure 5.6: The Impact of Offered Load (Scenario 2)

dling intra-path interference and congestion. CC reduces collisions more than HM. This is once again attributed to its correlated congestion-collision control reduces the negative impact of congestion on collisions. Combining HS, HM and CC, MSC has the lowest packet loss rate under all three levels of interference in all three topologies. **Fairness:** Both MSC and correlated congestion-collision control achieve good fairness under all three interference

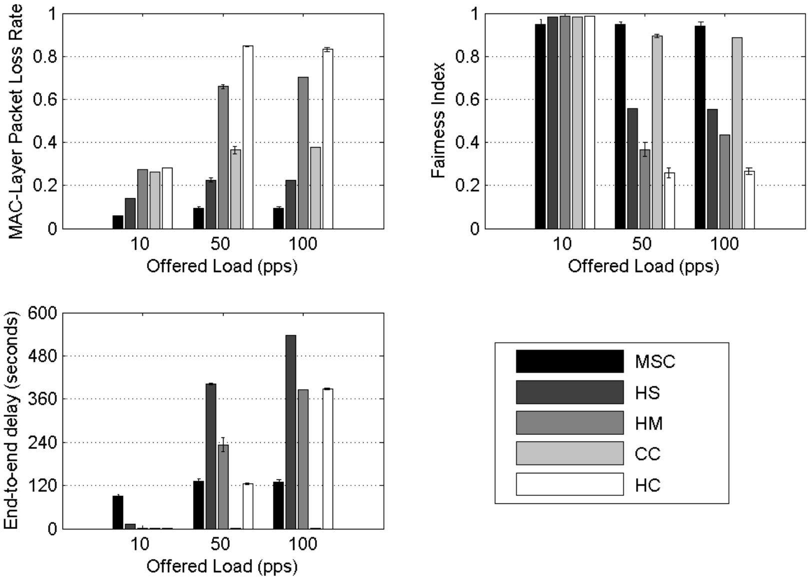

Figure 5.7: The Impact of Offered Load (Scenario 3)

levels due to their fairness-driven resource allocation. HM's fairness is again compromised by its high packet loss rate. HS improves fairness by coordinating intra-path transmissions and solving schedule conflicts between different paths. **End-to-end delay:** MSC achieves good end-to-end delay by inheriting the capability of throttling excessive traffic at the source node from correlated congestion-collision control. At the same time, due to its hybrid TDMA/CSMA scheduling component, MSC also incurs scheduling overhead. But MSC's delay is significantly shorter than that of HS. Furthermore, the delay of HS and hierarchical MG-Local control increase with a higher level of interference. This is because intense interference aggravates congestion, and results in longer queueing delay.

In summary, Fig. 5.11 shows the results of varying network topologies with an offered load of 50 pps and interference level of 15 paths. These results further validate our analysis for the random topology experiments (see Section 5.4.2). This in-depth study demonstrates that MSC combines the advantages of hybrid TDMA/CSMA, hierarchical MG-Local control and correlated congestion-collision control, and therefore achieves low packet loss rate, high fairness, and short end-to-end delay under different traffic load, interference and topologies. In the next section, we further evaluate MSC's performance by comparing it with three alternative methods.

In-Breadth Performance Study

So far, we have evaluated how each component of MSC contributes to the overall performance. The in-breadth study compares MSC to adaptive CSMA (AC), CSMA/CA-HD

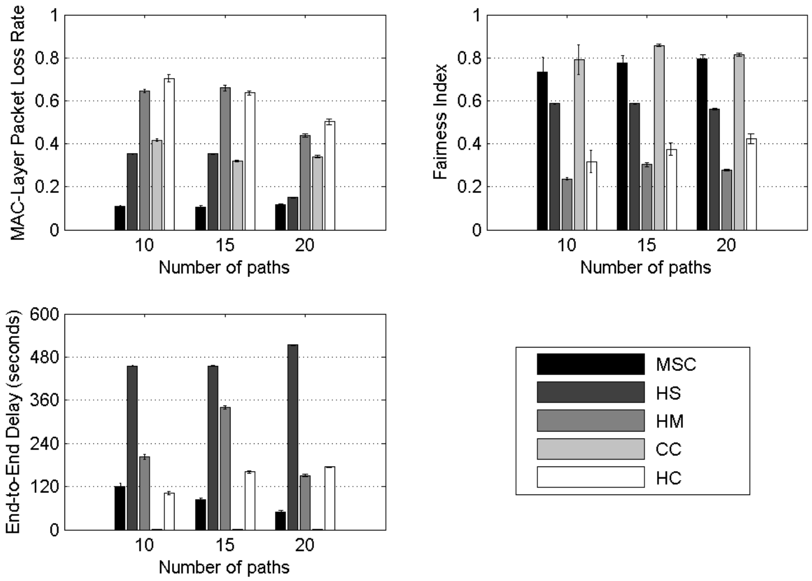

Figure 5.8: The Impact of Interference Levels (Scenario 1)

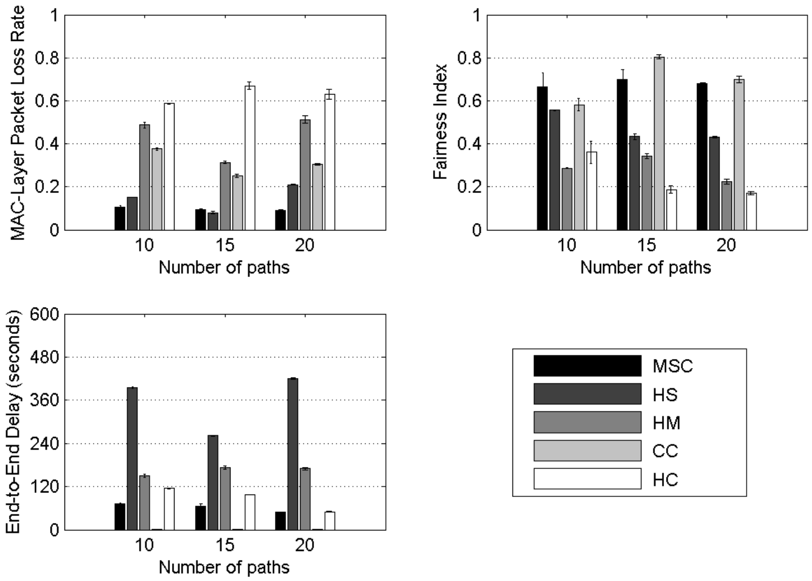

Figure 5.9: The Impact of Interference Levels (Scenario 2)

(HC) and CSMA/CA-ED (EC).

Experiments in Random Topologies We first compare MSC to AC, HC and EC in sixteen different routing topologies with a fixed offered load of 50 packets per second (pps). These topologies are the same as those used in in-depth experiments of Section 5.4.2. Fol-

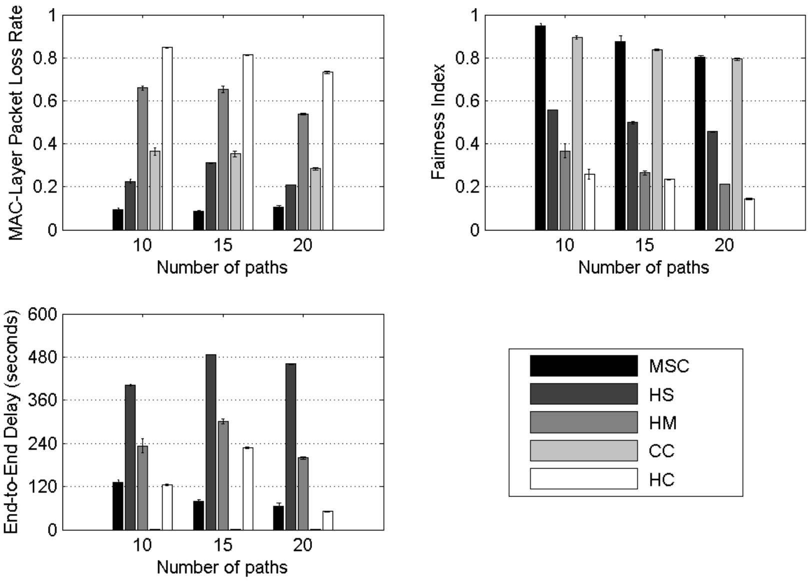

Figure 5.10: The Impact of Interference Levels (Scenario 3)

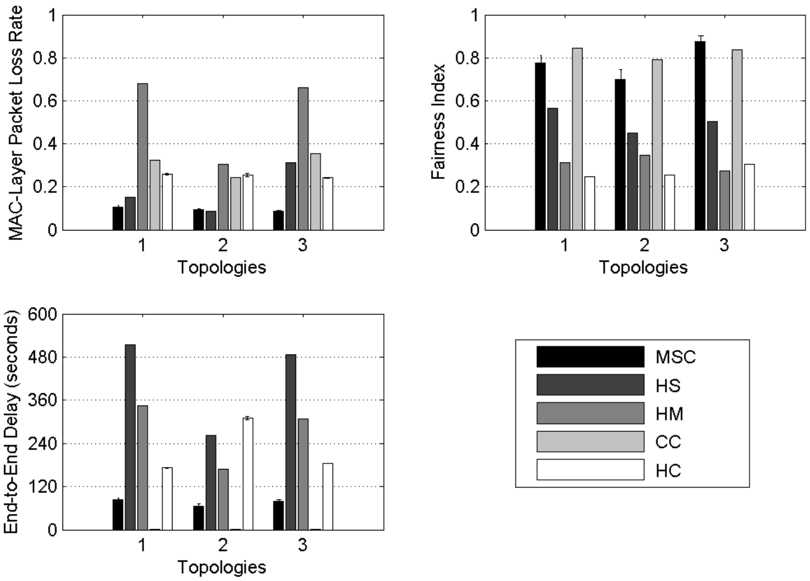

Figure 5.11: The Impact of Network Topologies

lowing the same method in Section 5.4.2, we sort the topologies of each sub-plot in Fig. 5.12 according to HC's performance for visual clarity. This is also because CSMA/CA-HD's performance varies significantly with topology changes.

First, MSC consistently outperforms AC, HC and EC by achieving the lowest packet loss rate among all 16 topologies as shown in Fig. 5.12. This is attributed to intra-path

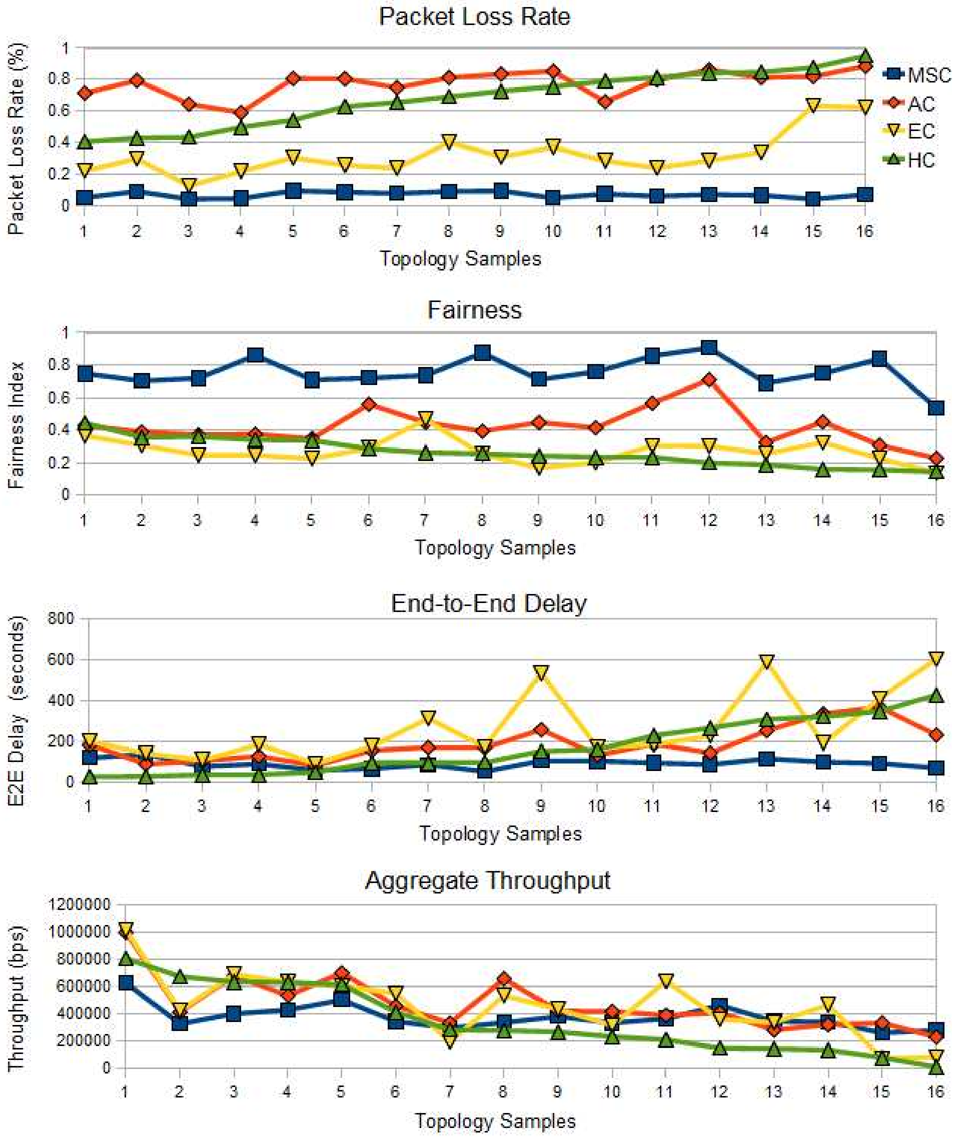

Figure 5.12: Random Topologies - In-Breadth Experiment

scheduling coordinating transmission within the same path, and inter-path contention control resolving schedule conflicts and minimizing collisions caused by various temporal and spatial interference scenarios. CSMA-ED (EC) ranks second because its larger carrier sensing range helps to reduce collisions caused by hidden terminals. CSMA-HD (HC) is subject to significant packet loss because its physical carrier sense range equals transmission range, and therefore cannot detect hidden terminals. Adaptive CSMA (AC) incurs even higher packet loss than HC in some of the sample topologies. Although AC aims to reduce collisions by applying RTS/CTS to reserve medium before transmitting data, hidden terminals cause collisions among RTS/CTS/DATA/ACK packets [88]. Furthermore, these collisions are aggravated because AC allows transmission over a link, which experiences more collisions, to last longer.

Second, MSC is consistently the fairest method compared with the other three candidates in all sample topologies. This is because hierarchical MG-Local control drives different paths to consume their fair share of limited bandwidth. AC ranks second, although it also aims to achieve fair allocation. Compared to MSC, adaptive CSMA does not consider the impact of wireless interference on fairness, and sacrifices short-term fairness by allowing lossy transmission to last longer. Consequently, AC causes more frequent starvation within a short period. Neither HC nor EC has a fairness policy, and perform poorly in most topologies.

Third, MSC achieves shorter end-to-end delay in most topologies. This is because correlated congestion-collision control reduces delay by throttling excessive traffic and reducing collisions. In comparison, AC controls congestion by adjusting source rate without reducing collisions. Moreover, AC allows transmission over a link, where collisions occur often, with a longer period and further increases queueing delay. An interesting observation is that HC achieves shorter or similar delay to MSC in Scenarios 1 to 7. This is because CSMA/CA-HD gives more transmission opportunities to a few paths that experience less severe interference, and starves the other. CSMA/CA-ED has the same problem. Compared to HC, EC's incapability of handling hidden terminals causes more collisions and retransmissions, which results in longer queueing delay than HC.

Finally, throughput varies significantly upon topology changes due to different interference levels/scenarios and transmission bottlenecks. As Fig. 5.12 shows, AC achieves higher throughput in a few scenarios because throughput maximization is its major design objective. EC achieves similar throughput to AC because it increases transmission aggressiveness by reducing carrier sensing range. However, EC's throughput gain is achieved by sacrificing fairness. In most cases, HC has lower throughput than AC and HC because it is more conservative in transmission to reduce hidden terminals. Compared to AC and EC, MSC achieves similar throughput in more than eight topologies. MSC achieves better throughput than HC in nearly half of the topology samples.

This experiment set demonstrates that MSC maintains consistent performance despite topology variations. It also shows that network topology does not influence MSC's capability to outperform AC, HC and EC in terms of minimizing packet loss, improving fairness and reducing end-to-end delay, while achieving reasonable throughput. Our future endeavor will focus on improving MSC's throughput.

Experiments With Controlled Network Topologies In the second set of experiments, we use the same three topologies in Section 5.4.2 to represent the cases where MSC achieves lower, similar and higher throughput than CSMA/CA-HD (HC), and compare MSC, AC,

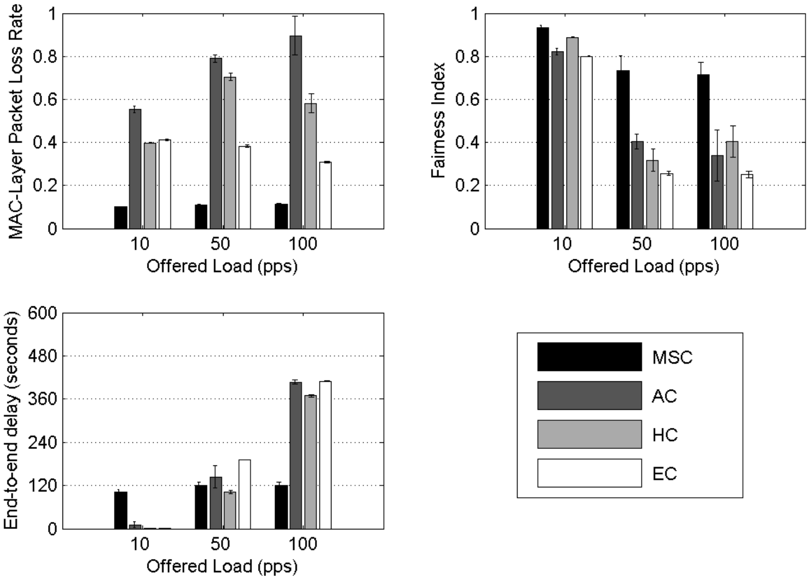

Figure 5.13: The Impact of Offered Load (Scenario 1)

HC and EC according to packet loss rate, fairness index and end-to-end delay. These three topologies are also denoted Topologies 1, 2 and 3 respectively. Besides topology, we study the impact of two other factors: offered loads and interference levels. Similar to the in-depth experiments, we use traffic loads of 10, 50 and 100 pps to address three situations: partially saturated, saturated and over-saturated traffic. By fixing the interference level at 15 paths, we run MSC, AC, HC and EC with all three traffic loads in all three topologies. For varying interference levels, we generate 10, 15 and 20 paths. With a fixed traffic load of 50 pps, MSC, AC, HC and EC are executed under all three interference levels for all three topologies.

First, Figs. 5.13, 5.14, and 5.15 show the results of varying offered loads in Topologies 1, 2 and 3.

Packet loss rate: MSC consistently achieves the lowest packet loss rate under different traffic loads, because hybrid TDMA/CSMA significantly reduces collisions by handling various interference scenarios. CSMA/CA-HD (HC) reduces hidden terminals and ranks second. Similar to our previous analysis, CSMA/CA-ED (EC) incurs higher packet loss rate than HC because its aggressive transmission leads to more collisions caused by hidden terminals. Adaptive CSMA (AC) experiences severe packet loss because its dynamic transmission length aggravates hidden terminals and causes more collisions among RTS/CTS/DATA/ACK. **Fairness:** MSC remains the fairest with different traffic loads, which is attributed to its fairness-driven bandwidth allocation and coordinated intra-path transmission. AC has lower fairness index, because its significant packet loss and dy-

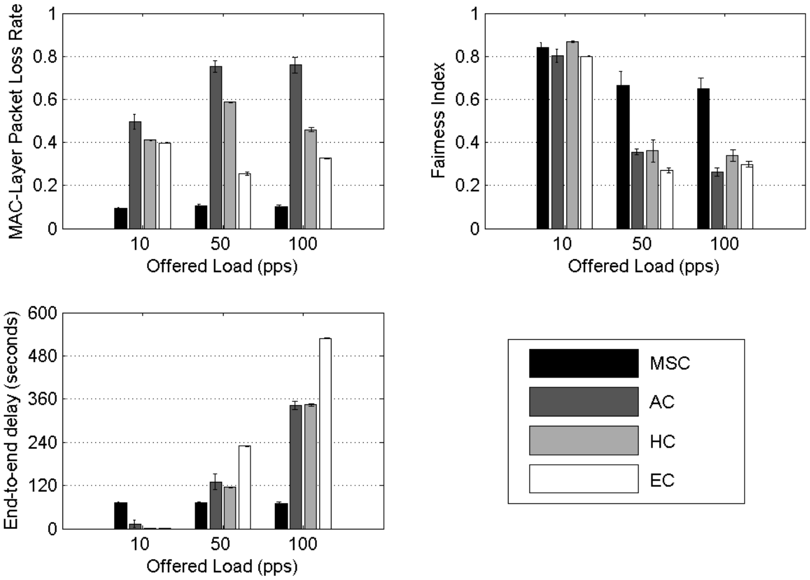

Figure 5.14: The Impact of Offered Load (Scenario 2)

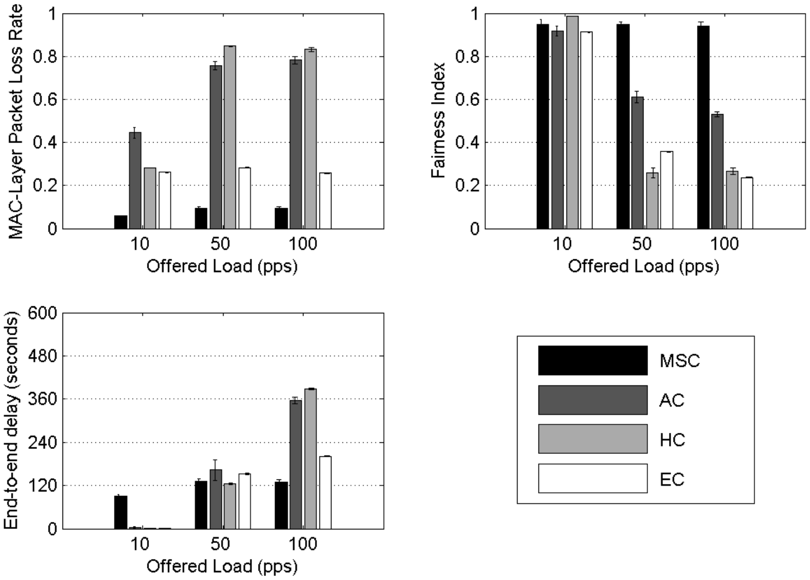

Figure 5.15: The Impact of Offered Load (Scenario 3)

namic transmission period compromises fairness. EC and HC inherit the unfair nature of CSMA/CA and perform poorly in most cases. **End-to-end delay:** With saturated traffic of 50 and 100 pps, MSC exhibits shorter end-to-end delay than AC, HC and EC. Furthermore, MSC maintains similar delay for all three loads by throttling excessive traffic and reducing collisions, while the delay of AC, HC and EC increases with higher traffic

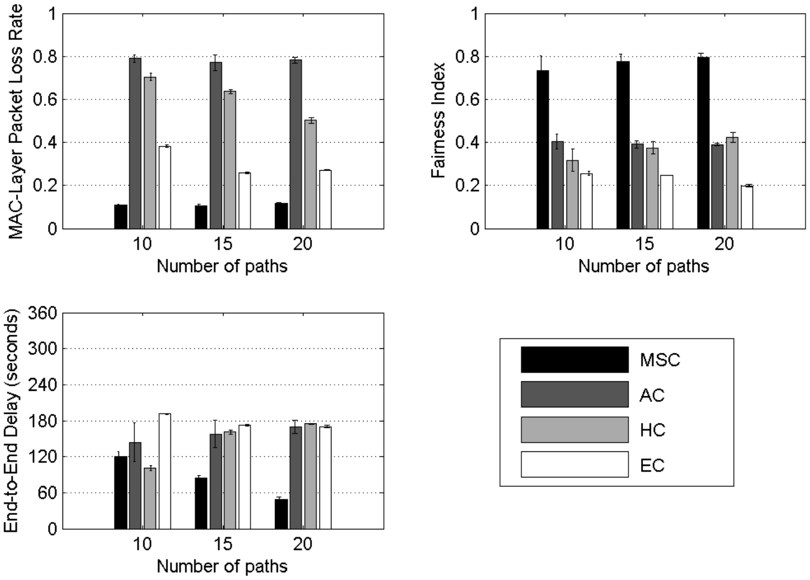

Figure 5.16: The Impact of Interference Levels (Scenario 1)

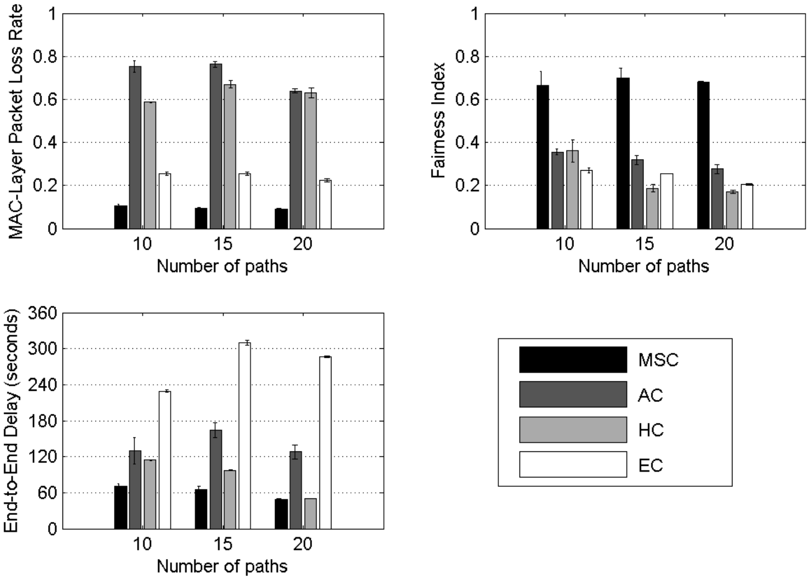

Figure 5.17: The Impact of Interference Levels (Scenario 2)

loads. Although AC also has source rate control, its significant packet loss requires more retransmissions and longer waiting time, which increases delay. For HC and EC, there is no congestion control. Both collisions and traffic overload contribute to longer delay.

Second, Figs. 5.16, 5.17 and 5.18 show the results of varying interference levels without changing the offered load (50 pps) in Topologies 1, 2 and 3.

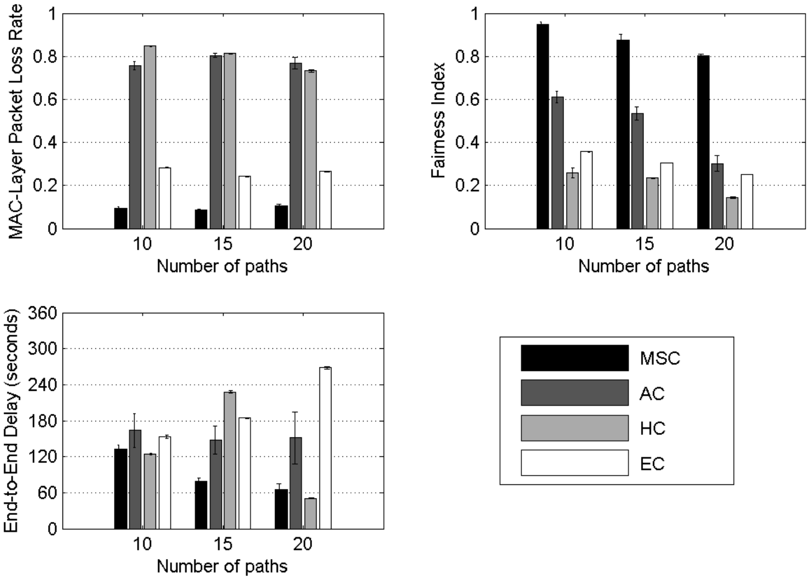

Figure 5.18: The Impact of Interference Levels (Scenario 3)

Packet loss rate: MSC consistently achieves lower packet loss rate than adaptive CSMA (AC), CSMA/CA-HD (HC) and CSMA/CA-ED (EC) under different levels of interference, which again demonstrates that hybrid TDMA/CSMA is effective for handling various interference scenarios. Different interference levels do not change the fact that AC and HC experience more packet loss than EC, because both methods cannot handle hidden terminals. **Fairness:** MSC still remains the fairest among all four methods despite interference increase, because hierarchical MG-Local guides each path to consume its fair share of bandwidth. In comparison, AC, HC and EC consistently perform poorly in fairness with different interference levels. **End-to-end delay:** MSC achieves shorter end-to-end delay than AC, HC and EC under different interference levels. Additionally, MSC's delay decreases when interference level increases because of correlated congestion-collision control. In contrast, AC does not consider the impact of collisions on congestion, and therefore increases queueing delay with frequent retransmissions. Although HC's delay is shorter than MSC when 10 paths are used, this shorter delay is at the expense of sacrificing fairness.

In summary, Fig. 5.19 shows the results of varying network topologies with an offered load of 50 pps and interference level of 15 paths. These results further validate our analysis for the random topology experiments (see Section 5.4.2). This in-breadth study demonstrates that MSC achieves better packet loss rate, higher fairness, and shorter end-to-end delay in all three topologies. Adaptive CSMA suffers from high packet loss, which compromises its fairness and end-to-end delay. CSMA/CA-ED is superior to CSMA/CA-HD in terms of reducing packet loss. Both EC and HC perform poorly in fairness and end-to-end

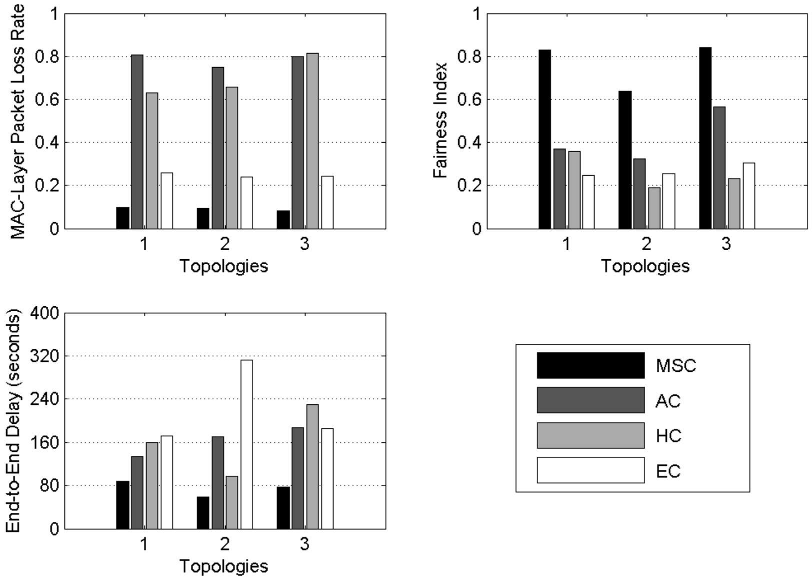

Figure 5.19: The Impact of Network Topologies

delay in all three topologies.

5.5 Chapter Summary

Multi-hop forwarding extends wireless coverage at the expense of degrading performance. To tackle unique multi-hop challenges, we propose a novel solution. This solution **first** offers hybrid TDMA/CSMA as a novel medium access control method to handle co-existing intra-path and inter-path interference. Hybrid TDMA/CSMA introduces a very simple and pure local per-path schedule algorithm to coordinate transmissions within the same path; solves schedule conflicts caused by common nodes by coordinating different paths to share conflicting schedules via randomization and controlled two-hop broadcasting; handles various temporal and spatial contention by a slotted adjustable CSMA/CA. **Second,** this solution effectively manages chaotic resource competition via hierarchical MG-Local. This method allocates limited bandwidth by taking into account fairness criteria, dynamic interference and scheduling efficiency; guides each path to efficiently utilize allocated resources by controlling their behaviors in terms of transmission, collisions and unnecessary idling. Its hierarchical control structure first determines the target states of G-Local optimization, and then uses these target states to guide multivariable control in terms of reducing the difference between a desired state and corresponding multivariable model. **Third**, we develop correlated congestion-collision control to handle the mutual impact of congestion and collisions in wireless networks. By throttling excessive traffic and reducing collisions, this method effectively reduces congestion. Our simulation results demonstrate that this solution

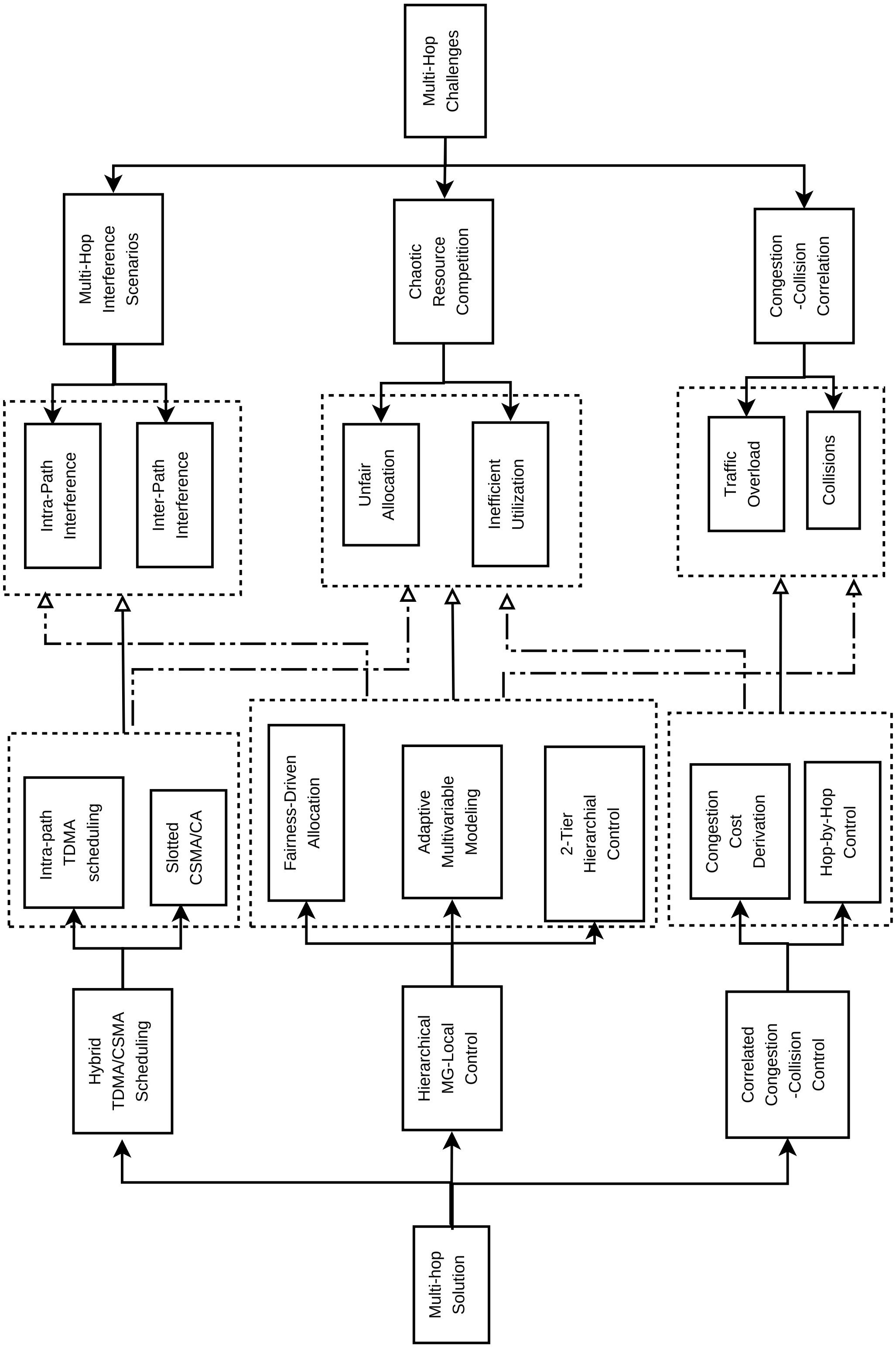
Multi-Hop Challenges
Multi-Hop Interference Scenarios
Chaotic Resource Competition
Congestion -Collision Correlation
Intra-Path Interference
Inter-Path Interference
Unfair Allocation
Inefficient Utilization
Traffic Overload
Collisions
Intra-path TDMA scheduling
Slotted CSMA/CA
Fairness-Driven Allocation
Adaptive Multivariable Modeling
2-Tier Hierarchial Control
Congestion Cost Derivation
Hop-by-Hop Control
Hybrid TDMA/CSMA Scheduling
Hierarchical MG-Local Control
Correlated Congestion -Collision Control
Multi-hop Solution

Figure 5.20: Multi-Hop Solution Structure

achieves lower packet loss rate, better end-to-end delay and fairness with competitive network throughput, compared to three alternative methods. The proposed multi-hop solution

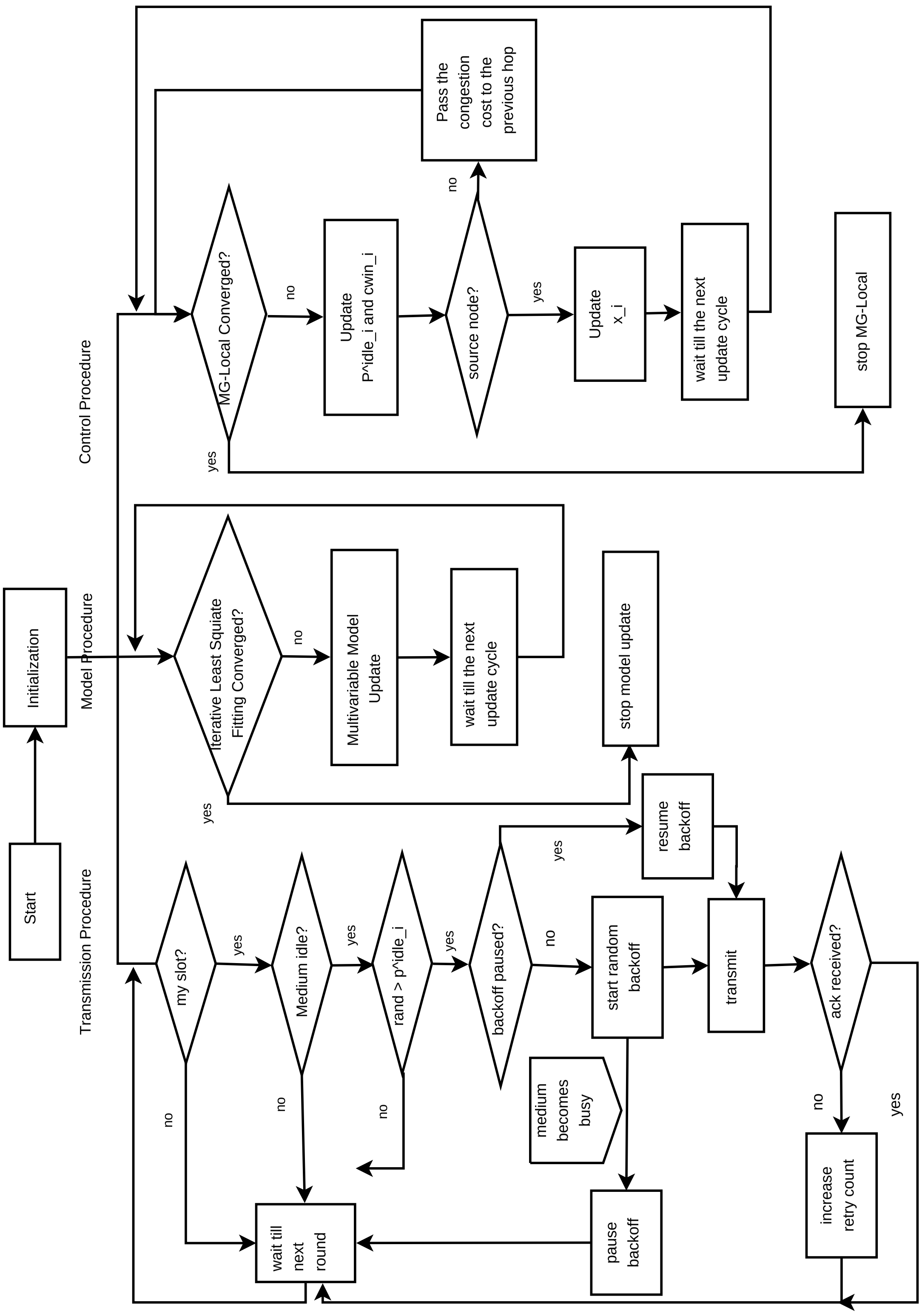

Figure 5.21: System Flow Diagram

requires message passing in two places. First, a two-hop controlled broadcasting is needed to solve schedule conflicts. Second, correlated congestion-collision control requires a node to pass its congestion cost to the previous hop on the same path. In the future work, we will explore cross-layer optimization to improve network throughput. The plan is described in the conclusion chapter.

Chapter 6

Conclusions

Wireless networking is a key enabling technology to ubiquitous computing and Internet access. The proliferation of new applications and rapid growth of wireless traffic poses great challenges for current technologies. To bridge the gap between the increasing user demand and limited delivery capability of wireless networks, we propose a novel framework of resource management in this thesis. This section summarizes our contributions, describes potential applications and outlines future directions.

6.1 Contributions

The proposed G-Local framework makes wireless networks more tractable, so that network operators/managers can allocate resources according to different user and system requirements, improve resource utilization to provide better network service, and reduce overhead in terms of control message passing and computational complexity. Specifically, our major contributions are:

- **G-Local Optimization** is a novel method of resource allocation that drives user consumption toward a desired fair share. G-Local optimization re-adjusts this level when it is not achievable due to dynamic interference, or imperfect scheduling. The re-adjustment is based on two cost functions of collisions and unnecessary idling. By intelligently exploiting local information, G-Local Optimization enables each user to derive the level of competition, and jointly maximizes resource utilization and minimizes the total consumption cost. Compared with the state-of-art global optimization algorithm, G-Local optimization 1) supports different fairness criteria; 2) suppresses the co-existence of conflicts and waste, 3) provides fine-tunable trade-off between fairness and efficiency, and 4) approaches a global optimum via local optimization with zero message passing.

- **Adaptive Multivariable Control** effectively controls network behavior in terms of transmissions, collisions and idling. It reifies the G-Local optimization to steer a network toward its desired state. Our method: 1) categorizes interference scenarios into five basic types; 2) differentiates dynamic causes of interference in four dimensions: time, space, frequency and intensity; 3) selects key control variables; and 4) models basic network behaviors as functions of the selected factors. These functions consider the impact of each factor as well as their correlations, and are obtained via iterative least square fitting. Compared with deterministic model-based optimization and stochastic approximation-based optimization, Adaptive Multivariable Control is adaptable to dynamic network conditions and faster in convergence.

- **Hybrid TDMA/CSMA scheduling** handles the co-existence of intra-path and inter-path interference in multi-hop wireless networks. The basic idea includes 1) enforcing each path to coordinate internal transmissions to eliminate collisions caused by intra-path interference via time-division-based scheduling; and 2) applying G-Local Optimization to guide different paths to a corresponding fair share via random-access-based scheduling. Compared with existing methods which require complex computation and frequent control-message passing, our method is simple in computation, light in overhead, and flexible to traffic/network-condition changes. These advantages are achieved by enabling each path to derive its own schedule locally, and promptly adjusting existing schedules upon traffic pattern changes.

- **Correlated congestion-collision control** explicitly handles the mutual impact of congestion and collisions in wireless networks. Our method explicitly models and controls the correlation between congestion and collisions. With this correlation model, each node computes a congestion cost based on its incoming/outgoing traffic, and local interference level. This congestion cost is passed back to the upstream node to adjust its outgoing traffic in order to alleviate congestion at the downstream node. Compared with the TCP-style congestion control, which tends to under-utilize wireless resources by solely adjusting source data rate based on some end-to-end feedback information, our method considers the impact of both traffic overload and collisions. Compared with back-pressure congestion control, which aggravates interference and leads to more severe congestion by giving higher priorities to the nodes with more backlogged data, our method alleviates the negative impact of collisions on congestion.

6.2 Practical Applications

The G-Local framework is a generic method for managing resources in wireless networks. As a first step, we apply it in both single-hop and multi-hop wireless ad hoc networks with fixed resource demand. This framework can be customized for different systems and purposes. We discuss a few areas in which G-Local resource management could be applied.

6.2.1 Handling Diverse Resource Demand

The G-Local framework can be customized to manage systems where users have different resource demands, and these demands change over time. Essentially, G-Local provides a mechanism to drive user resource consumption toward a pre-configured target, and re-adjust this target when it is not achievable due to interference, imperfect scheduling, or over-asking/under-asking. To handle dynamic resource demands, user requirements are used to set the target, and we acknowledge that they may not be reasonable or achievable.

Due to the difficulty of frequently recomputing fair allocations upon demand changes, we specify a minimum resource guarantee x_{min} for all users. This minimum bandwidth guarantee is used as the baseline to set the fairness cost C_f of resource consumption. Therefore, this fairness criteria requires that a user has to pay a higher per-unit cost as it requests more than its minimum resource guarantee, x_{min}. If a user's request is lower than the minimum resource guarantee, the fairness cost is zero. We formulate this demand-oriented G-Local in Eq. 6.1.

$$
\begin{aligned}
&\max V(x_i, coll_i, idle_i)\ ,\ where\\
V\ &= k \cdot (\log x_i - \frac{1}{x_i^d} \cdot x_i) - (1-k) \cdot (C_i^f + \frac{coll_i}{B} - \frac{idle_i}{B})\\
C_i^f &= \frac{x_i^d - x_{min}}{B},\ if x_i^d \geq x_{min}\\
C_i^f &= 0,\ otherwise
\end{aligned}
\tag{6.1}
$$

In Eq. 6.1, x_i^d is the resource demand of user i. C_i^f is the fairness cost and increases with the x_i^d if $x_i^d \geq x_{min}$.

6.2.2 Improving Quality of Service (QoS)

The G-Local framework can be applied to improve quality of service by providing differentiated services. Given N classes of traffic (e.g. the traffic categorization specified in IEEE 802.16 [24]), we first assign each class m a unique function $U(x_i^m)$ that characterizes the user-perceived utility (e.g. the mean opinion score). This helps to control the aggressiveness of each class's bandwidth consumption. Second, each class m is associated with a minimum bandwidth guarantee (denoted x_{min}^m), and a user i in each class m requires a certain amount of bandwidth (d_i^m). Third, we replace x_{min} in Eq. 6.1 with x_{min}^m and formulate the G-Local framework for QoS in Eq. 6.2.

$$
\begin{aligned}
&\max V(x_i^m, coll_i^m, idle_i^m)\ ,\ where\\
V\ &= k \cdot (\log x_i^m - \frac{1}{d_i^m} \cdot x_i^m) - (1-k) \cdot (\frac{d_i^m - x_{min}^m}{B} + \frac{coll_i^m}{B} + \frac{idle_i^m}{B})
\end{aligned}
\tag{6.2}
$$

To fulfill the QoS-oriented G-Local given in Eq. 6.2, we need to identify key control parameters that have major influences on the effective bandwidth consumption, collisions and unnecessary idling of a medium access protocol such as IEEE 802.11a/b/g, 802.11e, 802.11n and 802.16 etc. In this thesis we demonstrated how to control IEEE 802.11b. We describe possible ways to control the other protocols as follows.

IEEE 802.11e

IEEE 802.11e extends IEEE 802.11 medium access control with an enhanced distributed channel access (EDCA) to support QoS [41]. EDCA differentiates traffic types and maps them to different priority levels. Each priority level has its own EDCA parameters including the arbitrary inter-frame space ($AIFS$), the contention window (CW), the transmission opportunity ($TXOP$). $TXOP$ specifies the maximum number of multiple packets a node can transmit. These parameters can be used as control variables to model the network behavior. According to the method we proposed in Chapter 5, we can derive corresponding control policies to fulfill Eq. 6.2.

IEEE 802.11n

The recent amendment of IEEE 802.11 is IEEE 802.11n for higher throughput operation [39]. Besides standardizing enhancements at the physical layer, new mechanisms of medium access control are also defined including: aggregation, block acknowledgment, and reverse direction mechanism. Despite various hardware and protocol enhancements to improve network throughput, the MAC layer is still the performance bottleneck because interference increases with the number of wireless stations [97]. By selecting key MAC parameters including the aggregation frame size, contention parameters (if EDCA or DCF is used), we can develop multivariable control policies for the QoS-oriented G-Local.

IEEE 802.16

IEEE 802.16 supports wireless mesh networks, and offers two communication modes - point to multipoint (PMP) and mesh [24]. In the mesh mode, the subscriber stations (SS) are eligible to communicate directly among themselves without any base stations (BS). The mesh mode of IEEE 802.16 supports two modes of scheduling: centralized and distributed. Our method can be applied to the distributed scheduling by controlling the transmission duration, and holdoff time which determines the waiting time between two consecutive transmissions.

6.2.3 Allocating Multiple Channels

The G-Local framework can also be applied to design a probability-based multi-channel scheduling algorithm as in Eq. 6.3. We specify an access probability $p_{i,j}$, where i is the user and j is the channel.

$$\begin{aligned} &\max \sum_{j=1}^{N} V(x_{i,j}, coll_{i,j}, idle_{i,j}) \ , \ where \\ V \ &= k \cdot (\log x_{i,j} - \frac{1}{d_i/N} \cdot x_{i,j}) - (1-k) \cdot (\frac{d_i/N - x_f/N}{B} + \frac{coll_{i,j}}{B} + \frac{idle_{i,j}}{B}) \end{aligned} \tag{6.3}$$

$$\begin{aligned} x_{i,j} &= R(p_{i,j}) \\ coll_{i,j} &= CL(p_{i,j}) \\ idle_{i,j} &= CL(p_{i,j}) \end{aligned} \tag{6.4}$$

6.2.4 Managing Energy Consumption

Besides channel resources (bandwidth and frequency), another application of the G-Local framework is to manage energy consumption. Compared with bandwidth, energy in mobile, battery-based devices decreases over time. Instead of using a pre-configured x_i^f, we can use a pre-configured energy consumption rate e_i^f, which corresponds to the average energy consumed by successful transmissions and receptions. We also define e_i^{coll} to characterize the energy wasted on collisions. The G-Local optimization for energy consumption is formulated in Eq. 6.5. We can select transmit power, and sleeping schedule parameters as the control variables, and model their impact on e_i and e_i^{coll} via the adaptive multivariable control method introduced in Chapter 4:

$$\begin{aligned} &\max V(e_i, e_i^{coll}) \ , \ where \\ V \ &= k \cdot (\log e_i - \frac{1}{e_i^f} \cdot e_i) - (1-k) \cdot (\frac{e_i^{coll}}{B}) \end{aligned} \tag{6.5}$$

6.3 Future Direction

In the future, we are interesting in exploring the following two directions.

6.3.1 Cross-Layer Optimization: Incorporating Routing and Agile Communication Patterns

Cross-layer design is a promising paradigm to bridge the gap between network performance and interoperability. Many papers have shown that cross-layer design is appropriate to optimize wireless networks, where tight interaction exists among the physical, link, network and transport layers. The recent "layering as network decomposition" theory [15] further establishes a foundation for cross-layer design in mathematics and control theory. To take advantage of this methodology, we will incorporate routing and agile communication patterns in the G-Local framework to achieve more efficient utilization of network resources. Routing is an important network functionality, which finds one or more than one multi-hop paths between a source-destination pair. Poor routing causes unbalanced load distribution and makes fair resource allocation impossible. Furthermore, in real networks, both traffic and network conditions change dynamically. Frequently recomputing routes to adapt to these changes can be very costly and incur very long delays. In comparison to frequent route update, agile communication patterns (e.g. code migration, traffic relay, peer-to-peer communication etc.) are more flexible and easier to develop local solutions which ultimately reduce control overhead. Our future plan is to explore the advantages of routing and agile communication patterns to pursue more practical and effective resource management.

6.3.2 Application to Infrastructure-Free and Infrastructure-Based Mobile Networks

Resource management is more challenging in mobile networks due to frequent topology changes, dynamic user-mobility patterns, diverse user QoS (quality of service) requirements, time-varying traffic patterns and link quality. Extending G-Local resource management to mobile networks is the second direction. We are interested in both infrastructure-free (e.g. mobile adhoc networks) and infrastructure-based (e.g. wireless mesh networks) mobile networks. Infrastructure-free mobile networks are widely used in emergency scenarios (e.g. first-responder communication, rescue missions, battlefields, etc.), and require robust, reliable and fast transmissions. Furthermore, although the size of such networks is usually small, frequent topology changes caused by user movement have to be properly handled. In comparison, infrastructure-based mobile networks has a fixed backbone that connects mobile users to different gateways. A typical example is where mobile users require Internet services in a municipal mesh network. Despite the fixed backbone, infrastructure-based networks have to balance traffic loads from a significant number of mobile users and satisfy diverse requirements. Applying the G-Local framework to benefit both types of mobile networks is a very interesting problem.

Bibliography

[1] Norman Abramson. The ALOHA systemanother alternative for computer communications. In *Proceedings of the 1970 Fall Joint Computer Conference, AFIPS Conference*, pages 281–285, 1970.

[2] Arup Acharya, Archan Misra, and Sorav Bansal. MACA-P: A mac for concurrent transmissions in multi-hop wireless networks. In *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications (PERCOM'03)*, pages 505–509, 2003.

[3] Dimitri Bertsekas and Robert Gallager. *Data Networks*. Prentice-Hall, EngelWood Cliffs, N.J., 1992.

[4] Vaduvur Bharghavan, Alan Demers, Scott Shenker, and Lixia Zhang. MACAW: A media access protocol for wireless LANs. In *Proceedings of the Conference on Communications Architectures Protocols and Applications*, pages 212–225, Octobor 1999.

[5] Giuseppe Bianchi. Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE Journal on Selected Areas in Communications*, 18(3):535–547, 2000.

[6] R. Boorstyn, A. Kershenbaum, B. Maglaris, and V. Sahin. Throughput analysis in multihop CSMA packet radio networks. *IEEE Transactions on Communications*, 35(3):267–274, 1987.

[7] Steven A. Borbash and Anthony Ephremides. Wireless link scheduling with power control and sinr constraints. *IEEE Transactions on Information Theory*, 52(11):5106–5111, 2006.

[8] Raffaele Bruno, Marco Conti, Enrico Gregori, and Romano Fantacci. Throughput vs. temporal fair MAC protocols in multi-rate WLANs: Analysis and performance evaluation. In *Proceedings of the Fifty-Ninth Vehicular Technology Conference (VTC'04)*, pages 2017– 2021, 2004.

[9] Zhijun Cai, Mi Lu, and C.N Georghiades. Topology-transparent time division multiple access broadcast scheduling in multihop packet radio networks. *IEEE Transactions on Vehicular Technology*, 52(4):970984, 2003.

[10] Frederico Cali, Marco Conti, and Enrico Gregori. IEEE 802.11 wireless LAN: Capacity analysis and protocol enhancement. In *Proceedings of the Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies*, pages 142–149, 1998.

[11] John I. Capetanakis. Tree algorithm for packet broadcast channels. *IEEE Transactions on Information Theory*, 25(5):505–515, 1979.

[12] Matteo Cesanax, Daniela Maniezzoy, Pierpaolo Bergamoy, and Mario Gerla. Interference aware (IA) MAC:an enhancement to IEEE 802.11b DCF. In *Proceedings of the IEEE 58th Vehicular Technology Conference (VTC'03)*, pages 2799–2803, 2003.

[13] Prasanna Chaporkar, Koushik Kar, and Saswati Sarkar. Throughput gaurantees through maximal scheduling in wireless networks. In *Proceedings of 43rd Anuual Allerton Conference on Communication, Control and Computing*, pages 28–30, 2005.

[14] Chun cheng Chen, Eunsoo Seo, Hwangnam Kim, and Haiyun Luo. Select: Self-learning collision avoidance for wireless networks. *IEEE Transactions on Mobile Computing*, 7(3):305–321, 2008.

[15] Mung Chiang, Steven H. Low, A.Robert Calderbank, and John C. Doyle. Layering as optimization decomposition: A mathematical theory of network architectures. In *Proceedings of the IEEE*, pages 255–312, 2007.

[16] Claudio Cicconetti, Luciano Lenzini, and Enzo Mingozzi. Scheduling and dynamic relocation for IEEE 802.11s mesh deterministic access. In *Proceedings of the Fifth IEEE Communication Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON'08)*, pages 19–27, 2008.

[17] Cisco. Cisco visual networking index: Forecast and methodology, 2010-2015, 2010.

[18] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, Third Edition.* MIT Press, U.S.A., 2009.

[19] Jean-Pierre Corriou. *Process Control: Theory and Applications.* Springer-Verlag, London, UK, 2004.

[20] Javier del Prado Pavon and Sunghyun Choi. Link adaptation strategy for IEEE 802.11 WLAN via received signal strength measurement. In *Proceedings of 2003 IEEE International Conference on Communications (ICC03)*, pages 1108–1113, 2003.

[21] Jing Deng, Ben Liang, and P.K Varshney. Tuning the carrier sensing range of IEEE 802.11 MAC. In *Proceedings of the 47th IEEE Global Telecommunications Conference (GLOBECOM 2004)*, pages 2987–2991, December 2004.

[22] Nico M. Van Dijk. *Queueing Networks: A Fundamental Approach.* Springer, New York, US, 2011.

[23] Petar Djukic and Shahrokh Valaee. Delay aware link scheduling for multi-hop TDMA wireless networks. *IEEE/ACM Transactions on Networking*, 17(3):870–883, 2009.

[24] Carl Eklund, Kenneth L. Stanwood, and Stanley Wang. Ieee standard 802.16: A technical overview of the wirelessman air interface for broadband wireless access.

[25] Anthony Ephremedis and Trungdung Truong. A distributed algorithm for efficient and interference free broadcasting in radio networks,. In *Proceedings of the seventh ACM international symposium on Mobile ad hoc networking and computing (MobiHoc'06)*, pages 190–201, 2006.

[26] Kevin Fall and Kannan Varadhan. The ns mannual. http://www.isi.edu/nsnam/ns/doc/ns_doc.pdf, 2011.

[27] Chane L. Fullmer and J.J. Garcia-Luna-Aceves. FAMA-PJ: A channel access protocol for wireless LANs. In *Proceedings of the 1995 ACM Mobile Computing and Networking Conference*, page 10, 1995.

[28] J.J. Garcia-Luna-Aceves and Chane L. Fullmer. Floor acquisition multiple access (FAMA) in single-channel wireless networks. *Mobile Networks and Applications*, 4(3):157–174, 1999.

[29] J.J. Garcia-Luna-Aceves and Asimakis Tzamaloukas. Receiver-initiated collision avoidance in wireless networks. *Wireless Networks*, 8(2/3):249–263, 2002.

[30] David Garcia-Roger, Maria Jose Domenech-Benlloch, Jorge Martinez-Bauset, and Vicent Pla. Adaptive admission control in mobile cellular networks with streaming and elastic traffic. In *Proceedings of the 19th International Teletraffic Congress (ITC'07)*, pages 925–937, 2007.

[31] Jimmi Grnkvist. Traffic controlled spatial reuse tdma in multi-hop radio networks. In *Proceedings of the Ninth IEEE International Symposium on Personal, Indoor and Mobile Radio Communication*, pages 1203–1207, 1998.

[32] Jimmi Grnkvist. Comparison between scheduling models for spatial reuse tdma. In *Proceedings of the Second Anual Workshop of Affordable Wireless Services and Infrastructurensor Technologies (AWSI'04)*, page 5, 2004.

[33] Piyush Gupta and Panganamala R. Kumar. The capacity of wireless networks. *IEEE Transactions on Information Theory*, 46(2):388–404, 2000.

[34] Bruce Hajek and Galen Sasaki. Link scheduling in polynomial time. *IEEE Transactions on Information Theory*, 34(5):910–917, 1988.

[35] Martin Heusse, Franck Rousseau, Romaric Guilier, and Andrzej Duda. Idle sense: An optimal access method for high throughput and fairness in rate diverse wireless LANs. In *Proceedings of the 2005 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 121–132, 2005.

[36] IEEE. *IEEE Standard 802.11-1999, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, 1999.

[37] IEEE. *IEEE Standard 802.15.4d-1999, Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)*, 1999.

[38] IEEE. *802.3ae-2002 - IEEE Standard for Information technology– Local and metropolitan area networks– Part 3: CSMA/CD Access Method and Physical Layer Specifications - Media Access Control (MAC) Parameters, Physical Layer, and Management Parameters for 10 Gb/s Operation*, 2002.

[39] IEEE. *802.11n-2009 - IEEE Standard for Information technology– Local and metropolitan area networks– Specific requirements– Part 11: Wireless LAN Medium Access Control (MAC)and Physical Layer (PHY) Specifications Amendment 5: Enhancements for Higher Throughput*, 2009.

[40] IEEE. *IEEE Standard for Local and metropolitan area networks– Part 16: Air Interface for Broadband Wireless Access Systems Amendment 3: Advanced Air Interface*, 2009.

[41] Inanc Inan, Feyza Keceli, and Ender Ayanoglu. Analysis of the 802.11e enhanced distributed channel access function. *IEEE Transactions On Communications*, 57(6):1753–1764, 2009.

[42] Raj Jain. *The Art of Computer Systems Performance Analysis:Techniques for Experimental Design, Measurement, Simulation, and Modeling.* Wiley Computer Publishing, New York, U.S.A., 1991.

[43] Rajendera K. Jain, Dah-Ming Chiu, and William R. Hawe. A quantitative measure of fairness and discrimination for resource allocation in shared computer systems. Technical Report TR-301, DEC Research, 1984.

[44] Kyle Jamieson, Hari Balakrishnan, and Y. C. Tay. Sift: A mac protocol for event-driven wireless sensor networks. In *Proceedings of the Third European Workshop on Wireless Sensor Networks*, pages 1–23, 2006.

[45] Dong Geun Jeong and Wha Sook Jeon. Performance of an exponential backoff scheme for slotted-ALOHA protocol in local wireless environment. *IEEE Transactions on Vehicular Technology*, 44(3):470–479, 1995.

[46] Li Bin Jiang and Soung Chang Liew. Hidden-node removal and its application in cellular wifi networks. *IEEE Transactions on Vehicular Technology*, 56(5):2641–2654, 2007.

[47] Li Bin Jiang and Jean Walrand. Approaching throughput-optimality in distributed CSMA scheduling algorithm with collisions. *IEEE/ACM Transactions on Networking*, 19(3):816–829, 2011.

[48] Li Bin Jiang and Jean Walrand. A distributed CSMA algorithm for throughput and utility maximization in wireless networks. In *IEEE/ACM Transactions on Networking*, pages 357–362, 2011.

[49] Changhee Joo, Xiaojun Lin, and Nes B. Shroff. Understanding the capacity region of the greedy maximal scheduling algorithms in multi-hop wireless networks. *IEEE/ACM Transactions on Networking*, 17(4):1132–1145, 2009.

[50] Akimitsu Kanzaki, Takahiro Hara, and Shojiro Nishio. An adaptive tdma slot assignment protocol in ad hoc sensor networks. In *Proceedings of the 2005 ACM symposium on Applied computing*, pages 1160–1165, 2005.

[51] Akimitsu Kanzaki, Toshiaki Uemukai, Takahiro Hara, and Shojiro Nishio. Dynamic TDMA slot assignment for ad hoc networks. In *Proceedings of the 2003 International Conference on Advanced Information Networking and Appications (AINA'03)*, pages 330–339, 2003.

[52] Phil Karn. MACA – a new channel access method for packet radio. In *Proceedings of In ARRL/CRRL Amateur Radio 9th Computer Networking Conference*, pages 134–140, 1990.

[53] Elli Kartsakli, Jesus Alonso-Zarate, Alex Cateura, Christos Verikoukis, and Luis Alonso. *Contention-Based Collision-Resolution Medium Access Control Algorithms.* Nova Science Publishers Inc, Hauppauge, NY, US, 2009.

[54] Frank Kelly. Charging and rate control for elastic traffic. *European Transactions on Telecommunications*, 8(1):3337, 1996.

[55] EunKyung Kim and Young-Joo Suh. ATXOP: An adaptive TXOP based on the data rate to guarantee fairness for IEEE 802.11e wireless LANs. In *Proceedings of the Sixtieth Vehicular Technology Conference (VTC'04)*, pages 2678–2682, 2004.

[56] Leonard Kleinrock and Fouad A. Tobagi. Packet switching in radio channels: Part i-carrier sense multiple-access modes and their throughput-delay characteristics. *IEEE Transaction on Communications*, 23(12):1400–1416, 1975.

[57] Can Emre Koksal, Hisham Kassab, and Hari Balakrishnan. An analysis of short term fairness in wireless media access protocols. In *Proceedings of the 2000 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, pages 118– 130, 2000.

[58] Euihyeok Kwon, Doyun Hwang, and Jaesung Lim. An idle timeslot reuse scheme for ieee 802.15.3 high-rate wireless personal area networks. In *Proceedings of the IEEE 62nd Vehicular Technology Conference (VTC'05)*, pages 715–719, 2005.

[59] Peter Lancaster and Kes Salkauskas. *Curve and Surface Fitting: An Introduction.* Academic Press, London, UK, 1986.

[60] Hung-Cuong Le, Herv Guyennet, and Noureddine Zerhouni. A new contention access method for collision avoidance in wireless sensor networ. In *Proceedings of the Sixth International Conference on Networking (ICN'07)*, page 27, 2007.

[61] Jang-Won Lee, Mung Chiang, and Robert Calderbank. Utility-optimal medium access control: Reverse and forward engineering. In *Proceedings of the 25th IEEE International Conference on Computer Communications (INFOCOM'06)*, pages 1–13, 2006.

[62] Jinsung Lee, Junhee Lee, Yung Yi, Song Chong, Alexandre Proutiere, and Mung Chiang. Implementing utility-optimal CSMA. In *Proceedings of the 47th Annual Allerton Conference on Communication, Control, and Computing*, pages 102–111, 2009.

[63] Xiaojun Lin, Ness Shroff, and Rayadurgam Srikant. A tutorial on cross-layer optimization in wireless networks. *IEEE Journal on Selected Areas in Communications*, 24(8):1452–1463, 2006.

[64] Chen Liu, Mike MacGregor, and Janelle Harms. Improving multi-path routing performance in WSNs by tuning IEEE 802.11 parameters. In *Proceedings of 6th ACM International Symposium on Mobility Management and Wireless Access*, pages 142–146, 2008.

[65] Chen Liu, Mike MacGregor, and Janelle Harms. Optimal control of spatial, temporal and bandwidth contention in wireless ad hoc networks. In *Proceedings of 2009 IEEE International Conference on Wireless and Mobile Computing, Networking and Communications*, pages 263–269, 2009.

[66] Feilu Liu, Zhifeng Tao, Thanasis Korakis, Elza Erkip, and Shivendra Panwar. The hidden cost of hidden terminals. In *Proceedings of 2010 IEEE International Conference on Communications*, pages 1–6, 2010.

[67] Miguel Luis, Rodolfo Oliveira, Luis Bernardo, and Rui Dinis. Maximizing throughput-fairness tradeoff in MAC for ad hoc networks. In *Proceedings of the 2011 IEEE Conference of Wireless Communications and Networking Conference*, pages 357–362, 2011.

[68] Hui Ma, Jing Zhu, Sumit Roy, and Soo Young Shin. Joint transmit power and physical carrier sensing adaptation based on loss differentiation for high density IEEE 802.11 WLAN. *Computer Networks: The International Journal of Computer and Telecommunications Networking*, 52(9):1703–1720, 2008.

[69] James L. Massey. Collision-resolution algorithms and random-access communications. *Multiuser Communications.*

[70] Peter Mathys and Philippe Flajolet. Q-ary collision resolution algorithms in random-access systems with free or blocked channel access. *IEEE Transactions on Information Theory*, 31(2):217–234, 1985.

[71] Jeonghoon Mo and Jean Walrand. Fair end-to-end window-based congestion control. *IEEE/ACM Transactions on Networking*, 8(5):556–567, 2000.

[72] Christelle Molle, Fabrice Peix, and Herv Rivano. Cross-layer design for wireless mesh networks using column generation. Technical report, Institut National de Recherche en Informatique et Automatique (INRIA), 2008.

[73] KW Morton and David Mayers. *Numerical Solution of Partial Differential Equations, An Introduction.* Cambridge University Press, UK, 2005.

[74] Andrew Muir and J.J. Garcia-Luna-Aceves. An efficient packet sensing mac protocol for wireless networks. *Mobile Networks and Applications*, 52(4):221234, 1998.

[75] Thyagarajan Nandagopal, Tae-Eun Kim, Xia Gao, and Vaduvur Bharghavan. Achieving MAC layer fairness in wireless packet networks. In *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, pages 87–98, 2000.

[76] Jian Ni, Bo Tan, and Rayadurgam Srikant. Q-CSMA: queue-length based CSMA/CA algorithms for achieving maximum throughput and low delay in wireless networks. In *Proceedings of the 29th conference on Information communications*, page 250, March 2009.

[77] Paul Patrass, Albert Banchs, Pablo Serrano, and Arturo Azcorra. A control-theoretic approach to distributed optimal configuration of 802.11 WLANs. *IEEE Journal on Mobile Computing*, 10(6):897–910, 2011.

[78] Jun Peng, Liang Cheng, and Biplab Sikdar. A wireless MAC protocol with collision detection. *IEEE Transaction on Mobile Computing*, 6(12):1357–1369, 2007.

[79] Craig Prohazka. Decoupling link scheduling constraints in multihop packet radio networks. *IEEE Transactions on Computers*, 38(3):455458, 1989.

[80] Subramanian Ramanathan and Errol L. Lloyd. Scheduling algorithms for multihop radio networks. *IEEE Transactions on Networking*, 1(2):166–178, 1993.

[81] Injong Rhee, Ajit Warrier, Jeongki Min, and Lisong Xu. DRAND: Distributed randomized TDMA scheduling for wireless ad-hoc networks. In *Proceedings of the seventh ACM international symposium on Mobile ad hoc networking and computing (MobiHoc'06)*, pages 190–201, 2006.

[82] Lawrence G. Roberts. ALOHA packet system with and without slots and capture. *ACM SIGCOMM Computer Communication Review*, 5(2):28–42, 1975.

[83] R. Rozovsky and Panganamala Kumar. SEEDEX: A MAC protocol for ad hoc networks. In *Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking and computing*, pages 67–75, 2001.

[84] Claude Shannon. Communication in the presence of noise. In *Proceedings of the Institute of Radio Engineers*, pages 10–21, 1949.

[85] Nirmala Shenoy, Xiaojun Cao, Yoshihiro Nozaki, Stefan Hild, and Paul Chou. Neighbor turn taking mac a loosely scheduled access protocol for wireless networks. In *Proceedings of the IEEE 18th Symposium on Personal, Indoor and Mobile Radio Communications (PIMR'07)*, pages 1–5, 2007.

[86] Jingpu Shi, Ehsan Aryafar, Theodoros Salonidis, and Edward W. Knightly. Synchronized CSMA contention: Model, implementation and evaluation. In *Proceedings of the 29th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'09)*, pages 2052–2060, 2009.

[87] Julie Shor and Thomas G. Robertazzi. Traffic sensitive algorithms and performance measures for the generation of self-organizing radio network schedules. *IEEE Transactions On Communications*, 41(1):1621, 1993.

[88] Joao Luis Sobrinho, Roland de Haan, and Jose Manuel Brazio. Why RTS-CTS is not your ideal wireless LAN multiple access protocol. In *Proceedings of 2005 IEEE Wireless Communications and Networking Conference (WCNC'05)*, pages 81–87, March 2005.

[89] James Spall. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Transactions on Automatic Control*, 37(3):332–347, 1992.

[90] Rayadurgam Srikant. *The Mathematics of Internet Congestion Control*. Birkhuser, USA, 2004.

[91] Paul Starzetz, Martin Heusse, Franck Rousseau, and Andrzej Duda. Hashing backoff: A collision-free wireless access method. In *Proceedings of the 2009 IFIP International Conferences on Networking*, pages 429–441, 2009.

[92] Christian Schlegel Sumeeth Nagaraj, Sheehan Khan and Marat Burnashev. Differential preamble detection in packet-based wireless networks. *IEEE Transactions on Wireless Communications*, 8(2):599–607, 2009.

[93] Fabrizio Talucci, Mario Gerla, and Luigi Fratta. MACA-BI (MACA by invitation) a receiver oriented access protocol for wireless multihop networks. In *Proceedings of the 1997 IEEE Personal, Indoor and Mobile Radio Communications Symposium (PIMRC'97)*, pages 1–5, 1997.

[94] Andrew S. Tanenbaum. *Computer Networks (The Fourth Edition)*. Prentice Hall, U.S.A., 2002.

[95] Leandros Tassiulas and Anthony Ephremides. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. *IEEE Transactions on Automatic Control*, 36(12):1936–1948, 1992.

[96] Leandros Tassiulas and Saswati Sarkar. Maxmin fair scheduling in wireless networks. In *Proceedings of the 22th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'02)*, pages 763–772, 2002.

[97] Selvam Thanigasalam and S. Sricant. A frame aggregation scheduler for IEEE 802.11n. In *Proceedings of National Conference on Communications*, pages 1–5, 2010.

[98] Jean Tourrilhes. Piggydata: Reducing CSMA/CA collisions for multimedia and TCP connections. In *Proceedings of the Fiftieth Vehicular Technology Conference (VTC'99)*, pages 1675–1679, 1999.

[99] David Tse and Pramod Viswanath. *Fundamentals of Wireless Communication*. Cambridge University Press, Cambridge, UK, 2005.

[100] Vasileios Vitsas. Throughput analysis of a linear bakcoff scheme in wireless LANs. *IEEE Electronic Letters*, 39(1):99–100, 2003.

[101] Weizhao Wang, Yu Wang, XiangYang Li, WenZhan Song, and Ophir Frieder. Efficient interferenceaware tdma link scheduling for static wireless networks. In *Proceedings of the 12th Annual International Conference on Mobile Computing and Networking*, pages 262–273, 2006.

[102] Xin Wang. Throughput modelling and fairness issues in CSMA/CA based ad-hoc networks. In *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'05)*, pages 23–34, 2005.

[103] Xinzhou Wu, Rayadurgam Srikant, and James Perkins. Queue-length stability of maximal greedy schedules in wireless networks. In *Proceedings of Information Theory and Applications Inaugural Workshop*, pages 6–10, 2006.

[104] MyungJune Youn, Young-Yul Oh, Jaiyong Lee, and Yeonsoo Kim. IEEE 802.15.4 based qos support slotted CSMA/CA MAC protocol for wireless sensor networks. In *Proceedings of 2007 International Conference on Sensor Technologies and Applications*, pages 113–117, 2007.

[105] David Young. USAP: A unifying dynamic distributed multichannel TDMA slot assignment protocol. In *Proceedings of the 1996 Military Communication Conference (MILCOM'96)*, pages 235–239, 1996.

[106] David Young. USAP multiple access: Dynamic resource allocation for mobile multihop multichannel wireless networking. In *Proceedings of the 1999 Military Communication Conference (MILCOM'99)*, pages 271–275, 1999.

[107] Ahmad Zahedi and Kaveh Pahlavan. Throughput of a wireless LAN access point in presence of natural hidden terminals and capture effects. In *Proceedings of the Seventh IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, pages 397–401, 1996.

[108] Guokai Zeng, Yinlong Xu, Yafeng Wu, and Xi Wang. Weighted flow contention graph and its applications in wireless ad hoc networks. *Lecture Notes in Computer Science (Networking ICN 2005)*, 3421.