On Channel Access Design for Wireless Networks with Multi-Packet Reception

by

Ke Li

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Department of Computing Science
University of Alberta

Abstract

As wireless devices have emerged as a ubiquitous part of people's everyday lives, the demands for faster wireless communications become even more pressing. Fortunately, the advanced techniques of the physical layer such as multiple-input and multiple-output (MIMO), multi-user detection (MUD), advanced modulation, etc., make multi-packet transmission (MPT) and multi-packet reception (MPR) possible. It has been well recognized that the MPT/MPR technique can improve the performance of the wireless networks. However, novel algorithms at the medium-access control (MAC) and higher layers are needed to fully exploit the MPT/MPR capability. In this thesis, we study the behavior the MPT/MPR wireless network, evaluate its potential performance and design algorithms to efficiently and fairly manage the MPT/MPR networks.

We start from a single-hop scenario where uncoordinated nodes share a MPR channel and assess its performance by designing additive-increase multiplicative-decrease MAC (AIMD-MAC) to achieve the max-min fairness. We show that with an appropriate set of parameters, AIMD-MAC can be applied to distributed environments where the number of nodes and channel capacity are not constant to achieve at least 90% of the performance of the benchmark.

For multi-hop scenarios, we observe the M property of MPT/MPR networks, which profoundly changes the traditional understanding of managing a multi-hop wireless network. By identifying and investigating the M property, we propose novel algorithms to evaluate the MPT/MPR networks and demonstrate the relative importance of the MPT and MPR capacity limits. To efficiently manage the multi-hop flows traversing a MPT/MPR network, we design the AIMD backpressure MAC (AB-MAC) algorithm. Extensive simulations show that AB-MAC significantly outperforms IEEE 802.11 especially in dense networks.

To my husband Yang Chen and my parents Guoyuan Li and Qiongfang Lang.

Acknowledgements

There are so many people I would like to thank. Without their support, this work would not have been possible.

Foremost, I would like to express my deepest gratitude to my supervisors Dr. Ioanis Nikolaidis, Janelle Harms and Christian Schlegel. Without their tremendous support, guidance and advice I could not complete the thesis. No word can express my gratitude and appreciation to them. I am also very thankful to the examining committee members: Dr. Ekram Hossain, Hong Vicky Zhao, Martin Jagersand, Mike MacGregor and Csaba Szepesvari. Thanks for their valuable suggestions for the thesis.

I would like to thank Dr. Martha Streenstrup and Ehab Elmallah for the advice at our network group meetings. I would also like to thank Dr. Majid Ghanbarinejad and Dimitry Trukhachev for all the discussions and suggestions for my early work. My thanks go to my colleagures Dr. Baljeet Singh Malhotra, Israat Tanzeena Haque, Chen Liu, Zhiyu Wang, Mohamed Shazly who never hesitate to help when I countered problems.

I cannot thank enough my dad Guoyuan Li and my mom Qiongfang Lang for their understanding and unconditional support. I would also like to thank my brother Han Li. Thanks for taking care of our parents and keeping them companied for all those years. I thank my husband, Dr. Yang Chen. Your unconditional love and support are essential to my success today.

Finally, I would like to thank the financial support from the Computing Science Department at the University of Alberta, Provost Doctoral Entrance award from the University of Alberta and Profiling Alberta's Graduate Students Award from the Faculty of Graduate Studies at the University of Alberta.

Contents

List of Tables

List of Figures

List of Acronyms

AB-MAC additive-increase multiplicative-decrease backpressure MAC

AF after failure

AIMD additive-increase multiplicative-decrease

AP access probability

AS after success

BER bit-error rate

BP backpressure

CA collision avoidance

CDMA code division multiple access

CS carrier sense

CSMA/CA carrier sense multiple access with collision avoidance

CSMA Carrier-sense multiple-access

CTS confirm-to-send

CW contention window

CW contention window

CWND congestion window

DCA dynamic channel assignment

DEI destructive interference

ECG extended contention graph

EWMA exponentially weighted moving average

FDMA frequency-division multiple-access

GAIMD general AIMD

GAIMD general AIMD

GDP generic distributed probabilistic

GSM Global System Mobile

IMT International Mobile Telecommunications

IoE internet of everything

LAN local area network

LP linear programming

MACA Multiple access with collision avoidance

MAC medium-access control

MAI multiple-access interference

MC maximal clique

MIMO multiple-input and multiple-output

MPR multi-packet reception

MPT multi-packet transmission

MQSR multiqueue service room

MUD multi-user detection

NAV network allocation vector

OPET optimum packet scheduling for each traffic flow

PHY physical layer

PPS packets per slot

QOS quality of service

RBAR receiver-based autorate

RP-CDMA Random Packet CDMA

RTS request-to-send

RTT round-trip time

SIC successive interference cancellation

SNR signal to noise ratio

TDMA time-division multiple-access

TQD transmission quota distribution

WF-ASYM asymmetric wireless water-filling

WF wireless water-filling

W-LAN wireless local area networks

Chapter 1

Introduction

As wireless devices, such as laptops, tablets, smartphones, etc, have emerged as a ubiquitous part of people's everyday lives, the demands for faster wireless communications become even more pressing and designing protocols and mechanisms to exploit increased access capability is necessary.

A brief overview of the evolution of wireless networks in the past 20 years can remind us about the dramatic increase in data rate of wireless communications from a few Kbps typically in 90s to hundreds of Mbps in 2010. During the days of 1G technologies in the early 1980s, people relied on analog systems for voice calls. In the early 90s, we saw the rise of the Global System Mobile (GSM) and code division multiple access (CDMA) which allowed the 2G standards to reach a theoretically maximum speed of 50 Kbps [43]. For 3G network systems, the International Mobile Telecommunications (IMT)-2000 standard [45] specifies that the minimum transmission data rates should be 2 Mbps for stationary or walking users and 384 Kbps for a moving vehicle. In March 2008, the IMT-advanced specification [44] set peak speed requirements for 4G service to 100 Mbps for high mobility communication and 1 Gbps for low mobility communication.

On the side of demand, the explosive growth of cloud, social media, mobile computing and BigData increased the demand for higher speed networks to a whole new level. It has been forecast that the world will become the internet of everything (IoE) by 2020 with 25 to 50 billion devices connected to Internet [14,67]. However, in 2013, a report from Cisco estimated that 99.4 percent of physical objects that may one day be part of the IoE are still unconnected [14].

In order to meet the increasing demand of transmission rates, it is important to fully utilize advanced techniques of the physical layer (PHY) such as multiple-input and multiple-output (MIMO), multi-user detection (MUD), advanced modulation, etc., which make the simultaneous multi-packet transmission (MPT) [8, 38, 101] and

multi-packet reception (MPR) [19, 20, 30, 60, 83, 111] possible. It has been theoretically demonstrated that, with MPT and MPR techniques, the network capacity can improve, compared to the traditional collision channel which can only accommodate one transmission at a time from the receiver's perspective [28, 52, 81].

While the advancements in signal processing techniques introduce the potential to further improve the performance of wireless communication systems, it is unlikely that this improvement can be achieved by the physical layer alone. The channel model which influences the development of higher layer (medium-access control (MAC) and above) protocols is known as the *collision channel model*. It assumes that for the desired transmission, all other concurrent transmissions in the communication range are considered as destructive interference (DEI), or to simplify, noise. A large number of protocols have been proposed for the collision channel model [3, 17, 23, 26, 38, 50, 55, 94, 107] based on the principle that at most one packet transmission should occur within the receive range of the intended receiver. For instance, numerous protocols [11, 53, 103, 104, 110, 112] have been designed based on the carrier sense multiple access with collision avoidance (CSMA/CA) and/or exponential backoff mechanisms in wireless local area network (LAN)s or multi-hop ad hoc networks. Certain protocols apply the request-to-send (RTS)/confirm-to-send (CTS) handshake prior to data transfer in order to alleviate the so called hidden and exposed terminal problems and "reserve" the channel before a transmission starts. Any other nodes having received RTS and/or CTS inhibit their transmissions for a specified time period.

However, this principle of the collision channel model is too restrictive in wireless networks with MPT/MPR capabilities. Applying multiple-access interference (MAI)-avoidance protocols designed for collision channels does not exploit the capabilities of MPT/MPR networks because their primary objective is to reduce the MAI and prevent overlapping transmissions from happening. Therefore, novel algorithms are needed to embrace the MAI and encourage concurrent transmissions to the extent allowed by MPT/MPR. This thesis is concerned with protocols able to exploit the MPT/MPR capability. In MPT/MPR systems, the problem is not simply to decide which pair of nodes can access the channel. Instead, the problem is to decide the set of transmissions that can occur simultaneously. The size of the set should be regulated by the MPT and MPR limits. Exceeding the MPR limit results in collision potentially destructive to all concurrent transmissions. Equally, if the number of concurrent transmissions is less than MPR, the channel resource is underutilized. Additionally, if a number of multi-hop flows traverse the network, care should be taken to decide the content of the set, *i.e.,* how many packets should be chosen from

each flow, which will significantly affect the efficiency and fairness of the system, especially in multi-hop scenarios.

The fundamental conflict between fairness and efficiency has been observed by researchers [42, 58, 62] in multi-hop networks with conventional channel models (or equivalently, collision channel models). Both in the designs for single-hop flows [42,62] and in multi-hop flows [58], it is necessary to support fair channel access; otherwise, certain flows will be poorly treated or starved even if the total throughput is maximized. Additionally, the asymmetric information problem (which will be discussed in Section 2.3.2) suggests that the widely adopted CSMA/CA paradigm can result in severe unfair channel access among competitors. The max-min fairness is a well-known method to measure the fairness of wired networks [9] and was extended to wireless networks for single-hop flows by Huang *et al.* [42]. Regulating channel access of competing flows according to the max-min fairness prioritizes the mostly poorly treated flow. However, as we will discuss in Chapter 4, it is challenging to solve the max-min fairness problem in multi-hop MPT/MPR wireless networks.

The scope of the thesis is the design and performance evaluation of protocols in a time-slotted MPT/MPR wireless network capable of fully exploiting the MPT/MPR capabilities. Our contributions include:

- Starting from a single-hop scenario, we study how to achieve the max-min rate allocation among uncoordinated nodes contending for an MPR channel. This part of the study helps elucidate the performance of single-hop MPR systems. While immediate and error-less feedback is essential to achieve the optimal performance, it is also expensive to implement in a distributed environment. Therefore, we propose the AIMD-MAC algorithm to achieve the max-min fairness without needing feedback at every slot. Although this study is based on a static network scenario, we show that our algorithm can be applied to dynamic networks where the number of nodes and channel capacity are not constant.

- To study the multi-hop MPT/MPR wireless networks, we start from a centralized viewpoint and investigate the max-min fair problem. In order to compute the max-min rate allocation for conventional multi-hop wireless networks, a necessary and sufficient condition (Theorem 4.1) is proposed to determine whether a rate allocation is max-min fair. Based on Theorem 4.1, we develop a linear programming (LP) formulation, for use by the wireless water-filling (WF) algorithm, to compute the exact max-min rate allocation for the conventional multi-hop networks.

- The contention graph is widely utilized to describe the interference relationships among transmissions in conventional wireless networks [42]. Nonetheless, we observe that the contention graph of the MPT/MPR wireless network possesses characteristics that are different from its traditional understanding (*i.e.,* in conventional channels). More specifically, since more than one transmission can occur simultaneously even if they are interfering with each other, which is referred as *the M property* of the contention graph, the information contained in the contention graph alone is not sufficient to decide the feasibility of a schedule. We explore the MPT/MPR networks and study what information is needed to compute and schedule the max-min rate allocation.

- Based on the above observation, we examine the pair-wise interference relationship among links and design novel method to compute the max-min rate allocation for multi-hop MPT/MPR networks. We group the pair-wise interference relationships into eight patterns, based on which an approximation of the max-min allocation can be determined by the asymmetric wireless water-filling (WF-ASYM) algorithm we proposed. To realize the computed rate allocation, heuristic centralized scheduling algorithms, MDSATUR and LEX are proposed and evaluated for both homogeneous and non-homogeneous networks. To the author's best knowledge, the schedule construction problem in a multi-hop MPT/MPR wireless network has not been adequately covered in research literature thus far.

- Our study from the centralized perspective provides us with insights on the multi-hop MPT/MPR network. It demonstrates that the fair resource allocation is affected by all pair-wise interfering transmissions which are not necessarily traversing the same link. Additionally, numerical results show that increasing the value of the MPR limit to be greater than the MPT limit can improve the aggregate throughput while setting the MPT limit to be greater than that of MPR does not benefit the end-to-end throughput of a multi-hop flow. Based on these observations, we propose a MAC algorithm, the AB-MAC algorithm, for the applications in distributed systems, such as ad hoc networks. AB-MAC is designed, simulated and compared with the IEEE 802.11 standard in three simulation schemes. The three schemes are the mini-slot scheme, the random scheme and the prioritized scheme which demonstrate how different levels of node coordination during channel access can affect performance of AB-MAC and thus suggest potential directions to further improve AB-MAC.

This thesis is organized as follows. Background and related work on different

channel models are reviewed in Chapter 2. Chapter 3 presents the AIMD-MAC algorithm for a single-hop scenario with MPR channel. The multi-hop MPT/MPR wireless networks are studied by a centralized approach in Chapter 4 and the concrete schedule constructing algorithms are stated in Chapter 5. In Chapter 6, we propose a MAC algorithm to manage multi-hop flows in MPT/MPR wireless networks. Chapter 7 provides concluding remarks and outline of further research directions.

Chapter 2

Background and Related Work

In this chapter, we summarize various channel models most relevant to the development of our MPT/MPR models and compare their characteristics. Since the max-min fair rate allocation will be used as a benchmark in both single-hop and multi-hop wireless networks to evaluate the performance of the network system, we will discuss the difference between the conventional definition of the max-min fairness in wired networks and its counterpart in wireless networks. To demonstrate the challenges to solve the max-min problem in MPT/MPR wireless networks, we introduce *the M property* of the contention graph in MPT/MPR networks, which will have a persistent and significant impact in the subsequent exploration of MPT/MPR.

2.1 Evolution of Channel Models

We discuss and compare the collision channel models, the MPT/MPR channel models and the multi-channel models.

2.1.1 A Matrix Description of the Channel Models

The formal description of the multi-access channel models is presented in this section. Ghez *et al.* [31] generalize the collision model and describe the MPR channel by a stochastic matrix E (Eq. 2.1). Each element of the matrix ϵ_{nk} (Eq. 2.2) denotes the probability that k nodes can be decoded when n nodes are transmitting concurrently. The generalized channel model describes a more practical communication environment where certain transmitted signals can be decoded in the presence

of other signals.

$$E = \begin{bmatrix} \epsilon_{10} & \epsilon_{11} & & \\ \epsilon_{20} & \epsilon_{21} & \epsilon_{22} & \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \tag{2.1}$$

$$\epsilon_{nk} = P[k \text{ packets are correctly received} | n \text{ are transmitted}], n \geq 1, 0 \leq k \leq n \tag{2.2}$$

The ***collision channel model*** which can only accommodate one transmission can be described by Eq. 2.3. E_{COL} shows that if one packet is transmitted, *i.e.,* $n = 1$, one packet can be decoded; otherwise, *i.e.,* $n > 1$, zero packet can be decoded.

$$E_{COL} = \begin{bmatrix} 0 & 1 & 0 & \cdots \\ 1 & 0 & 0 & \cdots \\ 1 & 0 & 0 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \tag{2.3}$$

2.1.2 The K-MPR Channel Model

An interference outage model named the ***K-MPR*** channel model has been widely applied in protocol designs and theoretical analyses [20, 30, 35, 63]. If the number of concurrent transmissions does not exceed K, the receiver can extract all signals; on the other hand, if there are more than K signals arriving at the same time (an event we call ***interference outage***), a collision happens and all signals are destroyed. The matrix E_K (Eq. 2.4) captures the behavior of the K-MPR channel. It shows that $\epsilon_{nk} = 1$ if $n = k \leq K$; otherwise $\epsilon_{nk} = 0$.

$$E_K = \begin{bmatrix} 0 & 1 & 0 & & \\ 0 & 0 & 1 & & \\ \vdots & \vdots & \vdots & & \\ 0 & 0 & 0 & \cdots & 1 \\ 1 & 0 & 0 & & \\ \vdots & \vdots & \vdots & & \end{bmatrix} \tag{2.4}$$

The K-MPR channel model captures the important role of signal to noise ratio (SNR) of communication systems such as CDMA, assuming that every user's received power is balanced [86]. The normalized version of SNR is E_b/N_0 where E_b is bit energy and N_0 is noise power spectral density. E_b can be computed by dividing signal power S by bit rate R and N_0 is the ratio between noise power N and bandwidth W (Eq. 2.5) [86]. W/R is generally referred to as the ***processing gain***. A basic metric

measuring the performance of communications systems is bit-error rate (BER) which is a function of E_b/N_0. Therefore, this bit energy to noise power spectral density ratio (*i.e.,* E_b/N_0) is a standard quality measure to compare different digital communication systems.

$$\frac{E_b}{N_0} = \frac{S/R}{N/W} = \frac{S}{N}(\frac{W}{R}) \tag{2.5}$$

When K nodes are accessing the shared channel with equal power S, for any given node, there are $K-1$ interfering nodes with total interference equal to $(K-1)S$. Taking into account the interference, Eq. 2.5 can be rewritten as Eq. 2.6. For a communication system with certain required bit error rate (BER) and therefore, the corresponding E_b/N_0, the capacity of the system in terms of number of concurrent nodes can be denoted by K as in Eq. 2.7 [33].

$$\frac{E_b}{N_0} = \frac{W/R}{(K-1)+(N/S)} \tag{2.6}$$

$$K = 1 + \frac{W/R}{E_b/N_0} - N/S \tag{2.7}$$

Throughout this thesis, we assume that the MPR capability is realized by the K-MPR channel model with K indicating the channel capacity limit. Since we assume that the result of a reception is completely determined by the number of concurrent transmissions arriving at the receiver, from the receiver's viewpoint, the channel capacity is equivalent to the MPR capacity limit of the receiver.

2.1.3 MPT/MPR Channel Model

The MPR model extends the capacity of the receiving end. It has been theoretically demonstrated that in multi-hop scenarios, when MPR is coupled with the multi-packet transmission (MPT) capability, the system performance can be further improved [52]. We denote the transmitting and receiving capability limits by K_T and K_R, respectively, which means this transceiver can transmit K_T or receive K_R packets simultaneously. Since half-duplex radio is assumed throughout this thesis, a transceiver cannot transmit and receive at the same time. In a network system, if the transceiver's transmitting or receiving capacity limit is greater than 1 (*i.e.,* $K_T > 1$ or $K_R > 1$), we name the channel model of the network the ***MPT/MPR channel model***. In Chapter 4, we will study the behavior of multi-hop MPT/MPR wireless networks from a centralized viewpoint and design algorithms to compute and realize the max-min fairness allocation of a MPT/MPR system. In order to exploit the

MPT/MPR capability for distributed networks, in Chapter 6, we propose a MAC algorithm to manage multi-hop flows traversing a MPT/MPR network.

2.1.4 Comparison with the Multi-Channel Model

The multi-channel model is similar to the MPT/MPR channel in the sense that multiple transmissions can coexist. An example is the IEEE 802.11b [23] which provides configuration options of up to 14 channels. We use an example to elaborate the properties of the different channel models.

For a certain amount of frequency spectrum, users can use it as a single wide-band channel or divide it into multiple channels with narrower bandwidth, *i.e.,* the multi-channel model. Assuming a channel with the bandwidth equal to W, there are three ways to use this channel:

(1) The conventional (*i.e.,* collision) channel with bandwidth W and $K_T = K_R = 1$;

(2) The MPT/MPR channel with $K_T = K_R = K$ and bandwidth equal to W/K;

(3) The multi-channel model in which the single wide-band channel is divided into K narrower conventional channels each with the bandwidth equal to W/K.

Without specifying the technique implementing the MPT/MPR transceiver, we roughly assume the three channel models consume the same amount of bandwidth.

The most significant difference between the multi-channel model (3) and the other two channel models is that, in the multi-channel model, communication occurs in different channels and do not interfere with each other even if they are in the same vicinity. On the other hand, in the conventional (1) and MPT/MPR (2) channel models, all the transmissions share the same channel, thus, any two nodes in the interference range can hear each other.

As discussed in [87], there are pros and cons in the conventional channel model (1) with one wide-band and the multi-channel model (3) with separated narrower channels. We extend the discussion by taking into account the MPT/MPR channel model (2).

- In a multi-channel model, nodes exchange messages to negotiate channel usage. *i.e.,* which channel will be occupied by which pair of nodes (the transmitting and receiving terminals) for a certain amount of time. In MPT/MPR channels, all the nodes share the same channel and cooperate to avoid collisions at receivers. While it is difficult to tell which channel model's cooperation demands more

resources (*e.g.,* bandwidth and time), the ***multi-channel hidden terminal problem*** observed by So *et al.* [88] brings some insights.

The well-known hidden terminal problem [1] can be partially solved by the RTS/CTS exchange [11, 23]. However, this problem can become more complicated in multi-channel networks. Consider a multi-channel system which dynamically determines the channel assignment for the contending nodes for optimal channel usage. If only a single transceiver is allowed for a host, the RTS/CTS handshake can no longer mitigate the hidden terminal problem because the neighbors are not necessarily in the same channel. A node may miss the RTS/CTS sent by their neighbors which are in a different channel. Because of the presence of multi-channels, more effort is needed for nodes to cooperate. On the other hand, in MPT/MPR channels, the control messages, if any are needed, can be heard by any neighbor that is not transmitting.

- As a technical requirement to address coordination, multi-channel protocols usually specify one channel as the control channel where all the control messages are transmitted [88, 100]. While having all the nodes listening to the control channel for a certain period of time can alleviate the multi-channel hidden terminal problem, the control channel itself can become a bottleneck in a dense network.

- A rather minor technical requirement is that, when dividing the single wideband channel into multiple channels, "guard bands" are needed between adjacent channels to avoid the inter-channel interference. Because of the presence of the guard bands, not all the bandwidth can be used for communication, which can lead to capacity degradation.

This comparison helps illustrate that, without considerable rework, the protocols designed for multi-channels cannot be directly applied to MPT/MPR channels,.

2.2 Max-Min Fairness

For a network system where the users have unlimited demands, an intuitive notion of fairness is that every user obtains the same amount of resource as any other users contending for the same resource.

[1] The hidden terminal problem occurs when two nodes which are not in communication range of each other transmit to the same receiver concurrently, resulting in a collision at the receiver [92]. This problem was later studied in multi-channel networks and referred to as the multi-channel hidden terminal problem [88].

2.2.1 Wired Networks

In a wired network scenario where multiple flows share the links, such a fairness requirement is equivalent to maximizing the rate of the most poorly treated flow. This fairness metric is known as the ***max-min*** fairness [9].

Let f denote a flow and $\boldsymbol{f}$ represent the set of flows ($f \in \boldsymbol{f}$). In this thesis, we use a non-bold italic symbol (*e.g.,* f) to represent an element and a bold italic symbol (*e.g.,* $\boldsymbol{f}$) to represent a set. A bold italic symbol with a subscript denotes a subset. *e.g.,* $\boldsymbol{f}_l$ represents the set of flows traversing link l. The size of set $\boldsymbol{f}$ is denoted by $|\boldsymbol{f}|$. Let $\boldsymbol{\rho}$ be a rate allocation. The rate allocation of $\boldsymbol{f}$ is represented by $\boldsymbol{\rho_f}$. ρ_f ($\rho_f \in \boldsymbol{\rho_f}$) is the rate of flow f. The traffic through link l, denoted by R_l, is the sum of the rates of all the flows that traverse link l, *i.e.,* $R_l = \sum_{f \in \boldsymbol{f}_l} \rho_f$. C_l (packets/slot) represents the capacity of link l (*i.e.,* $R_l \leq C_l$). In wired networks, a rate vector $\boldsymbol{\rho_f}$ is ***feasible*** if $R_l \leq C_l$ for any link l of the network. If $f \in \boldsymbol{f}_l$, link l is a ***bottleneck link*** of f if 1) $R_l = C_l$ and 2) $\rho_f >= \rho_{f'}$ for any $f' \in \boldsymbol{f}_l$.

Proposition 2.1. *[9] In a wired network, a feasible vector $\boldsymbol{\rho_f}$ is max-min fair if and only if every flow has at least one bottleneck link.*

Prop. 2.1 suggests a method to evaluate whether a rate allocation is max-min fair. In wired networks, the flows traversing the same link compete for the resource of the link. If every flow traverses at least one bottleneck link, the rate allocation for the flows attains the max-min fairness. Based on Prop. 2.1 the water-filling algorithm [9] has been designed to compute the max-min fair rate allocation for wired networks.

2.2.2 Wireless Networks

To extend the max-min fairness from the wired networks to the wireless networks, first, we discuss the ***contention graph***. In a contention graph, every link is represented by a vertex. Two vertices of the contention graph are connected if the corresponding two links are interfering with each other. For example, Fig. 2.1(a) depicts a network of five single-hop flows and Fig. 2.1(b) shows its contention graph. Every link in Fig. 2.1(a) is represented by a vertex in Fig. 2.1(b). Two adjacent links are connected in the contention graph because the nodes cannot transmit and receive at the same time. Thus, adjacent links are interfering with each other.

Since the contention graph shows the interference relationships between links, we can use it to tell which flows are competing with each other, which can be precisely illustrated by the ***maximal clique (MC)***. In graph theory, a MC is a clique that

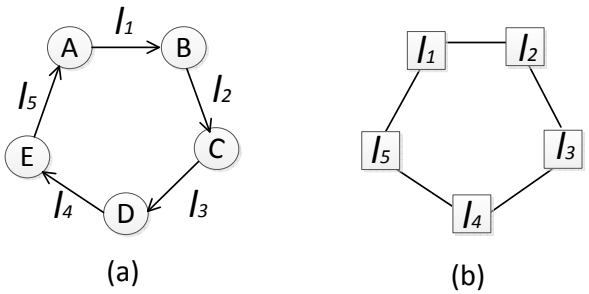


Fig. 2.1: A pentagonal network.

cannot be extended by including one more adjacent vertex, meaning it is not a subset of a larger clique. For example, in Fig. 2.1(b), every two connected vertices (*e.g.,* $\{l_1,\ l_2\}$, $\{l_2,\ l_3\}$, etc) compose a MC because for any given two vertices, there does not exist another vertex connecting to both vertices. Since the links (represented by the vertices in the contention graph) belonging to the same MC pair-wisely interfere with each other, the flows traversing the same MC (*i.e.,* going through any link belonging to the MC) are contending for the shared channel. It was observed by Huang *et al.* [42] that a MC in a wireless network is analogous to a link in a wired network in the sense that the flows traversing the same MC are contending for the resource of the MC. The study of [42] focuses on the max-min problem in single-hop wireless networks with conventional channels. We will extend the problem to multi-hop networks with MPT/MPR networks.

In MPT/MPR channels, a distinguishing feature of the contention graph significantly affects the behavior of the wireless network. The primary property of a contention graph in conventional channels (*i.e.,* $K_T = K_R = 1$) is that if any two links belong to the same MC, they can not transmit simultaneously. However, this principle does not hold for MPT/MPR channels due to the fact that a node can accommodate multiple transmissions or receptions. Thus, even links belonging to the same MC can transmit concurrently. We refer to this property as ***the M property*** of the contention graph in MPT/MPR networks. The M property changes our view of the contention graph and will have a persistent and significant impact in our subsequent exploration of MPT/MPR.

We will apply the max-min fairness metric in both single-hop and multi-hop scenarios of MPT/MPR wireless networks. We will demonstrate that it is not trivial to extend the max-min fairness from the wired networks to multi-hop MPT/MPR wireless networks.

2.3 Related Work

2.3.1 Medium-access Control Protocols

The MAC sub-layer is designed to efficiently allocate the communication medium to a multitude of nodes competing for the shared channel [9]. Considering a system with limited number of nodes ($1 \leq n \leq M$), Zhao *et al.* [111] define the channel capacity as

$$\eta \triangleq \max_{n=1,\ldots,M} C_n \tag{2.8}$$

where

$$C_n \triangleq \sum_{k=1}^{n} k\epsilon_{n,k} \tag{2.9}$$

The capacity from Eq. 2.8 is the maximum expectation of the number of successful receptions given the reception matrix Eq. 2.1. To approach this capacity, an intelligent strategy is needed to orchestrate the transmissions such that the number of transmissions is optimized. More specifically, the number of simultaneous transmissions should be n_0 such that $C_{n_0} = \eta$. The question is, without knowing the channel matrix Eq. 2.1, the number of nodes and transmission demand of each node (*i.e.,* the number of packets ready to transmit), how can we decide the size of the subset of nodes (*i.e.,* n_0) to access the channel at each slot such that the channel capacity is optimally used. If quality of service (QOS) needs to be considered, the dynamic allocation algorithm should decide not only the size, but also the content of the subset. For example, if the max-min fairness is considered, the multi-access protocol should give priority to the nodes with minor requests; if the delays of packet transmissions are concerned, the nodes with more strict constraints should be served first, etc..

MAC protocols are a class of schemes that are dedicated to coordinate nodes' accesses to the shared channel in a way that the channel capability is fully utilized and the QoS requirements of all the nodes are satisfied. A large number of MAC protocols have been designed for the collision channel [3, 17, 23, 26, 38, 50, 55, 94, 107]. Since the MPR capability makes it possible to accommodate up to K_R simultaneous transmissions, in order to fully exploit the MPR capability, MAC protocols based on MPR channels should cautiously embrace interference without exceeding the capacity limit. In the following, we explore recent research on MAC for the collision channel and the MPR channel.

MAC Protocols for the Collision Channel

- **Fixed allocation schemes**

  Classic fixed allocation schemes include time-division multiple-access (TDMA), frequency-division multiple-access (FDMA) and code division multiple access (CDMA). Fixed allocation schemes require the network system to provide centralized services. For instance, a high degree of synchronization is necessary in TDMA to coordinate the nodes and precise power control is required for CDMA cellular networks to solve the near-far problem [96]. Those controlling functions put a computational burden on networks. While FDMA is neither vulnerable to the timing problem of TDMA nor the near-far problem from CDMA, it requires higher-performing hardware compared to the other two schemes [12].

- **ALOHA**

  In contrast to the fixed strategies, more flexible methods are favored for their simplicity and adaptivity. ALOHA [3] developed in early 70s is a protocol using the "free-for-all" strategy. The channel resource is "free" to all nodes and packets are transmitted immediately after their arrivals. Every packet transmission will possibly collide with any other transmission within a time interval twice that of the packet transmission time. As demonstrated in [4], assuming messages are transmitted in packets with equal size and the transmission time of a packet is one slot, without considering the retransmission (*i.e.,* the colliding packet is lost), the maximum departure rate for ALOHA in an unslotted channel is $1/2e \approx 0.184$ packets per slot (PPS) when the attempting transmission rate is equal to $1/2$ PPS. In slotted ALOHA, because all transmissions are synchronized, the possible collision duration for a given packet transmission is reduced to one slot. The throughput of slotted ALOHA has a maximum rate of $1/e \approx 0.368$ PPS when the attempting transmission rate is 1 PPS [4].

  The (slotted) ALOHA protocol is attractive for its simplicity and low delay for bursty traffic under light load. However, it is also well-known for its stability problems [31,65,74]. The throughput collapses when the arrival rate approaches the channel capacity. The algorithms looking for an improved stability can be categorized as collision-recovery and collision-avoidance protocols.

- **Collision-recovery and collision-avoidance protocols**

  A class of collision-recovery protocols are referred to as splitting algorithms [9] (*e.g.,* tree algorithms [17] and first-come first-serve [26]) which apply different approaches to decompose the set of collided nodes. In ALOHA, if failed packets

need to be retransmitted, a random time interval will be placed before the retransmission. This random waiting period is necessary to spread the overlapped packets and reduce the collision possibility for the next transmission [3].

Failed transmissions are a significant source of waste in processing power and channel resources. Hence, researchers are more interested in finding strategies to prevent the potential collisions from happening. Carrier-sense multiple-access (CSMA) [55] uses the physical sensing scheme to monitor the shared channel. A node is only permitted to transmit when it senses that the carrier is free to avoid conflict with an undergoing transmission. IEEE 802.11 [23] employs collision avoidance (CA) by backing off for a random number of slot time. Optionally, the request-to-send (RTS)/confirm-to-send (CTS) handshake is applied to implement the ***virtual sensing*** scheme to reduce collisions and alleviate the hidden and exposed terminal problems [2]. Namely, the sender and the corresponding receiver use a short packet called RTS, which elicits a short packet called CTS, to reserve the channel if the channel is sensed free. The packets contain the network allocation vector (NAV) value indicating the duration of the transmission. Upon hearing these packets, the neighbors extract the NAV value and refrain from transmitting for the specified time period.

- **Application-oriented protocols**
  Numerous application-oriented protocols have evolved from IEEE 802.11 to provide more efficient services for certain applications such as wireless sensor networks and wireless local area networks (W-LAN)s [38, 50, 94, 107]. S-MAC [107] and T-MAC [94] are proposed for wireless sensor networks where efficient energy consumption is the primary concern. For W-LAN with devices equipped with multiple modulation schemes (and hence multiple data rates), Holland *et al.* [38] propose the receiver-based autorate (RBAR) protocol to dynamically select the appropriate modulation scheme matching the current channel state. The receiver estimates the channel quality according to the received RTS and selects the data rate using the latest channel estimation. The chosen data rate is then incorporated in CTS. Upon receiving this CTS, the sender will use the chosen rate to transmit the data packet.

[2]The exposed terminal problem occurs when two nodes which are in communication range of each other do not transmit concurrently due to carrier sense even though these two transmissions do not interfere with each other at the corresponding receivers, resulting in a waste of the channel resource [10].

MAC Protocols for MPR Channel

MPR channels are expected to provide higher performance; however, simply applying the MAC protocols designed for collision channels to MPR channels may not fully exploit the MPR capability. In the collision channel, a successful transmission implies the idleness of all other nodes. However, this is not true in MPR channels since a transmission can be successful in the presence of other transmissions. It is more difficult to infer the state of every other node, hence, it has been suggested that new protocols specifically designed for MPR channels are necessary to fully exploit the MPR capability [111].

- **MPR MAC protocols**

  Dan *et al.* [20] analyze the impact of MPR on CSMA with respect to throughput and delay in the scenario where a finite number of stations compete for an MPR channel. They propose a cross-layer designed CSMA for the K-MPR channel model, assuming that a node can precisely estimate the number of simultaneous transmissions by sensing the energy of the carrier. Based on the estimate of the number of concurrent transmissions, assuming the channel capacity K and the number of contending nodes are known, a waiting node will transmit with an appropriate probability such that the number of simultaneous transmissions will approach but not exceed K.

  Zhao *et al.* [111] propose the multiqueue service room (MQSR) protocol which supports QoS with respect to a delay requirement at each node. The MQSR protocol is a centralized scheme with the assumption that a central station assumes time-invariant and system-wide knowledge including the number of contending nodes, the channel reception matrix (Eq. 2.1), the traffic load and the maximum delay tolerable by each node. With this information, the controller grants the channel access to a set of nodes to maximize the throughput while satisfying the delay requirements of the nodes.

  Celik *et al.* [19] study the inefficiency of the conventional back-off schemes (*e.g.,* as in IEEE 802.11 CSMA/CA) and suggest that increasing (decreasing) the transmission probability (or equally, by modifying the contention window (CW)) after a successful (failed) transmission is unfair to distant nodes as they suffer more significantly from the power attenuation. By assigning distant nodes more transmission opportunities, the aggregate throughput can be improved assuming there are more nodes in further area. They propose the generic distributed probabilistic (GDP) protocol where a node is in one of the

two states: after success (AS) and after failure (AF). Nodes in AS state transmit with the probability p_{as} while nodes in AF state use the probability p_{af}. Various scenarios require different p_{as} and p_{af} values, which need to be set off-line, to achieve the maximum throughput.

- **MIMO MAC protocols**

  Another line of research on MAC protocols for MPR channels focuses on multiple access schemes realized by multiple antennas [18, 69, 78, 113]. Multiple antennas can be used for both uplink (several nodes transmit to one receiver) and downlink (one node transmits to multiple receivers) communications. Casari *et al.* [18] assume a completely connected network and propose a PHY-MAC cross-layer protocol to take advantage of multiple-input and multiple-output (MIMO) techniques and interference cancellation, making parallel transmissions possible. Instead of using RTS/CTS to reserve the channel for one transmission, the CTS control message is applied to grant channel access to multiple transmission requests (RTSs). NULLHOC [69] is a cross-layer protocol designed for multi-path MIMO communication channels where nodes are equipped with antenna arrays. NULLHOC extends the control messages of IEEE 802.11 (RTS/CTS) and allocates a fraction of the bandwidth for the control packets, which are used to collect the channel information of all the nodes involved in an ongoing transmission. The gathered information is then applied to regulate transmissions such that multiple transmissions in the same collision domain can coexist without disturbing one another.

- **CDMA MAC protocols**

  In code division multiple access (CDMA), a signal is spread using a pseudo-random noise (PN) code at the transmitter. To recover the signal, the receiver de-spreads the signal with the same PN code. By applying distinct PN codes, the receiver can receive multiple signals simultaneously. However, the near-far problem [3] can severely degrade the network throughput of a CDMA system [70].

  The control access CDMA (CA-CDMA) [70] utilizes the RTS/CTS packets over an out-of-band control channel to collect channel-gain information and solve the near-far problem in single-hop wireless ad hoc networks. The channel-gain information is used to adjust the transmission power of the data packet such that the concurrent transmissions at the intended receiver do not destroy the re-

[3] The near-far problem describes the condition that the signal from a transmitter near the base station is received with high power which makes it difficult to receive the signals sent from faraway transmitters [70].

ception. However, this work assumes that each terminal is equipped with two transceivers which allow a terminal to transmit and receive simultaneously over the control and data channels thus cannot be directly applied in half-duplex scenarios. CDMA sensor MAC (CSMAC) [61] seeks to reduce the energy consumption and the latency assuming that each node has two receivers and can estimate its location. By negotiating PN codes via RTS/CTS based handshake and applying frequency division, CSMAC enables concurrent transmissions. With the broadcast location information of the neighbors, a subset of nodes are chosen to be active. The reduced network decreases the energy consumption and the average latency of the packet transmissions.

The multiqueue service room (MQSR) MAC protocol [111] and the work of [20] require global information such as the number of contenders and the channel capacity. The generic distributed probabilistic (GDP) protocol [19] is a distributed algorithm but it does not compute the optimal access probability to access the channel. The MIMO MAC protocols [18, 69] need separate control channel to convey the RTS and CTS control packets while the CDMA MAC protocols [61, 70] assume two transceivers on each node. Among the large number of proposed MAC protocols, we have not seen any fully distributed design for MPR channels which can be adaptive to dynamic environments with various numbers of nodes (*i.e.,* competitors) and channel capacities. In Chapter 3, we will present AIMD-MAC to solve this problem. AIMD-MAC estimates the optimal access probability by manipulating the transmission history in the past and thus does not need the RTS/CTS-like control messages. We will demonstrate via simulation that, with an appropriate set of parameters, AIMD-MAC does not require the feedback at every slot to achieve the optimal performance and can be adaptive to a wide range of scenarios.

2.3.2 Protocols for Higher Layer Wireless Networks

In this section, we discuss several well-known problems affecting the performance of multi-hop flows in wireless networks and review recent research on the upper layer protocols for multi-hop wireless networks with MPT or MPR capabilities.

Hidden and Exposed Terminal Problems

The well-known hidden and exposed terminal problems in wireless networks have attracted considerable attention. Multiple access with collision avoidance (MACA) [53]

published in the 90s intends to alleviate the hidden and exposed terminal problems existing in the CSMA method. MACA removes the carrier sense (CS) part of CSMA/CA but extends the CA mechanism. Namely, MACA reserves the channel before transmission via a three-way handshake (*i.e.,* RTS/CTS/DATA). However, MACA cannot completely solve the hidden and exposed terminal problems and MACAW [11] was proposed to refine the MACA algorithm by constructing a five-phase dialogue for each transmission (*i.e.,* RTS/CTS/DS/DATA/ACK). MACAW significantly improves the performance of MACA but still does not fully solve the hidden and exposed terminal problems.

IEEE 802.11 [23] applies the four-way handshake (*i.e.,* RTS/CTS/DATA/ACK) as an option to mitigate the hidden-terminal problem. Recent efforts to solve the hidden-terminal problem include [103, 104, 110, 112]. The work of [104, 110] suggests using larger carrier sensing range at the cost of potentially more severe exposed terminal problem. Xu *et al.* [103] propose to shorten the transmitter-receiver distance so that RTS/CTS can work more effectively. However, this mechanism indeed reduces the effective transmission range. Another work by Zhu *et al.* [112] suggests striking for a good balance between the hidden and exposed terminal problems without completely solving them.

Asymmetric Information

The asymmetric information (AI) problem has been observed in many papers [11, 27, 51, 57, 71, 80, 85, 102] and it introduces interactions among flows in a multi-hop setting. Examples of this are illustrated in Fig. 2.2 in which solid lines represent that two nodes are in transmission range and the arrows show the directions of the transmissions.

(a) Can be solved by RRTS (b) Cannot be solved by RRTS

Fig. 2.2: The examples of the asymmetric information [11].

In Fig. 2.2(a), node A transmits to node B and node D transmits to node C. Without loss of generality, if l_2 wins the initial contention period, the channel will almost completely be occupied by l_2 thereafter. The reason is that once l_2 starts to transmit data packets, node A will repeatedly send RTS to node B because node A is not aware of the transmission on l_2. Node B will not reply with CTS since B needs

to defer the transmission to node C. Every time A sends a RTS without response, it needs to double its CW. Because the data packet is much longer than the control packet, A will exponentially increase its CW until it reaches the maximum length, which makes it even harder to compete with node D whose CW is set to the minimum value. The only chance A can win the channel is when A's RTS arrives at B after a complete data transmission and before C's next CTS.

Bharghavan [11] proposes to use the request-for-request-to-send packet (RRTS) to solve the problem. Since the problem exists because the transmitter (*i.e.,* node A) cannot hear the competing transmission (*i.e.,* the one on l_2), one solution is to ask the receiver (*i.e.,* node B) to send a RRTS during the next contention period if it has received repeated RTS requests. After received the RRTS, the transmitter (*i.e.,* A) immediately sends a RTS to request the channel. Similar approach is applied in distributed wireless ordering protocol (DWOP) [51] to support a FIFO scheduling.

However, in an asymmetric topology, as shown in Fig. 2.2(b), the AI problem persists. In Fig. 2.2(b), node C, the transmitter of l_2, can hear from the receiver of l_1, which gives C the complete information to compete for the channel. On the other hand, node A does not know the transmissions on l_2. RRTS does not work in this case because once node B is overwhelmed in the data transmission from C, it cannot hear the RTS from A and therefore does not know whether it should send RRTS or not.

Inter-flow and Intra-flow Contentions

Inter-flow and intra-flow contention is yet another cause for poor performance in multi-hop scenarios [58, 59, 80, 109]. A multi-hop flow needs to contend with other flows traversing in the neighborhood (*i.e.,* inter-flow contention) and a node on the path of the flow also competes with its upstream and downstream nodes on the same path (*i.e.,* intra-flow contention).

802.11 MAC interactions with ad hoc forwarding are studied in [59] which suggests that the performance of 802.11 closely depends on the topology of the network, the traversal distances (from the source to the destination) of the flows and the traffic pattern. Particularly, in a long chain of nodes, 802.11 can only achieve about half of the optimal capacity because nodes early in the chain starve later nodes. In [58], Li shows that it is significant to allocate the bandwidth for multi-hop flows from an end-to-end viewpoint and take into consideration the intra-flow correlation between upstream and downstream hops. Raniwala *et al.* [80] design a centralized algorithm to fairly allocate bandwidth to multi-hop flows. The intra-flow contention is taken care of by propagating the bandwidth allocation of each hop to all the nodes along the

flow and the minimum hop allocation decides the flow rate. Zhai *et al.* [109] propose optimum packet scheduling for each traffic flow (OPET) in which higher priority is assigned to the nodes which have just received a packet such that those nodes have a higher chance to forward the newly received packets. Additionally, OPET applies a backward-pressure mechanism, which alleviates both the inter-flow and intra-flow contentions, to restrict a node from transmitting if the backlog of its downstream node overflows. The backlog overflow information is communicated via an enhanced RTS/CTS-like handshake paradigm.

The above problems are inherent to the CSMA/CA-based designs and significantly affect the performance of the network. In Chapter 6, we propose the AB-MAC algorithm which organizes the channel access of multi-hop flows in a way that alleviates these problems without introducing significant overhead.

Protocols for Multi-hop MPR Wireless Networks

A number of upper layer protocols designed for wireless ad hoc networks with underlying MPT or MPR capability are discussed here.

Wang *et al.* [97] formulate an optimization problem for the joint routing and scheduling problem for multi-hop wireless networks with $K_T = 1$ and $K_R > 1$, *i.e.,* the MPR technique. In order to maximize the system throughput, they apply node-disjoint multi-path routing to alleviate the restrictions caused by single-packet transmitters. However, routing flows through different nodes cannot completely solve the problem caused by the limiting transmitting capacity. The experimental results of [97] show that the system throughput cannot linearly increase with the MPR capacity limit, which is what the numerical results presented in this thesis demonstrate. Another scheduling algorithm proposed to exploit the MPR capability is the work by Lv *et al.* [63], which studies link scheduling for ad hoc networks with successive interference cancellation (SIC) receivers. The authors also suggest that in order to cope with interference and attenuation in wireless systems, it is necessary to integrate the interference cancellation and rate adaptation.

Random Packet CDMA (RP-CDMA) [84] is a new CDMA technique which applies the multi-user detection (MUD) receiver and encodes the packet header by a common spreading code and the payload by a randomly generated spreading code. A link layer acknowledgement protocol is proposed by Mortimer *et al.* [68] based on the RP-CDMA technique to improve throughput and reliability for multi-hop wireless ad hoc networks. It is observed in this work that, as the system load increases, the MPR capabilities of the receivers are usually under-utilized while the transmit queues

overflow because the system throughput is limited by the transmitting capability of the sender. To solve this problem, the authors of [68] suggest using simultaneous transmissions to pair the MUD receivers with equally capable transmitters.

Motivated by these observations, in Chapter 4 and 5, we present an analysis to explain how K_R and K_T jointly affect the system performance and why it is necessary to have $K_R = K_T$ for the system throughput to scale linearly. To this end, we propose algorithms to calculate the max-min fairness rate allocation, which is used as the indicator of the system performance, for multi-hop wireless networks with advanced communication channels. As we will show, the task is not trivial.

2.3.3 Multi-Channel Protocols

As discussed in Section 2.1.4, the multi-channel model enables separately concurrent transmissions at the cost of the need to negotiate channel usage beforehand. For the sake of completeness, a couple of examples are illustrated here.

The proposals to explore the flexibility brought by the available channel set can be roughly categorized into two classes: 1) the protocols [36, 49, 72, 73, 100] which need multiple transceivers per host and 2) protocols [88, 93, 106] that focus on low-cost designs using only one transceiver per host. The implementation of multiple transceivers make it more convenient to monitor the channels at the cost of more complex hardware. For example, Wu *et al.* [100] propose the dynamic channel assignment (DCA) protocol assuming the available channel set consists of one control channel and n data channels and every node is equipped by one control transceiver and one data transceiver. So *et al.* [88] design the MMAC protocol which requires only one transceiver per host and solves the multi-channel hidden terminal problem.

2.3.4 AIMD Protocols

The widely studied additive-increase multiplicative-decrease (AIMD) algorithm is a feedback control algorithm used in the transport layer for congestion avoidance [46]. In the congestion avoidance state of the AIMD algorithm, the congestion window (CWND), which captures the as-of-yet unacknowledged transmission in transit to the destination, is increased by $\frac{1}{CWND}$ after every acknowledgement (ACK) and decreased by half if a congestion is detected. The frequency to update CWND is determined by the round-trip time (RTT) which can be estimated by Jacobson/Karels algorithm [46].

The seminal work by Chui and Jain [22] applies a simpler, binary-feedback model where a central station is monitoring the aggregate traffic load. The central station will send a one-bit signal to the nodes to identify whether the current aggregate load

is above or below the desired level. If the current channel usage exceeds the optimal level, the nodes will decrease their load; otherwise they will increase the load. Instead of using the increase-by-one decrease-by-half strategy in TCP, the increase term α and the decrease factor β are generalized. This AIMD algorithm [22] is referred to as the basic AIMD algorithm in following text of the thesis. With $\alpha > 0$ and $0 < \beta < 1$, the AIMD algorithm is proved to be able to converge to an efficient and fair state disregarding the initial setting. The efficiency is measured by the closeness to the desired load level and the fairness is evaluated by Jain's index [47] with the max-min criterion [9]. This generalized AIMD is later applied by Yang *et al.* [105] and denoted as the general AIMD (GAIMD).

The AIMD algorithms have attracted extensive research attention. Many modified versions have been proposed for different applications. For example, instead of using feedback to indicate the congestion, Rangwala *et al.* [79] apply the queue length to measure the levels of the congestion. More specifically, the exponentially weighted moving average (EWMA) of the instantaneous queue length coupled with multiple thresholds is used to signal congestion and the potential need to halve the flow rate.

Moving downward to the MAC sub-layer, and more relevant to the work in this thesis, AIMD algorithms have also been considered to organize nodes' accesses for efficient channel utilization and QoS requirements. Heusse *et al.* [37] count the number of successive idle slots and compare it against the optimal value (*i.e.,* the threshold), which is approximately constant for different number of contending nodes. The comparison results are used to decide whether the size of the CW should be increased additively or decreased multiplicatively. This protocol is called Idle Sense and is designed to improve throughput and short-term fairness. Hu *et al.* [41] apply a similar approach and propose the MAC contention control (MCC) protocol based on IEEE 802.11e [2] to maximize the bandwidth utilization and achieve proportional bandwidth allocation. Instead of adjusting the size of CW, MCC monitors the number of successive collisions or idle slots and updates the packet dequeueing rate following the AIMD strategy. Since each node needs to independently decide whether the channel is idle, these methods [37, 41] require that all the nodes can sense each other; otherwise RTS/CTS-like control messages are needed to coordinate the nodes.

The above AIMD-based algorithms are designed for conventional networks. In this thesis, we propose two MAC algorithms based on AIMD logic for MPT/MPR networks: AIMD-MAC for single-hop scenario and AB-MAC for multi-hop scenarios. In both algorithms, we apply the access probability (AP) to control the channel access and adopt the AIMD strategy to adjust the access probability. The advantages to applying the AIMD method include: 1) reduced collisions; 2) support for fairness and

3) alleviation of problems inherent in the CSMA/CA based algorithms discussed in Section 2.3.2.

2.3.5 Back-Pressure Protocols

In characterizing the capacity of a network and describing a strategy to achieve it, certain important results have been produced in the literature. One is the back-pressure (BP) routing algorithm, originally proposed by Tassiulas *et al.* [91]. It is applied in constrained queueing networks to achieve the maximum throughput and is analyzed via the Lyapunov drift theorem [29, 76, 77].

The BP algorithm mimics how water flows through a network of pipes via pressure gradients to reach its destination. In order to create the corresponding "pressure gradients" in communication networks, the BP algorithm prioritizes the links with high ***differential backlog***. Simply speaking, the differential backlog is the difference of the backlogs of the same flow at two consecutive nodes. By repeatedly designating the links with high differential backlog to transmit, the BP routing algorithm artificially constructs similar types of gradients — the congestion gradients. Therefore, with the BP routing algorithm, the packets of different flows will traverse the network via congestion gradients and eventually exit the network by reaching their destinations.

We formally introduce the BP algorithm using the framework described in [75] which consists of three stages: 1) determine the optimal flow; 2) compute the link rate matrix and 3) finalize the link rate. Consider a synchronous network system with a given topology graph. The network consists of N nodes and L links. A set of flows $\boldsymbol{f}$ arrive at arbitrary nodes heading to different destinations with the path undecided. Node i has an array of queues to store the backlogs of the flows crossing the node. $Q_i^{(f)}(t)$ $(t = 1, 2, 3, ...)$ denotes the length of the backlog of flow f at node i at slot t. Because the packets exit the system once they reach their destinations, $Q_i^{(f)}(t) = 0$ if node i is the destination of flow f. The rate of a link from node i to node j at slot t is denoted by $\mu_{i,j}(t)$. Let $\boldsymbol{\mu}(t) = \{\mu_{i,j}(t),\ i = 1, ..., N,\ j = 1, ..., N\}$ be the link rate matrix at slot t, the goal of the BP routing algorithm is to select the link rate matrix to achieve the maximum throughput while stabilizing the queueing network. BP ignores the details of the MAC layer by assuming a predefined set of feasible link rate matrices. Let $\Im$ represent the set of all feasible link rate matrices such that when the links are transmitting with the rate specified by a link rate matrix belonging to $\Im$, no collisions occur. In other words, $\Im$ is the constraint for the active link set with which the medium access problem is taken care of.

(1) **Determine the optimal flow:**

The first step of the BP routing algorithm is to decide the optimal flow of a link. For a given link i to j, there can be more than one flow crossing the link. The set of flows crossing link i to j is denoted by $\boldsymbol{f}_{i,j}$. The value of $Q_i^{(f)}(t) - Q_j^{(f)}(t)$ is the ***differential backlog*** of flow f on link (i,j) at slot t. The optimal flow of link i,j at slot t, denoted by $f_{i,j}^*(t)$, is the flow with the largest differential backlog, as shown in Eq. 2.10.

$$f_{i,j}^*(t) = \arg\max_{f \in \boldsymbol{f}_{i,j}} (Q_i^{(f)}(t) - Q_j^{(f)}(t)) \tag{2.10}$$

If link (i,j) is assigned to transmit at slot t, only the packets of flow $f_{i,j}^*(t)$ have the potential to be selected.

(2) **Compute the link rate matrix:**
At this step, BP computes the link rate matrix by solving a ***max-weight*** problem (Eq. 2.12). The weight of link (i,j) is denoted by $w_{i,j}(t)$ and is computed by Eq. 2.11.

$$w_{i,j}(t)) = \max\{Q_i^{(f_{i,j}^*(t))}(t) - Q_j^{(f_{i,j}^*(t))}(t), 0\} \tag{2.11}$$

Max-weight problem:

$$\max \quad \sum_{i=1}^{N} \sum_{j=1}^{N} \mu_{i,j}(t) w_{i,j}(t) \tag{2.12a}$$

$$\text{subject to} \quad \boldsymbol{\mu}(t) \in \Im \tag{2.12b}$$

The max-weight problem finds the link rate matrix $\boldsymbol{\mu}(t)$ which can maximally "release" the congestion pressure of the network. The potentially active link set is constrained by Eq. 2.12(b), which guarantees that the schedule will be collision-free.

(3) **Finalize the link rate:**
After the link rate matrix is found, the final step is to assign the link rate to the optimal flow of that link. Particularly, a flow should refrain from transmitting if the current different backlog is non-positive, even though the flow is assigned as the optimal flow. Eq. 2.13 excludes the possibility that a flow transmits to a neighbor with an equal or higher backlog. It is also possible that a node does not have enough packets to transmit, *i.e.,* $Q_i(f)(t) < \sum_{j=1}^{N} \mu_{i,j}^{(f)}(t)$. This situation is called a ***queue underflow***. In this case, node i will send a null packet to occupy the assigned transmission quota. It is shown that the null packets will not

jeopardize the stability of the system because it only happens when the backlog is low [75].

$$\mu_{i,j}^{(f)}(t) = \begin{cases} \mu_{i,j}(t), & \text{if } f = f_{i,j}^{*}(t) \text{ and } Q_i^{(f)}(t) - Q_j^{(f)}(t) > 0 \\ 0, & \text{otherwise} \end{cases} \tag{2.13}$$

From the above description, it can be observed that the BP algorithm needs global backlog information to decide the set of active links at every slot, namely, to solve the max-weight problem (Eq. 2.12). Additionally, since the routing is dynamically determined by the differential backlog, the end-to-end delay of a packet can be very long when the congestion gradients are not strong enough and the packet takes a loopy tour before it eventually reaches the destination. Therefore, researchers [24, 32, 48, 66, 75–77, 90, 108] have paid increasing attention to improve the BP algorithm.

A joint routing and power allocation policy is proposed by Neely *et al.* [76] to stabilize the system and provide bounded average delay guarantee. Ying *et al.* [108] propose a mechanism integrating the BP algorithm and the shortest-path routing to shorten the end-to-end delay. To lower the complexity of the BP algorithm (caused by solving the max-weight problem), a framework using the randomized, iterative algorithm is designed by Tassiulas [90], which is later extended by Giaccone *et al.* [32]. This framework does not require solving the max-weight problem; however, it still needs the global backlog knowledge. A number of distributed protocols [24, 66] are designed based on the framework of [32, 90] and various levels of local information exchange among the nodes. In [24], the conflict graph is organized by a tree structure and the weights of the components are communicated from the leaves to the root. In [66], a gossip mechanism [13, 54] is built up to perform the local information exchange.

In Chapter 6, we will propose a MAC algorithm AB-MAC for multi-hop MPT/MPR wireless networks, which adopts a modified BP strategy to alleviate the intra-flow contention and uses the AIMD method to regulate the channel access of each node. The combination of AIMD and BP algorithms avoids the need to solve the max-weight problem and takes care of the MAC layer.

Chapter 3

AIMD-MAC: A Distributed MAC Protocol for K-MPR channels

In Section 2.3.1, a number of MAC protocols designed for K-MPR channels are reviewed. Despite recent interest in MAC with MPR capability, little research has considered applying the well-known AIMD algorithm [22] to tackle this problem. As discussed in Section 2.3.4, AIMD was proved to be able to achieve the max-min fairness. We apply the AIMD strategy in K-MPR channels and design the AIMD-MAC protocol to exploit the MPR capacity while supporting max-min fairness among uncoordinated nodes.

This chapter is organized as follows. Section 3.1 shows the system model and Section 3.2 introduces the benchmark S-ALOHA*. The AIMD-MAC algorithm is presented in Section 3.3. We discuss the reasons why the basic AIMD strategy [22] cannot be directly applied to our system model and how we tackle this challenge by using the *transmission history*. Section 3.4 discusses the simulation results, which show that under light load, AIMD-MAC outperforms S-ALOHA* by achieving shorter delay. When the system load is heavy, AIMD-MAC achieves the same performance as S-ALOHA*, with respect to throughput, delay and fairness. Additionally, in Section 3.5, we analyze the behavior of AIMD-MAC and illustrate why AIMD-MAC can obtain fairness among uncoordinated nodes. By observing the distribution of the access probabilities, we demonstrate that AIMD-MAC is adaptive in dynamic environments without the need to reconfigure its parameter set. Section 3.6 is the summary.

3.1 System Model

We introduce a single-hop scenario (Fig. 3.1) built on the K-MPR channel. As discussed in Section 2.1.2, since only the receiving side has multi-packet capability in

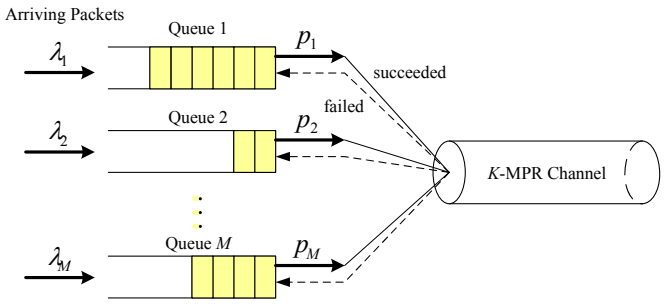


Fig. 3.1: A single-hop scenario with K-MPR channel.

the K-MPR channel model (*i.e.,* $K_T = 1$), we use K to represent the multi-packet reception capability in this chapter. Assume a buffered slotted ALOHA system where M independent nodes access the common channel. Data is transmitted in packets of equal (unit) size. The slot time is defined as the transmission time for a single packet. Each node is equipped with an unlimited queue to store backlog packets. Nodes apply the FCFS queueing service discipline. Therefore, a newly arriving packet is placed at the end of the queue. The packet at the head of the queue is transmitted in the next slot, t, with probability $p_m(t)$. A failed (collided) packet remains at the head of the queue until it is successfully transmitted. Note that we do not consider the case where the same node attempts multiple concurrent transmissions, or attempts transmissions for queued packets out-of-order. Hence, for a single node, at most one packet departure (*i.e.,* successful transmission) is possible in each time slot.

The queue length of node m ($m = 1, 2, ..., M$) at the beginning of slot t ($t = 1, 2, 3...$) is represented by $Q_m(t)$. Szpankowski [89] describes the evolution of Q_m by the stochastic recurrence

$$Q_m(t+1) = (Q_m(t) - y_m(t))^+ + x_m(t) \tag{3.1}$$

where $X^+ = max\{0, X\}$. $x_m(t)$ represents the number of newly arriving packets during slot t at node m. When the traffic load is modeled by a Poisson process, the quantities $\{x_m(t), t = 1, 2, 3...\}$ are Poisson-distributed random variables with first moment equal to λ_m. $\lambda = \sum_{m=1}^{M} \lambda_m$ is the aggregate system mean arrival rate and λ_{ATT} is the aggregate channel attempt rate.

At the beginning of slot t, if $Q_m(t) > 0$, node m transmits one packet with the access probability ***access probability (AP)*** [1] equal to $p_m(t)$ and is idle with probability $1 - p_m(t)$; if $Q_m(t) = 0$, node m is idle with probability one. We assume

[1]Note that through the thesis, AP is an acronym for access probability rather than the access point.

that an 1-bit feedback from the receiver to the transmitter is immediately available at the end of each transmission which is one (if the transmission was successful) or zero (if the transmission failed). In the case of success, $y_m(t) = 1$; otherwise $y_m(t) = 0$. At the end of the slot, $Q_m(t)$ is updated by the stochastic recurrence (Eq. 3.1). The ***access probability vector (APV)*** $[p_1(t), p_2(t), ..., p_M(t)]$ characterizes the access policy across all nodes of the system at the t–th slot.

3.2 S-ALOHA*

Since the transmissions at each node is controlled by the AP, we present the S-ALOHA* to evaluate the performance of such a system if every node knows the optimal AP to maximize the aggregate throughput over the K-MPR channel. S-ALOHA* will be applied as a benchmark to evaluate the performance of our distributed algorithm. S-ALOHA* works for the systems with specific requirements as follows:

(i) The number of nodes M is constant and known *a priori*;

(ii) Nodes always have packets ready to transmit;

(iii) The channel capacity K is also constant and known beforehand.

A system satisfying the above three requirements is named a ***specified system***; otherwise, it is called a ***unspecified system***.

For a specified system where $p_1(t) = p_2(t) = ... = p_M(t) = p, (t = 1, 2, 3...)$, Ghanbarinejad *et al.* [30] present Eq. 3.2 to compute the expected aggregate throughput

$$R(p, M, K) = \sum_{n=1}^{K} n \binom{M}{n} p^n (1-p)^{M-n} \tag{3.2}$$

and define the optimal access probability p^* (Eq. 3.3). Hence, the optimal aggregate throughput R^* can be expressed by Eq. 3.4.

$$p^*(M, K) = \arg\max_{0 \le p \le 1} R(p, M, K) \tag{3.3}$$

$$R^* = R(p^*(M, K), M, K) \tag{3.4}$$

In a *specified system* with the precomputed optimal AP (*i.e.,* p^*), S-ALOHA* is a strategy in which every node uses p^* as the AP when determining whether to transmit or not. Fig. 3.2 shows that the aggregate throughput of the system model (Fig. 3.1) with $M = 30$, $K = 10$ and a Poisson arrival process with aggregate mean arrival

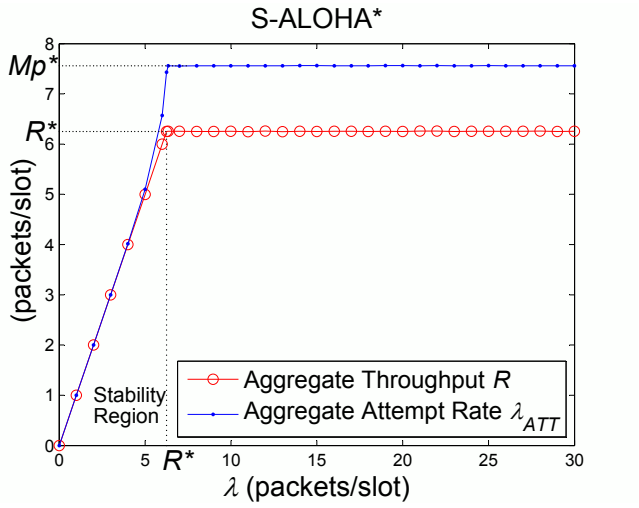


Fig. 3.2: The aggregate throughput of S-ALOHA*.

rate λ. In this case, it can be computed numerically that $p^* = 0.25$ and $R^* = 6.3$. As shown in Fig. 3.2, $[0, R^*]$ is the stability region of S-ALOHA* because, as long as the aggregate load is limited to the bound (*i.e.,* $\lambda \leq R^*$), the queue growth is bounded. S-ALOHA* is optimal in the sense that $R = \lambda$ within the stability region (*i.e.,* $\lambda \in [0, R^*]$) and $R = R^*$ beyond the stability region (*i.e.,* $\lambda > R^*$). Because the transmissions are controlled by a probability, even with the optimal AP, collisions still occur. Thus, the aggregate attempt rate λ_{ATT} is slightly greater than R. λ_{ATT} is equal to Mp^* after the system is saturated since M active nodes transmit with the AP equal to p^*.

A relevant question is: in an *unspecified system* where M, K and p^* are unknown or dynamic, how can we achieve the optimal performance in a distributed environment? We will solve this problem in this chapter.

3.3 AIMD-MAC Protocol

S-ALOHA* can only be applied in a *specified system* since it requires p^* before the node enters the network. After a new node enters the system, all nodes which are already in the system have to update their APs with a new p^* because the number of accessing nodes M is changed (Note that p^* is a function of M and K as shown in Eq. 3.3). If any node fails to invoke the updating, the system will crash since the aggregate demand exceeds the channel capacity. Similarly, if any node leaves the system, such updating failure will result in a waste of channel resources.

Therefore, we propose the AIMD-MAC protocol. AIMD-MAC is a distributed algorithm designed for fully uncoordinated nodes accessing the common K-MPR

channel in a single-hop scenario. AIMD-MAC's objective is to grant access to the channel at each slot in a manner such that the expected aggregate throughput is maximized while attaining long-term fairness across all nodes. To demonstrate AIMD-MAC's performance, we compare it against the benchmark S-ALOHA*.

The AIMD-MAC protocol consists of two steps at each node: 1) updating the transmission history and 2) adjusting the AP. The basic idea is that, without the knowledge of the number of active nodes and the channel capacity, each node collects its local transmission history to estimate the most recent channel quality ("collision level"). Based on the collected information, each single node can independently and dynamically set a suitable AP such that the number of concurrent transmissions approaches but does not exceed the channel capacity K.

The AIMD-MAC protocol performs the two steps periodically. The updating period is specified by a preset parameter ***update cycle (UC)*** (Fig. 3.3), denoted by τ. The indices -1 and -2 indicate the past two UCs, which specify the time interval of the transmission history involved in computing the AP of the current UC. The AIMD-MAC algorithm runs at the first slot of every UC, starting from the time at which a node enters the system. Because the AP is constant during one UC, we use the superscript (*e.g.*, $p^{(i_c)}$) to denote parameters for different UCs. (In this sub-section we describe the operation of a single node and to enhance clarity, we have dropped the subscript m identifying the node.) The value of τ plays a significant role in the adaptivity of AIMD-MAC which will be fully discussed in Section 3.5.

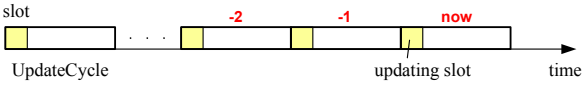


Fig. 3.3: The update cycle.

The *transmission history* records the most recent local transmission information and consists of two components, the ***success ratio*** v (Eq. 3.5) and the ***utilization ratio*** u (Eq. 3.6).

3.3.1 Success Ratio (v)

$$v^{(i_c)} = \begin{cases} \frac{S^{(i_c)}}{T^{(i_c)}}, & T^{(i_c)} \neq 0 \\ -1, & T^{(i_c)} = 0 \end{cases} \tag{3.5}$$

In Eq. 3.5, S is the number of successful transmissions and T is the number of attempted transmissions. i_c ($i_c = -1, -2$) is the index for the UC relative to the current UC (as showed in Fig. 3.3). Obviously, for one UC, $S \leq T$ ($S \geq 0,\ T \geq 0$) and $0 \leq v \leq 1$ if $T \neq 0$. v characterizes the recent channel conditions at the MAC layer which are classified into three cases.

(i) Good channel quality for increasing AP: if the collision level was low, there were few collided (failed) packets and v would be close to 1;

(ii) Poor channel quality for decreasing AP: if the collision level was high, S was small compared to T. Thus v would approach 0;

(iii) Channel quality is unknown and AP is not updated: if $T = 0$, there was no attempted transmission from the considered node during the past UC. In this case, the channel condition is unknown to the node and the AP value stays the same because either increasing or decreasing the AP has a 50% chance to jeopardize the channel. v is assigned the special value -1 to indicate "undefined channel".

It is natural to consider v as the index of the channel quality and adjust AP according to the above categorization. Chiu and Jain's seminal work [22] explains that the basic AIMD algorithm converges to efficiency and fairness if 1) a unique command (*i.e.,* either increase or decrease the load) is sent to all nodes by the base station at every slot and 2) every node receives this error-free feedback and reacts accordingly. Hence, in order to achieve the optimal performance in distributed environments, it is important that all nodes achieve an agreement on the channel condition. Only with the agreement of the current channel evaluation, can each node applies the same AP adjusting operation independently. However, it is not trivial to find a unique success ratio threshold for the nodes to make such a decision.

Since the transmissions are random events, each node may have a slightly different value of v even for the same UC. If a uniform threshold of success ratio is applied, the comparison between v and the threshold may produce different results at different nodes. This disparity will lead to inconsistent channel evaluations and various AP adjustments among these nodes, which will negatively affect system performance. To solve this problem, we propose a simple method to avoid this biased channel estimation caused by the uniform threshold.

It was observed from our earlier simulations that, although a general threshold for v does not exist, there exists a certain level of consistency to how the relation of two successive success ratios (essentially, the slope of the success ratio "function") reveals the shared channel condition. An improved channel condition (increases from one UC to the next) results in an increase in success ratios for all nodes while a compromised channel condition leads to a decrease in all success ratios. This observation inspired the idea to compare $v^{(-1)}$ and $v^{(-2)}$ (*i.e.,* the success ratio for (-1)–th and (-2)–th UCs, respectively) at each single node to determine the channel condition trend. The following three cases explain the new approach:

(I) Increasing AP: if $v^{(-2)} < v^{(-1)}$ ($v^{(-1)} \neq -1$), the channel quality improved from (-2)–th to (-1)–th UC;

(II) Decreasing AP: if $v^{(-2)} > v^{(-1)}$ ($v^{(-1)} \neq -1$), the channel quality decreased from (-2)–th to (-1)–th UC;

(III) Maintaining AP: if $v^{(-1)} = -1$, the channel condition is unknown.

This is a simple approach to get a relatively consistent measure of the recent channel condition from each node independently, and at negligible storage cost. A more thorough analysis of why this method can have the nodes uniformly respond to the channel quality will be presented in Section 3.5.

3.3.2 Utilization Ratio (u)

$$u^{(i_c)} = \frac{T^{(i_c)}}{T_w^{(i_c)} + T^{(i_c)}} \tag{3.6}$$

While the employment of success ratio makes AIMD-MAC converge to efficiency, utilization ratio (Eq. 3.6) is applied to further reduce collisions.

At the beginning of a slot, a node generates a random number which is uniformly distributed in $[0, 1]$. If this number is less than $p^{(i_c)}$ the node will transmit the first packet of its queue, if it is not empty. $T_w^{(i_c)}$ records the number of slots in the i_c-th UC when the node wastes the transmission opportunity. *i.e.,* the generated random number is less than $p^{(i_c)}$ but the node does not have a packet to transmit. The expectation of $T_w^{(i_c)} + T^{(i_c)}$ is $p^{(i_c)} \times UC$. Let us assume that the current value of the AP, $p^{(i_c)}$ represents the fraction of "channel time" allocation to a node. Given the definition of Eq. 3.6, we can consider u as a measure of the waste of the given allocation, which could happen *e.g.,* because of low queue occupancy. Specifically, if a node is allocated more channel time than needed, u will be less than one.

The utilization ratio is introduced to prevent the unnecessary channel time allocation to one node. If $u = 1$, there was no waste of allocated channel time, which suggests that this node's AP proportionally reflects the traffic contribution from this node. However, if $u < 1$, the number of attempted transmissions from the node could be very small even if it has a relatively large AP. If the queue now starts to accumulate packets (*e.g.,* poor channel condition or bursty traffic load) and the AP is high, this node will abruptly send packets to the channel at high rate. The channel will not be able to accommodate the sudden increase of traffic which would lead to collisions. These collisions will be eventually resolved by AP adjustments but could result in increasing of average packet delay.

This negative effect is negligible when λ is high because u will be equal to one. But when the traffic load is light, the average packet delay could be an order of magnitude larger if the unnecessary channel time allocation is not taken care of. A simple method to avoid this case (a large value of AP with $u < 1$) is to add one more condition for a node to increase AP: a node can only increase AP in Case I when the utilization ratio is equal to one ($u = 1$).

The combined information from v and u enables the node to dynamically adjust the AP to approach the same performance as S-ALOHA*.

```
AIMD-MAC  (τ, α, β)
1   v^(-2) ← v^(-1)
2   u^(-2) ← u^(-1)
3   v^(-1) ← T^(-1) ≠ 0? S^(-1)/T^(-1) : -1
4   u^(-1) ← T^(-1)/(T^(-1) + T_w^(-1))
5   if v^(-1) = 0
6       p ← MAX(p^(-1) × 0.5, α)
7   elseif v^(-1) > 0
8       if v^(-1) ≥ v^(-2)
9           if u^(-1) = 1
10              p ← MIN(p^(-1) + α, 1.0)
11          end
12      else
13          p ← MAX(p^(-1) × β, α)
14      end
15  end
16  p^(-1) ← p
17  return p
```

Fig. 3.4: The AIMD-MAC protocol.

3.3.3 Pseudocode

Fig. 3.4 presents the pseudocode description of the AIMD-MAC protocol. τ, α and β are three preset parameters. The steps are periodically executed. Lines 1-4 update the transmission history and lines 5-15 adjust the AP periodically. The length of one UC is τ slots. α is the minimum AP applied by any node in the system and it is also the increment amount when a node's AP is increased. Naturally, α should be a positive fraction less than one and close to zero. β is the decrease rate of AP, when it is so decided.

Additive Increase The AP updating proceeds when $v^{(-1)} \geq 0$. A negative v (as defined in (Eq. 3.5)) represents an unknown channel condition and AIMD-MAC does

not change AP in this case (Case III). Line 7 indicates a positive success ratio and line 8 conditions on a channel improvement trend (Case I). Line 9 further signals that this node has to increase AP because it has used up its assigned channel time quota for the last UC. The AP linearly increases (line 10) for perfect resource utilization (*i.e.*, $u = 1$).

Multiplicative Decrease If $v^{(-1)} = 0$ (line 5), then all packet transmissions failed during the (-1)–th UC. This condition suggests a serious collision level and a slow recovery will negatively affect packet delay. In order to quickly decrease AP, the AP is multiplied by 0.5. If $v^{(-1)}$ is positive but the channel condition worsens (Case II) (line 12: $v^{(-1)} < v^{(-2)}$), then the AP is reduced less aggressively, by multiplying it by β, which is a constant greater than 0.5 but less than 1.

3.4 Performance Evaluation

The AIMD-MAC protocol is simulated in various settings, including synchronized and unsynchronized UCs, homogeneous and heterogenous systems. To show the performance of the AIMD-MAC under different traffic loads, we apply both a Poisson arrival process and a bursty traffic model in the simulations.

3.4.1 Simulation Scenarios

Systems with Synchronized and Unsynchronized UCs

For the purpose of extensive simulations, we discuss the synchronism of the UCs. We specify a system as ***synchronized*** when the UCs from all nodes start at the same slot such that the AP updating procedures proceed simultaneously. In this case, transmission histories from all nodes always cover exactly the same time interval. A more general circumstance is when nodes enter the system at different time slots. The UCs may be staggered (Fig. 3.5) and the transmission histories from different nodes cover overlapped but not aligned time intervals. We model the ***unsynchronized*** environment by specifying node m's starting time I_m as a discrete random variable uniformly distributed over the slot interval $[0, I_{MAX}]$.

Homogeneous and Heterogeneous Systems

Second, both homogeneous and heterogeneous systems are modeled. For the ***homogeneous*** environment, $\lambda_m = \lambda/M, m = 1, 2, ..., M$; for the ***heterogeneous*** environ-

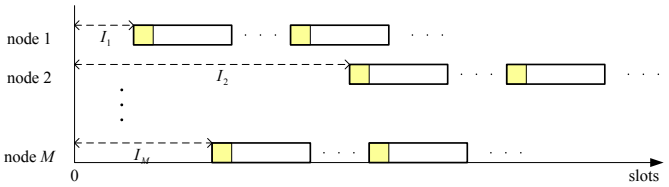


Fig. 3.5: The unsynchronized system.

ment, λ_m is uniformly distributed in the interval $[0, 2\frac{\lambda}{M}]$, such that the average across all is $\frac{\lambda}{M}$.

Poisson and Binary Source Traffic Models

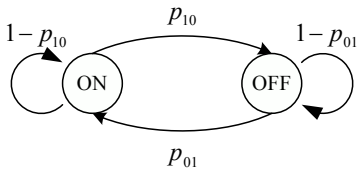


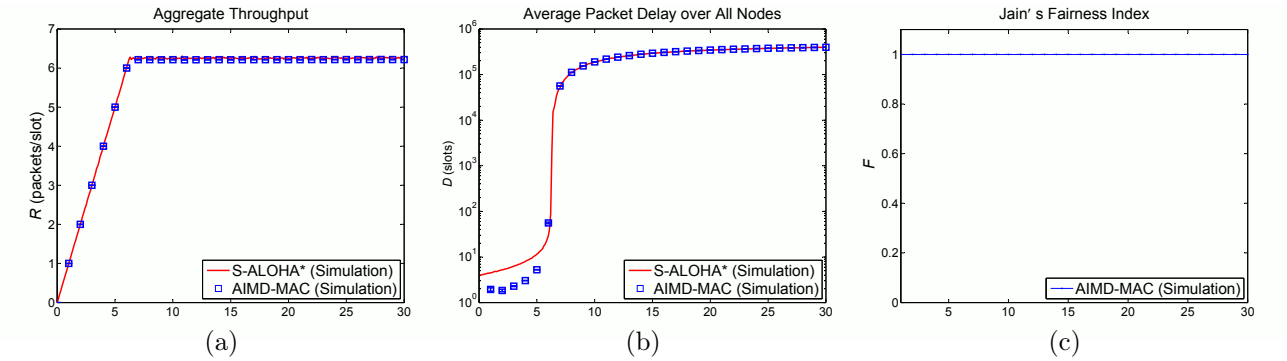
Fig. 3.6: The binary source model.

Third, to accurately show the performance of AIMD-MAC in various environments, we model traffic loads with different statistical characteristics. Poisson models and binary source models are studied. In the ***Poisson model***, the packet inter-arrival times at node m are exponentially distributed with mean arrival rate λ_m. In the ***binary source model*** (Fig. 3.6) [40], a node (source) is in one of two states at each slot: ***ON*** or ***OFF***. The initial state of a node is decided randomly. The state transition probability from ON to OFF and from OFF to ON is p_{10} and p_{01}, respectively. With probability $1 - p_{10}$ or $1 - p_{01}$, a node stays in the ON or OFF state. If node m is in the ON state, packets arrive at the node following a Poisson process with parameter λ'_m (Eq. 3.7); otherwise, this node is idle. With the ON-state arrival rate λ'_m, the mean arrival rate of node m over the entire simulation time is λ_m. This is applied to have a controllable aggregate load λ for convenient comparison.

$$\lambda'_m = \begin{cases} \lambda_m \times \frac{p_{10}+p_{01}}{p_{01}}, & \text{if } m \text{ is in ON state} \\ 0, & \text{if } m \text{ is in OFF state} \end{cases} \tag{3.7}$$

3.4.2 Simulation Results

We present a network system example with $M = 30$, $K = 10$, $\tau = 10$, $\alpha = 0.025$, $\beta = 0.9$ and $I_{MAX} = 10000$. If α is close to one, AP will fluctuate around the optimal



(a) (b) (c)

Fig. 3.7: The homogeneous synchronized system with Poisson arrivals.

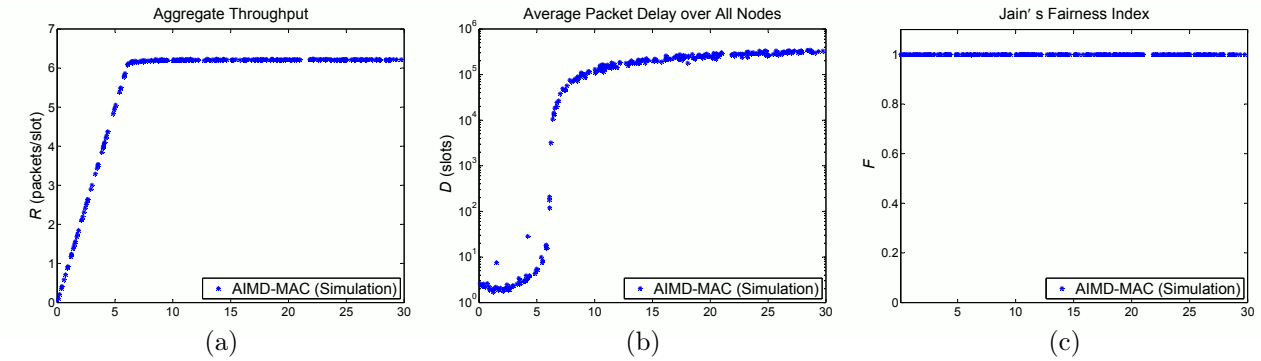
AP $p*$ with a large amplitude. For a similar reason, β should be a positive fraction close to one. (A more detailed parameter analysis will be presented in the next Section.) Nodes enter the system with a random AP (*i.e.*, $p_m \in [0,1],\ m = 1, ..., M$). Two network scenarios are considered, 1) the homogeneous synchronized system with Poisson arrival traffic (Fig. 3.7) and 2) a heterogeneous unsynchronized system with bursty traffic (Fig. 3.8). We consider the following metrics:

- **Aggregate Throughput** R: The total number of successful packet transmissions per slot from all nodes;
- **Packet Delay** D: The difference in time slots between the arrival and departure of the same packet. The average packet delay is computed over all the departed packets from all nodes;
- **Fairness** F: Fairness is evaluated by Jain's fairness index (Eq. 3.8) [47].

$$F = \frac{(\sum_{m=1}^{M} x_m)^2}{M \times \sum_{m=1}^{M} x_m^2} \tag{3.8}$$

where $x_m = \frac{R_m}{O_m}$. R_m is the per node throughput of node m and O_m is the fair share defined by the Jain's fairness criterion.

Fig. 3.7(a) shows an exact match of the aggregate throughput between AIMD-MAC and S-ALOHA*. We performed 10 runs of AIMD-MAC in the homogeneous system with the error bar (which are too small to be displayed) denoting the standard deviation. Even in the heterogeneous system with bursty traffic (Fig. 3.8(a)), the system can approximately achieve the optimal result. The result implies that even without knowing the number of nodes and channel capacity, AIMD-MAC can stabilize the AP at $p*$ for various traffic load.



(a) (b) (c)

Fig. 3.8: The heterogenous unsynchronized system with bursty traffic.

Fig. 3.7(b) displays the average packet delay on a log scale. Before the system is saturated (*i.e.,* $\lambda < 6.3$), the delay of AIMD-MAC is lower than that of S-ALOHA* because AIMD-MAC can adaptively increase the nodes' AP when the load is not heavy; however, S-ALOHA* applies a static strategy which keeps the same AP regardless of the current traffic. The average delay of AIMD-MAC shows a similar trend in the heterogeneous system (Fig. 3.8(b)).

The fairness index is almost equal to one in Fig. 3.7(c) and Fig. 3.8(c). This high fairness index is a direct result of using relative success ratios instead of a single universal threshold for adjusting APs.

The above simulations show that AIMD-MAC behaves similarly to the benchmark with respect to the throughput, delay and fairness in both the homogeneous synchronized model with Poisson arrival traffic and the heterogeneous unsynchronized model with bursty traffic. Note that the second model is not only a binary bursty traffic model. It also simulates a heterogeneous network where each node has different arrival rate and enters the system at various slot. Regardless of the characteristics of the network, compared to the Poisson model, AIMD-MAC shows persistent performance, which confirms the robustness of the algorithm.

3.5 The Adaptivity of AIMD-MAC

Having seen the performance of AIMD-MAC in one scenario with one parameter set, it is worth seeing whether AIMD-MAC can be adaptive for a dynamic system where the number of nodes M and the channel capacity K are not constant. In this section, we will introduce the oscillatory behavior of AIMD-MAC and demonstrate how the value of τ (*i.e.,* the update cycle (UC)) affects the distribution of the APs. We will show that with an appropriate set of parameters, AIMD-MAC can achieve

high performance in a wide range of scenarios without the requirement of error-less feedback at every slot.

3.5.1 The Oscillatory Behavior

The basic AIMD algorithm converges to the max-min fair rate allocation when all the nodes react uniformly to the command sent from a central observer [22]. AIMD-MAC assumes that error-less feedback about the status (successful or failed) of the reception is sent back to the transmitter after each transmission. If the node did not transmit at the beginning of a slot, it would not receive the feedback at the end of the slot. Because not all the nodes have the feedback at every slot, the convergence condition in the basic AIMD [22] does not hold in AIMD-MAC. Therefore, AIMD-MAC resorts to comparing the success ratios of two continuous UCs when updating the AP. Simply speaking, the AP of a node multiplicatively decreases if the success ratio decreases from its last value; otherwise, the AP additively increases. The additive-increase and multiplicative-decrease mechanism based on the values of the successive success ratios plays a significant role in having the nodes converge to a fair state.

In Section 3.2, we discussed the relationship between the expected aggregate throughput R, the AP p, the number of nodes M and the reception capacity K by Eq. 3.2. The optimal AP p^* to achieve the maximal throughput is expressed by Eq. 3.3 and the optimal aggregate throughput R^* is described by Eq. 3.4. From Eq. 3.3 and Eq. 3.2, for every M-K pair, p^* can be numerically computed. Fig. 3.9 plots the optimal APs for the systems with 10, 30 and 100 nodes and various channel capacities. The channel capacity K is normalized by M in the x-axis. p^* is monotonically increasing and is close to the value of K/M in all scenarios shown in Fig. 3.9. However, since we assume a non-stable environment, the values of K and M are unknown to the nodes and may change from time to time. Thus, a node cannot estimate p^* by K/M and the key objective of the AIMD-MAC algorithm is to adaptively estimate the value of p^* in the dynamic wireless environment to approach the optimal performance.

Assuming that all the M nodes apply the AP p in the K-MPR channel and the nodes always have packets to transmit, the expectation of success ratio v (Eq. 3.5) can be calculated by the following equation.

$$v(M, K, p) = \sum_{i=0}^{K-1} \binom{M-1}{i} p^i (1-p)^{M-1-i} \tag{3.9}$$

Eq. 3.9 is the first K terms of a binomial expression and it expresses that in the K-MPR channel, a transmission will be successful if there are less than K concurrent

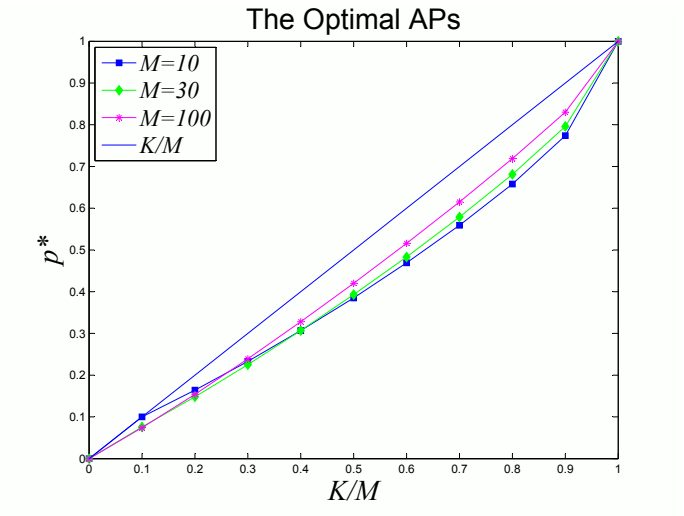


Fig. 3.9: Optimal access probabilities.

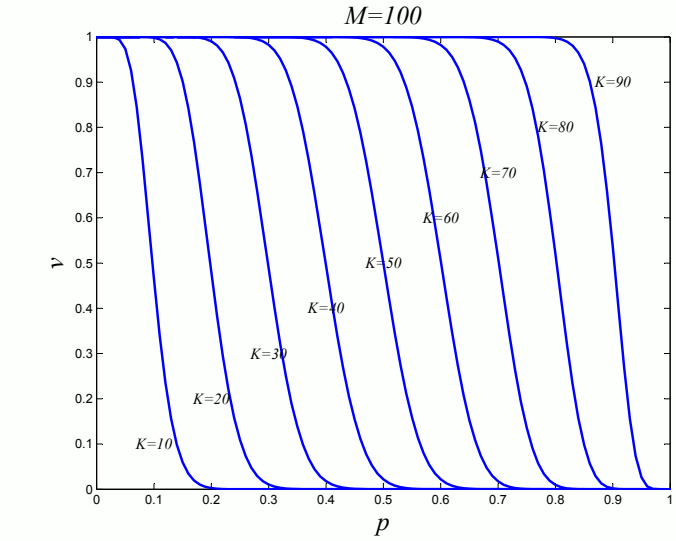


Fig. 3.10: Success ratios *vs.* access probability.

transmissions. Fig. 3.10 plots v *vs.* p for a 100-node system with channel capacities from 10 to 90. v is monotonically decreasing as p is increasing. When the nodes apply a larger p, the channel will have higher collision level and the nodes will observe a lower success ratio. By definition, AIMD-MAC increases the APs when v is increasing and decreases the APs when v is decreasing. Because Eq. 3.9 is a decreasing function, the increase of the AP will result in more collisions in the next UC and nodes will observe lower success ratio. According to AIMD-MAC, nodes will then decrease their APs. The procedure will repeat thereafter. Fig. 3.11 demonstrates the interaction between the success ratio and the AP.

Because the oscillatory behavior of AIMD-MAC, the nodes will have approximately equal number of increase and decrease operations regardless of the value of their APs. Because the decrease operation is implemented by multiplying by a factor less than one, those nodes with higher APs will decrease more than those with lower APs, which guarantees that the system will eventually attain a fair state.

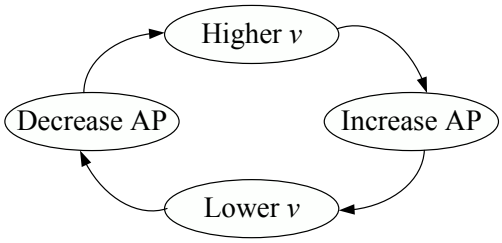


Fig. 3.11: The interaction between the success ratio and the AP.

$$\tilde{p} = \frac{\alpha}{1-\beta} \tag{3.10}$$

The value of AP at which the increasing amount equals the decreasing amount is called the ***equilibrium point*** and is denoted by $\tilde{p}$ in Eq. 3.10. $\tilde{p}$ is a function of α and β while p^* (Eq. 3.3) is a function of M and K. The value of $\tilde{p}$ is not necessarily close to the value of p^* because M and K are unknown to the nodes when setting the parameters of AIMD-MAC. If AIMD-MAC always has the APs oscillate around $\tilde{p}$, the performance will degrade when the value of $|\tilde{p} - p^*|$ is large. In other words, for AIMD-MAC to be adaptive in dynamic environments, the APs should not fluctuate around $\tilde{p}$ even though it is the equilibrium point. Instead, the value of the APs should always vary around p^*, within an appropriate interval, regardless of the value of the equilibrium point.

3.5.2 The Relationship Between UC and the Distribution of the APs

The above goal (*i.e.,* having the APs varying around p^* rather than $\tilde{p}$) is achieved by setting UC (τ) to be a relatively small value. In AIMD-MAC, nodes refrain from updating the AP if there was no transmission initiated from the node during the last UC. The success ratio is assigned to -1 (Eq. 3.5) to indicate the idle state. For a given value of AP, if τ is small, there is lower probability that a transmission will occur during one UC and therefore, higher probability that the node will not change its AP. P_{idle} (Eq. 3.11) computes the probability that a node will be idle during a UC when AP and UC are equal to p and τ, respectively.

$$P_{idle} = (1-p)^{\tau} \tag{3.11}$$

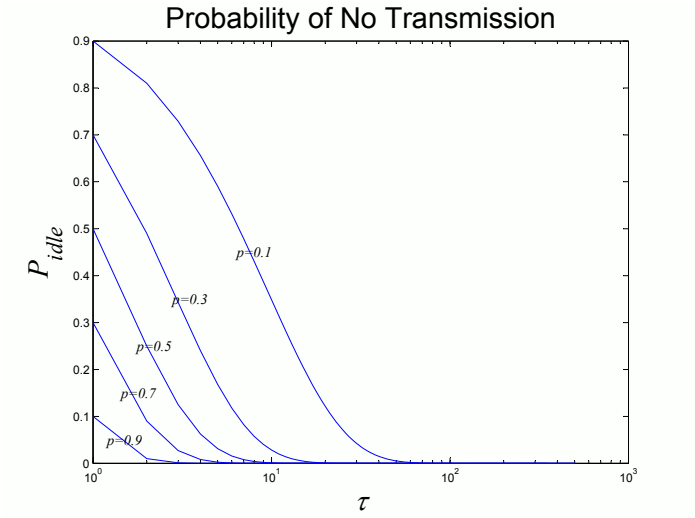
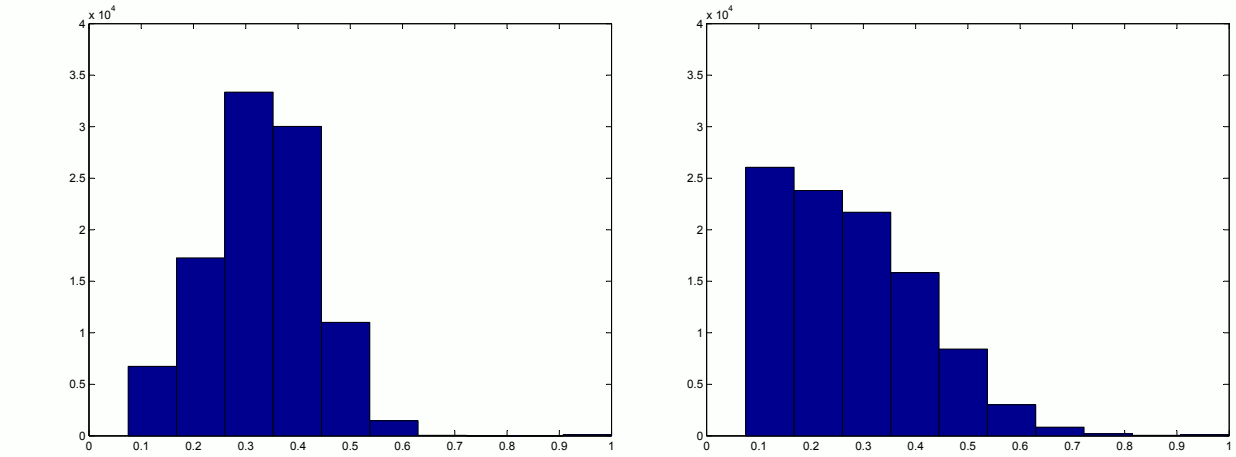


Fig. 3.12: The probability of no transmission for one update cycle.

Fig. 3.12 demonstrates the relationship between P_{idle} and τ for different values of p (0.1, 0.3, 0.5, 0.7 and 0.9). With a fixed τ, P_{idle} is decreasing as p increasing, which implies that the nodes with lower APs have greater probability to be idle and thus maintain this low AP for a longer time. On the other hand, the nodes with large APs have greater chance to transmit and their APs are updated more frequently. When τ is near 100, P_{idle} approaches zero regardless of the AP applied. Therefore, when τ is small, especially when τ is less than 10, more nodes will apply a small AP. If $\tilde{p}$ is greater than p^*, most nodes will apply an aggressive AP and the system performance will downgrade. This is the primary cause of poor performance in experiments. To

solve this problem, we can set τ to a small value so that the APs will drift to a smaller value to resolve the collision.



(a) $\tau = 100$ $R/R^* = 0.4$. (b) $\tau = 10$ $R/R^* = 0.9$.

Fig. 3.13: The histogram of the APs when $M = 100,\ K = 30$.

Fig. 3.13 shows the distribution of the APs when $M = 100$, $K = 30$, $\alpha = 0.075$ and $\beta = 0.9$, in which case $p^* = 0.24$ and $\tilde{p} = 0.75$. When UC is set to 100 (Fig. 3.13(a)), few nodes have AP greater than 0.6 even though the equilibrium point $\tilde{p}$ is 0.75. When $M = 100$, $K = 30$ and p is greater than 0.6, according to Fig. 3.9, the success ratio is close to zero. According to AIMD-MAC, the node which observed a zero success ratio should decrease the AP by half. Therefore, the APs are mostly upper bounded by 0.6. However, most nodes still apply an AP much greater than p^*, which leads to frequent collisions and poor performance ratio (*i.e.,* $R/R^* = 0.4$). Fig. 3.13(b) shows the distribution of AP when τ is set to 10. In this case, many nodes apply a lower AP because the nodes with lower AP are idle for a large probability and will keep the low AP for a longer time. With this AP distribution, collisions are largely avoided and the performance ratio reaches 90%.

3.5.3 Simulation Results

Because of the above attributes of AIMD-MAC, we demonstrate that with an appropriate set of parameters, AIMD-MAC can be applied to a wide range of scenarios to attain a sufficiently good performance. We investigate homogeneous synchronized systems (as in Fig. 3.7) with different number of nodes (*i.e.,* M) and various channel capacities (*i.e.,* K). The parameters are set as: $\alpha = 0.025$ $\beta = 0.9$ and $\tau = 10$. In each case, the aggregate throughput of AIMD-MAC (R) is compared with that of

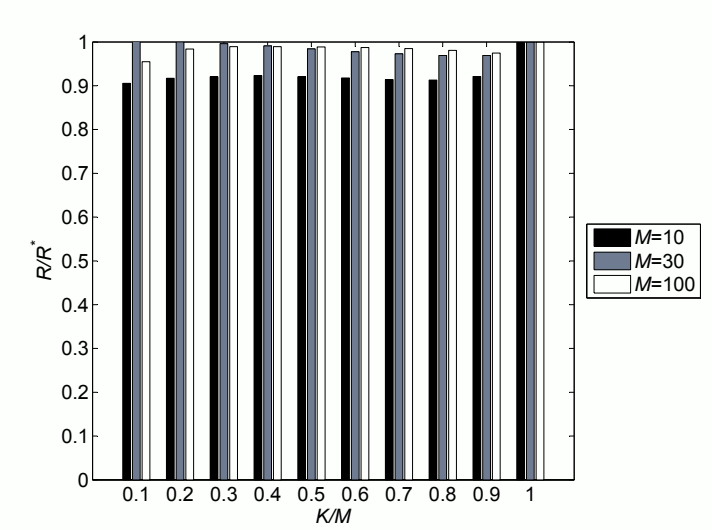


Fig. 3.14: The performance ratios when $M = 10, 30, 100$, $K = 1, ..., M$.

S-ALOHA* (R^*) when the system is saturated. The performance ratios R/R^* are plotted in Fig. 3.14. To display all the cases in one figure, the channel capacity K is normalized by the number of nodes M and represented by the x-axis. Note that Fig. 3.14 plots error bars with 99% two-side confidence interval which are too small to be distinguished. The performance ratios are greater than 0.9 in all the scenarios which suggests that AIMD-MAC can efficiently exploit the MPR capability in dynamic wireless environments. Since the UC is set to 10 (*i.e.,* $\tau = 10$), AIMD-MAC is invoked every 10 slots. Thus, the AIMD-MAC does not need the feedback at every slot. Instead, an aggregate feedback for the past 10 slots can be sent back to the transmitter at the end of every UC.

3.6 Summary

The AIMD-MAC algorithm is inspired by the basic AIMD algorithm [22] for its capability to achieve efficiency and fairness at the same time without the need for coordination among nodes. However, the basic AIMD cannot be directly applied to the wireless network to solve the medium access control problem because it is not trivial to guarantee that the nodes can receive accurate binary feedback at every slot. Without the timely feedback for all the nodes, the convergence property of the basic AIMD does not hold. We propose AIMD-MAC to tackle the random access problem of a wireless network while exploiting the MPR capacity. AIMD-MAC is a fully distributed MAC protocol which does not need feedback at every slot. In our simulations, as long as the nodes can receive the transmission history for every 10

slots, AIMD can achieve 90% of the optimal performance (defined by S-ALOHA*) in a dynamic wireless environments.

Chapter 4

Modelling Challenges of multi-hop MPT/MPR Networks

Having discussed how to exploit the MPR capability in the MAC layer of a single-hop scenario, starting from this chapter, we study how the MPT/MPR capability will affect a multi-hop wireless network. As mentioned in Section 2.3.2, it has been observed in the literature [63, 68, 97] that the MPR capability alone cannot linearly scale the system performance and the problem can be approached by bringing in the MPT capability. The need for MPT capability in a multi-hop system should also be intuitively clear. For example, consider a node that only relays traffic. Even if the node has MPR capability, it cannot utilize it effectively if it cannot transmit at the same rate, because in the long run the input and output rates of traffic traversing the node should match.

The inherent conflict between optimizing the channel utilization and achieving fairness has been recognized in [71]. The broadcasting nature of the wireless network leads to location-dependent contentions among multi-hop flows following paths intersecting or close to each other. Therefore, in multi-hop wireless networks the locations of the bottlenecks are dependent on the paths of the flows. To achieve fairness, the channel resource of the bottleneck should be fairly distributed among all the flows traversing the bottleneck. As we will see in this chapter, it is challenging to compute a fair allocation in mullti-hop MPT/MPR wireless networks, especially when the MPT and MPR capacities are unequal.

Therefore, in this chapter, our goal is to study how to fairly and efficiently manage multi-hop flows in MPT/MPR networks. The efficiency will be measured by the end-to-end throughput of the flows. To evaluate the fairness we use the max-min fairness and design novel algorithms to determine the max-min rate allocation for the flows.

The structure of this chapter is as follows. First, in Section 4.1, we model the

multi-hop MPT/MPR wireless networks via topology, flow and contention graphs. Section 4.2 introduces the advanced channel model and Section 4.3 defines the max-min fairness for multi-hop MPT/MPR wireless networks. Because *the M property* of the contention graph built for MPT/MPR networks significantly affects the understanding of the max-min problem, we first study the problem in conventional channel models (*i.e.,* $K_T = K_R = 1$) in Section 4.4. In Section 4.4, we propose a necessary and sufficient condition (Theorem 4.1) for max-min fairness in conventional networks. Based on Theorem 4.1, the wireless water-filling (WF) algorithm is introduced to compute the max-min rate allocation for multi-hop flows in conventional wireless networks. The result of WF can be scaled to symmetric networks (*i.e.,* $K_T = K_R = K,\ K = 1, 2, 3...$). We prove that the max-min allocation computed by WF is schedulable if the contention graph is a perfect graph and the minimum number of slots to realize the allocation can be computed as a by-product of WF. The challenges in computing the max-min fair allocation for general MPT/MPR networks (*i.e.,* $K_T, K_R \geq 1$) are discussed in Section 4.5. Due to *the M property*, even interfering transmissions can occur simultaneously. Thus, new rules are needed to regulate interfering transmissions so that the MPT and MPR capacity limits are not exceeded. We group the mutual interference relationships into eight patterns which are used to identify and normalize interfering traffic load. This procedure allows us to generate a set of novel constraints to manage interfering transmissions, based on which the WF-ASYM algorithm is stated to approximate the max-min allocation when K_T is not qual to K_R. Section 4.6 is the summary.

4.1 The Topology, Flow and Contention Graphs

We motivate and explain our model via a simple example. Consider a multi-hop wireless network of nine nodes which are equipped with half-duplex radios. Every node can either transmit or receive multiple packets simultaneously but not both. The topology graph of the network is shown by Fig. 4.1 (a). For simplicity, we assume symmetric communication links between nodes. However, our model can be adaptive to asymmetric networks to capture the irregular behavior of the links.

Five flows traverse the network as shown in the flow graph Fig. 4.1 (b) *e.g.,* flow 3 (f_3) is from D to F by passing through E. The flow graph extends the topology graph with directed edges corresponding to the links crossed by flows. For the rest of the paper, a ***link*** refers to the ability to communicate between two nodes and an ***arc*** represents a directed transmission in the flow graph. The direction of an arc is defined by the flow crossing it. If a link of the topology graph is crossed by n ($n > 1$)

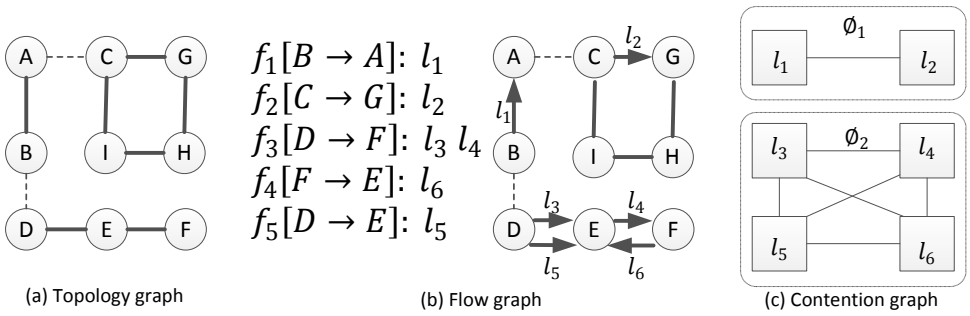


Fig. 4.1: The topology graph, flow graph and contention graph.

flows, we use n separate arcs in the flow graph to represent the link. *e.g.,* since link D to E in the topology graph is traversed by both f_3 and f_5, it is represented by both l_3 and l_5 in the flow graph.

Fig. 4.1 (c) is the contention graph which is built upon the flow graph. In the contention graph, every arc (*i.e.,* directed edge) of the flow graph is denoted by a vertex (We use the term ***node*** for the topology and flow graphs and ***vertex*** for the contention graph.). Two vertices are connected if, for the corresponding two arcs in the flow graph, one arc interferes with another. The interference between arcs is defined by edges (both directed and undirected) of the flow graph. In Fig. 4.1 (c), l_1 and l_2, which compose the maximal clique (MC) ϕ_1, are connected because C is in the interference range of A and l_2's transmission will impair l_1's reception at node A. However, although node B and D are in the interference range of each other, l_1 and l_3 (l_5) are not connected in the contention graph since l_3 (l_5) will not interfere with l_1 at node A and l_1 will not interfere with l_3 (l_5) at node E. l_3, l_4, l_5 and l_6 are mutually connected because they are adjacent to the same node and they compose MC ϕ_2.

The contention graph we described differs from its traditional definition [42] in the following ways:

(i) A link of the topology graph may correspond to multiple vertices in the contention graph, if it is traversed by multiple flows.

(ii) The interference relationship between vertices in the contention graph is related to the direction of the corresponding arcs in the flow graph.

(iii) ***The M property***: as mentioned in Section 2.2.2, in MPT/MPR channels, two vertices connected in the contention graph do not necessarily mean that the two corresponding arcs in the flow graph cannot be active simultaneously.

4.2 The Advanced Channel Model

It is worth noting that, in multi-hop wireless networks, a MPT/MPR channel model is not necessarily more advantageous than the collision channel model (*i.e.*, $K_T = K_R = 1$) with respect to the end-to-end throughout unless K_R is greater than one. To capture the requirement for K_R to be greater than one, we introduce the ***advanced channel model*** (Def. 4.1) and illustrate this model by Fig. 4.2.

Definition 4.1. ***Advanced Channel Model****: A channel model is named an advanced channel model if its MPR capacity limit K_R is greater than one.*

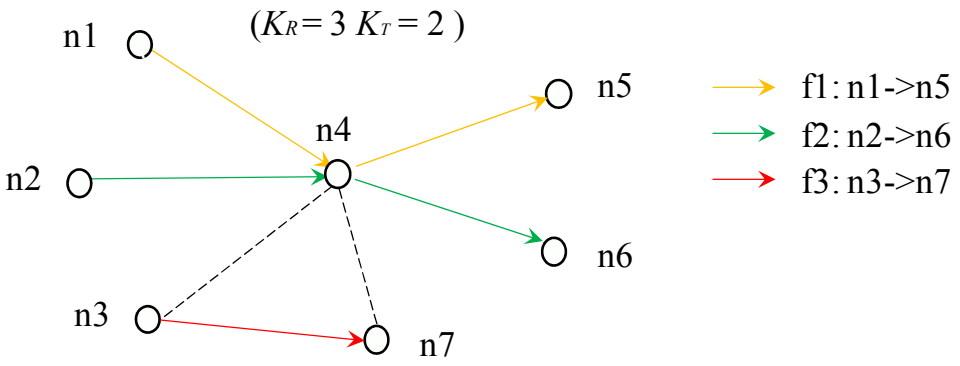


Fig. 4.2: An example of the advanced channel model.

Two nodes are in ***transmission range*** (denoted by solid lines) if they can decode the signals from each other. Two nodes are in ***interference range*** (denoted by dashed lines) but not in transmission range if they can sense the signals from each other but cannot decode them. In Fig. 4.2, three flows traverse the network. Node n4 is the relay of flow f1 and f2. Flow f3 does not cross n4 but n3 is located inside the interference range of n4. Thus, n4 needs to handle three inbound transmissions including the traffic from n3. If n4 has $K_T = 2$ and $K_R = 3$, n1, n2 and n3 can transmit simultaneously compared to the case when $K_T = K_R = 2$. Generally speaking, because a relay or a receiver of a wireless network needs to handle the interfering transmissions in addition to the desired transmissions, the overall performance can be improved if the relays or receivers have higher reception capacity than transmission capacity, *i.e.*, $K_R > K_T$. On the other hand, if $K_R < K_T$, the additional traffic cannot be decoded by the receiver and therefore, the end-to-end throughput cannot benefit from the greater transmission capacity limit. These conjectures will be confirmed in Section 5.3 by numerical results.

4.3 Max-Min Fairness for Multi-Hop Wireless Networks

In Section 2.2, we discussed the max-min fairness for wired networks and identified the analogy between a link of a wired network and a MC of a wireless network. In this section, we present the definition of max-min fairness for multi-hop MPT/MPR wireless networks.

Denote a MC by ϕ ($\phi \in \boldsymbol{\phi}$). For simplicity, we also refer to each member of ϕ as the corresponding arc in the flow graph. Def. 4.2 defines the max-min fair rate allocation for the set of flows $\boldsymbol{f}$.

Definition 4.2. ***Max-Min Fair Allocation****: In multi-hop MPT/MPR wireless networks, the rate allocation, $\boldsymbol{\rho_f}$, is max-min fair if*

(i) $\boldsymbol{\rho_f}$ *is feasible;*

(ii) while maintaining $\boldsymbol{\rho_f}$ *feasible,* ρ_f *cannot be increased without decreasing any* $\rho_{f'}$ *which is not greater than* ρ_f*.*

According to the analogy between a link of a wired network and a MC of a wireless network, in wireless networks, a rate allocation $\boldsymbol{\rho_f}$ is *feasible* if the capacity of any MC of the contention graph is not exceeded. Due to *the M property* of the contention graph in MPT/MPR wireless networks, the feasibility of the rate allocation leads to different constraints of the flow rates in conventional ($K_T = K_R = 1$) and MPT/MPR ($K_T,\ K_R \geq 1$) networks. In conventional channel models, we can normalize the capacity to be "1" for each maximal clique (*i.e.,* $C_\phi = 1$), which is not necessarily true in MPT/MPR networks. Therefore, we will investigate the max-min problem in conventional networks and MPT/MPR networks separately in the following two sections.

4.4 Max-Min Fairness in Conventional Multi-Hop Wireless Networks

A well-known algorithm to solve the max-min problem in wired networks is the basic water-filling algorithm [9]. This algorithm iteratively examines each unsaturated link and fairly distributes the available bandwidth of each link among the non-bottlenecked flows crossing the link. According to Prop. 2.1, a max-min fair allocation is achieved if every flow crosses at least one bottleneck link [1]. Based on the analogy between

[1]We assume all flows are greedy; *i.e.,* they can use up any allocation provided to them; equivalently, they can be thought of as having infinite demands.

a link of a wired network and a MC of a conventional wireless network, Huang *et al.* [42] propose algorithms to assign max-min fair allocation to single-hop flows. We extend Prop. 2.1 to multi-hop flows traversing conventional wireless networks ($K_T = K_R = 1$) and propose Theorem 4.1.

Based on Def. 4.2 and the necessary and sufficient condition (Theorem 4.1) for $\boldsymbol{\rho_f}$ to be max-min fair, we present next the wireless water-filling (WF) algorithm to compute the max-min fair allocation for multi-hop wireless networks when $K_T = K_R = 1$, denoted by $\boldsymbol{\rho_f}(1)$. Let $\boldsymbol{\rho_f}(K)$ be the max-min allocation when $K_T = K_R = K$. Since any arc scheduled to transmit one packet when $K_T = K_R = 1$ can transmit K packets when $K_T = K_R = K$, we can derive that $\boldsymbol{\rho_f}(K) = K\boldsymbol{\rho_f}(1)$. Thus, $\boldsymbol{\rho_f}(1)$ can be scaled to compute $\boldsymbol{\rho_f}(K)$.

4.4.1 A Necessary and Sufficient Condition for Max-Min Fairness

A couple of definitions are introduced before Theorem 4.1. Def. 4.3 defines a saturated MC ϕ with respect to a given rate allocation $\boldsymbol{\rho_f}$. Def. 4.4 further gives the conditions for MC ϕ to be a bottleneck MC for flow f. Note that $\boldsymbol{f}_\phi$ is the set of flows traversing ϕ (specifically, traversing an arc belonging to ϕ).

Following the definition of *clique feasible* in [7], let C_ϕ be the capacity of ϕ and $R_\phi = \sum_{l \in \boldsymbol{l}_\phi} \rho_l$ be the aggregate traffic traversing the arcs of ϕ. In conventional wireless networks a rate vector $\boldsymbol{\rho_f}$ is *feasible* if $R_\phi \leq C_\phi$ for any MC ϕ of the contention graph (*i.e.*, $\forall\ \phi \in \boldsymbol{\phi}$).

Definition 4.3. ***Saturated Maximal Clique****: For a feasible rate allocation $\boldsymbol{\rho_f}$ and a maximal clique ϕ, if $R_\phi = C_\phi$ the MC ϕ is called saturated.*

In conventional networks, the MC saturation condition can be expressed by one equation, $R_\phi = C_\phi$, however, the same definition does not apply in MPT/MPR networks. In Section 4.5 we will discuss the challenges to find the MC saturation conditions when multi-packets transmissions are allowed.

Definition 4.4. ***Bottleneck Maximal Clique****: For a given rate allocation $\boldsymbol{\rho_f}$, a maximal clique ϕ is a bottleneck maximal clique of a flow f if*

(i) ϕ is saturated with respect to $\boldsymbol{\rho_f}$;

(ii) flow f traverses ϕ (i.e., $f \in \boldsymbol{f}_\phi$);

(iii) flow f is a maximal rate flow crossing ϕ, i.e., $\rho_f = \max\{\rho_i | i \in \boldsymbol{f}_\phi\}$.

Additionally, all flows satisfying Def. 4.4 (iii) are called *bottlenecked flows* with respect to MC ϕ.

With the definition of the bottleneck MC (Def. 4.4), Theorem 4.1 provides a necessary and sufficient condition for a rate allocation $\boldsymbol{\rho_f}$ to be max-min fair in multi-hop wireless networks with conventional channel models.

Theorem 4.1. *In a multi-hop wireless network with conventional channels (i.e., $K_T = K_R = 1$), a feasible allocation set $\boldsymbol{\rho_f}$ is **max-min fair** if and only if each flow has a bottleneck MC.*

Proof. We prove the necessary condition by a contradiction: assume that for the max-min fair allocation $\boldsymbol{\rho_f}$ there exists a flow f_o not bottlenecked with respect to any ϕ. Denote the MCs traversed by f_o by $\boldsymbol{\phi}_{f_o}$ and the rate of f_o by ρ_{f_o}. Let $\boldsymbol{\phi}_{f_o} = \boldsymbol{\phi}_{f_o}^{\{b\}} \cup \boldsymbol{\phi}_{f_o}^{\{nb\}}$. $\boldsymbol{\phi}_{f_o}^{\{b\}}$ and $\boldsymbol{\phi}_{f_o}^{\{nb\}}$ denote the sets of bottleneck and non-botttleneck MCs traversed by f_o, respectively. If $\boldsymbol{\phi}_{f_o}^{\{nb\}} \neq \emptyset$, then $\delta^{\{nb\}} = \min_{\phi \in \boldsymbol{\phi}_{f_o}^{\{nb\}}} \{C_\phi - R_\phi\}$. If $\boldsymbol{\phi}_{f_o}^{\{b\}} \neq \emptyset$, for each $\phi \in \boldsymbol{\phi}_{f_o}^{\{b\}}$ there exists, by Def. 4.4 a flow, f', with rate higher than f_o. Let us define the following equation:

$$\delta^{\{b\}} = \min_{\phi \in \boldsymbol{\phi}_{f_o}^{\{b\}}} \{\rho_{f'} - \rho_{f_o} | f' \in \boldsymbol{f}_\phi,\ \rho_{f'} > \rho_{f_o}\} \tag{4.1}$$

We consider the following case:

(i) Both $\boldsymbol{\phi}_{f_o}^{\{nb\}}$ and $\boldsymbol{\phi}_{f_o}^{\{b\}}$ are non-empty then $\delta = min\{\delta^{\{nb\}}, \delta^{\{b\}}\}$
If $\delta^{\{b\}}$ is smaller than $\delta^{\{nb\}}$, we note that in each MC $\phi \in \boldsymbol{\phi}_{f_o}^{\{b\}}$, $\rho_{f'} - \rho_{f_o}$ rate can be reallocated from flow f' to flow f_o without reducing the rates of flows with rates lower than or equal to ρ_{f_o}. Note, in this case, the flows over non-bottlenecked MCs are not affected, since there is more available capacity there (*i.e.*, $\delta^{\{nb\}} > \delta^{\{b\}}$).

(ii) $\boldsymbol{\phi}_{f_o}^{\{nb\}}$ is empty but $\boldsymbol{\phi}_{f_o}^{\{b\}}$ is non-empty then $\delta = \delta^{\{b\}}$
We note that in each MC $\phi \in \boldsymbol{\phi}_{f_o}^{\{b\}}$, $\rho_{f'} - \rho_{f_o}$ rate can be reallocated from flow f' to flow f_o without reducing the rates of flows with rates lower than or equal to ρ_{f_o}.

(iii) $\boldsymbol{\phi}_{f_o}^{\{b\}}$ is empty but $\boldsymbol{\phi}_{f_o}^{\{nb\}}$ is non-empty then $\delta = \delta^{\{nb\}}$

We can increase the allocation of f_o by δ without reducing any of the allocation of flows traversing the same MC as f_o with rate lower than or equal to ρ_{f_o}. Hence ρ_f cannot be a max-min allocation.

The sufficient condition is straightforward. If every flow traverses at least one bottleneck MC, to increase the rate of any flow f, the rate of another flow f' which crosses the bottleneck MC of f must be decreased. By definition of the bottleneck MC, $\rho'_f \leq \rho_f$. Thus, $\boldsymbol{\rho_f}$ satisfies the conditions of max-min fairness. □

4.4.2 The Linear Programming Problem LP_{WF}

According to Theorem 4.1, we can solve the max-min problem by establishing at least one bottleneck for every flow. Therefore, before introducing the WF algorithm, we present the linear programming (LP) problem LP_{WF} (Eq. 4.2) to determine which MC becomes a bottleneck MC at each iteration of the WF algorithm.

$\rho_\phi = LP_{WF}(\phi,\ \widetilde{\boldsymbol{f}}_\phi,\ \boldsymbol{l}_\phi,\ \boldsymbol{\rho_{l_\phi}})$:

$$\max \quad \sum_{l \in \{l | l \in \boldsymbol{l}_\phi \wedge l \in \boldsymbol{l}_f \forall f \in \widetilde{\boldsymbol{f}}_\phi\}} \rho_l \tag{4.2a}$$

$$\text{subject to} \quad \rho_{l_{\phi,i}} = \rho_{l_{\phi,j}} \ \forall\ l_{\phi,i}, l_{\phi,j} \in \{l | l \in \boldsymbol{l}_\phi \wedge l \in \boldsymbol{l}_f \ \forall\ f \in \widetilde{\boldsymbol{f}}_\phi\} \tag{4.2b}$$

$$\rho_{l_{\phi,i}} = \rho_{f_{\phi,k}} \text{ if } l_{\phi,i} \in \boldsymbol{l}_{f_{\phi,k}} \wedge f_{\phi,k} \notin \widetilde{\boldsymbol{f}}_\phi \tag{4.2c}$$

$$\sum_{i=1}^{|\boldsymbol{l}_\phi|} \rho_{l_{\phi,i}} \leq 1 \tag{4.2d}$$

The objective of LP_{WF} is to calculate the fair share of an unsaturated MC ϕ for all the non-bottlenecked flows crossing ϕ when $K_T = K_R = 1$. The flows crossing ϕ (*i.e.,* $\boldsymbol{f}_\phi$) have unlimited demands and may include both bottlenecked flows and non-bottlenecked flows, denoted by $\widetilde{\boldsymbol{f}}_\phi$. $\boldsymbol{\rho_{f_\phi}}$ denotes the rate allocation of $\boldsymbol{f}_\phi$. $f_{\phi,k}$ is the k-th flow of $\boldsymbol{f}_\phi$ and $\rho_{f_{\phi,k}}$ represents the rate of $f_{\phi,k}$. $\boldsymbol{l}_\phi$ denotes the arcs of ϕ and $\boldsymbol{\rho_{l_\phi}}$ represents the rate allocation of $\boldsymbol{l}_\phi$. Let $l_{\phi,i}$ be the i-th arc of $\boldsymbol{l}_\phi$ and $\rho_{l_{\phi,i}}$ represent the rate of $l_{\phi,i}$. $\boldsymbol{l}_{f_{\phi,k}}$ represents the arcs constituting $f_{\phi,k}$. Note that $\rho_{l_{\phi,i}} = \rho_{f_{\phi,k}}$ if $l_{\phi,i} \in \boldsymbol{l}_{f_{\phi,k}}$.

Since the rate of a flow is determined by the rate of arcs composing the flow, the objective function Eq. 4.2a maximizes the sum of the arc rates of the non-bottlenecked flows crossing ϕ. Due to the flow conservation constraint, all the arcs of a flow should be assigned the same rate. According to Def. 4.4 and Theorem 4.1, all the flows bottlenecked at the same MC should have the same rate. Therefore, all the arcs of the non-bottlenecked flows crossing ϕ should be assigned the same rate, which is denoted by ρ_ϕ and is defined as the solution to LP_{WF}. Eq. 4.2b shows this ***equality constraint***. Any two arcs belonging to a non-bottlenecked flow, not necessarily the same flow, crossing ϕ should have the same rate. Guaranteeing the pair–wise equality is equivalent to assigning those arcs the same rate. If an arc belongs to a flow which has

been bottlenecked at earlier iterations of WF (*i.e.,* $l_{\phi,i} \in \boldsymbol{l}_{f_{\phi,k}}$ and $f_{\phi,k} \notin \widetilde{\boldsymbol{f}}_\phi$), its rate is determined by the rate of that flow (*i.e.,* $\rho_{l_{\phi,i}} = \rho_{f_{\phi,k}}$) and should not be changed. Eq. 4.2c specifies the rates of these fixed arcs. When $K_T = K_R = 1$, the resource of a MC can be considered as one unit. Thus, the node transmitting/receiving capability limits and the half-duplex restriction can be expressed by the ***node constraints*** (*i.e.,* Eq. 4.2d).

LP_{WF} fairly distributes the residual bandwidth of an unsaturated MC ϕ among all the arcs belonging to non-bottlenecked flows within ϕ. With the new fair share ρ_ϕ, ϕ will become saturated. If ϕ is identified as the bottleneck MC by WF, all the non-bottlenecked flows crossing ϕ will be bottlenecked at ϕ at the same rate ρ_ϕ. As we will see in Fig. 4.3, LP_{WF} is invoked repeatedly in the WF algorithm until all the flows are bottlenecked by at least one bottleneck MC.

4.4.3 The Wireless Water-Filling (WF) Algorithm

Having obtained ρ_ϕ of an unsaturated MC ϕ, we can present the wireless WF algorithm (Fig. 4.3) to compute the max-min fair allocation for wireless networks when $K_T = K_R = 1$. The parameters of WF include the set of MCs $\boldsymbol{\phi}$, the set of flows $\boldsymbol{f}$ and the set of arcs $\boldsymbol{l}$.

```
WF (φ, f, l)
1   φ̃ = φ; f̃ = f;
2   ρ_f is set to NULL;
3   Each element of ρ_φ is set to +∞;
4   while f̃ is not empty
5       for φ ∈ φ̃
6           ρ_φ = LP_WF(φ, f̃, l, ρ_f)
7       end
8       ρ*_φ = min_{φ∈φ̃}(ρ_φ);
9       for φ ∈ φ̃
10          if ρ_φ == ρ*_φ
11              for f ∈ f̃
12                  if (f ∈ f_φ) ρ_f = ρ*_φ; end
13              end
14              break;
15          end
16      end
17      φ̃ = {φ is not saturated};
18      f̃ = {f | f does not cross any saturated MC};
19  end;
20  return ρ_f;
```

Fig. 4.3: The wireless water-filling (WF) algorithm.

$\widetilde{\boldsymbol{\phi}}$ and $\widetilde{\boldsymbol{f}}$ denote the unsaturated MCs and the non-bottlenecked flows, which are initialized by $\boldsymbol{\phi}$ and $\boldsymbol{f}$, respectively, and are updated at the end of every iteration (line 17, 18). $\boldsymbol{\rho_f}$ is the rate allocation for $\boldsymbol{f}$ and $\boldsymbol{\rho_\phi}$ is the set of ρ_ϕ for each $\phi \in \boldsymbol{\phi}$. If not all the flows are bottlenecked (line 4), LP_{WF} calculates the fair share ρ_ϕ for every unsaturated MC (line 6). The minimum value of these shares is denoted by ρ^*_ϕ at line 8. For any unsaturated MC ϕ (line 9), if ρ_ϕ is equal to ρ^*_ϕ (line 10), it implies that the MC ϕ will be saturated by the rate ρ^*_ϕ at this iteration. If it is true, all the non-bottlenecked flows crossing ϕ are assigned the rate ρ^*_ϕ and become bottlenecked at line 12. The *break* statement at line 14 guarantees that WF saturates one MC at each iteration. Upon termination, $\widetilde{\boldsymbol{f}}$ is empty and all the flows are bottlenecked because every flow crosses at least one bottleneck MC. $\boldsymbol{\rho_f}$ records the resulting allocation for the flows of the network.

Because WF saturates one MC at each iteration, it invokes LP_{WF} at most $|\boldsymbol{\phi}| \times |\boldsymbol{f}|$ times. There is a "hidden" cost of finding the MCs which is a well-known hard problem [42]. To obtain a rate allocation for the case $K_T = K_R = K$, we can first use WF to compute the max-min rate allocation $\boldsymbol{\rho_f}$ for the case $K_T = K_R = 1$ and then scale the rate allocation by a factor of K (*i.e.,* $K\boldsymbol{\rho_f}$).

4.4.4 The Schedulability of WF

Definition 4.5. ***Schedulability****: Consider a synchronous time-slotted network with the flow set $\boldsymbol{f}$, a rate vector $\boldsymbol{\rho_f}$ is schedulable if there exists one discrete schedule such that any flow $f \in \boldsymbol{f}$ can achieve the rate ρ_f. Denoting the minimum number of slots to realize $\boldsymbol{\rho_f}$ by N, it means that for every N slots, every arc on the path of flow f can transmit $\rho_f N$ (an integer) packets for any $f \in \boldsymbol{f}$.*

It has been observed that not all max-min fair rate allocations are schedulable [25, 42]. Fig. 2.1 shows the case when a max-min fair rate allocation cannot be realized. Since all the MCs in this example have a size of two, the max-min rate of every flow is 0.5 packets/slot. However, when we try to arrange transmissions, there does not exist a discrete schedule with which every flow can realize this max-min rate. Indeed, since at most two arcs can be active at a slot, the maximum rate of the flows is 2/5 packets/slot.

It is known that the schedulability of an allocation is decided by the feasibility of the rate allocation and whether the contention graph is perfect [7]. A ***perfect graph*** is a graph G such that for every induced subgraph of G, the clique number equals the chromatic number [34]. For a rate allocation $\boldsymbol{\rho}$, [7] defines that $\boldsymbol{\rho}$ is ***clique feasible*** for the contention graph if for any clique, the sum of the traffic traversing this clique

does not exceed the capacity of the clique (which is 1 in conventional channels). Proposition 8 of [7] further states that for the contention graph, ***feasibility*** (graph feasibility) and ***clique feasibility*** are equivalent if and only if the contention graph is a perfect graph. Therefore, *for a perfect contention graph and a rate allocation* $\boldsymbol{\rho}$*, if* $\boldsymbol{\rho}$ *is clique feasible, there exists a feasible schedule such that every arc corresponding to a vertex of the contention graph can achieve the rate specified by* $\boldsymbol{\rho}$. Applying this result, we can prove that for perfect contention graphs, the max-min allocation computed by WF is schedulable. We start from the conventional channel model ($K_T = K_R = 1$) which we will extend to the case $K_T = K_R = 2,\ 3, ...$

Theorem 4.2. *For a wireless network with multi-hop flows and* $K_R = K_T = 1$*, if the contention graph of the network is a perfect graph, there exists a schedule that realizes the max-min allocation computed by WF.*

Proof. Let i denote the index of the iterations of the WF algorithm. At the i-th iteration, clique $\phi^{(i)}$ becomes bottlenecked and the flows traversing $\phi^{(i)}$ are bottlenecked at iteration i, if they have not already been bottlenecked at a previous iteration. Let $\boldsymbol{l}_{\phi^{(i)}}$ represent the set of arcs of $\phi^{(i)}$. $|\boldsymbol{l}_{\phi^{(i)}}|$ is the size of $\phi^{(i)}$. The arcs of the flows that are bottlenecked at the i-th iteration are denoted by $\boldsymbol{\tau}_i$ and the set of arcs bottlenecked during the first i iterations is denoted by $\mathbf{T}_i$ (*i.e.,* $\mathbf{T}_i = \cup_{j=1}^{i} \boldsymbol{\tau_j}$). (If a flow is bottlenecked, we say that all the arcs on the path of the flow are bottlenecked). The max-min rate of the flows bottlenecked at the i-th iteration, or equivalently, the rate of $\boldsymbol{\tau}_i$, is obtained at the i-th iteration and is not changed in the successive iterations. Therefore, at the end of the i-th iterations, WF obtains i max-min rates for $\{\boldsymbol{\tau}_1, ..., \boldsymbol{\tau}_i\}$. The max-min rates computed during the first i iterations are represented by $\boldsymbol{\rho}^{(i)}$ (*i.e.,* $|\boldsymbol{\rho}^{(i)}| = i$) where $\rho_j^{(i)}$ stores the max-min rate of $\boldsymbol{\tau}_j$ for $j = 1, ..., i$. Now we use induction to show that $\boldsymbol{\rho}^{(i)}$ maintains *clique feasibility* for the contention graph for any iteration of WF. Let $\boldsymbol{\rho}$ be the rate set obtained at the final iteration of WF, the following proof will show that this max-min rate set $\boldsymbol{\rho}$ is *clique feasible*.

Step 1: First, we show that $\boldsymbol{\rho}^{(1)} = \{\rho_1^{(1)}\}$ is clique feasible. Since all the arcs have rate 0 initially, all the MCs have full capacity which is equal to one. To compute $\rho_1^{(1)}$ (*i.e.,* the max-min rate of $\boldsymbol{\tau}_1$), Eq. 4.3 selects the MC with maximal size, denoted by $\phi^{(1)}$, and divides the capacity one by the size of ϕ_1 (*i.e.,* $|\boldsymbol{l}_{\phi^{(1)}}|$). $\phi^{(1)}$ satisfies $\phi^{(1)} = \text{argmin}_{\phi \in \boldsymbol{\phi}} \frac{1}{|\boldsymbol{l}_\phi|}$ and $\rho_1^{(1)} = \frac{1}{|\boldsymbol{l}_{\phi^{(1)}}|}$.

Let indicator function $\mathbb{1}_{\boldsymbol{\tau}_1}(l) = 1$ if arc $l \in \boldsymbol{\tau}_1$; otherwise $\mathbb{1}_{\boldsymbol{\tau}_1}(l) = 0$. To examine the clique feasibility of $\boldsymbol{\rho}^{(1)}$, it is necessary to compute the aggregate traffic of every MC and evaluate if it exceeds the capacity limit. For any MC $\phi \in \boldsymbol{\phi}$, the rate of arc l of ϕ can be represented by $\rho_1^{(1)} \mathbb{1}_{\boldsymbol{\tau}_1}(l)$ (*i.e.,* the rate of l is equal to $\rho_1^{(1)}$ if l is

bottlenecked at the first iteration) and the aggregate traffic of ϕ is $\sum_{l\in\boldsymbol{l}_\phi}\rho_1^{(1)}\mathbb{1}_{\boldsymbol{\tau}_1}(l)$ (Eq. 4.4). Because all the arcs of $\phi^{(1)}$ are bottlenecked at the first iteration, $\boldsymbol{l}_{\phi^{(1)}}\subseteq\boldsymbol{\tau}_1$ and $\sum_{l\in\boldsymbol{l}_{\phi^{(1)}}}\rho_1^{(1)}\mathbb{1}_{\boldsymbol{\tau}_1}(l)=|\boldsymbol{l}_{\phi^{(1)}}|\rho_1^{(1)}=|\boldsymbol{l}_{\phi^{(1)}}|\frac{1}{|\boldsymbol{l}_{\phi^{(1)}}|}=1$. Thus Eq. 4.4 holds when $\phi\in\{\phi^{(1)}\}$. If the aggregate traffic for any $\phi\in\boldsymbol{\phi}\setminus\{\phi^{(1)}\}$ is greater than 1, there exists a MC ϕ' such that $\sum_{l\in\boldsymbol{l}_{\phi'}}\rho_1^{(1)}\mathbb{1}_{\boldsymbol{\tau}_1}(l)=|\boldsymbol{l}_{\phi'}\cap\boldsymbol{\tau}_1|\frac{1}{|\boldsymbol{l}_{\phi^{(1)}}|}>1$, *i.e.,* $\frac{1}{|\boldsymbol{l}_{\phi^{(1)}}|}>\frac{1}{|\boldsymbol{l}_{\phi'}\cap\boldsymbol{\tau}_1|}\geq\frac{1}{|\boldsymbol{l}_{\phi'}|}$, which contradicts that $\phi^{(1)}=\text{argmin}_{\phi\in\boldsymbol{\phi}}\frac{1}{|\boldsymbol{l}_\phi|}$. Therefore Eq. 4.4 holds at the first iteration and $\boldsymbol{\rho}^{(1)}$ is clique feasible.

$$\rho_1^{(1)}=\min_{\phi\in\boldsymbol{\phi}}\{\frac{1}{|\boldsymbol{l}_\phi|}\} \tag{4.3}$$

$$\sum_{l\in\boldsymbol{l}_\phi}\rho_1^{(1)}\mathbb{1}_{\boldsymbol{\tau}_1}(l)\begin{cases}=1 & \phi\in\{\phi^{(1)}\}\\ \leq 1 & \phi\in\boldsymbol{\phi}\setminus\{\phi^{(1)}\}\end{cases} \tag{4.4}$$

Step 2: Assume $\boldsymbol{\rho}^{(i)}=\{\rho_1^{(i)},...,\rho_i^{(i)}\}$ is clique feasible at the i-th iteration, *i.e.,* Eq. 4.5 holds. The i-th iteration computes the max-min rate for $\boldsymbol{\tau}_i$ and saturates $\phi^{(i)}$. Therefore, at the end of the i-th iteration, $\{\rho_1^{(i)},...,\rho_i^{(i)}\}$ are the max-min rates for $\{\boldsymbol{\tau}_1,...,\boldsymbol{\tau}_i\}$ and $\{\phi^{(1)},...,\phi^{(i)}\}$ are saturated MCs. Eq. 4.5 computes the aggregate traffic of all the MCs by counting in the max-min rates obtained during the first i iterations. The aggregate traffic of the MC is equal to 1 if the MC is saturated; otherwise it is not greater than 1.

$$\sum_{l\in\boldsymbol{l}_\phi}\sum_{j=1}^{i}\rho_j^{(i)}\mathbb{1}_{\boldsymbol{\tau}_j}(l)\begin{cases}=1 & \phi\in\{\phi^{(1)},...,\phi^{(i)}\}\\ \leq 1 & \phi\in\boldsymbol{\phi}\setminus\{\phi^{(1)},...,\phi^{(i)}\}\end{cases} \tag{4.5}$$

Step 3: In order to show that $\boldsymbol{\rho}^{(i+1)}=\{\rho_1^{(i)},...,\rho_{i+1}^{(i+1)}\}$ is clique feasible, it is necessary to compute each element of $\boldsymbol{\rho}^{(i+1)}$ (computed at line 8 of Fig. 4.3), which is shown by Eq. 4.6. For the flows bottlenecked during the first i iterations, their max-min rates are not changed in the successive iterations. Therefore, $\{\rho_1^{(i+1)},...,\rho_i^{(i+1)}\}=\{\rho_1^{(i)},...,\rho_i^{(i)}\}$.

$$\rho_j^{(i+1)}=\begin{cases}\rho_j^{(i)} & j=1,...,i\\ \min_{\phi\in\boldsymbol{\phi}}\{\frac{1-\sum_{k=1}^{i}\rho_k^{(i)}|\boldsymbol{\tau}_k\cap\boldsymbol{l}_\phi|}{|\boldsymbol{l}_\phi-\mathbf{T}_i|}\} & j=i+1\end{cases} \tag{4.6}$$

At the $(i+1)$-th iteration, WF computes the max-min rate $\rho_{i+1}^{(i+1)}$ for $\boldsymbol{\tau}_{i+1}$. In the second case of Eq. 4.6, $\boldsymbol{\tau}_k$ represents the arcs bottlenecked at the k-th iteration. $\boldsymbol{\tau}_k\cap\boldsymbol{l}_\phi$ are the arcs of ϕ which were bottlenecked at the k-th iteration. Thus, $\rho_k^{(i)}|\boldsymbol{\tau}_k\cap\boldsymbol{l}_\phi|$ represents a portion of the clique capacity of ϕ that was "consumed" at the k-th iteration. The numerator represents the residual clique capacity of ϕ which can be

allocated to the non-bottleneck arcs of ϕ (*i.e.,* $\boldsymbol{l}_\phi - \mathbf{T}_i$). Thus, the quotient is the result of evenly allocating the residual capacity of ϕ over all the non-bottleneck arcs of ϕ. The minimum quotient over all the MCs is $\rho_{i+1}^{(i+1)}$, with which $\phi^{(i+1)}$ becomes saturated. $\phi^{(i+1)}$ can be expressed by $\phi^{(i+1)} = \text{argmin}_{\phi \in \boldsymbol{\phi}} \frac{1-\sum_{k=1}^{i} \rho_k^{(i)} |\boldsymbol{\tau}_k \cap \boldsymbol{l}_\phi|}{|\boldsymbol{l}_\phi - \mathbf{T}_i|}$.

$$\sum_{l \in \boldsymbol{l}_\phi} \sum_{j=1}^{i+1} \rho_j^{(i+1)} \mathbb{1}_{\boldsymbol{\tau}_j}(l) = \sum_{l \in \boldsymbol{l}_\phi} \sum_{j=1}^{i} \rho_j^{(i+1)} \mathbb{1}_{\boldsymbol{\tau}_j}(l) + \sum_{l \in \boldsymbol{l}_\phi} \rho_{i+1}^{(i+1)} \mathbb{1}_{\boldsymbol{\tau}_{i+1}}(l) = \sum_{l \in \boldsymbol{l}_\phi} \sum_{j=1}^{i} \rho_j^{(i)} \mathbb{1}_{\boldsymbol{\tau}_j}(l) + \sum_{l \in \boldsymbol{l}_\phi} \rho_{i+1}^{(i+1)} \mathbb{1}_{\boldsymbol{\tau}_{i+1}}(l) \quad (4.7a)$$

$$= \begin{cases} \sum_{l \in \boldsymbol{l}_\phi} \sum_{j=1}^{i} \rho_j^{(i)} \mathbb{1}_{\boldsymbol{\tau}_j}(l) = 1 & \phi \in \{\phi^{(1)}, ..., \phi^{(i)}\} \\ \sum_{l \in \boldsymbol{l}_{\phi^{(i+1)}}} \left(\sum_{j=1}^{i} \rho_j^{(i)} \mathbb{1}_{\boldsymbol{\tau}_j}(l) + \rho_{i+1}^{(i+1)} \mathbb{1}_{\boldsymbol{\tau}_{i+1}}(l) \right) = 1 & \phi \in \{\phi^{(i+1)}\} \\ \sum_{l \in \boldsymbol{l}_\phi} \sum_{j=1}^{i} \rho_j^{(i)} \mathbb{1}_{\boldsymbol{\tau}_j}(l) + \sum_{l \in \boldsymbol{l}_\phi} \rho_{i+1}^{(i+1)} \mathbb{1}_{\boldsymbol{\tau}_{i+1}}(l) \leq 1 & \phi \in \boldsymbol{\phi} \setminus \{\phi^{(1)}, ..., \phi^{(i+1)}\} \end{cases} \quad (4.7b)$$

Eq. 4.7 shows the aggregate traffic of every MC when the arcs of $\boldsymbol{\tau}_j$ have the rate $\rho_j^{(i+1)}$ for $j = 1, ..., i+1$. Because $\{\rho_1^{(i+1)}, ..., \rho_i^{(i+1)}\} = \{\rho_1^{(i)}, ..., \rho_i^{(i)}\}$, the original expression can be converted to the last expression of Eq. 4.7a. Eq. 4.7b further divides the equation into three cases. Because the MCs belonging to $\{\phi^{(1)}, ..., \phi^{(i)}\}$ are saturated before the $(i+1)$-th iteration and the rates for their arcs are not changed anymore, the aggregate traffic of ϕ for any $\phi \in \{\phi^{(1)}, ..., \phi^{(i)}\}$ can be expressed by the first case of Eq. 4.7b. According to Eq. 4.5, it is equal to 1. For $\phi^{(i+1)}$, the aggregate traffic is shown by the second case of Eq. 4.7b. Replacing $\rho_{i+1}^{(i+1)}$ by Eq. 4.6, the aggregate traffic of $\phi^{(i+1)}$ is equal to one. Suppose to the contrary, that the third case of Eq. 4.7b does not hold. Then, there exists a MC ϕ' in $\boldsymbol{\phi} \setminus \{\phi^{(1)}, ..., \phi^{(i+1)}\}$ such that $\sum_{l \in \boldsymbol{l}_{\phi'}} \sum_{j=1}^{i} \rho_j^{(i)} \mathbb{1}_{\boldsymbol{\tau}_j}(l) + \sum_{l \in \boldsymbol{l}_{\phi'}} \rho_{i+1}^{(i+1)} \mathbb{1}_{\boldsymbol{\tau}_{i+1}}(l) > 1$, from which we can derive that $\rho_{i+1}^{(i+1)} > \frac{1-\sum_{k=1}^{i} \rho_k^{(i)} |\boldsymbol{\tau}_k \cap \boldsymbol{l}_{\phi'}|}{|\boldsymbol{l}_{\phi'} - \mathbf{T}_i|}$. However, this contradicts Eq. 4.6. Based on the above reasoning, Eq. 4.7 holds.

Based on the three steps, the allocation is clique feasible at any iteration of WF. Therefore, the max-min rate allocation $\boldsymbol{\rho}$ obtained at the final iteration of WF is clique feasible. According to Proposition 8 of [7], $\boldsymbol{\rho}$ is feasible. □

Denote the max-min schedule of a network with $K_T = K_R = 1$ by $\boldsymbol{S}(1)$. Lemma 4.1 states a method to construct a feasible schedule in symmetric networks based on the max-min schedule of the conventional network.

Lemma 4.1. *For a wireless network with multi-hop flows, if the contention graph of the network is a perfect graph, scale the schedule $\boldsymbol{S}(1)$ by a factor $K > 1$ and denote the new schedule by $\boldsymbol{S}(K)$. $\boldsymbol{S}(K)$ is feasible for the same topology and flows when $K_T = K_R = K$.*

Proof. According to Theorem 4.2, a feasible max-min schedule $\boldsymbol{S}(1)$ can be constructed based on the solution of WF when the contention graph is a perfect graph.

Since both the transmission and reception capacity limits are scaled by a factor K, every transmission of $\boldsymbol{S}(1)$ can be replaced by K concurrent transmissions without exceeding the limits. Thus, $\boldsymbol{S}(K)$ is feasible in the network with the same flows when $K_T = K_R = K$. □

For a slot-synchronized system, the least number of slots needed to realize $\boldsymbol{\rho}$ is the smallest integer, denoted by N_{min}, such that $\boldsymbol{\rho}N_{min}$ is an integer set. N_{min} is called the ***minimal scheduling cycle***. When $K_T = K_R = 1$, a feasible schedule to realize $\boldsymbol{\rho}$ is to assign $\rho_i N_{min}$ slots to the arcs of rate ρ_i for $i = 1, ..., |\boldsymbol{\rho}|$. Next, we show that N_{min} can be calculated as a byproduct of the WF algorithm.

4.4.5 Minimal Scheduling Cycle

The computation of N_{min} is based on Lemma 4.2.

Lemma 4.2. *For $i = 1, 2, ..., |\boldsymbol{\rho}|$, the max-min rate ρ_i computed at the i-th iteration of WF can be expressed by Eq. 4.8, where $f(j,i)$, $a(j,i)$ and $g(j,i)$ are shown by Eq. 4.9, Eq. 4.10 and Eq. 4.11, respectively.*

$$\rho_i^{(i)} = \frac{f(1,i) - \sum_{j=1}^{i-1} a(j,i)\left(f(1,j) + \sum_{t=1}^{j-1} f(1,t)g(t,j)\right)}{f(1,i+1)} \tag{4.8}$$

$$f(j,i) = \begin{cases} \prod_{t=j}^{i-1} |\boldsymbol{l}_{\phi^{(t)}} - \mathbf{T}_{t-1}| & \text{if } j < i \\ 1 & \text{if } j \geq i \end{cases} \tag{4.9}$$

$$a(j,i) = \begin{cases} |\boldsymbol{\tau}_j \cap \boldsymbol{l}_{\phi^{(i)}}| f(j+1,i) & \text{if } j < i \\ 0 & \text{if } j \geq i \end{cases} \tag{4.10}$$

$$g_(j,i) = \begin{cases} -a(j,i) - \sum_{t=j+1}^{i-1} a(t,i)g(j,t) & \text{if } j < i \\ 0 & \text{if } j \geq i \end{cases} \tag{4.11}$$

Proof. Note that $f(j,i)$ (Eq. 4.9) has the property shown by Eq. 4.12.

$$f(j,i) = f(j,k)f(k,i), \quad j \leq k \leq i \tag{4.12}$$

We apply induction to prove Lemma 4.2.

- Initially, all arcs are non-bottleneck arcs and $\mathbf{T}_0 = \emptyset$. When $i = 1$, according to Eq. 4.3, $\rho_1^{(1)} = \frac{1}{|\boldsymbol{l}_{\phi^{(1)}}|}$. Based on Eq. 4.9, $\frac{f(1,1)}{f(1,2)} = \frac{1}{|\boldsymbol{l}_{\phi^{(1)}}|}$. Therefore, $\rho_1^{(1)} = \frac{f(1,1)}{f(1,2)}$ and Eq. 4.8 holds for $i = 1$.

- Assume Eq. 4.8 holds for $\rho_k^{(k)}$, $k = 1, ..., i$.

- Now we show that for $i+1$, $\rho_{i+1}^{(i+1)}$ (Eq. 4.6) can be expressed in the format of Eq. 4.8. Note that $\rho_k^{(k)} = \rho_k^{(i)}$ for $k = 1, ..., i$. Replacing $\rho_k^{(i)}$ in Eq. 4.6 by $\rho_k^{(k)}$ of Eq. 4.8 for $k = 1, ..., i$, $\rho_{i+1}^{(i+1)}$ can be expressed by Eq. 4.13.

$$\rho_{i+1} = \frac{1 - \sum_{j=1}^{i} \frac{f(1,j) - \sum_{k=1}^{j-1} a(k,j)\left(f(1,k) + \sum_{t=1}^{k-1} f(1,t)g(t,k)\right)}{f(1,j+1)} |\boldsymbol{\tau}_j \cap \boldsymbol{l}_{\phi^{(i+1)}}|}{|\boldsymbol{l}_{\phi^{(i+1)}} - \mathbf{T}_i|} \tag{4.13}$$

According to Eq. 4.9 and 4.12, $\frac{1}{|\boldsymbol{l}_{\phi^{(i+1)}} - \mathbf{T}_i|} = \frac{f(1,i+1)}{f(1,i+2)}$. Thus, Eq. 4.13 becomes Eq. 4.14.

$$\rho_{i+1} = \frac{f(1,i+1) - \sum_{j=1}^{i} \frac{f(1,i+1)|\boldsymbol{\tau}_j \cap \boldsymbol{l}_{\phi^{(i+1)}}|}{f(1,j+1)} \left(f(1,j) - \sum_{k=1}^{j-1} a(k,j) \left(f(1,k) + \sum_{t=1}^{k-1} f(1,t)g(t,k)\right)\right)}{f(1,i+2)} \tag{4.14}$$

Because $f(j+1, i+1) = \frac{f(1,i+1)}{f(1,j+1)}$ (Eq. 4.12), Eq. 4.15 holds, which can be converted to Eq. 4.16 because $a(j, i+1) = f(j+1, i+1)|\boldsymbol{\tau}_j \cap \boldsymbol{l}_{\phi^{(i+1)}}|$ (Eq. 4.10).

$$\rho_{i+1} = \frac{f(1,i+1) - \sum_{j=1}^{i} f(j+1,i+1)|\boldsymbol{\tau}_j \cap \boldsymbol{l}_{\phi^{(i+1)}}| \left(f(1,j) - \sum_{k=1}^{j-1} a(k,j) \left(f(1,k) + \sum_{t=1}^{k-1} f(1,t)g(t,k)\right)\right)}{f(1,i+2)} \tag{4.15}$$

$$\rho_{i+1} = \frac{f(1,i+1) - \sum_{j=1}^{i} a(j,i+1) \left(f(1,j) - \sum_{k=1}^{j-1} a(k,j) \left(f(1,k) + \sum_{t=1}^{k-1} f(1,t)g(t,k)\right)\right)}{f(1,i+2)} \tag{4.16}$$

The term $\sum_{k=1}^{j-1} a(k,j) \left(f(1,k) + \sum_{t=1}^{k-1} f(1,t)g(t,k)\right)$ in Eq. 4.16 can be rearranged as $\sum_{k=1}^{j-1} f(1,k) \left(a(k,j) + \sum_{t=k+1}^{j-1} a(t,j)g(k,t)\right)$. According to Eq. 4.11, $-g(k,j) = a(k,j) + \sum_{t=k+1}^{j-1} a(t,j)g(k,t)$ and Eq. 4.17 holds, which completes the proof.

$$\rho_{i+1} = \frac{f(1,i+1) - \sum_{j=1}^{i} a(j,i+1) \left(f(1,j) + \sum_{k=1}^{j-1} f(1,k)g(k,j)\right)}{f(1,i+2)} \tag{4.17}$$

□

Re-express Eq. 4.8 by $\frac{\mathcal{N}(i)}{\mathcal{D}(i)}$ where $\mathcal{N}(i)$ and $\mathcal{D}(i)$ are the numerator and denominator of Eq. 4.8, respectively. Since $f(j,i)$, $a(j,i)$ and $g(j,i)$ are integers, $\mathcal{N}(i)$ and $\mathcal{D}(i)$ are also integers. Thus, $\frac{\mathcal{D}(i)}{gcd(\mathcal{N}(i),\mathcal{D}(i))}$ is the minimum number of slots needed such that $\rho_i^{(i)}$ can be realized in discrete systems. Therefore, the least common multiplier of $\frac{\mathcal{D}(i)}{gcd(\mathcal{N}(i),\mathcal{D}(i))}$ for $i = 1, ..., |\boldsymbol{\rho}|$ is the least number of slots needed to schedule $\boldsymbol{\rho}$, *i.e.,* Eq.

4.18 holds. Since $\boldsymbol{l}_{\phi^{(i)}}$, $\boldsymbol{\tau}_i$ and $\mathbf{T}_i$ can be computed at the i-th iteration of WF, N_{min} can be obtained as a by-product of WF.

$$N_{min} = lcm\left(\frac{\mathcal{D}(1)}{gcd(\mathcal{N}(1),\mathcal{D}(1))}, ..., \frac{\mathcal{D}(|\boldsymbol{\rho}|)}{gcd(\mathcal{N}(|\boldsymbol{\rho}|),\mathcal{D}(|\boldsymbol{\rho}|))}\right) \tag{4.18}$$

4.5 Max-Min Fairness in MPT/MPR Multi-Hop Wireless Networks

We investigate next the characteristics of the MPT/MPR wireless networks and show that, when multiple transmissions participating in the same MC can occur simultaneously (*i.e.,* the M property), even determining the feasibility of a rate allocation and the saturation status of a MC are challenging. In this section, we study the feasibility of a rate allocation by introducing the eight interference patterns and illustrate the challenges to find a necessary and sufficient condition for a saturated MC in MPT/MPR networks. Based on the eight interference patterns we propose the WF-ASYM algorithm to approximate the max-min rate allocation in MPT/MPR multi-hop wireless networks.

4.5.1 Eight Pair-wise Interference Patterns

We start from studying the conditions of a feasible rate allocation in MPT/MPR networks. When the transmitting or receiving capability limit is greater than 1 (*i.e.,* $K_T, K_R \geq 1$), more than one transmission of the same MC can occur concurrently. Thus, the capacity of the MC cannot be simply assumed as one unit and Eq. 4.2d is not valid to determine the feasibility of a rate allocation. Since a successful transmission means that it can be correctly transmitted and received by the corresponding transmitter and receiver of the arc, a rate allocation $\boldsymbol{\rho_f}$ being feasible implies that both the transmitter and receiver of any arc can accommodate the traffic load specified by $\boldsymbol{\rho_f}$. This traffic load includes the traffic of the arc itself (referred to as the ***desired traffic***) and the ***interfering traffic*** carried by the interfering arcs. Therefore, for $\boldsymbol{\rho_f}$ to be feasible, any arc (or more specifically, the transmitter and receiver of any arc) should be able to handle both the desired and the interfering traffic.

For a given arc l, the MC in which l parcitipates contains all the interfering arcs of l which are in pair–wise interference relationship themselves. To guarantee that arc l can accommodate the traffic load specified by $\boldsymbol{\rho_f}$, we need to examine every MC containing l because different MC contains different interfering arcs of l. Therefore,

to evaluate the feasibility of $\boldsymbol{\rho_f}$, we examine every MC ϕ in $\boldsymbol{\phi}$. Then for any ϕ, we compute the aggregate traffic of every arc $l \in \boldsymbol{l}_\phi$, including both the desired traffic carried by l and the interfering traffic carried by any other arcs of ϕ. If the aggregate traffic can be handled by both the transmitter and receiver of l, which can be any arc of any MC, the feasibility of $\boldsymbol{\rho_f}$ is satisfied.

Starting with a single MC ϕ (*i.e.,* $\boldsymbol{\phi} = \{\phi\}$) of size two (*i.e.,* $\boldsymbol{l}_\phi$=$\{l_{\phi,i}, l_{\phi,j}\}$), in which case one arc $l_{\phi,i}$ only has one interfering arc $l_{\phi,j}$ within ϕ. In the MC ϕ, the aggregate traffic of $l_{\phi,i}$ consists of the transmissions carried by $l_{\phi,i}$ (*i.e.,* desired traffic) and the interfering transmissions contributed by $l_{\phi,j}$ (*i.e.,* interfering traffic). The aggregate traffic is calculated at the transmitter (tx_i) and receiver (rx_i) of $l_{\phi,i}$ separately. Note that not every contributor to the aggregate traffic is using tx_i as its transmitter and rx_i as its receiver. For example, in Fig. 4.4 (b), tx_i is the receiver of $l_{\phi,j}$. The transmissions of $l_{\phi,i}$ always denotes tx_i as transmitter and rx_i as receiver; however, the transmissions of an interfering arc $l_{\phi,j}$ may or may not involve tx_i and/or rx_i. Thus, it is necessary to identify the combinations of how $l_{\phi,j}$ interferes with $l_{\phi,i}$.

We identify eight patterns (Fig. 4.4) and derive the equations (Eq. 4.19 and 4.20) to compute the normalized aggregate traffic of $l_{\phi,i}$ within ϕ, according to the interference pattern in which $l_{\phi,j}$ interferes with $l_{\phi,i}$. Fig. 4.4 shows the eight interference patterns from $l_{\phi,i}$'s viewpoint, *i.e.,* each case shows how $l_{\phi,j}$ interferes with $l_{\phi,i}$.

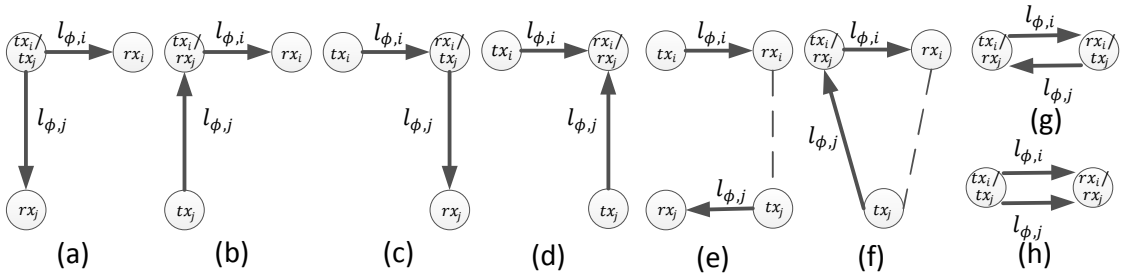


Fig. 4.4: The eight pair-wise interference patterns.

There are three ways in which $l_{\phi,j}$ can interfere with $l_{\phi,i}$:

(A) $l_{\phi,j}$ only interferes with $l_{\phi,i}$ at tx_i (Fig. 4.4(b))

Note that a transmitting node will not be affected by an overheard signal. Thus, $l_{\phi,j}$ will interfere with tx_i if 1) $tx_i = tx_j$ (*i.e.,* $l_{\phi,i}$ and $l_{\phi,j}$ share the same transmitter) or 2) $tx_i = rx_j$ (*i.e.,* $l_{\phi,j}$'s receiver is tx_i). If $tx_i = tx_j$, the signal sent by tx_j will be received by rx_i because rx_i is in the transmission range of tx_j (or tx_i). Thus, the only case of group A is when $l_{\phi,j}$'s receiver is the transmitter of $l_{\phi,i}$ and rx_i is not in the interference range of tx_j, which is depicted by Fig. 4.4(b).

(B) $l_{\phi,j}$ only interferes with $l_{\phi,i}$ at rx_i (Fig. 4.4(c), (d) and (e))
A receiver will be affected when it is in the interference or transmission range of a transmitter, which is captured by Fig. 4.4(d) and (e). Since a node cannot perform reception and transmission concurrently (half-duplex radio), rx_i will need to share the resource with tx_j if $rx_i = tx_j$. Thus, group B include Fig. 4.4(c), (d) and (e).

(C) $l_{\phi,j}$ interferes with $l_{\phi,i}$ at both tx_i and rx_i (Fig. 4.4(a), (f), (g) and (h))
Fig. 4.4(a), (f), (g) and (h) compose the cases of group C. Fig. 4.4(a) and (h) show the patterns when $tx_i = tx_j$ but $l_{\phi,i}$ and $l_{\phi,j}$ may or may not share the same receiver; Fig. 4.4(f) and (g) include the patterns when $tx_i = rx_j$ but $l_{\phi,j}$ may or may not apply rx_i as it transmitter.

Therefore, the eight patterns include all the cases that $l_{\phi,j}$ can interfere with $l_{\phi,i}$, which are described in the following list.

(a) $l_{\phi,j}$ interferes with $l_{\phi,i}$ at both the transmitter and the receiver of $l_{\phi,i}$ as $tx_i = tx_j$ and tx_j is in the interference range of rx_i;

(b) $l_{\phi,j}$ interferes with $l_{\phi,i}$ at the transmitter of $l_{\phi,i}$ as $tx_i = rx_j$;

(c) $l_{\phi,j}$ interferes with $l_{\phi,i}$ at the receiver of $l_{\phi,i}$ as $rx_i = tx_j$;

(d) $l_{\phi,j}$ interferes with $l_{\phi,i}$ at the receiver of $l_{\phi,i}$ as $rx_i = rx_j$;

(e) $l_{\phi,j}$ interferes with $l_{\phi,i}$ at the receiver of $l_{\phi,i}$ as tx_j is in the interference range of rx_i;

(f) $l_{\phi,j}$ interferes with $l_{\phi,i}$ at both the transmitter and the receiver of $l_{\phi,i}$ as $tx_i = rx_j$ and tx_j is in the interference range of rx_i;

(g) $l_{\phi,j}$ interferes with $l_{\phi,i}$ at both the transmitter and the receiver of $l_{\phi,i}$ as $tx_i = rx_j$ and $tx_j = rx_i$;

(h) $l_{\phi,j}$ interferes with $l_{\phi,i}$ at both the transmitter and the receiver of $l_{\phi,i}$ as $tx_i = tx_j$ and $rx_i = rx_j$.

Fig. 4.4 shows the non-commutative property of the interference relationships. While $l_{\phi,j}$ interferes with $l_{\phi,i}$ in one type, $l_{\phi,i}$ can interfere with $l_{\phi,j}$ in another type or may not interfere with $l_{\phi,j}$. For instance, in case (e), $l_{\phi,j}$ interferes with $l_{\phi,i}$ at rx_i but $l_{\phi,i}$ does not interfere with $l_{\phi,j}$ because tx_i is not in the interference range of

rx_j. However, if either $l_{\phi,j}$ interferes with $l_{\phi,i}$ or $l_{\phi,i}$ interferes with $l_{\phi,j}$ in any type of Fig. 4.4, we say that $l_{\phi,i}$ and $l_{\phi,j}$ are ***dependent***; otherwise, they are ***independent***. *In MPT/MPR wireless networks, if two arcs are dependent, they are connected in the contention graph.* Since the interactions between flows caused by overhearing are considered when constructing the pair–wise interference patterns, edges of the flow graph with no traffic (*i.e.,* undirected edges) will influence whether vertices of the contention graph interfere or not (*e.g.,* Fig. 4.4 (e) and (f)).

When $K_T = K_R = 1$, the capacity of one MC is 1 unit because at most one packet transmission can occur at a slot within one MC. In MPT/MPR channels, since nodes can transmit or receive multiple packets simultaneously, we can normalize each contribution of the aggregate traffic going through tx_i or rx_i to the range of 0 to 1 by dividing respectively by K_R or K_T. Specifically, if the node is the transmitter, the traffic load at the node should be normalized by K_T; if the node is the receiver or overhearing the transmission, the traffic load should be normalized by K_R. Let $C_{TX}(i,j)$ ($C_{RX}(i,j)$) be the normalizing coefficient for the traffic of $l_{\phi,j}$ at $tx_i(rx_i)$. For a rate allocation $\boldsymbol{\rho_f}$ to be feasible, it is necessary that Eq. 4.19 and Eq. 4.20 hold. $\rho_{l_{\phi,i}}$ is the rate of $l_{\phi,i}$. Note that the rate of an arc is equal to the rate of the flow crossing the arc (*i.e.,* $\rho_{l_{\phi,i}} = \rho_f$, $l_{\phi,i} \in \boldsymbol{l}_f$, $\rho_f \in \boldsymbol{\rho_f}$).

$$TX: \quad \rho_{l_{\phi,i}} \times C_{TX}(i,i) + \rho_{l_{\phi,j}} \times C_{TX}(i,j) \leq 1 \tag{4.19}$$

$$RX: \quad \rho_{l_{\phi,i}} \times C_{RX}(i,i) + \rho_{l_{\phi,j}} \times C_{RX}(i,j) \leq 1 \tag{4.20}$$

Eq. 4.19 and 4.20 express the normalized aggregate traffic crossing $l_{\phi,i}$ when $l_{\phi,j}$ is the only interfering arc in ϕ. Eq. 4.19 is the normalized aggregate traffic going through the transmitter of $l_{\phi,i}$ (*i.e.,* tx_i). $\rho_{l_{\phi,i}} \times C_{TX}(i,i)$ is the normalized traffic load of $l_{\phi,i}$ passing through tx_i, *i.e.,* the desired traffic; $\rho_{l_{\phi,j}} \times C_{TX}(i,j)$ is the normalized traffic load that $l_{\phi,j}$ contributes to $l_{\phi,i}$ at tx_i, *i.e.,* the interfering traffic. $C_{TX}(i,i)$ ($C_{TX}(i,j)$) is the normalizing coefficient of the traffic load of $l_{\phi,i}$ ($l_{\phi,j}$) crossing tx_i. If Eq. 4.19 holds, it implies that $l_{\phi,i}$ ($l_{\phi,j}$) can achieve the rate $\rho_{l_{\phi,i}}$ ($\rho_{l_{\phi,j}}$) without exceeding the MPT/MPR capacity of tx_i.

Similarly, Eq. 4.20 is the normalized aggregate traffic going through the receiver of $l_{\phi,i}$ (*i.e.,* rx_i). $\rho_{l_{\phi,i}} \times C_{RX}(i,i)$ is the normalized traffic load of $l_{\phi,i}$ passing through rx_i, *i.e.,* the desired traffic; $\rho_{l_{\phi,j}} \times C_{RX}(i,j)$ is the normalized traffic load that $l_{\phi,j}$ contributes to $l_{\phi,i}$ at rx_i, *i.e.,* the interfering traffic. $C_{RX}(i,i)$ ($C_{RX}(i,j)$) is the normalizing coefficient of the traffic load of $l_{\phi,i}$ ($l_{\phi,j}$) crossing rx_i. If Eq. 4.20 holds, it implies that $l_{\phi,i}$ ($l_{\phi,j}$) can achieve the rate $\rho_{l_{\phi,i}}$ ($\rho_{l_{\phi,j}}$) without exceeding the MPT/MPR

capacity of rx_i.

Table. 4.1: The normalizing coefficients when $i \neq j$.

	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)
$C_{TX}(i,j)$	$\frac{1}{K_T}$	$\frac{1}{K_R}$	0	0	0	$\frac{1}{K_R}$	$\frac{1}{K_R}$	$\frac{1}{K_T}$
$C_{RX}(i,j)$	$\frac{1}{K_R}$	0	$\frac{1}{K_T}$	$\frac{1}{K_R}$	$\frac{1}{K_R}$	$\frac{1}{K_R}$	$\frac{1}{K_T}$	$\frac{1}{K_R}$

Because in our notation convention, the transmissions of $l_{\phi,i}$ always have tx_i as transmitter and rx_i as receiver, $C_{TX}(i,i) = 1/K_T$ and $C_{RX}(i,i) = 1/K_R$ always hold. When $i \neq j$, the coefficients $C_{TX}(i,j)$ and $C_{RX}(i,j)$ used to normalize the aggregate traffic are decided by the interference pattern in which $l_{\phi,j}$ interferes with $l_{\phi,i}$, as summarized in Table. 4.1. Eq. 4.21–4.36 list the conditions for $\boldsymbol{\rho_f}$ to be feasible for every case of Fig. 4.4.

(a)

$$TX: \quad \rho_{l_{\phi,i}} \times \frac{1}{K_T} + \rho_{l_{\phi,j}} \times \frac{1}{K_T} \leq 1 \tag{4.21}$$

$$RX: \quad \rho_{l_{\phi,i}} \times \frac{1}{K_R} + \rho_{l_{\phi,j}} \times \frac{1}{K_R} \leq 1 \tag{4.22}$$

(b)

$$TX: \quad \rho_{l_{\phi,i}} \times \frac{1}{K_T} + \rho_{l_{\phi,j}} \times \frac{1}{K_R} \leq 1 \tag{4.23}$$

$$RX: \quad \rho_{l_{\phi,i}} \times \frac{1}{K_R} + \rho_{l_{\phi,j}} \times 0 \leq 1 \tag{4.24}$$

(c)

$$TX: \quad \rho_{l_{\phi,i}} \times \frac{1}{K_T} + \rho_{l_{\phi,j}} \times 0 \leq 1 \tag{4.25}$$

$$RX: \quad \rho_{l_{\phi,i}} \times \frac{1}{K_R} + \rho_{l_{\phi,j}} \times \frac{1}{K_T} \leq 1 \tag{4.26}$$

(d)

$$TX: \quad \rho_{l_{\phi,i}} \times \frac{1}{K_T} + \rho_{l_{\phi,j}} \times 0 \leq 1 \tag{4.27}$$

$$RX: \quad \rho_{l_{\phi,i}} \times \frac{1}{K_R} + \rho_{l_{\phi,j}} \times \frac{1}{K_R} \leq 1 \tag{4.28}$$

(e)

$$TX: \quad \rho_{l_{\phi,i}} \times \frac{1}{K_T} + \rho_{l_{\phi,j}} \times 0 \leq 1 \tag{4.29}$$

$$RX: \quad \rho_{l_{\phi,i}} \times \frac{1}{K_R} + \rho_{l_{\phi,j}} \times \frac{1}{K_R} \leq 1 \tag{4.30}$$

(f)

$$TX: \quad \rho_{l_{\phi,i}} \times \frac{1}{K_T} + \rho_{l_{\phi,j}} \times \frac{1}{K_R} \leq 1 \tag{4.31}$$

$$RX: \quad \rho_{l_{\phi,i}} \times \frac{1}{K_R} + \rho_{l_{\phi,j}} \times \frac{1}{K_R} \leq 1 \tag{4.32}$$

(g)

$$TX: \quad \rho_{l_{\phi,i}} \times \frac{1}{K_T} + \rho_{l_{\phi,j}} \times \frac{1}{K_R} \leq 1 \tag{4.33}$$

$$RX: \quad \rho_{l_{\phi,i}} \times \frac{1}{K_R} + \rho_{l_{\phi,j}} \times \frac{1}{K_T} \leq 1 \tag{4.34}$$

(h)

$$TX: \quad \rho_{l_{\phi,i}} \times \frac{1}{K_T} + \rho_{l_{\phi,j}} \times \frac{1}{K_T} \leq 1 \tag{4.35}$$

$$RX: \quad \rho_{l_{\phi,i}} \times \frac{1}{K_R} + \rho_{l_{\phi,j}} \times \frac{1}{K_R} \leq 1 \tag{4.36}$$

For example, if $l_{\phi,j}$ interferes with $l_{\phi,i}$ in type (a), for an allocation vector $\boldsymbol{\rho_f}$ to be fessible, it is necessary that Eq. 4.21 and Eq. 4.22 hold. Because $l_{\phi,i}$ and $l_{\phi,j}$ share the same transmitter (*i.e.,* $tx_i = tx_j$), at tx_i, both $\rho_{l_{\phi,i}}$ and $\rho_{l_{\phi,j}}$ should be normalized by K_T in Eq. 4.21. Because the transmissions of $l_{\phi,j}$ can be heard by rx_i, at rx_i, both $\rho_{l_{\phi,i}}$ and $\rho_{l_{\phi,j}}$ should be normalized by K_R in Eq. 4.22. In this case, $C_{TX}(i,j) = 1/K_T$, $C_{RX}(i,j) = 1/K_R$ as indicated in the second column of Table. 4.1.

Similarly, if $l_{\phi,i}$ and $l_{\phi,j}$ fit case (b), for $\boldsymbol{\rho_f}$ to be feasible, it is necessary that Eq. 4.23 and Eq. 4.24 hold. Because the transmitter of $l_{\phi,i}$ is the receiver of $l_{\phi,j}$, at tx_i, $\rho_{l_{\phi,i}}$ should be normalized by K_T and $\rho_{l_{\phi,j}}$ should be normalized by K_R in Eq. 4.23. Because tx_j is not in the interference range of rx_i, the receiver of $l_{\phi,i}$ will not be affected by the transmission of $l_{\phi,j}$. Thus, in Eq. 4.24, the coefficient of $\rho_{l_{\phi,j}}$ is 0. In this case, $C_{TX}(i,j) = 1/K_R$ and $C_{RX}(i,j) = 0$ as shown in the third column of Table. 4.1.

Similar reasoning can be applied to the remaining interference patterns (c) to (h).

As mentioned earlier, for a rate allocation $\boldsymbol{\rho_f}$ to be feasible, the arc of any MC should be able to accommodate the aggregate traffic within that MC. Eq. 4.37 and

Eq. 4.38 show that, for $\boldsymbol{\rho_f}$ to be feasible, the normalized aggregate traffic going through any end (*i.e.,* transmitter or receiver) of any arc of any MC should be less than one. As we will show in Section 5.3, the rate allocation computed restricted to these conditions closely approximates the max-min allocation. Unfortunately, these constraints are neither necessary nor sufficient for determining whether ϕ is saturated with respect to $\boldsymbol{\rho_f}$.

$$\sum_{j=1}^{|\boldsymbol{l}_\phi|} C_{TX}(i,j)\rho_{l_{\phi,j}} \leq 1,\ i = 1, ..., |\boldsymbol{l}_\phi| \tag{4.37}$$

$$\sum_{j=1}^{|\boldsymbol{l}_\phi|} C_{RX}(i,j)\rho_{l_{\phi,j}} \leq 1,\ i = 1, ..., |\boldsymbol{l}_\phi| \tag{4.38}$$

4.5.2 Challenges to Determining the Saturation Status of MC

The challenges are specific to the M property of the contention graph built in MPT/MPR wireless networks. We will revisit this property again in Section 5.1 when we present the scheduling algorithm for MPT/MPR channels. In this section, we discuss the obstacles to find the necessary and sufficient conditions to determine whether ϕ is saturated with respect to $\boldsymbol{\rho_f}$. More specifically, we will illustrate why the equality of Eq. 4.37 and 4.38 are neither necessary nor sufficient conditions for ϕ to be saturated.

Challenges to Determining the Necessary Conditions

Fig. 4.5 depicts an example of two flows. We will compute the max-min allocation using Eq. 4.37 and Eq. 4.38 and show that, for this example, these two constraints are over constraining.

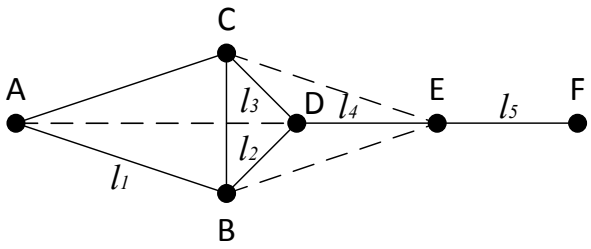


Fig. 4.5: An example of two flows. f_1: A->B->D. f_2: C->D->E->F.

In Fig. 4.5, f_1 traverses arc l_1 and l_2 and f_2 traverses arc l_3, l_4 and l_5. These five arcs compose one MC because they are pair-wise dependent. Assuming $K_T = 2$ and $K_R = 5$ and applying Eq. 4.37 and Eq. 4.38, we construct Eq. 4.39 to 4.43 for the

five arcs. For instance, Eq. 4.39a is the constraint of the transmitter of l_1 constructed according to Eq. 4.37 when $\boldsymbol{l}_\phi = \{l_1,\ l_2,\ l_3,\ l_4,\ l_5\}$ and $i = 1$. Similarly, Eq. 4.39b is the constraint of the receiver of l_1 created by Eq. 4.38.

$$TX(l_1): \ \frac{\rho_{l_1}}{K_T} \leq 1 \tag{4.39a}$$

$$RX(l_1): \ \frac{\rho_{l_1}}{K_R} + \frac{\rho_{l_2}}{K_T} + \frac{\rho_{l_3}}{K_R} + \frac{\rho_{l_4}}{K_R} + \frac{\rho_{l_5}}{K_R} \leq 1 \tag{4.39b}$$

$$TX(l_2): \ \frac{\rho_{l_1}}{K_R} + \frac{\rho_{l_2}}{K_T} \leq 1 \tag{4.40a}$$

$$RX(l_2): \ \frac{\rho_{l_1}}{K_R} + \frac{\rho_{l_2}}{K_R} + \frac{\rho_{l_3}}{K_R} + \frac{\rho_{l_4}}{K_T} + \frac{\rho_{l_5}}{K_R} \leq 1 \tag{4.40b}$$

$$TX(l_3): \ \frac{\rho_{l_3}}{K_T} \leq 1 \tag{4.41a}$$

$$RX(l_3): \ \frac{\rho_{l_1}}{K_R} + \frac{\rho_{l_2}}{K_R} + \frac{\rho_{l_3}}{K_R} + \frac{\rho_{l_4}}{K_T} + \frac{\rho_{l_5}}{K_R} \leq 1 \tag{4.41b}$$

$$TX(l_4): \ \frac{\rho_{l_2}}{K_T} + \frac{\rho_{l_3}}{K_R} + \frac{\rho_{l_4}}{K_R} \leq 1 \tag{4.42a}$$

$$RX(l_4): \ \frac{\rho_{l_2}}{K_R} + \frac{\rho_{l_3}}{K_R} + \frac{\rho_{l_4}}{K_R} + \frac{\rho_{l_5}}{K_T} \leq 1 \tag{4.42b}$$

$$TX(l_5): \ \frac{\rho_{l_5}}{K_T} + \frac{\rho_{l_4}}{K_R} \leq 1 \tag{4.43a}$$

$$RX(l_5): \ \frac{\rho_{l_5}}{K_R} + \frac{\rho_{l_4}}{K_R} \leq 1 \tag{4.43b}$$

Since the two flows are bottlenecked by the same MC, according to max-min fairness, the two flows should have the same rate. Because of the flow conservation property, all the arcs on the path of a flow should have the same rate. Thus, the rates of the five arcs should be the same. Solving Eq. 4.39 to 4.43, the max-min rate of these arcs is 10/13 packets/slot. Nonetheless, we will present a schedule (Table. 4.2) to show that the five arcs can achieve a higher rate satisfying the rules of max-min fairness.

Table. 4.2 shows a schedule of 11 slots for the five arcs. For example, at slot 1, l_1 and l_4 can each transmit two packets without causing any collision. Similarly, at slot 6, arc l_2, l_3 and l_5 can transmit 2, 1 and 2 packets, respectively. With this

Table. 4.2: A schedule when $K_T = 2$ $K_R = 5$.

Flow ID	1		2		
Arcs	l_1	l_2	l_3	l_4	l_5
Slot 1	2			2	
Slot 2	2			2	
Slot 3	2			2	
Slot 4	2			2	
Slot 5	2			2	
Slot 6		2	1		2
Slot 7		2	2		1
Slot 8		1	2		2
Slot 9		2	1		2
Slot 10		2	2		1
Slot 11		1	2		2

schedule, both f_1 and f_2 can achieve the rate of 10/11 packets/slot, which is higher than the rate (*i.e.,* 10/13) computed by Eq. 4.37 and Eq. 4.38. This disparity can be attributed to the fact that the constraints depicted by Eq. 4.37 and Eq. 4.38 treat the interfering transmissions the same as the desired transmissions as if both need to be decoded by the receiver. However, the interfering transmissions are different from the desired transmissions in that the overheard signal will not affect a transmitting node.

For example, in Eq. 4.40b, the five arcs are normalized according to their interference patterns at the receiver of l_2 (*i.e.,* node D). ρ_{l_4} is normalized by K_T because node D is the transmitter of l_4 and all the other arcs' rates are normalized by K_R because their transmissions are either received or overheard by node D. Each term of Eq. 4.40b represents the time fraction (between 0 and 1) the arc needs at node D and these time fractions cannot overlap with each other (Note that MPT/MPR has been captured by the normalizing coefficients). However, Eq. 4.40b is over constraining because at node D, the overheard transmission from l_1 will not interfere with the transmission on l_4. As shown in Table. 4.2, l_4 can transmit two packets (which fully occupies all the resource of node D because $2/K_T = 1$) while overhearing the transmissions on l_1.

Intuitively, for any transmitter n, if the overheard transmissions can be scheduled within the time interval when node n is assigned to transmit, the overheard transmissions should be ignored by node n; otherwise, the portion of the overheard transmission which cannot be scheduled within the time interval (when node n is designated to transmit) should be considered as the interfering transmissions at node

n because the node will decode the received signal when it is not transmitting. However, for a linear programming problem, which portion of the interfering transmission can be omitted and which portion should be taken into consideration is unknown beforehand. Thus, in Eq. 4.37 and Eq. 4.38, we cautiously assume that any overheard transmission should be treated as desired transmission and counted in the constraint to determine whether a MC is saturated or not.

Nonetheless, for conventional channel (*i.e.,* $K_T = K_R = 1$), the fact that two arcs belong to the same MC means that these two arcs cannot be active simultaneously. In the example of Fig. 4.5, it means that l_1 and l_4 cannot be active concurrently, which is precisely captured by Eq. 4.37 and Eq. 4.38.

Challenges to Determining the Sufficient Conditions

Eq. 4.37-4.38 are also not sufficient conditions for ϕ to be saturated. For example, when we normalize a traffic load of an arc by K_R, we assume that the transmitter of the arc can transmit K_R packets at a slot (*i.e.,* $K_T \geq K_R$). This assumption does not hold if $K_T < K_R$. Therefore, in this case the constraints are not sufficient and the actual arc rate may be determined by $\min(K_T, K_R)$. However, we can not normalize the traffic by $\min(K_T, K_R)$ neither, because even when $K_T < K_R$, it is possible the aggregate traffic at a receiver should be normalized by K_R owing to the fact that transmissions could be originating from different transmitters. Even if one arc cannot transmit K_R packets per slot, the aggregate traffic from several transmitters can benefit from the higher reception capacity of the receiver. If we normalized the aggregate traffic by $\min(K_T, K_R)$, we would rule out the potential that an MPR receiver could handle multiple separate transmissions simultaneously, including those interfering flows.

(a) Example 1: $\frac{\rho_{l_1}}{K_T} + \frac{\rho_{l_2}}{K_R} \leq 1$ and $\frac{\rho_{l_1}}{K_R} + \frac{\rho_{l_2}}{K_T} \leq 1$

(b) Example 2: $\frac{\rho_{l_1}}{K_T} \leq 1$, $\frac{\rho_{l_1}}{K_R} + \frac{\rho_{l_2}}{K_R} \leq 1$ and $\frac{\rho_{l_2}}{K_T} \leq 1$

Fig. 4.6: Two examples of two single-hop flows.

We use two examples in Fig. 4.6 to illustrate the dilemma. Assume $K_T = 2$ and $K_R = 3$. In both examples, there is only one MC composed by l_1 and l_2. The conditions derived by Fig. 4.37 and Fig. 4.38 are described in the caption of each example. Solving the constrains for Example 1, the rate allocation is $\{1.2, 1.2\}$.

However, this allocation cannot be realized. Because node B is either transmitting or receiving at a slot and the maximum rate of any arc is 2 packets/slot ($K_T = 2$), for every n slots, at most $2n$ packets can be delivered either by l_1 or l_2. Thus, the maximum aggregate rate of the two arcs is at most 2 packets/slot, which is smaller than the sum of the rate allocation. The reason that the allocation cannot be realized is caused by the conditions. For example, the constraint $\frac{\rho_{l_1}}{K_R} + \frac{\rho_{l_2}}{K_T} \leq 1$ normalizes ρ_{l_1} by K_R assuming the transmitter of l_1 can send K_R packets at a slot. However, this is not true because $K_T < K_R$ in this example.

Additionally, we cannot normalize the rate by $\min(K_T, K_R)$. A counterexample is given by Example 2. If we normalize the arc rate by $\min(K_T, K_R)$, the condition $\frac{\rho_{l_1}}{K_R} + \frac{\rho_{l_2}}{K_R} \leq 1$ of Example 2 should be replaced by $\frac{\rho_{l_1}}{K_T} + \frac{\rho_{l_2}}{K_T} \leq 1$, because $K_T < K_R$. Solving the new set of conditions, the max-min allocation is $\{1, 1\}$. However, this is not true because both flows can achieve the rate 1.5 packets/slot. For example, at the first slot, let l_1 transmit two packets and l_2 transmit one packets. At the second slot, let l_2 transmit two packets and l_1 transmit one packet. Normalizing by $\min(K_T, K_R)$ incorrectly excludes the possibility that node B can handle the transmissions from both node A and C simultaneously.

Apparently, this problem does not exist if $K_T = K_R = 1$ because in this case $\min\{K_T, K_R\} = K_R = K_T$.

4.5.3 The WF-ASYM Algorithm for $K_T \neq K_R$

```
WF-ASYM  (φ, f, K_T, K_R, Δ)
1    f̃ = f;
2    ρ_f is set to NULL;
3    while f̃ is not empty
4        for f ∈ f̃
5            ρ_f = ρ_f + Δ;
6            if for any φ ∈ φ Eq. 4.37 or 4.38 does not hold
7                flow f is bottlenecked;
8                ρ_f = ρ_f − Δ;
9            end
10       end
11       f̃ = {f | f is not bottlenecked};
12   end;
13   return ρ_f;
```

Fig. 4.7: The WF-ASYM algorithm.

Despite the challenges to compute a necessary and sufficient saturation condition, Eq. 4.37 or 4.38 can be applied as an approximated feasibility constraint in

MPT/MPR networks. We design the heuristic method WF-ASYM (Fig. 4.7) to approximate the max-min rate allocation in MPT/MPR wireless networks when K_T is not necessarily equal to K_R. Assume a set of multi-hop flows $\boldsymbol{f}$ with unlimited demands traversing a MPT/MPR wireless network and the set of MCs is denoted by $\boldsymbol{\phi}$. WF-ASYM applies the basic water-filling method and repeatedly increases the flows' rates by a small amount, denoted by Δ. If the increment of a flow rate (line 5) results in the violation of Eq. 4.37 or 4.38, it means that the current rate allocation (after line 5 is executed) is not feasible. Thus, the increment of the flow rate is reversed by line 8 and this flow is identified as bottlenecked. WF-ASYM continues to increase the rate of the non-bottlenecked flows until all the flows are bottlenecked.

4.6 Summary

In this chapter, we study the MPT/MPR multi-hop wireless networks from a centralized viewpoint. With the objective to evaluate the potential of a MPT/MPR network, we model the system via flow and contention graphs. The M property of the contention graph built for MPT/MPR networks is that even the interfering arcs can transmit simultaneously. This property fundamentally alters the rules to control the arcs and leads to the fact that the traditional approaches to measure and manage a wireless network are not applicable to the MPT/MPR wireless networks.

We extend the max-min fairness and its water-filling algorithm to the MPT/MPR wireless networks. Based on the analogy between a link of a wired network and a MC of a wireless network, WF is designed to compute the exact max-min allocation in a conventional channel ($K_T = K_R = 1$). In order to determine the max-min allocation for advanced channels, the pair-wise interference relationships are categorized into eight patterns. With the classification of the interactions between arcs, the constraints to regulate the traffic of each arc are constructed by taking into account the MPT/MPR capacity limits and the half-duplex radio property. Based on the constraints built upon the eight pair-wise interference patterns, WF-ASYM is designed to approximate the max-min allocation in asymmetric networks ($K_T \neq K_R$). Unfortunately, these constraints are not necessary and sufficient to determine the saturation status of a MC. The challenges to find the necessary and sufficient condition via linear expressions are demonstrated via examples.

We will design scheduling algorithms to realize a given rate allocation in the next chapter where the performance of WF and WF-ASYM will be evaluated and compared with the scheduling algorithms.

Chapter 5

MPT/MPR Schedule Constructions

In the previous chapter, we discussed how to compute the max-min allocation in MPT/MPR networks. The next question is: how to realize such an allocation computed by WF (or WF-ASYM) or, more generally, how to realize any given allocation in a slotted and synchronous system? To the author's best knowledge, the schedule construction problem in a multi-hop MPT/MPR wireless network has not been thoroughly discussed. In this chapter, we study how to convert a given rate allocation, which specifies the rate of every flow, into a concrete schedule in multi-hop MPT/MPR wireless network.

The MPT/MPR channel model restrains the transmissions by the MPT and MPR capacity limits. In a more practical wireless environment, the traffic load of a given arc is not only decided by K_T of the transmitter and K_R of the receiver but also affected by the channel quality of the arc. Under a poor channel condition, the receiver of the arc may not be able to handle K_R packets at the same time. To incorporate the non-homogeneous characteristics of a wireless network, we define the ***effective reception capacity limit*** $\hat{K}_R(l)$ to indicate the channel quality of arc l. The value of $\hat{K}_R(l)$ should be a non-negative integer which is not greater than K_R. For a perfect channel, $\hat{K}_R(l) = K_R$. $\hat{K}_R(l) = 0$ implies that no packet carried by arc l can be successfully decoded by the receiver of l.

In this chapter, a greedy scheduling algorithm MDSATUR is presented in Section 5.1 which computes a discrete schedule to implement a given allocation. Section 5.2 demonstrates the LEX scheme to evaluate an asymmetric MPT/MPR network based on the lexicographical optimality. The numerical results in Section 5.3 illustrate the performance of WF, WF-ASYM and LEX and explain how the value of K_T and K_R affect the end-to-end throughput. Discrete schedules are constructed applying MDSATUR and LEX in both homogeneous and non-homogeneous networks. Section 5.4 is the summary.

5.1 The MDSATUR algorithm

A schedule designates which arcs should be active at which slot and how many packets it should accommodate without causing any conflict with the goal to fully utilize the channel resource. The length of this schedule is named the ***scheduling cycle*** and the schedule will be repeated for every scheduling cycle.

5.1.1 Extended Contention Graph

Traditionally, the arc scheduling problem can be interpreted as the vertex-coloring problem of the contention graph. Since an arc may need to transmit more than one packet for every scheduling cycle, in both conventional and advanced channels, it is necessary to extend the basic contention graph to reflect the fact that each arc may need to transmit for multiple slots, or equivalently, each vertex of the contention graph may need to have more than one color. For this purpose, we introduce the ***extended contention graph (ECG)*** which extends the contention graph according to a given allocation.

Fig. 5.1: An example of the extended contention graph.

Fig. 5.1 shows an example of the ECG in a conventional channel model ($K_T = K_R = 1$). This example uses the topology graph of Fig. 4.1 (a). There are two flows f_1 and f_3 crossing the network as shown in Fig. 4.1(b). The contention graph built based on the topology and flow information is shown in Fig. 5.1 (a) where every arc is represented by exactly one vertex. Assuming the max-min fair rate allocation is $\{1, 1/2\}$ packets/slot, one way to implement this allocation is, for every two slots (*i.e.,* the length of the scheduling cycle is two) let arc l_1 be active for two slots and arc l_3 and l_4 each be active for one slot (because l_3 and l_4 are dependent and they cannot be active concurrently in a conventional channel model). In order to reflect the fact that l_1 transmits for two slots for every scheduling cycle, the vertex of l_1 in Fig. 5.1 (a) is replaced by a clique of size two in Fig. 5.1 (b). We use a clique to capture the fact that an arc is dependent with itself. By constructing the ECG, the process of scheduling is conceptually simplified: designate every vertex one slot to transmit without causing conflicts.

5.1.2 MDSATUR

The DSATUR algorithm is a well-known heuristic proposed by Brelaz [15] to approximate the vertex-coloring problem in which colors are represented by integers starting from 1. Given a graph, the number of different colors which a vertex is adjacent to is called the ***saturation degree*** of the vertex. The approach of the DSATUR algorithm is to repeatedly select the vertex with the largest saturation degree and assign it with the least possible (lowest-numbered) color. DSATUR can be applied to the ECG to construct a discrete schedule for a conventional channel. However, for more general channel models, a novel approach is needed for the scheduling problem due to *the M property* (Section 2.2.2) of the contention graph in MPT/MPR channels. We encountered the M property in Section 4.5.2 when elaborating the challenges to find the necessary and sufficient saturation conditions. Here, we explain its impact from the viewpoint of a scheduling algorithm.

For example, in Fig. 4.4 (a), arc $l_{\phi,i}$ and $l_{\phi,j}$ are interfering with each other because they share the same transmitter. However, if $K_T = K_R = 2$, $l_{\phi,i}$ and $l_{\phi,j}$ can transmit concurrently, which implies that in the corresponding ECG the connected vertices can have the same color. This example shows that *the fundamental principle of DSATUR (i.e., connect vertices cannot have the same color) does not apply in the vertex-coloring problem if the ECG is built from a MPT/MPR channel model.*

For MPT/MPR channel models, when choosing a color for a vertex v, the scheduling algorithm *cannot* simply select color based on the interference relationship contained in the ECG and exclude all the colors that have been given to a vertex connected to v. Two vertices connected in the ECG means that the two corresponding arcs share a certain resource. Whether these two (or more) arcs can transmit concurrently in one slot depends on if the shared resource is adequate to accommodate all the arcs. Therefore, the scheduling algorithm needs the knowledge of both the ECG and the flow graph (*e.g.,* Fig. 4.1 (b)). The ECG shows which arcs are sharing a resource while the flow graph specifies which resource is being shared by which arcs. If the arcs are sharing the transmitter, the transmission capacity limit K_T should be examined; otherwise, the reception capacity limit K_R is the one to be considered.

The greedy scheduling algorithm MDSATUR (Fig. 5.2) extends the DSATUR algorithm to non-homogeneous wireless networks under MPT/MPR channels. Since we assume a time-slotted network, an arc can only accommodate an integer number of packets and each element of $\boldsymbol{\rho_f}$ must be converted to an integer. For example, to schedule the max-min allocation of WF (Fig. 4.3), $\boldsymbol{\rho_f}$ should be replaced by $\tilde{\boldsymbol{\rho}}_{\boldsymbol{f}} = \boldsymbol{\rho_f} \times N_{min}$ where N_{min} is the minimal scheduling cycle as discussed in Section

4.4.5.

The parameters of MDSATUR include K_T, K_R, the effective reception capacity limits for all the arcs of $\boldsymbol{l}$, denoted by $\hat{K}_R(\boldsymbol{l})$ and the integer allocation $\tilde{\boldsymbol{\rho}}_{\boldsymbol{f}}$. MDSATUR returns the schedule $\boldsymbol{S}$ and the length of the scheduling cycle N_{sch}. By repeating the schedule $\boldsymbol{S}$ for very N_{sch} slots, flow $f \in \boldsymbol{f}$ can achieve the rate of $\tilde{\rho}_f/N_{sch}$ packets/slot.

```
MDSATUR (K_T, K_R, K̂_R(l), ρ̃_f)
1    Build the ECG based on ρ̃_f;
2    Assign a vertex of maximal degree to slot 1. Break ties arbitrarily;
3    while not every vertex is assigned a slot
4      Choose the vertex l_next with maximum saturation degree. Break ties arbitrarily.
5      Set slot index t to 0;
6      while (true)
7        t = t + 1;
8        if (l_act(t) == ∅)
9          break;
10       end
11       if (C+_rx(t) ≤ K_R and C+_tx(t) ≤ K_T)
12         if (tx(l_next) ≠ rx(l_act(t)) and rx(l_next) ≠ tx(l_act(t)) ∀ l_act ∈ l_act(t))
13           break;
14         end
15       end
16     end
17     Schedule l_next to slot t;
18   end
19   N_sch = t
20   return N_sch and the schedule S;
```

Fig. 5.2: The MDSATUR algorithm.

The first three steps (line 1 to 4) follow DSATUR to choose the next vertex (*i.e.,* l_{next}) to be assigned a slot. Because l_{next} can be scheduled to any slot (including possibly the slots occupied by other arcs), the loop from line 6 to 16 examines all the slots starting from slot $t = 1$. l_{next} can be active at slot t if its transmission does not conflict with any of the arcs, denoted by $\boldsymbol{l}_{act}(t)$, which have already been assigned to slot t. Each active arc is denoted by $l_{act}(t) \in \boldsymbol{l}_{act}(t)$. Let $rx(l)$ and $tx(l)$ be the transmitter and receiver of arc l, respectively. $\boldsymbol{hx}(l)$ is the set of nodes which are in interference range of $tx(l)$. Note that $tx(l) \notin \boldsymbol{hx}(l)$ and $rx(l) \in \boldsymbol{hx}(l)$. The nodes which are adjacent to l_{next} or any active arc of slot t are represented by $\boldsymbol{n}_{act}(t) = \{n | n \in \{tx(l), rx(l)\} \ \forall \ l \in \{l_{next}, \ \boldsymbol{l}_{act}(t)\}\}$. $\boldsymbol{n}_{act}(t)$ contains all the active nodes of slot t and the potentially active nodes, *i.e.,* the transmitter and receiver of l_{next}. Specifically, l_{next} can be assigned to slot t if one of the following two conditions is satisfied.

1. There is no active arc at slot t, *i.e.,* $\boldsymbol{l}_{act}(t) = \emptyset$, which is indicated by line 8. If the condition holds, the loop breaks at line 9 and l_{next} is assigned to slot t.

2. Otherwise, $\boldsymbol{l}_{act}(t) \neq \emptyset$. In this case, l_{next} is assigned to slot t if the following constraints are satisfied.

   - *The transmission/reception capacity constraint*:
     In a non-homogeneous MPT/MPR wireless network, the activations of the arcs are restricted by the nodes' transmission and effective reception capacity limits. In order to prevent conflict, it is necessary to examine the aggregate inbound and outbound traffic of any active node (*i.e.,* $\forall\ n \in \boldsymbol{n}_{act}(t)$). Let $\mathcal{C}_{tx}(n,t)$ (Eq. 5.1) and $\mathcal{C}_{rx}(n,t)$ (Eq. 5.2) be the aggregate outbound and inbound traffic of node n at slot t, respectively. $\mathbb{1}_{\{l_{next},\ \boldsymbol{l}_{act}(t)\}}(l)$ is an indicator function. It is equal to one if $l \in \{l_{next},\ \boldsymbol{l}_{act}(t)\}$; otherwise it is equal to zero. In Eq. 5.1, $\mathcal{C}_{tx}(n,t)$ is the sum of all the outbound transmissions from an active node n at slot t. In Eq. 5.2, $\mathcal{C}_{rx}(n,t)$ is the sum of all the inbound (either received or overheard) transmissions at node n in slot t. When computing the aggregate inbound traffic, the load of arc l is scaled by the factor $\frac{K_R}{\hat{K}_R(l)}$ if the channel is poor and $\hat{K}_R(l) < K_R$. Intuitively, the reciprocal of the factor, *i.e.,* $\frac{\hat{K}_R(l)}{K_R}$, can be considered as a measure of the channel quality of arc l, which is equal to one if the channel is perfect. The load of arc l is normalized (or divided) by the channel quality index when computing the aggregate inbound traffic. Note that $\hat{K}_R(l) = 0$ means arc l cannot carry any transmission, or equivalently, arc l cannot be active.

     $\mathcal{C}^+_{tx}(t)$ and $\mathcal{C}^+_{rx}(t)$ represent the maximum outbound and inbound load over all the active nodes at slot t, respectively. If $\mathcal{C}^+_{tx}(t) \leq K_T$ and $\mathcal{C}^+_{rx}(t) \leq K_R$ (line 11 of Fig. 5.2), the transmission/reception capacity constraints are satisfied. Next, the half-duplex radio constraint should be examined.

$$\mathcal{C}_{tx}(n,t) = \sum_{l \in \{l | l \in \boldsymbol{l} \wedge tx(l) = n\}} \mathbb{1}_{\{l_{next},\ \boldsymbol{l}_{act}(t)\}}(l), \quad n \in \boldsymbol{n}_{act}(t) \tag{5.1}$$

$$\mathcal{C}_{rx}(n,t) = \sum_{l \in \{l | l \in \boldsymbol{l} \wedge n \in \boldsymbol{hx}(l)\}} \frac{K_R}{\hat{K}_R(l)} \mathbb{1}_{\{l_{next},\ \boldsymbol{l}_{act}(t)\}}(l), \quad n \in \boldsymbol{n}_{act}(t) \tag{5.2}$$

$$\mathcal{C}^+_{tx}(t) = \max_{n \in \boldsymbol{n}_{act}(t)} \mathcal{C}_{tx}(n,t) \tag{5.3}$$

$$\mathcal{C}^+_{rx}(t) = \max_{n \in \boldsymbol{n}_{act}(t)} \mathcal{C}_{rx}(n,t) \tag{5.4}$$

   - *The half-duplex radio constraint*:
     Since we assume a half-duplex radio is equipped on every node, to have l_{next}

become active at slot t, the transmitter (receiver) of l_{next} cannot be the receiver (transmitter) of any active arc $l_{act}(t)$ as show in line 12. If it is true, the transmission of l_{next} will not conflict with any arc which has already been assigned to slot t. Therefore, l_{next} can be assigned to slot t as well.

MDSATUR repeats until every vertex (or arc) is assigned to a slot. Upon the termination of MDSATUR, a feasible schedule, denoted by $\boldsymbol{S}$ is found and the value of t is equal to the length of the scheduling cycle. If the size of the ECG is $|V|$, it is known that the time complexity of DSATUR is $O(|V|^2)$ [15]. For every step of DSATUR, it determines the vertex with the maximal saturation degree. This vertex is then assigned to the least indexed unused color, which can be done in one operation. However, in MDSATUR, we need to exam all the used slots which can be shared by the considered vertex (or arc) in MPT/MPR networks.

Since the number of used slots is in the order of $|V|$, the time complexity of MDSATUR is $O(|V|^3)$. If the flow set is $\boldsymbol{f}$ and the node set is $\boldsymbol{n}$, the size of the contention graph is on the order of $|\boldsymbol{f}||\boldsymbol{n}|^2$. Thus, the time complexity of MDSATUR is $O((|\boldsymbol{f}||\boldsymbol{n}|^2)^3)$.

5.2 The LEX Scheme

According to Lemma 4.1, based on the solution of WF, a feasible schedule can be constructed for a symmetric network with $K_T = K_R = K$ if the contention graph is a perfect graph. However, for a more general system, *e.g.,* when $K_T \neq K_R$ or in cases where the contention graph is not a perfect graph, it is also necessary to compute a feasible schedule and the aggregate throughput. To this end, we propose the LEX (Fig. 5.3) scheme which can approximate the max-min schedule when $K_T \neq K_R$. The computation of the aggregate throughput can be trivially determined from the constructed schedule.

The LEX scheme is designed based on the ***lexicographical optimality***, which is formally described as follows.

Definition 5.1. ***Lexicographical Optimality*** *[82]: Given two feasible M-dimensional rate allocations $\boldsymbol{\rho}1$ and $\boldsymbol{\rho}2$ both arranged in increasing order. $\boldsymbol{\rho}1$ is lexicographically greater ($>_{lex}$) than $\boldsymbol{\rho}2$ if there exists $i \in [1,\ M]$ such that $\rho 1_i > \rho 2_i$ and $\rho 1_j = \rho 2_j$ for all $j < i$. If $\boldsymbol{\rho}1 >_{lex} \boldsymbol{\rho}2$, $\boldsymbol{\rho}2$ is lexicographically less than $\boldsymbol{\rho}2$, i.e.,$\boldsymbol{\rho}2 <_{lex} \boldsymbol{\rho}1$. A feasible allocation $\boldsymbol{\rho}$ is lexicographically optimal if every other feasible allocation is not lexicographically greater than $\boldsymbol{\rho}$.*

The max-min fairness and the lexicographical optimality have been use interchangeably in many references [16,21]. It is proved by Sarkar *et al.* [82] that a feasible max-min allocation is lexicographically optimal but the lexicographical optimality is not a sufficient condition for max-min fairness. Based on this conclusion, the LEX scheme is designed to find an approximate max-min schedule when $K_T \neq K_R$, using the WF and MDSATUR algorithms presented earlier.

```
LEX  (φ, f, l, K_T, K_R, K̂_R(l), Δ)
1    K_min = min(K_T, K_R);
2    [ρ_f, N_min] = WF(φ, f, l);
3    ρ_f = K_min × ρ_f;
4    ρ̃_f = ρ_f N_min;
5    S^+ = ∅;
6    ρ_f^+ = ρ_f;
7    N_sch^+ = 0;
8    N_lim = N_min + Δ;
9    for (any integer allocation ρ̃'_f satisfying ρ̃'_f >_lex ρ̃_f and can be realized in N_lim slots)
10     [S, N_sch]=MDSATUR(K_T, K_R, K̂_R(l), ρ̃'_f);
11     if (ρ̃'_f / N_sch >_lex ρ_f^+)
12       ρ_f^+ = ρ̃'_f / N_sch;
13       S^+ = S;
14       N_sch^+ = N_sch;
15     end
16   end
17   return ρ_f^+, S^+ and N_sch^+;
```

Fig. 5.3: The LEX scheme.

As shown in Fig. 5.3, the parameters of LEX include the set of MC $\boldsymbol{\phi}$, the set of flows $\boldsymbol{f}$, the set of arcs $\boldsymbol{l}$, K_T, K_R, $\hat{K}_R(\boldsymbol{l})$ and a non-negative integer Δ. LEX returns $\boldsymbol{\rho}_{\boldsymbol{f}}^+$, $\boldsymbol{S}^+$ and N_{sch}^+. The approximate max-min allocation is $\boldsymbol{\rho}_{\boldsymbol{f}}^+$ which can be realized by infinitely repeating the schedule $\boldsymbol{S}^+$ for every N_{sch}^+ slots. The aggregate throughput of the schedule can be computed by $\sum_{f \in \boldsymbol{f}} \rho_f^+$, which will be used to evaluate the system performance of MPT/MPR networks when $K_T \neq K_R$, as we do in the next section.

LEX consists of two steps:

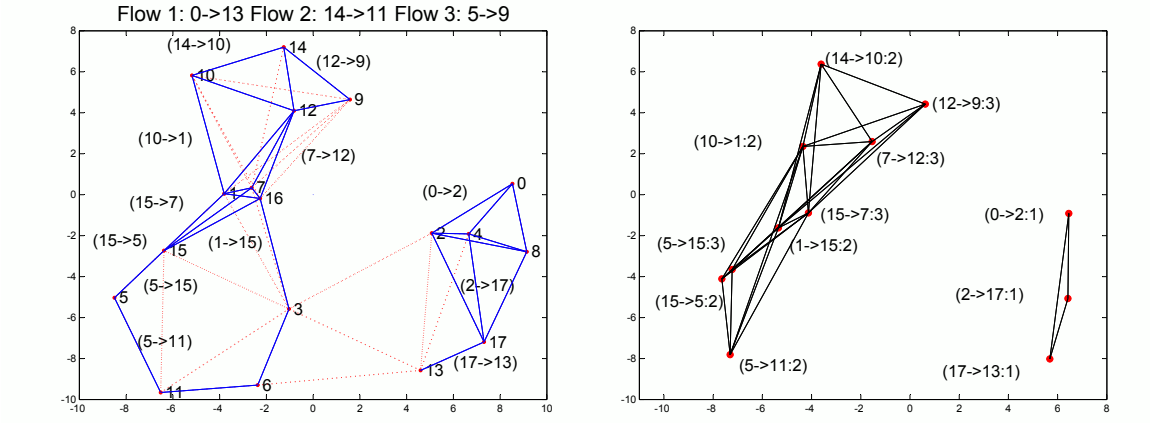
(i) *Find a lower feasible bound of the exact max-min allocation when* $K_T \neq K_R$: The lower feasible bound means an allocation which is lexicographically less than the exact max-min allocation. To find the lower bound, LEX applies WF to compute the rate allocation $\boldsymbol{\rho}_{\boldsymbol{f}}$ which is then scaled by $K_{min} = \min\{K_T, K_R\}$. Because MDSATUR only accepts integer allocations, we multiply $\boldsymbol{\rho}_{\boldsymbol{f}}$ by N_{min}

and denote the integer allocation by $\tilde{\boldsymbol{\rho}}_{\boldsymbol{f}}$ at line 4. The allocation $\boldsymbol{\rho}_{\boldsymbol{f}}$ serves as the lower feasible bound of the exact max-min allocation.

(ii) *Search the lexicographically greater allocations within certain range*:
At line 9, LEX searches all the integer allocations $\tilde{\boldsymbol{\rho}}'_{\boldsymbol{f}}$ which may be lexicographically greater than $\tilde{\boldsymbol{\rho}}_{\boldsymbol{f}}$. To this end, it applies MDSATUR to schedule $\tilde{\boldsymbol{\rho}}'_{\boldsymbol{f}}$ and returns a discrete feasible schedule $\boldsymbol{S}$ and the scheduling cycle N_{sch} (line 10). If the new rate allocation (*i.e.*, $\frac{\tilde{\boldsymbol{\rho}}'_{\boldsymbol{f}}}{N_{sch}}$) is lexicographically greater than the current lexicographically optimal allocation (denoted by $\boldsymbol{\rho}^{+}_{\boldsymbol{f}}$), *i.e.*, $\frac{\tilde{\boldsymbol{\rho}}'_{\boldsymbol{f}}}{N_{sch}} >_{lex} \boldsymbol{\rho}^{+}_{\boldsymbol{f}}$, $\boldsymbol{\rho}^{+}_{\boldsymbol{f}}$ is replaced by $\frac{\tilde{\boldsymbol{\rho}}'_{\boldsymbol{f}}}{N_{sch}}$ at line 12 as the new candidate for the approximate max-min rate allocation.

However, there are an unlimited number of potential integer allocations $\tilde{\boldsymbol{\rho}}'_{\boldsymbol{f}}$. In order to control the running time, we set an integer Δ to limit the search range. Specifically, LEX only searches among feasible allocations whose schedule construct can be realized in $N_{min} + \Delta$ (denoted by N_{lim} at line 8) slots. Since LEX searches all the integer allocations within Δ more slots, each flow can transmit at most $K_R\Delta$ more packets. Hence the number of flow allocation combinations within Δ more slots is at most $(K_R\Delta)^{|\boldsymbol{f}|}$, which determines the worst case complexity of LEX. By increasing the value of Δ, we can enlarge the search range and hence the precision of the result.

5.3 Numerical Examples



(a) Flow 1 ($0 \Rightarrow 13$: $0 \to 2 \to 17 \to 13$); flo.. 2 ($14 \Rightarrow 11$: $14 \to 10 \to 1 \to 15 \to 5 \to 11$); flow 3 ($5 \Rightarrow 9$: $5 \to 15 \to 7 \to 12 \to 9$).

(b) The contention graph.

Fig. 5.4: The topology graph and the contention graph.

In the following we make a number of observations on the impact of MPT/MPR illustrated by means of a sample random topology depicted in Fig. 5.4(a), consisting of 18 uniformly randomly placed nodes. Solid edges between nodes represent the ability to communicate (connectivity) while faint (red) lines indicate that two nodes are within the interference range of each other but not close enough to ensure connectivity. Three unidirectional flows cross the network with the paths of which are indicated in the figure captions. Each edge of the topology graph can correspond to multiple vertices in the contention graph (Fig. 5.4(b)), one for each flow traversing it (which indirectly also defines the direction of traversal).

The vertices of the contention graph are the new labels (representing arcs with traffic) further extended to index each flow traversing the particular arc. Notice, for example, that the vertex labeled as $(14 \rightarrow 10 : 2)$ in Fig. 5.4(b) to indicate the flow 2 traversing it. In Fig. 5.4(a), link $(5-15)$ is traversed by flow 2 and flow 3 in opposite directions, resulting in two vertices $(15 \rightarrow 5 : 2)$ and $(5 \rightarrow 15 : 3)$ in the contention graph Fig. 5.4(b).

The maximal cliques of example are shown in Table. 5.1. The columns represent the arcs traversed by each flow for all flows (1 to 3) and the rows indicate in which maximal cliques (MC 1 to MC 4) each arc participates. For example, *e.g.,* flow 1 consists of arc $0 \rightarrow 2$, $2 \rightarrow 17$ and $17 \rightarrow 13$. The four MCs are displayed in the next four rows. From the row checkmarks one can identify the cardinality of each MC, *e.g.,* MC 2 has a size of six (corresponding to arcs $10 \rightarrow 1$, $1 \rightarrow 15$, $15 \rightarrow 5$, $5 \rightarrow 11$, $5 \rightarrow 15$ and $15 \rightarrow 17$).

Table. 5.1: The maximal cliques.

Flow ID	1			2					3			
Arcs	$0 \rightarrow 2$	$2 \rightarrow 17$	$17 \rightarrow 13$	$14 \rightarrow 10$	$10 \rightarrow 1$	$1 \rightarrow 15$	$15 \rightarrow 5$	$5 \rightarrow 11$	$5 \rightarrow 15$	$15 \rightarrow 7$	$7 \rightarrow 12$	$12 \rightarrow 9$
MC 1	✓	✓	✓									
MC 2					✓	✓	✓	✓	✓	✓		
MC 3				✓	✓	✓				✓	✓	✓
MC 4					✓	✓			✓	✓	✓	

5.3.1 The relative impact of K_T and K_R

Since the network can have different transmission and reception capacity limits, it is interesting to know whether they are equally significant in improving the overall performance. More specifically, the question is how the aggregate throughput scales with the value of K_T or K_R. In this section, we use the algorithms proposed earlier (*i.e.,* WF (Fig. 4.3), WF-ASYM (Fig. 4.7) and LEX (Fig. 5.3)) to explore the

behavior of the multi-hop wireless networks under MPT/MPR channel models.

Increasing K_R, Keeping K_T Constant

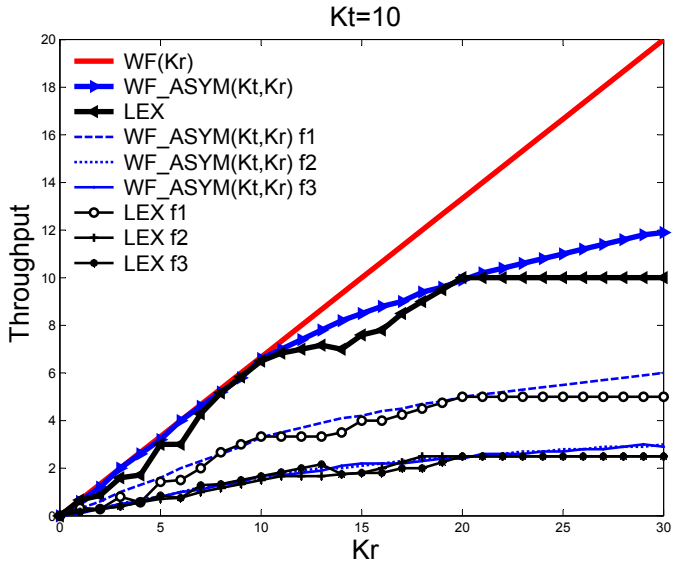


Fig. 5.5: The end-to-end throughput for $K_T = 10$ and $K_R = 1, ..., 30$.

In Fig. 5.5, we set the value of K_T to 10 and increase the value of K_R from 0 to 30. The y-axis represents the end-to-end throughput measured in packets/slot. The thick curves show the aggregate throughput of the three flows while the thin curves represent the throughput of each flow. Since WF calculates the rate allocation when $K_T = K_R$, the solid thick line (red) shows the aggregate throughput when applying WF(K_R), which means that in Fig. 4.3, the resulting rate allocation is scaled by K_R. The aggregate throughput of WF (K_R) is linearly increasing with the value of K_R which results in a straight line. This straight line serves as a benchmark against which the performance of the asymmetric networks (*i.e.*, $K_T \neq K_R$) is compared.

The thick curve (blue) with the right-pointing triangle markers is the aggregate throughput computed by WF-ASYM(K_T, K_R). This curve drifts away from the line of WF(K_R) after $K_R > K_T = 10$ because once K_R is greater than K_T, the end-to-end throughput is restricted by the transmission capacity limit. However, increasing the value of K_R can still improve the aggregate throughput when K_R is greater than K_T. The reason was discussed earlier when introducing the definition of the advanced channel model in Section 4.2. Due to the broadcast nature of the wireless network, a node which serves as a relay or receiver may need to decode both the desired and overheard transmissions. Therefore, by increasing K_R the relays or receivers can handle more interference and yield greater throughput.

The thick curve (black) with the left-pointing triangle markers shows the aggregate throughput of LEX with $\Delta = 2$. We assume perfect channel quality for all arcs, *i.e.,* $\hat{K}_R(l) = K_R$ for any $l \in \boldsymbol{l}$. Note that LEX approximates the max-min schedule for the asymmetric cases. The reason for the gap between the curves of WF-ASYM and LEX is twofold. First, WF-ASYM computes the approximate max-min rate allocation which may not be feasible. Second, LEX is a heuristic algorithm which may not always return the lexicographically optimal schedule. Nonetheless, in this example, the schedules computed by LEX have two desirable attributes:

- They are efficient because the curve of LEX is close to that of WF-ASYM up to $K_R = 20$. When K_R is greater than 20 the curve of LEX becomes flat because the rate allocation of the point ($K_R = 20,\ K_T = 10$) is equal to the max-min rate allocation of the system when $K_T = 10$ and $K_R = \infty$. The reason will be discussed shortly.

- They appear to be fair because the throughput of $f1$ is approximately twice that of $f2$ or $f3$. This is the desired fairness in the sense that $f1$ does not compete with $f2$ or $f3$, which are bottlenecked by the same maximal clique (*i.e.,* MC2 in Table. 5.1).

The curve of LEX becomes flat when $K_R > 2K_T = 20$ because when K_R is twice the value of K_T, K_T becomes the bottleneck of the network. For example, when $K_T = 10$ and $K_R = 20$, the optimal schedule of $f1$ is straightforward. For every two slots, arc $0 \rightarrow 2$ and $17 \rightarrow 13$ transmit concurrently at 10 packets/slot at the first slot. No collision will occur because node 2 can receive 20 packets at a slot. At the second slot, arc $2 \rightarrow 17$ transmits at 10 packets/slot. Therefore, the end-to-end throughput of $f1$ is $10/2 = 5$ packets/slot. The performance cannot be further improved by increasing K_R because the most busy node, node 2 (the node is either transmitting or receiving at full speed at every slot of the scheduling cycle), has two inbound arcs, *i.e.,* arc $0 \rightarrow 2$ and arc $17 \rightarrow 13$ (overheard by node 2). The number of inbound arcs of the busy node determines to what extend the end-to-end throughput can be improved by increasing the value of K_R. Hence, in this example, the throughput of $f1$ cannot be increased further once $K_R = 2K_T$. This is property is not captured by WF-ASYM whose curve keeps increasing when $K_R > 2K_T = 20$. The reason is discussed in Section 4.5.2: Eq. 4.37 and 4.38 are not sufficient conditions to determine a saturated MC and therefore, the rate allocation computed by WF-ASYM, is an approximate max-min allocation.

Increasing K_T, Keeping K_R Constant

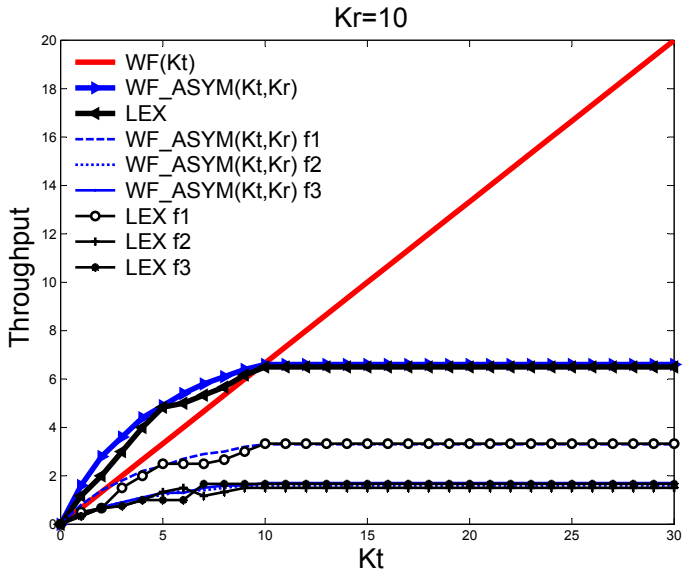


Fig. 5.6: The end-to-end throughput for $K_R = 10$ and $K_T = 1, ..., 30$.

We have observed that increasing K_R separately can improve the aggregate throughput while supporting a certain level of fairness. However, the behavior of both WF-ASYM and LEX are dramatically different when increasing K_T while keeping K_R constant, as shown in Fig. 5.6. The line of WF(K_T) is the same as the line of WF(K_R) in Fig. 5.5. The curve of WF-ASYM is above the line of WF when $K_T < K_R = 10$, which is consistent with the earlier observation that increasing K_R to a value greater than K_T can improve the aggregate throughput. The causes for the gap between WF-ASYM and LEX are similar as in the previous example. Fig. 5.6 is different from Fig. 5.5 in that, when K_T is greater than K_R, the curves of LEX become flat. Those flat lines imply that when K_T is greater than K_R, the increase of K_T cannot further improve the throughput. The reason is intuitively clear. When increasing K_T to a value greater than K_R, the increased traffic cannot be handled by the receiver and hence will not benefit the end-to-end throughput.

A larger network is depicted by Fig. 5.7 which has 40 randomly located nodes. Three flows traversing the network:

- flow 1 ($34 \Rightarrow 26$: $34 \rightarrow 38 \rightarrow 7 \rightarrow 29 \rightarrow 2 \rightarrow 6$);
- flow 2 ($25 \Rightarrow 21$: $25 \rightarrow 10 \rightarrow 12 \rightarrow 29 \rightarrow 2 \rightarrow 26 \rightarrow 21$);
- flow 3 ($11 \Rightarrow 31$: $11 \rightarrow 33 \rightarrow 5 \rightarrow 38 \rightarrow 16 \rightarrow 12 \rightarrow 9 \rightarrow 35 \rightarrow 31$).

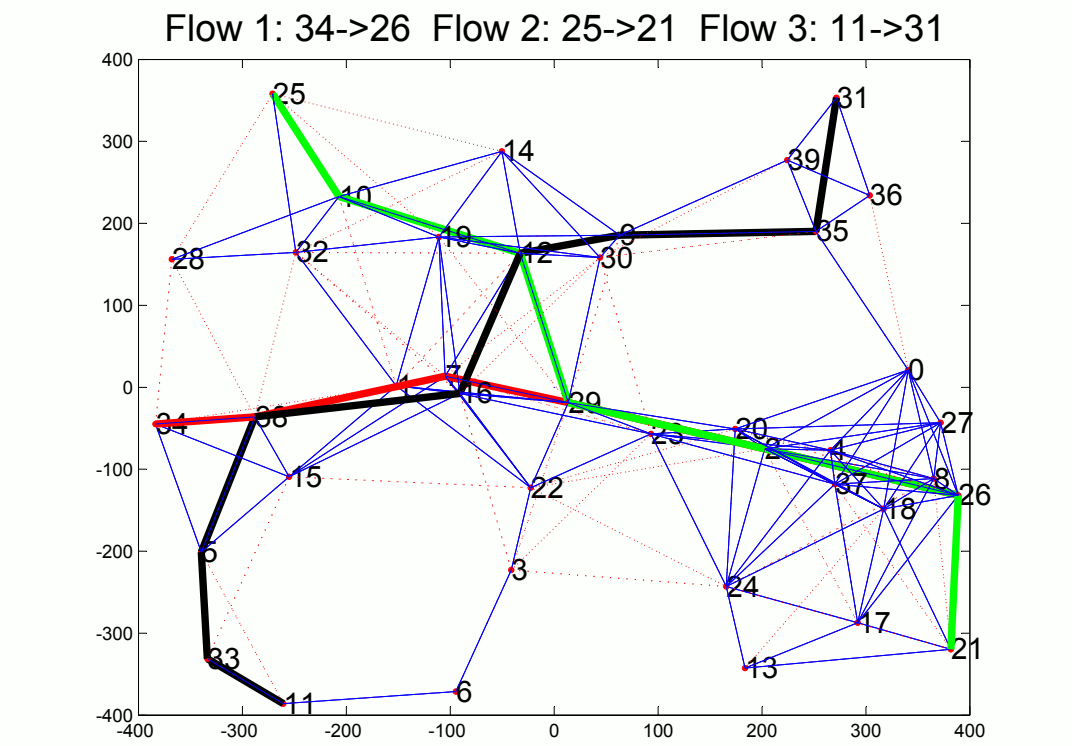


Fig. 5.7: A network of 40 random nodes.

Compared to Fig. 5.4(a), the three flows are bottlenecked by the same MC and therefore have the same max-min rate. Fig. 5.8 and 5.9 illustrate the relative impact of K_R and K_T on the aggregate throughput, respectively. The curves of Fig. 5.8 and 5.9 show a similar trend as the example of Fig. 5.4(a) except that the three flows have approximately the same rate when either WF-ASYM or LEX is applied. This is the desired fairness because these flows are bottlenecked by the same MC and have the same max-min rate. It is worth noting that in Fig. 5.8, the curves of LEX are not flat after $K_R > 2K_T$, as in Fig. 5.5. The reason is that the maximum number of inbound arcs of the network is greater than two (*e.g.,* in Fig. 5.7 the inbound arcs of node 29 include: 12->29, 7->29, 16->12, etc) which implies that the aggregate throughput can be further improved when $K_R > 2K_T$.

After studying of the end-to-end throughput and the fairness of MPT/MPR networks, next, we illustrate the schedules computed by MDSATUR and LEX. We will present two scheduling examples when the channels are perfect and one example when some arcs correspond to non-perfect channels.

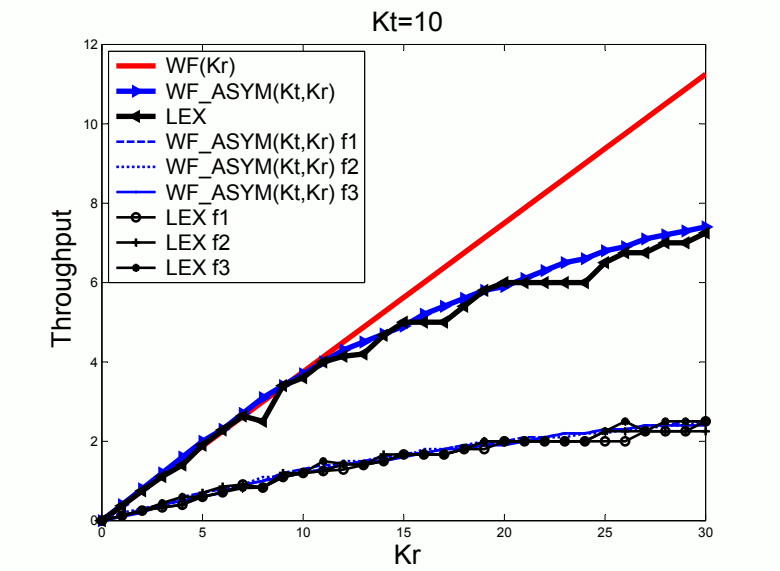


Fig. 5.8: The end-to-end throughput for $K_R = 10$ and $K_T = 1, ..., 30$.

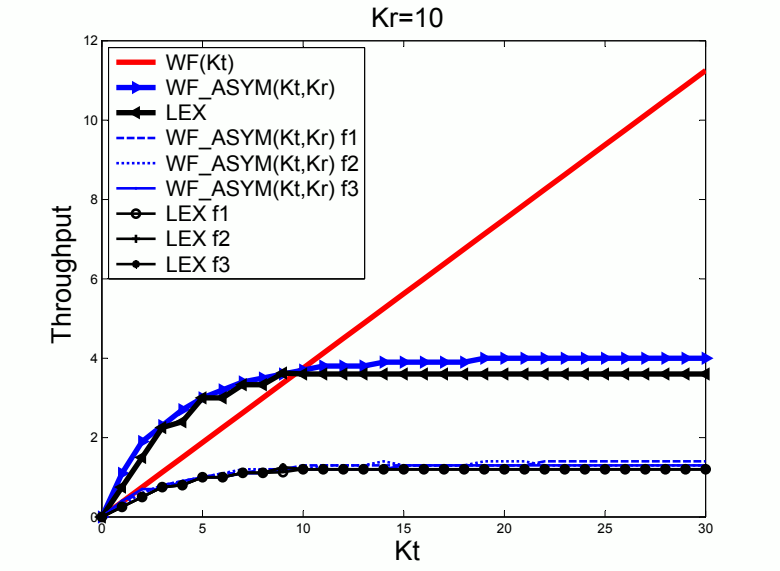


Fig. 5.9: The end-to-end throughput for $K_T = 10$ and $K_R = 1, ..., 30$.

5.3.2 Homogeneous Network

Firstly, we show the schedule of the basic scenario when $K_T = K_R = 1$ and the channels are perfect for any arc, *i.e.,* $\hat{K}_R(l) = K_R$ for any $l \in \boldsymbol{l}$. In Table. 5.2, the first and second rows represent the flows and the arcs crossed by each flow, respectively. The max-min fair allocation is computed by WF and displayed in the row of *flow rate*. The max-min rate of flow 1 is 2/6 packets/slot and, for flow 2 and 3, it is 1/6 packets/slot.

The following six rows list the schedule computed by MDSATUR, based on the allocation {1/3,1/6,1/6}. The scheduling cycle is six slots. The row of each slot shows the number of packets each arc should transmit at that slot. *e.g.,* for slot 1, arc $0 \rightarrow 2$ and $1 \rightarrow 15$ should transmit one packet concurrently. Referring to Fig. 5.4(a), arc $0 \rightarrow 2$ and arc $1 \rightarrow 15$ are independent which is the reason they can coexist in a conventional channel model. Every column of the last six rows shows the schedule for each arc. For instance, arc $17 \rightarrow 13$ can transmit one packet at slot 3 and one packet at slot 6. Therefore, for every six slots, every arc of flow 1 can transmit 2 packets. With this schedule, flow 1 can achieve the rate of 2/6 packets/slot. Similar reasoning applies to flow 2 and 3.

Table. 5.2: The scheduling example ($K_T = K_R = 1$).

Flow ID	1			2					3			
Arcs	$0 \rightarrow 2$	$2 \rightarrow 17$	$17 \rightarrow 13$	$14 \rightarrow 10$	$10 \rightarrow 1$	$1 \rightarrow 15$	$15 \rightarrow 5$	$5 \rightarrow 11$	$5 \rightarrow 15$	$15 \rightarrow 7$	$7 \rightarrow 12$	$12 \rightarrow 9$
flow rate	2/6 (packets/slot)			1/6 (packets/slot)					1/6 (packets/slot)			
Slot 1	1					1						
Slot 2		1			1							
Slot 3			1							1		
Slot 4	1			1					1			
Slot 5		1						1			1	
Slot 6			1				1					1

If we set K_T to 2 and K_R to 3, we need to apply LEX, which can compute the approximate max-min schedule when $K_T \neq K_R$. Table. 5.3 shows the schedule for this case. The scheduling cycle is three slots. By repeating this schedule, every flow doubles its end-to-end throughput, compared to the case when $K_T = K_R = 1$. For example, at slot 1, $0 \rightarrow 2$ can transmit two packets while $17 \rightarrow 13$ transmit one packet. Observing Fig. 5.4(a), node 2 will be interfered with by node 17. However, since node 2 can handle three packets at one slot, their reception is collision-free (as it should be). Similarly, the five arcs of flow 2 and 3 (*i.e.,* $14 \rightarrow 10$, $1 \rightarrow 15$, $5 \rightarrow 11$, $5 \rightarrow 15$ and $12 \rightarrow 9$) can transmit one packet simultaneously at slot 1. Becasue $K_R = 3$, no node will be overwhelmed by the concurrent transmissions in its

neighborhood.

Table. 5.3: A scheduling example ($K_T = 2$ $K_R = 3$).

Flow ID	1			2					3			
Arcs	$0 \to 2$	$2 \to 17$	$17 \to 13$	$14 \to 10$	$10 \to 1$	$1 \to 15$	$15 \to 5$	$5 \to 11$	$5 \to 15$	$15 \to 7$	$7 \to 12$	$12 \to 9$
flow rate	2/3 (packets/slot)			1/3 (packets/slot)					1/3 (packets/slot)			
Slot 1	2		1	1		1		1	1			1
Slot 2		2			1		1			1		
Slot 3			1								1	

5.3.3 Non-Homogeneous Network

To illustrate how the scheduling algorithm works for a non-homogeneous network, we assume that reception capacity of arc $1 \to 15$ is impaired. More specifically, we assume $\hat{K}_R(1 \to 15) = 1$ while $K_R = 3$. Table. 5.4 shows the schedule with non-perfect channel qualities. Compare the rows of *Slot* 1 in Table. 5.3 and Table. 5.4. In Table. 5.4, arc $5 \to 15$ and arc $5 \to 11$ are idle at slot 1. This is because when $\hat{K}_R(1 \to 15) = 1$ node 15 can only decode one packet at a time, which deprives $5 \to 15$ and $5 \to 11$'s transmission opportunities at slot 1. As a result, the end-to-end throughput of flow 2 and 3 suffers from the non-perfect channel quality of $1 \to 15$. Even though flow 1's throughput is increased compared to the last example, the overall performance is downgraded with respect to both the aggregate throughput and the max-min fairness.

Table. 5.4: The scheduling examples ($K_T = 2$ $K_R = 3$) under non-perfect channel.

Flow ID	1			2					3			
Arcs	$0 \to 2$	$2 \to 17$	$17 \to 13$	$14 \to 10$	$10 \to 1$	$1 \to 15$	$15 \to 5$	$5 \to 11$	$5 \to 15$	$15 \to 7$	$7 \to 12$	$12 \to 9$
flow rate	3/4 (packets/slot)			1/4 (packets/slot)					1/4 (packets/slot)			
Slot 1	2		1	1		1						1
Slot 2		2			1			1		1		
Slot 3	1		2						1		1	
Slot 4		1					1					

5.4 Summary

In this chapter, we state the MDSATUR algorithm to calculate a discrete schedule for a given rate allocation in MPT/MPR networks. To approximate the max-min schedule for general MPT/MPR networks when K_T is not necessarily equal to K_R,

we present the LEX scheme. Numerical examples are presented to evaluate the performance of WF, WF-ASYM, MDSATUR and LEX. The impact of K_T and K_R is examined separately applying WF, WF_SYM and LEX. The experiments show that having K_R greater than K_T can upgrade the end-to-end throughput to a certain extent. However, the throughput cannot benefit from the efforts to increase K_T to a value greater than K_R. It is also shown that the discrete schedules computed by LEX can achieve efficiency and fairness at the same time, measured against the max-min allocation computed by WF and WF-ASYM. The discrete schedules created by MDSATUR and LEX are shown in examples of both homogeneous and non-homogeneous networks.

Chapter 6

Schemes for MPT/MPR Multi-Hop Wireless Networks

In Chapter 3 we proposed the AIMD-MAC algorithm for a single-hop scenario where a number of nodes compete for a single channel to communicate to one MPR receiver. In Chapter 4 and 5 we studied the MPT/MPR multi-hop wireless network from a centralized viewpoint. In this chapter, we develop multiple schemes based on the AIMD backpressure scheduling (AB-MAC) algorithm which integrates medium access control and scheduling in MPT/MPR ad hoc multi-hop wireless network.

A backoff mechanism avoids collisions by regulating the number of slots a node should defer its transmission; however, if applied to MPT/MPR networks, we would need to also estimate the number of packets a node should concurrently transmit after the backoff timer runs out. In the discussion of AIMD-MAC in Chapter 3, we observed that by having every node adaptively adjust its access probability (AP) according to the transmission history, the nodes can reach an "agreement" to efficiently and fairly use the channel. We extend in this chapter the AIMD approach to multi-hop scenarios. The basic idea is to control the transmissions on every arc of every flow via dynamically adjusting the APs.

We emphasize that the MPT/MPR channel model is different from the multi-channel model [88, 100] in that, in MPT/MPR channels, the nodes in transmission range can hear each other because they are not separated by channels. Therefore, for any transmitter, all the receivers (within transmission range of the considered transmitter) are exposed to the signal sent by the transmitter. *Whether the concurrent transmissions can be successfully decoded at the desired receiver is decided by the total number of packets arriving at the receiver and its MPR capacity limit.* We endow every node with its own, single AP regardless of the number of flows crossing the node, the adjustment of which satisfies the MPR limits of the receivers.

Specifically, node will use its AP to compute the number of packets it should concurrently transmit, which is referred to as the ***transmission quota***. Also the collection of packets sent concurrently to a node is called a ***bundle***. The next question is: how do we assign this transmission quota to the flows crossing the node? We adopt the strategy of backpressure (BP) algorithm by [75]. According to BP, only flows with positive differential backlog can be assigned the transmission quota. In order to fairly treat the flows, the transmission quota distribution process prioritizes the most poorly treated flow based on local flow information.

Combining the AIMD and BP methods, we propose in this chapter a hybrid additive-increase multiplicative-decrease backpressure MAC (AB-MAC) algorithm. AB-MAC regulates a number of flows traversing a MPT/MPR wireless network with the goal of maximizing the end-to-end throughput of the flows while supporting a certain level of fairness. The integration of AIMD and BP has the following desired properties.

- **AIMD regulates the MAC layer**
  The BP routing algorithm [75, 91] sees the MAC layer as a black-box by assuming a predefined set of globally feasible schedules, which is denoted by $\Im$ in Eq. 2.12. $\Im$ includes all the feasible schedules that are restricted by the topology and the capacity limits of the network and are known to the designers beforehand. Here, the AIMD component of AB-MAC generates a locally feasible schedule (hence distributed in nature) on-the-fly by dynamically adjusting the AP according to the neighbors' reception status.

- **Local computation of AP**
  The BP algorithm involves solving the max-weight problem (Eq. 2.11) to compute the optimal link rate matrix, which needs global information and can be NP-hard [90]. On the other hand, the AP adjustment operation is performed locally and only requires constant computation time. Note that this means we will only be able to approximate the optimal performance.

- **Reduction of the intra-flow contention**
  The differential backlog needed by BP is the backlog difference between two successive nodes (the upstream node and the downstream node) of a flow. Restricting the candidate flows to the flows with positive differential backlogs guarantees that a node will not transmit to its downstream node if the downstream node has greater backlog. This principle prevents the upstream nodes from monopolizing the channel and overflowing the downstream nodes and therefore,

alleviates the intra-flow contention problem, which can cause poor performance in multi-hop networks [58, 59, 80, 109].

In Section 6.2 we introduce three simulation schemes to explore the performance of AB-MAC, *the mini-slot scheme*, *the random scheme* and *the prioritized scheme* and compare against CSMA/CA. The three simulation schemes demonstrate how different levels of the node coordination affect the performance of AB-MAC and the potential directions to further improve AB-MAC in MPT/MPR networks.

6.1 The AB-MAC Algorithm

In the following, the three components of AB-MAC: 1) exchange of local information, 2) computation of AP and 3) distribution of transmission quota are discussed in detail.

6.1.1 Local Information Exchange

To update the AP, a node requires the neighbors' latest reception status; and also, to select the potential flows (the flows with positive differential backlog), it requires the backlog information for each flow of its downstream nodes. Therefore, an important component of the AB-MAC algorithm is the exchange of local information between immediate neighbors.

The local information is exchanged in two ways: 1) end-of-reception acknowledgement and 2) piggybacked local flow status. To update the AP, the node needs the recent reception status of its immediate neighbors. Thus, at the end of a reception, the receiver broadcasts an acknowledgement to signify the status of the reception. The local flow status is piggybacked in the header of regular data packets and exchanged between neighbors.

End-of-Reception Acknowledgment

Since the AP updating (either increase or decrease) is decided by the neighbors' recent reception status, it is important that the nodes have the latest reception status before the AP adjustment. Thus, at the end of a reception, a receiver needs to broadcast its reception status via a binary acknowledgement. If the number of arriving packets at the receiver did not exceed K_R, the reception was successful and the receiver broadcasts a positive acknowledgement denoted by ACK [1]; if the number of arriving

[1] We use the term ACK (NAK) for what is a single bit feedback to indicate success or failure.

packets exceeded K_R, the reception failed and the receiver broadcasts a negative acknowledgement denoted by NAK. Note that ACK is different from its traditional meaning in CSMA/CA because it does not include the packet information. The sole purpose of the ACK is to inform the neighbors (including the sender) of the latest reception status seen by the receiver. The ACK is of constant length regardless of the number of packets decoded by the receiver. Similarly, the purpose of the NAK is to inform the neighbors that the latest reception has failed.

The reception of the ACK/NAKs follows the same receiving model of the data packets: if the total number of ACK/NAKs is less than or equal to K_R, the ACK/NAKs can be decoded by the node; otherwise, the ACK/NAKs will collide and cannot be decoded. Due to the half-duplex property, at the end of slot, if a node is sending ACK/NAKs, it cannot receive the ACK/NAKs from other nodes. Since a node needs to broadcast ACK/NAK at the end of a receiving slot, only nodes which are in transmitting or idle state can receive the ACK/NAKs from others at the end of a slot.

To keep track of the transmission status, every node maintains three sets: intended-receivers, receive-ACK-from and receive-NAK-from.

- **intended-receivers**: This set contains all the intended receivers of the bundle of packets transmitted at the current slot. If the node does not transmit at the slot, this set is empty.
- **receive-ACK-from**: This set contains the transmitters of all the successfully decoded ACKs at the end of slot.
- **receive-NAK-from**: This set contains the transmitters of all the successfully decoded NAKs at the end of slot.

Note that for any node, the intersection of receive-ACK-from and receive-NAK-from is empty because a receiver cannot be successful and failed at the same slot. The set receive-ACK-from (receive-NAK-from) does not necessarily belong to the set intended-receivers because a node can receive ACK/NAK from a node which was not its intended receiver. In Section 6.1.2, we will discuss how to use the received ACK/NAKs to update the AP.

Local Flow Status Piggybacking

The neighbors exchange local flow status (Table. 6.1) by including this information in the header of each data packet. The id of the node is denoted by n. d_n is the number of active nodes in one-hop distance of node n. This number can be obtained

by maintaining a neighbor table. If a packet is received from a new neighbor, an entry is added; if a neighbor has not been heard for a certain amount of time, the corresponding entry is expunged.

Table. 6.1: The local flow status embedded in the header.

n	node id
d_n	degree of node n
f	flow id
$Q_n^{(f)}$	backlog of the flow at node n

The last two elements of the header are the id and the backlog length of flow f. In order to guarantee that the header has constant (and small) length regardless of the number of flows crossing a node, each regular packet includes a single flow's status in the header. Flow status is carried in a round-robin fashion. Thus, the length of the header will not significantly affect the performance, and additionally because of MPT/MPR, a node can transmit multiple packets and hence multiple flow status updates in a slot, sending the status of many flows ($> K_R$) may require multiple slots. Nonetheless, as we will show in the simulation of 100 nodes (Section 6.2.3), even when the number of flows is greater than the value of K_R, our schemes still outperforms the benchmark, *i.e.,* the impact of sending flow status updates in round-robin fashion is not significant.

Each node maintains a flow table for all the flows crossing the node, containing the local and the collected flow status from its neighbors (by extracting the piggybacked information from the header of received data packets). The information entry for flow f at node n is illustrated in Table. 6.2. $Q_n^{(f)}$ is the length of the backlog of flow f at node n. n'_f is the downstream node of flow f. (*e.g.,* in Fig. 6.1, n_2 is the downstream node of f_2 at n_1.) $Q_{n'_f}^{(f)}$ is the backlog of flow f on node n'_f and $\psi_n^{(f)}$ is the differential backlog of flow f between n and n'_f (*i.e.,* $\psi_n^{(f)} = Q_n^{(f)} - Q_{n'_f}^{(f)}$). $\mathcal{T}_n^{(f)}$ is the total packets node n has transmitted so far for flow f, which is used to decide the priority levels of the potential flows when distributing the transmission quota.

Table. 6.2: The information in the entry of flow f at node n.

$Q_n^{(f)}$	backlog of flow f
n'_f	downstream node of f
$Q_{n'_f}^{(f)}$	backlog of flow f at n'_f
$\psi_n^{(f)}$	differential backlog of flow f
$\mathcal{T}_n^{(f)}$	number of transmitted packets for flow f

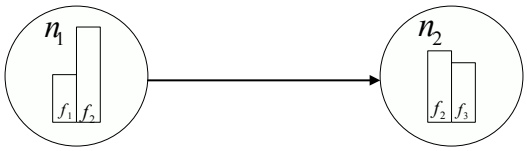


Fig. 6.1: An example of local information exchange.

Extensive experiments show that, due to the constantly changing status of the neighbors, it is essential to feed the node with timely information of its neighbors. As a result, AB-MAC requires that a node does not transmit in two successive slots, *i.e.,* only after a reception or idle slot, can a node transmit. This transmission constraint ensures that every node has the chance to listen after its transmission so that it can extract the latest flow status from the data packets received form its neighbors. Note that in a multi-hop flow traversing a path of nodes equipped with half-duplex radios, the relays should interchange between transmitting and receiving states over time such that the flow can traverse smoothly from the source to the destination [2].

Fig. 6.1 shows an example of a sub-network. n_1 and n_2 are in transmission range. n_1 is the relay for flows f_1 and f_2 and n_2 is the relay of f_2 and f_3. f_2 is traversing from n_1 to n_2. n_1 maintains a flow table of two entries, one for f_1 and one for f_2, with every entry including all the attributes listed in Table. 6.2. Once n_1 successfully receives a data packet from n_2, n_1 extracts the four attributes listed in Table. 6.1. With the value of f, n_1 looks up its flow table. For example, if it is f_2, n_1 then checks the node id in the header, which is n_2 in this case. If the transmitter (*i.e.,* n_2) is the downstream node of f_2 as recorded in the flow table, n_1 will update the backlog of the flow for the downstream relay. In this example, n_1 will update $Q_{n_2}^{(f_2)}$ of its flow table by the value of $Q_{n_2}^{(f_2)}$ extracted from the header of the received packet. The differential backlog $\psi_{n_1}^{(f_2)}$ will be updated accordingly.

Next, we discuss how a node can utilize the knowledge about its neighborhood to control the action of the node.

6.1.2 AP Computation

The basic idea of the AIMD approach is to decrease the value of AP multiplicatively after a collision; otherwise, increase the AP additively. The increments probe for the channel capacity limit while the reduction decreases the chance of collisions. We introduce the all-receiver-considered AIMD (ARC-AIMD) strategy to update the APs.

[2]This can limit the throughput of single-hop flows.

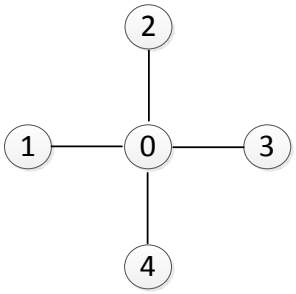


Fig. 6.2: An example of the ARC-AIMD method.

Fig. 6.2 shows an example topology to explain the ARC-AIMD method. Since a node's transmission will not only be heard by the desired receiver but all the nodes in transmission range, to adjust the AP of a node n, it is useful to take into consideration the latest reception status of all the neighbors of node n. For example, in Fig. 6.2, if any neighbor of node 0 (*e.g.,* node 1) just experienced a failed reception, the AP of node 0 should be multiplicatively decreased because the transmission from node 0 can be heard by node 1; on the other hand, if all the intended receivers of node 0 respond by an ACK at the end of slot, node 0 should increase the AP additively to probe for higher transmission opportunities.

For a node n_o, denote the three sets intended-receivers, receive-ACK-from and receive-NAK-from by $\boldsymbol{n}_{\mathbb{I}}(n_o)$, $\boldsymbol{n}_{\mathbb{A}}(n_o)$ and $\boldsymbol{n}_{\mathbb{N}}(n_o)$, respectively. At the end of a non-receiving slot (receiving nodes need to broadcast ACK/NAK and hence cannot receive at the same time), n_o can receive ACK/NAKs from its neighbors. With the latest reception status of the neighbors, n_o can decide whether it should increase or decrease its AP (*i.e.,* p_{n_o}) following the ARC-AIMD algorithm (Fig. 6.3). Note that if $|\boldsymbol{n}_{\mathbb{A}}(n_o)| + |\boldsymbol{n}_{\mathbb{N}}(n_o)| > K_R$, the acknowledgement arriving at n_o exceeds the reception capacity limit and cannot be decoded by n_o.

Each packet transmission from n_o can have three possible dispositions at the intended receiver n_r: 1) the packet is successfully decoded by the receiver n_r; 2) the packet collides with other arriving packets at the receiver n_r (named the 2nd-hop collision at n_r) and 3) n_r is in transmitting state and ignores the packet (named the 1st-hop collision at n_r). The successful reception (case 1) can be confirmed by an ACK from n_r and a 2nd-hop collision (case 2) can be confirmed by a NAK from n_r. If n_r does not respond by neither an ACK nor NAK, it implies that a 1st-hop collision occurs at n_r because n_r ignores the packets while transmitting and will not broadcast any acknowledgement after a transmission. Normally, if $\boldsymbol{n}_{\mathbb{I}}(n_o) \setminus (\boldsymbol{n}_{\mathbb{A}}(n_o) \cup \boldsymbol{n}_{\mathbb{N}}(n_o))$ is not empty, a 1st-hop collision occurs in at least one of n_o's intended receivers.

ARC-AIMD ($\boldsymbol{n}_{\mathbb{I}}(n_o)$, $\boldsymbol{n}_{\mathbb{A}}(n_o)$, $\boldsymbol{n}_{\mathbb{N}}(n_o)$)
1 **if** (($|\boldsymbol{n}_{\mathbb{I}}(n_o)| > 0$) **and**
(($|\boldsymbol{n}_{\mathbb{N}}(n_o)| > 0$) **or** ($|\boldsymbol{n}_{\mathbb{A}}(n_o)| + |\boldsymbol{n}_{\mathbb{N}}(n_o)| > K_R$) **or** ($|\boldsymbol{n}_{\mathbb{I}}(n_o) \setminus (\boldsymbol{n}_{\mathbb{A}}(n_o) \cup \boldsymbol{n}_{\mathbb{N}}(n_o))| > 0$)))
2 $p_{n_o} = \max\{p_{n_o} \times \beta, \alpha\}$;
3 **else**
4 $p_{n_o} = \min\{p_{n_o} + \alpha, 1.0\}$;
5 **end**
6 **return** p_{n_o};

Fig. 6.3: The ARC-AIMD algorithm.

In the ARC-AIMD algorithm, n_o decreases the AP by the decreasing factor β ($0 < \beta < 1$) after a transmission (*i.e.,* $|\boldsymbol{n}_{\mathbb{I}}(n_o)| > 0$) if one of the three conditions holds: 1) n_o receives NAKs (*i.e.,* $|\boldsymbol{n}_{\mathbb{N}}(n_o)| > 0$), 2) ACK/NAKs collide at n_o (*i.e.,* $|\boldsymbol{n}_{\mathbb{A}}(n_o)| + |\boldsymbol{n}_{\mathbb{N}}(n_o)| > K_R$), or, 3) a 1st-hop collision happens in at least one of n_o's intended receivers (*i.e.,* $|\boldsymbol{n}_{\mathbb{I}}(n_o) \setminus (\boldsymbol{n}_{\mathbb{A}}(n_o) \cup \boldsymbol{n}_{\mathbb{N}}(n_o))| > 0$). These three conditions include all the possible causes for which n_o does not receive ACKs from all its intended neighbors. Decreasing the AP under these conditions guarantees that n_o will only keep transmitting to a node n_r if n_o has confirmation that n_r can decode the packets from n_o. If n_o receives a NAK, it means at one of n_o's neighbor, the arriving packets (including the packets from n_o) exceed the MPR capacity limit. Since the packets transmitted from n_o can be heard by all its neighbors, n_o should decrease its AP to prevent future collisions at this neighbor even if this neighbor was not one of n_o's intended receivers. On the other hand, if none of these conditions holds or n_o did not transmit, it means that either n_o has received the ACKs from all of its intended receivers or n_o has no impact on the channel at the current slot. In either case, n_o increases the AP by the minimum AP α ($0 < \alpha < 1$) and hence probes for more transmission opportunities.

Comparing with the RTS/CTS in CSMA/CA, ARC-AIMD has two advantages.

- **Alleviates the hidden terminal problem**
  Unlike CSMA/CA, in which transmissions are regulated by exchanging control messages within one-hop distance, the ARC-AIMD method actually permits nodes within two-hop distance to communicate indirectly. For example, in Fig. 6.2, the AP of node 1 is adjusted according to the reception status of node 0, which is in turn decided by the APs of node 1, 2, 3 and 4. Thus, by informing node 1 of the reception status of node 0, node 1 learns about whether node 2, 3 and 4 are too aggressive or not, as the neighbors transmitting to node 0. Note that we do not claim that the ARC-AIMD method can guarantee a collision-free network. Even if every node applies the optimal AP, collisions will occur due

to the fact that the transmissions are random events. However, we will show that the ARC-AIMD method can greatly reduce the overhead caused by the RTS/CTS handshake mechanism while efficiently utilizing the network.

- **Avoids the asymmetric information problem**
  The asymmetric information problem significantly affects the system performance for two reasons. 1) As shown in Fig. 2.2(a), because the data packet is much longer than the control packets (*i.e.,* RTS/CTS packets), the node ready to transmit (*i.e.,* node A) could lose the competition for channel access if it is not aware of the transmissions sent by a competitor (*i.e.,* node D) and repeatedly sends RTS, which will lead to a large contention window (CW) at node A. 2) In Fig. 2.2(b), node C knows when the transmission on l_1 ends because it can hear CTS and ACK from node B. Thus, node C will send RTS to node D when the transmission on l_1 ends. However, node A is not aware of the transmission on l_2 because it cannot receive the message from node C. Since the data packet is much longer than the control packets, there is a high chance that node A will repeatedly send RTS when node C is transmitting. Because A cannot obtain the CTS from B when C is transmitting, node A will end up with a large CW and be unable to use the channel.

  However, if the ARC-AIMD method is applied instead of the RTS/CTS and backoff mechanisms, in Fig. 2.2(a), because neither B and C will have failed reception (both node B and C have only one neighbor which is transmitting), node A and D will always have their APs equal to 1, *i.e.,* node A and D can transmit concurrently. In Fig. 2.2(b), the AP of A is determined by the reception status at node B and the AP of C is determined by the reception status at node B and D. Since node D only has one neighbor, it will not have failed receptions. Thus, if a collision occurs at node B, both node A and C will decrease their APs.

Parameter Configuration

In this section, we discuss how to set the values of α and β used to regulate AIMD dynamics, depending on the density of the network and the value of K_T and K_R. From the discussion of the parameters of AIMD-MAC in Section 3.5, we recognized that the equilibrium point of the AP, denoted by $\tilde{p}$, can be computed by Eq. 3.10. If $\tilde{p}$ is close to the optimal AP, denoted by p^*, the APs of the nodes tend to oscillate around p^*, which can lead to desirable performance. Unfortunately, even for the single-hop scenario, we do not have a closed formula to compute p^*. Therefore, in

order to appropriately set α and β for AB-MAC for a multi-hop network, we apply a heuristic approach.

Fig. 3.9 shows that, the value of p^* can be approximated by the value of K/M where K is the reception capacity of the receiver and M is the number of nodes in one-hop distance of the receiver. In other words, M is the degree of the receiver. This approximation can be interpreted as: the reception capability of the receiver K should be evenly distributed over all the M nodes competing for the resource. In a multi-hop scenario, however, any node n_o may have multiple potential receivers, each with different degree. Hence, the problem is, which node's degree should be chosen as the one to approximate the optimal AP of n_o, denoted by $p^*_{n_o}$.

A conservative approach is to select the greatest degree over all the active neighbors. Formally, $p^*_{n_o}$ is approximated by the Eq. 6.1 where $\mathcal{D}_{n_o} = \max\{d_n | n \in \boldsymbol{n}_{\mathbb{1}}(n_o)\}$. K_T and K_R are the transmission and reception capacity limits. d_n can be obtained from packets sent by node n.

$$p^*_{n_o} \approx \frac{K_R}{\mathcal{D}_{n_o} K_T} \tag{6.1}$$

To allow the APs to oscillate around the approximated $p^*_{n_o}$, combining Eq. 3.10 and Eq. 6.1 leads to the relationship between α and β represented by Eq. 6.2. Eq. 6.2 implies that the nodes with busy neighbors (*i.e.,* nodes with high degree) will have a lower equilibrium point and therefore lower value of AP.

$$\frac{\alpha}{1-\beta} = \frac{K_R}{\mathcal{D}_{n_o} K_T} \tag{6.2}$$

With Eq. 6.2, if we know the value of α, the value of β can be computed accordingly. As mentioned earlier, the AP is increased by α to probe for the channel capacity limit. Thus, the value of α determines how fast a node will reach the channel capacity limit. We set an integer parameter τ to configure the value of α as shown in Eq. 6.3. We will show in the next section that τ is the number of increment operations needed for a node to be able to transmit one more packet. From Eq. 6.2 and 6.3, the value of β is expressed by Eq. 6.4.

$$\alpha = \frac{1}{\tau K_T} \tag{6.3}$$

$$\beta = 1 - \frac{\mathcal{D}_{n_o}}{\tau K_R} \tag{6.4}$$

Hence, AB-MAC has three parameters K_T, K_R and τ to jointly decide the value of α and β. We follow our earlier observation in Section 3.5.3 and set τ to 10 in the experiments of this chapter. The value of AP is a fraction between 0 and 1. Next, we

illustrate how to convert the fraction into an integer, which consequently expresses the upper-bound of the number of packets a node can transmit.

6.1.3 Transmission Quota Distribution

Knowing the value of p_{n_o}, the number of packets n_o can transmit in the next slot is computed by Eq. 6.5. Γ_{n_o} is referred as the ***transmission quota*** of node n_o. $\mathfrak{X}$ is equal to 1 with probability $(K_T \times p_{n_o} - \lfloor K_T \times p_{n_o} \rfloor)$ and is equal to 0 with probability $(1 - (K_T \times p_{n_o} - \lfloor K_T \times p_{n_o} \rfloor))$. Since p_{n_o} is multiplied by K_T when computing the transmission quota, if $p_{n_o} \leq (1 - 1/K_T)$, after τ consecutive increasing operations, Γ_{n_o} will be increased by 1.

$$\Gamma_{n_o} = \lfloor K_T \times p_{n_o} \rfloor + \mathfrak{X} \tag{6.5}$$

The next question is, how to distribute the Γ_{n_o} packets among all the flows crossing n_o. We adopt the strategy suggested in the backpressure (BP) algorithm in [75] which only selects the flows with positive differential backlog as the potential flows to be assigned the transmission quota. This BP method prevents upstream nodes from starving the downstream nodes and therefore relieves the intra-flow contention problem. Additionally, every flow records the number of packets it has sent out by updating $\mathfrak{T}_{n_o}^{(f)}$ in the flow table (Table. 6.2). The potential flows are then sorted in increasing order of $\mathfrak{T}_{n_o}^{(f)}$. $\mathfrak{T}_{n_o}^{(f)}$ reflects the amount of channel resource flow f has used at node n_o, thus, by prioritizing the most poorly treated flow (*i.e.,* the one at the head of the sorted potential flows), we can prioritize the most poorly treated flow at node n_o. The transmission quota distribution (TQD) algorithm (Fig. 6.4) formally describes the procedure to distribute the transmission quota Γ_{n_o} to the flows crossing n_o, denoted by $\boldsymbol{f}_{n_o}$.

$$\psi_{n_o}^{(f)} = \psi_{n_o}^{(f)} - 2t_{n_o}^{(f)} \tag{6.6}$$

$$\boldsymbol{f}_{n_o}^{*} = \{f | \psi_{n_o}^{(f)} > 0 \wedge f \in \boldsymbol{f}_{n_o}\} \tag{6.7}$$

$$\mathfrak{T}_{n_o}^{(f)} = \mathfrak{T}_{n_o}^{(f)} + t_{n_o}^{(f)} \tag{6.8}$$

$t_{n_o}^{(f)}$ represents the number of packets flow f should transmit from n_o and it is set to 0 initially. $\boldsymbol{t}_{n_o}$ represents the set of transmission quotas for all the flows traversing n_o and is returned by TQD. If Γ_{n_o} is less than one, the procedure terminates. Otherwise, at line 3, the differential backlog $\psi_{n_o}^{(f)}$ is updated by Eq. 6.6. If n_o transmitted $t_{n_o}^{(f)}$

```
TQD (Γ_{n_o}, f_{n_o})
1   t_{n_o}^{(f)} = 0 for any flow f ∈ f_{n_o};
2   while Γ_{n_o} > 0
3       Update ψ_{n_o}^{(f)} by Eq. 6.6;
4       Select candidate flow set f*_{n_o} according to Eq. 6.7;
5       if f*_{n_o} ≠ ∅
6           Update Υ_{n_o}^{(f)} for any f ∈ f*_{n_o} by Eq. 6.8;
7           f*_{n_o} = arg min_{f∈f*_{n_o}} (Υ_{n_o}^{(f)}). Break ties arbitrarily.
8           t_{n_o}^{(f*_{n_o})} = t_{n_o}^{(f*_{n_o})} + 1;
9           Γ_{n_o} = Γ_{n_o} − 1;
10          f*_{n_o} = ∅;
11      else
12          break;
13      end
14  end
15  return t_{n_o};
```

Fig. 6.4: The transmission quota distribution (TQD) algorithm.

packets for flow f, the downstream node will increase the backlog by $t_{n_o}^{(f)}$ packets, if it is not the destination of flow f. Thus, $\psi_{n_o}^{(f)}$ should be decreased by $2t_{n_o}^{(f)}$.

The potential flow set $\boldsymbol{f}_{n_o}^*$ contains the flows which can transmit according to Eq. 6.7. If there does not exist such a flow, the procedure terminates. Note that the flows with empty backlog cannot have a positive differential backlog, therefore, those empty flows will not waste the transmission quota.

$\Upsilon_{n_o}^{(f)}$ is the total number of packet n_o has transmitted for flow f. It is updated at line 6 by counting in the newly determined value of $t_{n_o}^{(f)}$ (Eq. 6.8). The flow with the minimum $\Upsilon_{n_o}^{(f)}$ is selected (denoted by $f_{n_o}^*$) at line 7 and its transmission quota (*i.e.*, $t_{n_o}^{(f_{n_o}^*)}$) is increased by one (line 8). Accordingly, the total transmission quota Γ_{n_o} is decreased by one (line 9). At this point, one unit of the transmission quota is assigned to a flow and the candidate flow set $\boldsymbol{f}_{n_o}^*$ is cleared (line 10) before the next round of quota distribution starts. Each iteration of the TQD algorithm allows the most poorly treated potential flow to transmit one more packet. The procedure repeats until either Γ_{n_o} is zero or there are no more potential flows satisfying Eq. 6.7.

Since Γ_{n_o} cannot be greater than K_T, the *while* loop can repeat at most K_T times. Inside each iteration, we need to find the candidate flow with the minimum $\Upsilon_{n_o}^{(f)}$. Thus, the time complexity of TQD is $O(K_T|\boldsymbol{f}_{n_o}|)$.

```
AB-MAC (K_T, K_R, τ, I)
1    Update n_I(n_o), n_A(n_o), n_N(n_o) and the flow table (Table. 6.2);
2    p_{n_o}=ARC-AIMD(n_I(n_o), n_A(n_o), n_N(n_o));
3    Generate a uniformly distributed random value x, x ∈ [0,1];
4    if ((n_o did not transmit) or (|n_I(n_o) \ (n_A(n_o) ∪ n_N(n_o))| > 0 and x < 0.5))
5        Compute the transmission quota Γ_{n_o} by Eq. 6.5;
6        t_{n_0} = TQD(Γ_{n_o}, f_{n_o});
7        P = ∅;
8        for each f ∈ f_{n_o}
9            while t_{n_o}^{(f)} > 0
10               Select one packet P from flow f to be transmitted;
11               t_{n_o}^{(f)} = t_{n_o}^{(f)} − 1;
12               I = I mod (|f_{n_o}|) + 1;//Round-robin selection
13               Select the I-th flow in the flow table and encode the header as Table. 6.1;
14               Add the header to packet P and P = {P, P};
15           end
16       end
17   end
```

Fig. 6.5: The AB-MAC algorithm at node n_o.

6.1.4 The Pseudo-Code of AB-MAC

Fig. 6.5 describes the AB-MAC algorithm at node n_o, which runs at the end of each slot and computes the content of the ready-to-transmit queue $\mathbf{P}$ for the following slot. If $\mathbf{P}$ is not empty, n_o will encapsulate all the packets in $\mathbf{P}$ and transmit at the next slot. The first step is to update the three sets $\boldsymbol{n}_{\mathbb{I}}(n_o)$, $\boldsymbol{n}_{\mathbb{A}}(n_o)$ and $\boldsymbol{n}_{\mathbb{N}}(n_o)$ and the flow table based on the received data packets or ACK/NAKs. p_{n_o} is then updated using the ARC-AIMD algorithm.

In Section 6.1.1 we explained that in multi-hop flows, a node should not transmit in two consecutive slots so that it can accomplish two things 1) receive updated local information of its neighbors and 2) give a chance for next hop to transmit to accomplish multi-hop forwarding. Thus, if n_o does not transmit at the current slot, it can transmit in the next slot as shown by the first condition of line 4. The purpose of the second condition is to alleviate the 1st-hop collisions. A 1st-hop collision occurs because the intended receiver is transmitting in the same slot. For example, if both n_o and its intended receiver n_r transmits at the odd-indexed slots, n_r can never decode the packets from n_o. We alleviate this problem by flipping a coin when a 1st-hop collision is detected at one of n_o's intended receivers. If $|\boldsymbol{n}_{\mathbb{I}}(n_o) \setminus (\boldsymbol{n}_{\mathbb{A}}(n_o) \cup \boldsymbol{n}_{\mathbb{N}}(n_o))| > 0$ holds (*i.e.,* 1st-hop collision occurs), n_o generates a normally distributed value x between 0 and 1. If $x < 0.5$, n_o can transmit in two consecutive slots, which indeed swaps n_o's transmitting slot from odd to even or from even to odd.

The transmission quota Γ_{n_o} is computed by Eq. 6.5 and distributed according to

TQD. The ready-to-transmit queue $\mathbf{P}$ is initially empty. To implement the round-robin flow selection, the last parameter of AB-MAC I points to the flow (in the flow table) to be included in the header. I is set initially to 0 and updated at line 12. n_o selects the data packets as assigned by TQD and encodes the status of one flow (indicated by I) in the header of each packet. The finalized packet is added into $\mathbf{P}$.

Because AB-MAC combines ARC-AIMD and TQD and ARC-AIMD only needs constant time, the time complexity of AB-MAC is equivalent to TQD, which is $O(K_T|\boldsymbol{f}_{n_o}|)$.

6.2 Simulations

We present the results of simulations performed on two topologies and three schemes. The two topologies show how AB-MAC functions in networks with various densities and the three coordination access schemes exhibit how different levels of coordination affect the performance of AB-MAC. The first topology has 18 nodes with three flows traversing the network while the second topology has 100 nodes with various numbers of flows. The performance of AB-MAC is compared against IEEE 802.11 DCF [23] (both with and without RTS/CTS) in the three schemes, namely, *the mini-slot scheme*, *the random scheme* and *the prioritized scheme*.

6.2.1 The Three Schemes

- **The mini-slot scheme**

  We first introduce *the mini-slot scheme*. With the ready-to-transmit queue $\mathbf{P}$ computed by the AB-MAC algorithm, a node will contend for channel access at the beginning of a slot if $\mathbf{P}$ is not empty. We name the node with non-empty $\mathbf{P}$ a *contending node*. The contending nodes may not start transmitting at exactly the same time instant, due to the clock drift existing in distributed environments. For example, two neighboring nodes n_0 and n_1 are both contending nodes at the same slot. n_0 starts transmitting slightly earlier than n_1 due to the different clock drifts occurs at n_0 and n_1. If the gap between the transmitting time of n_0 and n_1 is large enough, n_1 will switch to receiving mode once it senses the signal from n_0 and will not transmit at the current slot. However, if the gap is relatively small, n_1 will transmit before it decodes the signal from n_0. Thus, n_0 and n_1 will transmit at the same slot. In conventional networks, multiple transmissions starting at the same slot will collide at the receiver if they are in the transmission range of the receiver. In MPT/MPR networks,

however, multiple concurrent transmissions do not necessarily lead to collision if the receiver can handle all the arriving packets. In order to simulate the channel access contending process in MPT/MPR wireless networks, we propose *the mini-slot scheme* in which C_m mini-slots are located at the beginning of a slot.

When starting transmitting, each contending node randomly chooses an integer from 1 to C_m to identify which mini-slot it selects to transmit. If a node (*e.g.,* n_0) starts transmitting at the i-th mini-slot ($1 \leq i \leq C_m$), the neighbor of n_0 (*e.g.,* n_1) switches to receiving mode if 1) n_1 is not a contending node or 2) n_1 chooses a mini-slot j which is greater than i. However, if multiple contending nodes choose the same mini-slot, they will transmit simultaneously at the current slot. Also, the concurrent transmissions may or may not cause collision at a receiver, depending on the number of arriving packets and the MPR capacity limit of the receiver. We assume that when C_m is relatively small, which is set to 4 in the following simulations, the contention period is negligible compared to the packet transmission time.

The mini-slot scheme mimics the channel contending procedure and is applied to show the performance of AB-MAC when nodes lack coordination beyond honoring slot boundaries.

- **The random scheme**

  Because only the nodes with non-empty **P** will contend to transmit and **P** is decided by the value of AP and the differential backlog in AB-MAC, the total number of contending nodes has been limited by the AB-MAC algorithm running at each node before the contention starts. Ideally, assume the mini-slot scheme can solve the channel access problem within 1-hop distance for these limited contending nodes. *i.e.,* two connected contending nodes will not choose the same mini-slot and hence will not transmit at the same slot. Thus, we consider a simulation scenario where the 1st-hop collision will not occur because the intended receiver will not transmit at the same slot as the transmitter. This simulation scenario is named *the random scheme* which is proposed to evaluate the performance of AB-MAC if the 1st-hop collisions could be resolved.

  It is worth noting that the random scheme is not necessarily a benefit for MPT/MPR networks because it can stop connected nodes from transmitting simultaneously even if the concurrent transmissions can be accommodated by the intended receiver of the transmitters. For example, if three nodes n_0, n_1 and n_2 are contending nodes and are mutually connected. In the random scheme,

if n_0 transmits, n_1 and n_2 will both switch to receiving mode and hence no 1st-hop collision will happen. However, if both n_0 and n_1 need to transmit to n_2, in the random scheme n_0 and n_1 cannot transmit simultaneously even if n_2 can handle the packets from both n_0 and n_1. Thus, the random scheme may waste the MPR capability at the benefit of avoiding 1st-hop collisions.

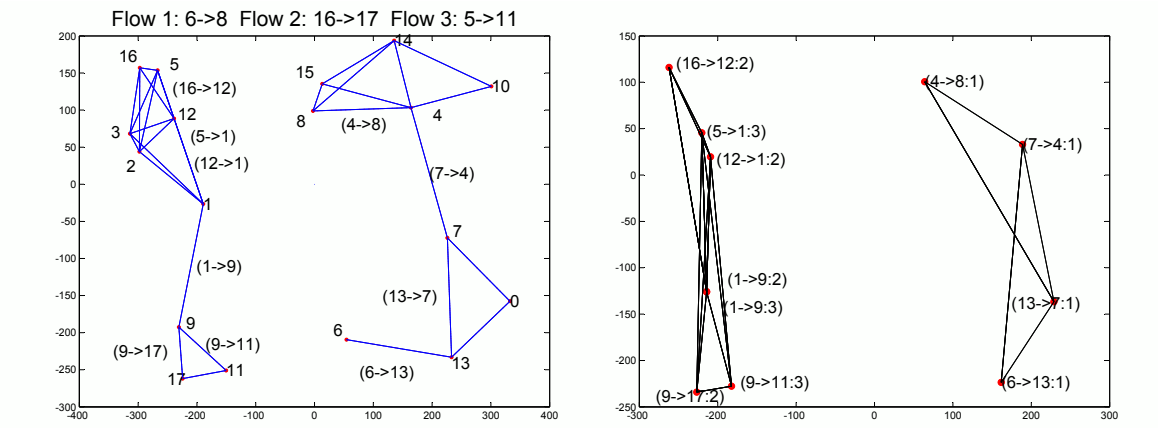
- **The prioritized scheme**
  If further node coordination is possible during channel access, the performance of AB-MAC can be further improved. All the contending nodes have non-empty ready-to-transmit queue $\mathbf{P}$. However, the total packets stored in $\mathbf{P}$ (*i.e.,* the length of $\mathbf{P}$) can be various in different contending nodes. If the channel is won by a node with small $\mathbf{P}$, the channel resource may be wasted because this node does not have enough packets in $\mathbf{P}$. Ideally, if the channel contention procedure prioritizes the contending nodes with large $\mathbf{P}$, it can maximize the overall channel utilization efficiency and hence improve the system performance. Therefore, we propose *the prioritized scheme* to evaluate the performance of AB-MAC if nodes can coordinate to prioritize the ones with large $\mathbf{P}$ when contending for channel access.

  Specifically, in the prioritized scheme, we consider a central station with the knowledge of the length of $\mathbf{P}$ for all the contending nodes. At the beginning of a slot, nodes start to transmit in descending order of the length of $\mathbf{P}$ (ties broken arbitrarily). When a node occupies the channel, all its neighbors (within 1-hop distance) switch to receiving mode and do not transmit at the slot. Comparing to the random scheme, the prioritized scheme has higher coordination level because it guarantees that, in a neighborhood, the channel is always won by the contending node with the largest $\mathbf{P}$.

These three schemes exhibit increasing level of node coordination. We will show the performance of AB-MAC in the three schemes next.

6.2.2 An 18-Nodes Topology

Fig. 6.6(a) shows the topology of 18 random nodes and three flows with their paths depicted in the caption. The interference range is set to be equal to the transmission range, which is 200 distance units. Two nodes within transmission range are in one-hop distance of each other and are connected by a solid line in Fig. 6.6(a). We assume that a transmission from a node outside the transmission range will not impact the reception. Therefore, a packet is successfully received by the receiver if two conditions



(a) Flow 1 ($6 \Rightarrow 8$: $6 \to 13 \to 7 \to 4 \to 8$); flow 2 ($16 \Rightarrow 17$: $16 \to 12 \to 1 \to 9 \to 17$); flow 3 ($5 \Rightarrow 11$: $5 \to 1 \to 9 \to 11$).

(b) The contention graph.

Fig. 6.6: The 18-node topology graph and the contention graph.

are satisfied: 1) the transmitter is within transmission range of the receiver; 2) the number of concurrent packets (sent from the transmitters in transmission range) at the receiver does not exceed K_R. Fig. 6.6(b) shows the contention graph. In this section, we study the performance of this 18-node example in the mini-slot scheme, which is the worst case scenario because there is no coordination during channel access contention.

First, we compare the aggregate end-to-end throughput of AB-MAC against that of IEEE 802.11b distributed coordination function (DCF) [23]. In the simulations, DCF is implemented based on the *WiFi* model of the ns-3 simulator [1] with the link rate (denoted by P) set to 1Mbps. In order to present a clear comparison, the PHY layer is simplified by assuming zero propagation delay and the physical layer convergence protocol (PLCP) header and preamble overheads are omitted. The data packets (including the headers of the higher-layer control protocols and the payload) for both AB-MAC and DCF are set to 1500 bytes. The length of a slot is equal to $1500 \times 8 \times 10^6/P$ seconds. The end-to-end throughput of the simulation is measured in packets/slot. However, in order to compare the performance of the protocols under different link rates, the throughput is converted to Mbps in several charts we are about to see.

In Fig. 6.7, AB-MAC is simulated on MPT/MPR channels where $K_T = 5$ and K_R varies from 5, 6, ... ,10 to 1000. In Section 5.3.1, we have seen that increasing K_T to a value greater than K_R cannot improve the end-to-end throughput of a multi-hop flow. Therefore, we do not consider the case when $K_T > K_R$. Since DCF is running under conventional channels where $K_T = K_R = 1$, in order to present a

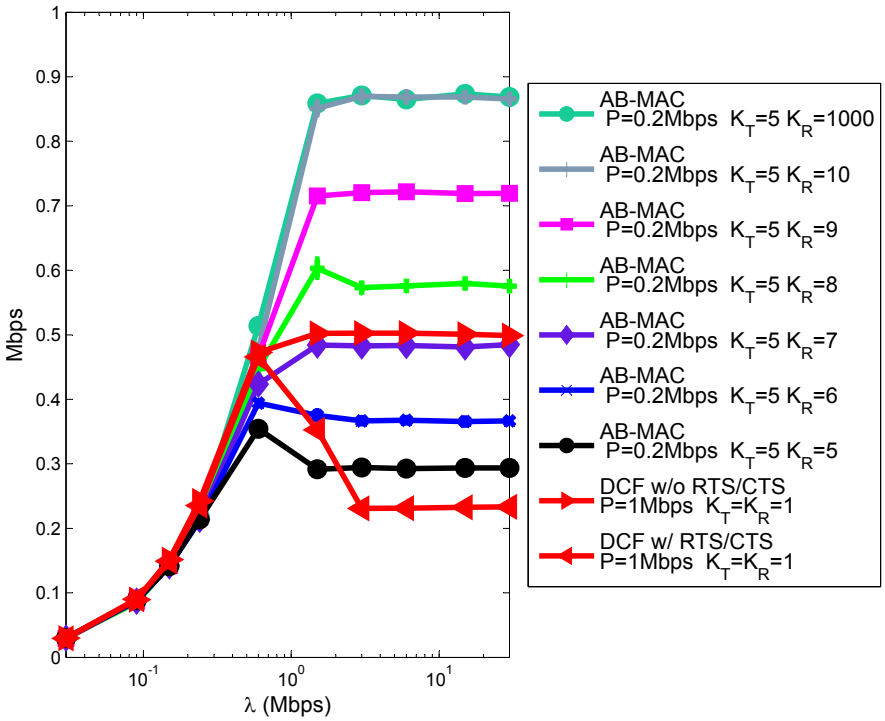


Fig. 6.7: The aggregate throughput of AB-MAC and DCF.

reasonable comparison, the link rate of AB-MAC is set to 0.2Mpbs, which is $1/K_T$ of the link rate of DCF. This configuration of DCF can also be considered as an idealized (*i.e.,* the overhead caused by the cooperation among different channels (Section 2.1.4) is free) multi-channel example which consists of K_T separate channels each with the same speed as the MPT/MPR channel of AB-MAC. Since the request-to-send and clear-to-send (RTS/CTS) handshake mechanism is an option for DCF, we run DCF both with and without the RTS/CTS mechanism by configuring the *RtsCtsThreshold* parameter of ns-3.

Fig. 6.7 shows the aggregate throughput of the three flows described in Fig. 6.6(a). Every flow has the same arrival rate at the source and the aggregate arrival rate λ is shown by the x-axis. Every data point in Fig. 6.7 is the mean of 10 runs lasting 10 seconds. The error bars are the standard deviations and are smaller than the line markers in the figure. Note that $K_R = 1000$ is a means to express "infinite" receiving capacity. Several observations can be made based on Fig. 6.7.

- Predictably, the aggregate throughput of AB-MAC is increasing with the value of K_R. After K_R is 10 (twice the value of K_T), increasing K_R cannot further improve the throughput. The reason was earlier elaborated in Section 5.3.1. Observing Fig. 6.6(a), over all the active nodes, the maximum number of con-

current inbound arcs is two, which implies that when $K_R > 2K_T$, the bottleneck of the system shifts to the transmitting side, *i.e.,* the value of K_T.

- The throughput of AB-MAC stablizes at the maximal level after the system is saturated (when λ is about 1Mbps). However, if RTS/CTS is applied, the throughput of DCF collapses under heavy congestion. This disparity can be attributed to the asymmetric information problem and the intra-flow contention problem we discussed earlier. We take the behavior of flow 1 as an example. In Fig. 6.6(a), the packets of flow 1 are stalled at node 13. The reason is that, in the channel access competition between node 6 and node 13, node 6 has an advantage because node 6 is the source and has fewer competitors.

  Node 6 can win the channel when both node 7 and node 13 are not transmitting because in this case node 6's RTS request will be replied to by node 13. On the other hand, node 13 can win the channel when node 6, 7 and 4 are not transmitting because node 13's RTS can be responded to by node 7 only when node 4 is not transmitting. As we explained in Section 2.3.2, because of the asymmetric information problem, whoever wins the channel first can monopolize the channel. In this example, node 7 will finish transmitting before node 4 since node 4 has nothing to transmit before it receives the packet from node 7. Thus, in the contention between node 6 and node 13, node 6 always gets the CTS from node 7 first. At the moment node 13 loses the channel to node 6, node 6's CW will be set to the minimum value while node 13 will have the maximum CW. The only chance node 13 can win the channel is when it sends RTS during the small interval after node 6 ends it transmission and before it sends another RTS.

  Fig. 6.8 shows the end-to-end throughput of flow 1. Three cases are compared as listed in the legend of the figure. If RTS/CTS is applied, the throughput of flow 1 (the curve with circle markers) dramatically drops after the system is saturated. However, the throughput can stabilize at the maximum level if RTS/CTS is not applied.

  As shown in Fig. 6.7, if RTS/CTS is not applied, DCF can reach an aggregate throughput similar to that of AB-MAC when $K_T = 5$ and $K_R = 7$. However, as we will see in the next example, the performance of DCF (either with or without RTS/CTS) is closely related to the density of the network, while AB-MAC can be adaptive to various density levels.

- Furthermore, since the flow with zero or negative differential backlog cannot

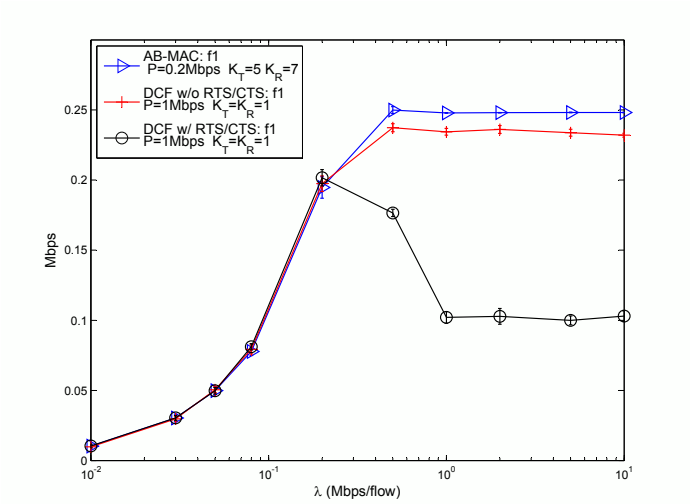


Fig. 6.8: The end-to-end throughput of flow 1.

transmit, we observed in the simulations that the backlog of a flow will monotonically decrease from the source to the destination. Therefore, if the buffer of arrival packets at the source of a flow is limited by a threshold, all the buffers along the path of the flow are limited as well. In other words, the queueing system exhibits stable queues by controlling the source of each flow.

Fig. 6.9 compares the aggregate throughput between AB-MAC and LEX (Fig. 5.3), which approximates the max-min rate allocation. The discrepancies between the observed values of AB-MAC and LEX can be attributed to: 1) AB-MAC is a distributed algorithm while LEX assumes global knowledge and applies a centralized computation and 2) similar to 1-hop S-ALOHA* (Section 3.2), AB-MAC is a probabilistic approach. Even with the optimal AP known to all the nodes, the aggregate throughput S-ALOHA* can achieve about 60% of the channel capacity. According to Fig. 6.9, the gap between the curves of AB-MAC and LEX is monotonically decreasing as K_R increases, which coincides with the observation in [74, 84].

Fig. 6.10 plots the performance of AB-MAC with bursty traffic, which is implemented by a binary source model same as depicted in Fig. 3.6. Fig. 6.10 shows that AB-MAC is adaptive to bursty traffic.

6.2.3 A 100-Node Topology

In this section, we compare the performance of AB-MAC and DCF in a larger network where 100 nodes are randomly located in a 1600 by 1600 square. The transmission range is 200 distance units. The topology graph is depicted in Fig. 6.11. We run simulations of three schemes. Fig. 6.12 (mini-slot scheme), Fig. 6.13 (random scheme)

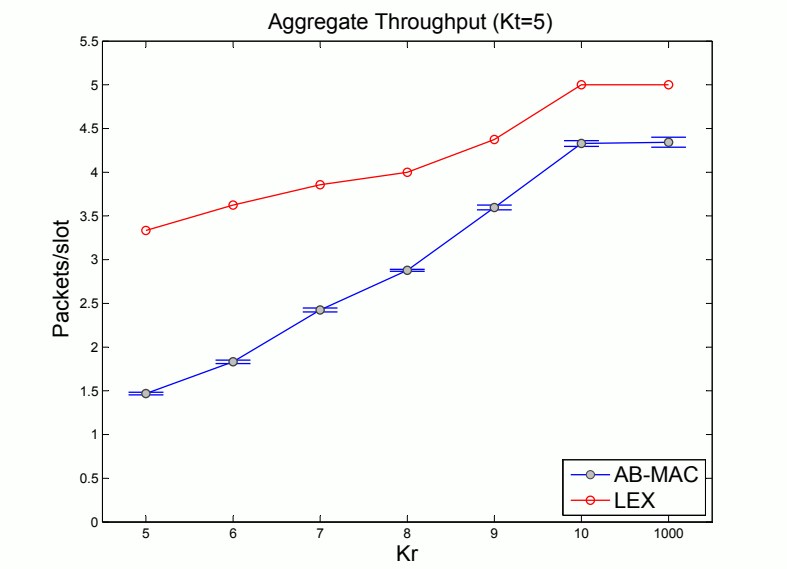


Fig. 6.9: The aggregate throughput of AB-MAC and LEX.

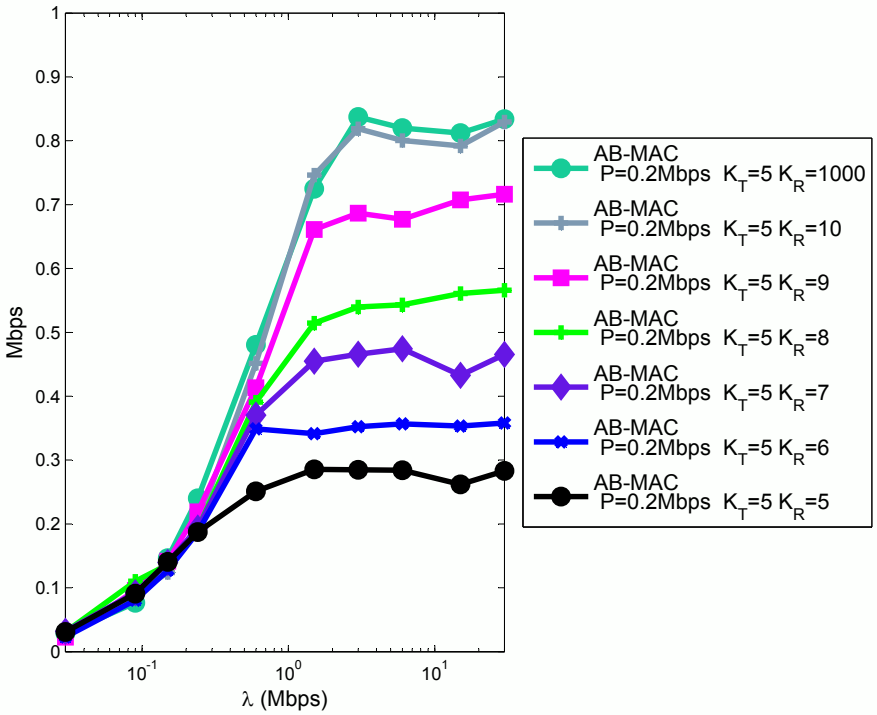


Fig. 6.10: The aggregate throughput of AB-MAC with bursty traffic.

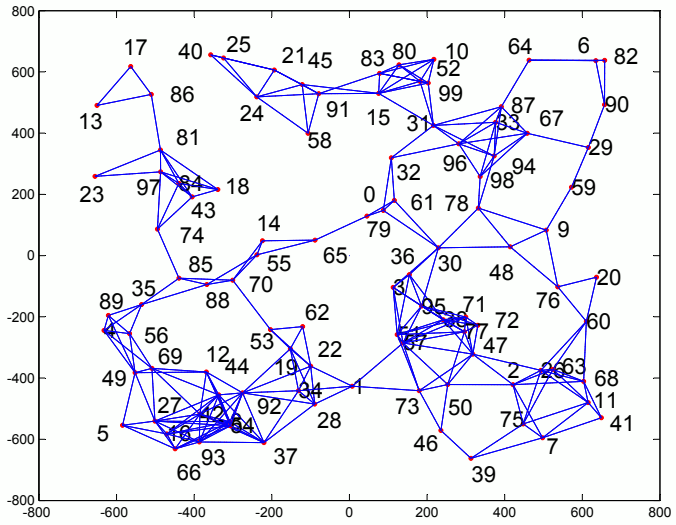
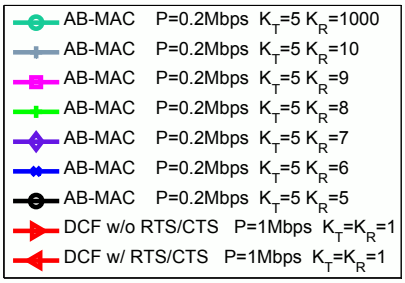


Fig. 6.11: The 100 node random topology.

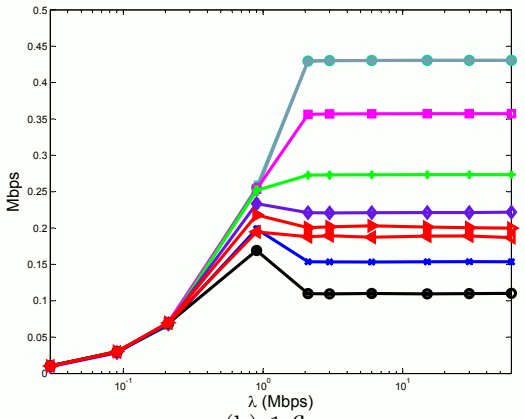
and Fig. 6.14 (prioritized scheme) illustrate the aggregate throughput of 1, 2, 10 and 20 flows in the network depicted in Fig. 6.11. These examples will show that while DCF cannot resolve collisions in dense networks, AB-MAC is adaptive to a network with different number of flows.

Fig. 6.12(b), Fig. 6.13(b) and Fig. 6.14(b) show the throughput of one flow from node 5 to node 6 following a predefined shortest path in Fig. 6.11. The end-to-end throughput of the single flow shows similar performance in the three schemes because four mini-slots are enough to resolve the channel access problem among the limited (by AB-MAC) contending nodes. The curves of DCF (either with or without RTS/CTS) are located between the curves of $K_R = 6$ and $K_R = 7$ of AB-MAC. Note that the link rate of AB-MAC is 1/5th that of DCF. This result shows that, in a network with single flow, while the normalized (by the channel capacity) performances are similar between AB-MAC and DCF, the aggregate throughput can be improved by increasing the MPR capacity limit when applying AB-MAC.

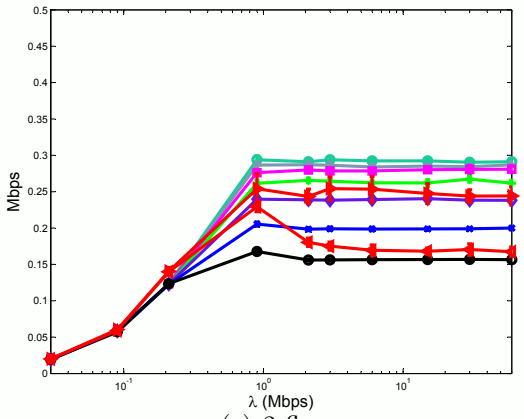
If we add one more flow from node 1 to 13, the performance is plotted in Fig. 6.12(c), Fig. 6.13(c) and Fig. 6.14(c). In the mini-slot and random schemes, the aggregate throughput decreases compared to the single-flow example because the two flows traverse the same bottleneck maximal clique where the channel needs to be shared. If RTS/CTS is not applied, DCF's performance is close to AB-MAC with $K_R = 6$ and $K_R = 7$ in Fig. 6.13(c) and Fig. 6.12(c), respectively. On the other hand, if RTS/CTS is applied, DCF's performance is similar to AB-MAC with $K_T = K_R = 5$. However, in the prioritized scheme, the aggregate throughput of



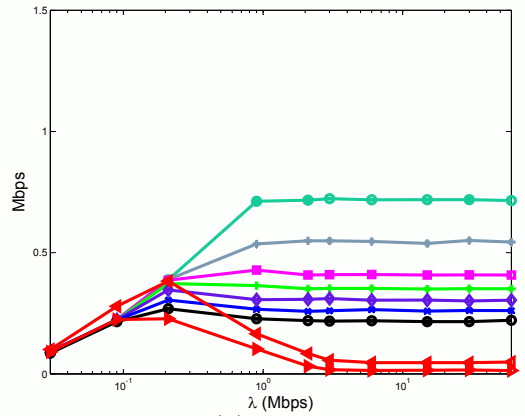
(a) Legend



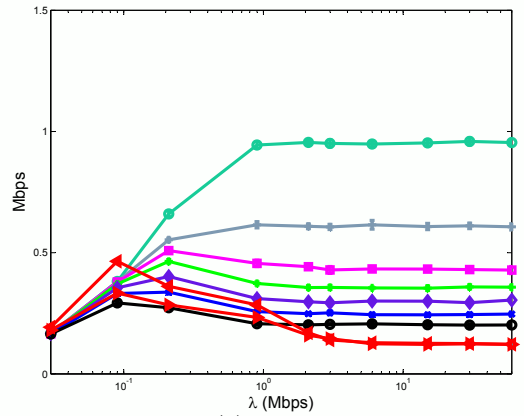
(b) 1 flow



(c) 2 flows

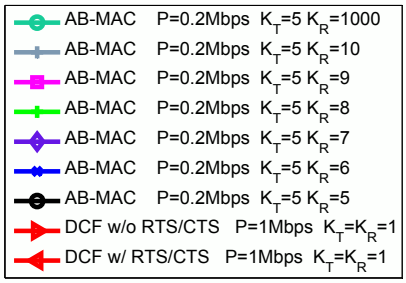


(d) 10 flows

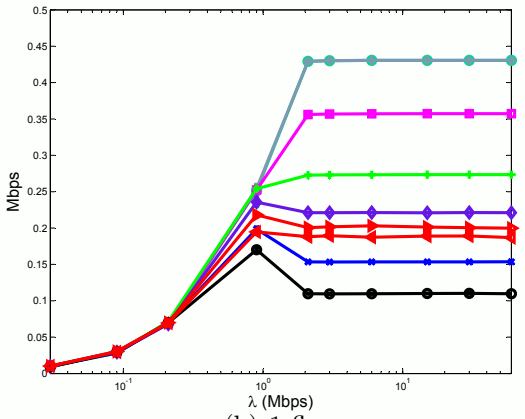


(e) 20 flows

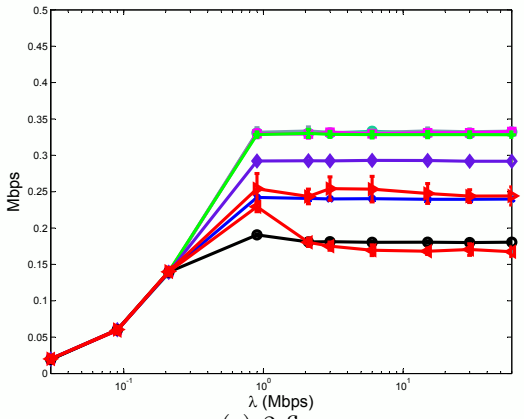
Fig. 6.12: The aggregate throughput for various flows in the mini-slot scheme.



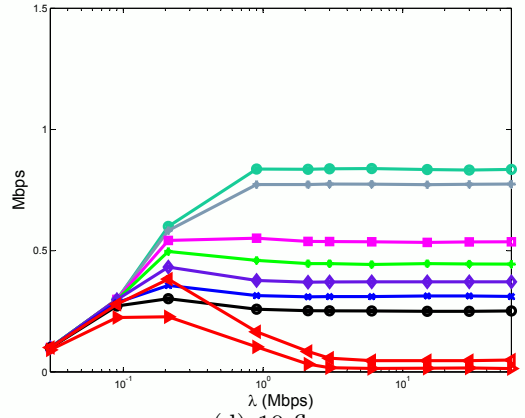
(a) Legend



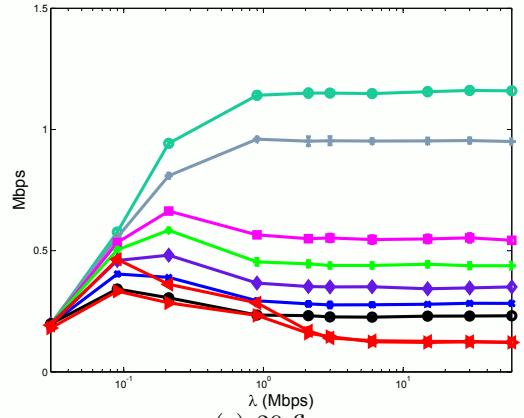
(b) 1 flow



(c) 2 flows

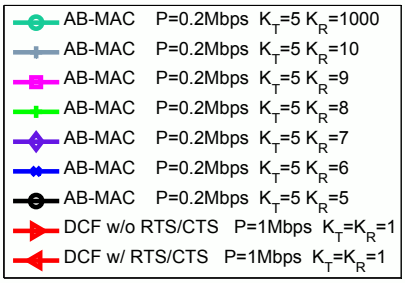


(d) 10 flows

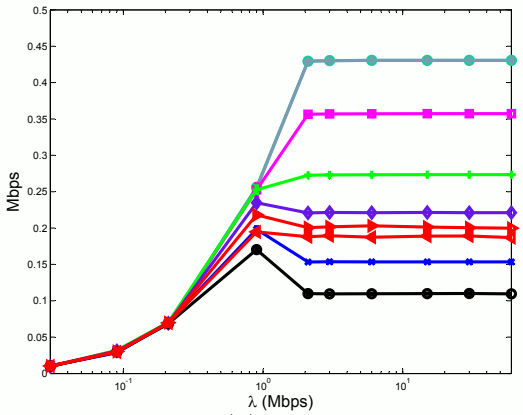


(e) 20 flows

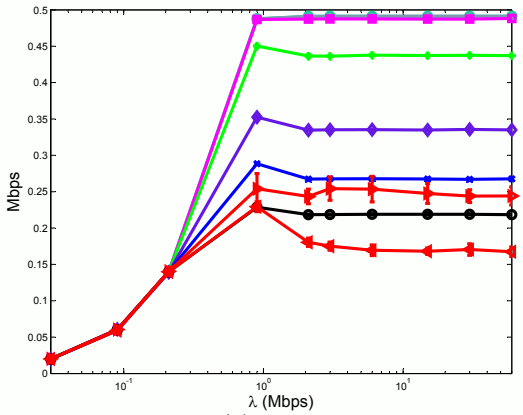
Fig. 6.13: The aggregate throughput for various flows in the random scheme.



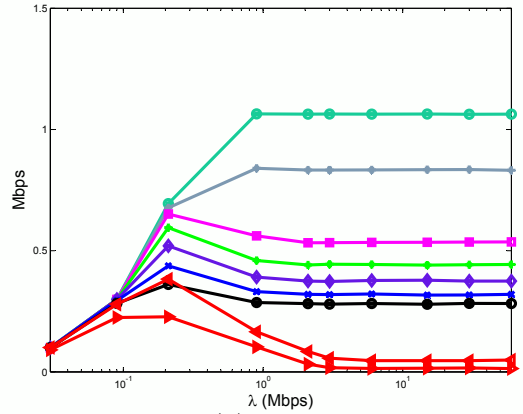
(a) Legend



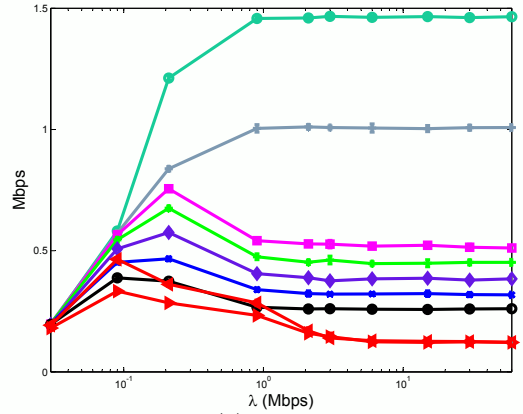
(b) 1 flow



(c) 2 flows



(d) 10 flows



(e) 20 flows

Fig. 6.14: The aggregate throughput for various flows in the prioritized scheme.

the two flows is largely increased. Note that in the prioritized scheme, AB-MAC prioritizes the nodes with large ready-to-transmit queue. Thus, the results show that, in the 2-flow example, coordinating nodes during channel access according to their ready-to-transmit queue can greatly increase the channel efficiency. We will see that in more dense networks, contending nodes coordination has a more significant impact on the overall performance.

As the number of flows increases to 10 or 20 (Fig. 6.12(d), Fig. 6.12(e), Fig. 6.13(d), Fig. 6.12(d), Fig. 6.14(d) and Fig. 6.14(e)), the aggregate throughput of AB-MAC largely increases compared to the single-flow and two-flow examples. The performance upgrades significantly because those sparsely located flows can benefit from the spatial reuse of the bandwidth (*i.e.,* nodes sufficiently far apart can transmit concurrently) and hence result in higher aggregate throughput.

The examples with 10 or 20 flows show that the performance of AB-MAC increases with the nodes coordination level. In the mini-slot scheme (Fig. 6.12), the contending nodes do not coordinate during channel access and AB-MAC presents the lowest performance of the three schemes. Specially when the traffic is light, the DCF with RTS/CTS outperforms AB-MAC in Fig. 6.12(d) and Fig. 6.12(e) even when K_R is unlimited. When the MPR capacity is not a constraint, concurrent transmissions will not collide because the number of arriving packets is too large. Thus, the performance is mostly affected by the 1st-hop collision in which the intended receiver is transmitting at the same slot as the transmitter. Because the neighboring contending nodes do not coordinate in the mini-slot scheme, in dense networks, the failed transmissions caused by the 1st-hop collisions significantly affect the aggregate throughput and degrade the performance of AB-MAC under light traffic. However, when the traffic is heavy, the overall performance is dominated by the collisions caused by the limited MPR capacity. Thus, AB-MAC can still outperform DCF, which collapses in saturated dense networks.

Nonetheless, if the 1st-hop collisions are eliminated by coordination, we can see that in the random scheme, even under light traffic AB-MAC can outperform DCF. Also, the aggregate throughput of AB-MAC is increased compared to the mini-slot scheme. If further coordination is allowed to prioritize the heavily loaded contending nodes, the overall performance can be greatly improved as we can see in the prioritized scheme.

Fig. 6.12, Fig. 6.13 and Fig. 6.14 illustrate the performance of AB-MAC in both sparse and dense networks under both light and heavy traffic load. It shows that AB-MAC significantly alleviates the inherent problem in CSMA/CA such as the asymmetric information problem and the intra-flow problem. The experiments also

show that by coordinating contending nodes during channel access, we can largely improve the performance of AB-MAC. In the mini-slot scheme, contending nodes do not coordinate and AB-MAC exhibits the lowest performance which is outperformed by DCF in dense networks under light traffic. However, when the 1st-hop collisions are resolved in the random scheme, AB-MAC can outperform DCF in any scenario. In the prioritized scheme, the channel efficiency is largely improved by prioritizing the nodes with large ready-to-transmit queue. This scheme shows another coordination direction which can increase the performance of AB-MAC.

6.3 Summary

In this chapter, we propose three schemes using AB-MAC to regulate transmissions of the multi-hop flows in MPT/MPR wireless networks. The AB-MAC algorithm consists of three major components: 1) exchange of local information, 2) computation of local access probabilities and 3) distribution of transmission quota across the flows. Applying the access probability to handle the MAC layer mitigates several fundamental problems caused by the RTS/CTS handshake method and can conveniently compute the transmission quota for each node. AIMD and BP strategies compensate for each other because, while AIMD can take care the MAC layer in a distributed environment, the BP method can alleviate the intra-flow contention and prevent the upstream nodes from monopolizing the channel. To complete the AB-MAC, a coordination scheme is additionally needed, for which three alternatives are considered. The three schemes include the mini-slot scheme, the random scheme and the prioritized scheme which exhibit increasing level of node coordination during channel access. Simulation results show that while DCF cannot resolve collisions in dense networks, AB-MAC is adaptive in various scenarios. Generally, increasing the value of K_R can improve the overall performance of AB-MAC. While the mini-slot scheme mimics the channel contention procedure, the random scheme and the prioritized scheme show different directions in which the performance of AB-MAC can be further improved.

Chapter 7

Conclusion and Future Directions

In this thesis, we investigate MPT/MPR wireless networks and design algorithms to fairly and efficiently regulate the transmissions of multi-hop flows. In this chapter, we summarize the contributions in both single-hop and multi-hop scenarios and present future directions.

7.1 Contributions

7.1.1 Single-hop scenario

For the single-hop scenario, we design the AIMD-MAC algorithm which can be run in a distributed environment to manage uncoordinated nodes accessing a shared K-MPR channel. The basic characteristics of AIMD-MAC are:

- **Relaxation of the requirements for feedback**
  The basic AIMD algorithm [22] is known to be able to achieve max-min fairness assuming every node can obtain a uniform and error-less binary feedback at every slot. However, the strict requirement for the feedback from the central station can be difficult to satisfy in distributed networks. AIMD-MAC extends the basic AIMD algorithm by introducing another parameter, the update cycle UC, with which the nodes do not need feedback at every slot.

- **Introduction of the transmission history**
  Without the immediate feedback at every slot, the binary feedback collected at every UC needs to be processed before it can be used to decide whether to increase or decrease the access probability (AP). The transmission history is therefore introduced to interpret the collected information. The transmission history consists of the success ratio and the utilization ratio. Applying the success ratio helps the nodes to have uniform operations (*i.e.,* either additively

increase or multiplicatively decrease the AP) at every update cycle, which is helpful towards achieving the max-min fairness. Incorporating the utilization ratio further reduces the delay under light traffic load.

- **Adaptivity to dynamic environments**
  Simulations show that AIMD-MAC outperforms S-ALOHA* under light traffic load because the packets have a shorter average delay and achieves the same performance as S-ALOHA* when the system is saturated. By tuning the length of the update cycle, we can adjust the distribution of the AP. With an appropriate setting of parameters, AIMD-MAC can be applied in a dynamic environment where the number of nodes and the channel capacity are not constant. Extensive simulations show that AIMD-MAC achieves at least 90% of the performance of S-ALOHA* over a wide range of scenarios without the need to adjust the parameters.

7.1.2 Multi-hop scenario

We first present a centralized study to evaluate the performance of the multi-hop MPT/MPR wireless networks and then develop a MAC algorithm to regulate multi-hop flows traversing the network. The steps towards the goal are:

- **Max-min fair in conventional multi-hop wireless networks**
  Huang *et al.* [42] suggest the analogy between a link in wired networks and a maximal clique (MC) in wireless networks and find the max-min rate allocation for single-hop flows in wireless networks under conventional channels. We further extend the max-min fairness to multi-hop wireless networks. Based on the proof (Theorem 4.1) of the necessary and sufficient condition for a rate allocation to be max-min fair, we propose the WF algorithm to compute the max-min rate allocation for multi-hop flows traversing a network with $K_T = K_R = 1$. WF repeatedly solves a LP problem (*i.e.,* LP_{WF}) to determine the rates with which the flows become bottlenecked and terminates when all the flows are bottlenecked, which is the sufficient condition for a rate allocation to be max-min fair.

- **Schedulability of the max-min allocation computed by WF**
  We prove that if the contention graph of the network is a perfect graph, there exists a discrete schedule to realize the max-min rate allocation computed by the WF algorithm. Additionally, we demonstrate that the minimum number of slots needed to realize the max-min rate allocation can be computed as a

by-product the WF algorithm. Since a schedule for the case $K_T = K_R = 1$ can be directly applied to the cases $K_T = K_R = K$ ($K > 1$) by scaling by a factor of K, we can compute a feasible schedule for the network with $K_T = K_R = K$ by first computing the max-min allocation when $K_T = K_R = 1$ and then scaling the rate allocation by a factor of K.

- **Challenges for the max-min allocation when $K_T \geq 1,\ K_R \geq 1$**
  When $K_T \geq 1,\ K_R \geq 1$, it is not trivial to determine the necessary and sufficient condition for a MC to be saturated, which is a substantial obstacle to extend the WF algorithm to MPT/MPR networks. The fundamental reason behind these challenges is that in MPT/MPR wireless networks, even arcs belonging to the same MC can be active simultaneously, which contradicts the primary attributes of the MC in a conventional channel model. We elaborate these challenges via examples and demonstrate why the problem cannot be solved as a LP problem.

- **Introduction of the eight pair-wise interference patterns**
  In order to tackle the max-min problem in MPT/MPR wireless networks, we group the non-commutative interference relationships of the arcs into eight patterns. Each pattern describes how one arc interferes with another arc. Based on the eight pair-wise interference patterns, we can determine the normalizing coefficients at the transmitter and receiver of every arc, which can be applied to "eliminate" the effect of the MPT/MPR capacity (*i.e.,* by multiplying the arc rates with the normalizing coefficients) when determining the saturation status of a MC. With the saturation constraints, WF-ASYM is developed to compute an approximated max-min rate allocation in MPT/MPR networks.

- **Construction of discrete schedules for MPT/MPR wireless networks**
  We present the MDSATUR algorithm to compute a discrete schedule to realize a given rate allocation. Since a MC of a MPT/MPR system is distinct from its traditional definition in that the arcs belonging to the same MC can be active simultaneously (*i.e.,* the M property), the information from the contention graph alone is not sufficient to decide the next slot to be allocated to a given arc. Thus, MDSATUR extracts the information from both contention graph and flow graph and generates a collision free schedule which can fully utilize the MPT/MPR capacity.

  In addition to max-min fairness, another fairness metric — the lexicographical optimality — is applied to evaluate the performance of the MPT/MPR

networks. The LEX scheme, which combines WF and MDSATUR, outputs a discrete schedule with which the flows can achieve an approximate max-min fairness.

- **Study observation of the relative significance of K_T and K_R**
  We run WF, WF-ASYM and LEX on a random example and show how the system performance depends on the value of K_T and K_R separately. The broadcast nature of the wireless network usually imposes a heavier burden on the receiving side of the nodes. As demonstrated by Fig. 4.2, a relay may need to receive more packets that those it needs to transmit to keep a balanced inbound and outbound traffic. Therefore, applying a K_R greater than K_T can improve the end-to-end throughput as long as K_T has not become a bottleneck. On the other hand, increasing K_T to a value greater than K_R does not benefit the system performance because the receiver cannot handle more than K_R packets at a time.

- **Design of MAC protocol for MPT/MPR networks**
  For the distributed MPT/MPR wireless networks we propose the AB-MAC algorithm, which combines the idea of the AIMD-MAC and the backpressure methods, to organize multi-hop flows traversing a network. A well studied scheduling algorithm in multi-hop networks is the backpressure algorithm. The backpressure algorithm is a centralized algorithm which consider the MAC layer as a black-box, *i.e.,* assuming all the feasible schedules are known beforehand. AB-MAC combines the backpressure algorithm with the AIMD method to manage the MAC layer in a distributed environment. Specifically, it applies the ARC-AIMD to regulate the access probabilities. This combination avoids the hidden terminal problem and the asymmetric information problem inherited in RTS/CTS-based algorithms and eliminates the need to solve the max-weight problem. To complete AB-MAC, we introduce three schemes to evaluate how different levels of node coordination affect the performance of AB-MAC. The three coordination schemes include the mini-slot scheme, the random scheme and the prioritized scheme. Simulations show that regulating nodes during channel access can further improve the performance of AB-MAC.

7.2 Other Efforts

As in much scientific research, not all the experiments and efforts turn into exciting results. During our study of the MPT/MPR networks, we have experienced

many unsuccessful trials before we finally discovered the distinct characteristics of the MPT/MPR network and developed algorithms for it. In this section, we summarize some of the "failures" we encountered which highlight challenges and might be useful for future research.

- **Analysis of AIMD-MAC**
  The AIMD-MAC algorithm achieves the max-min fairness in K-MPR networks by adaptively adjusting the access probability (AP) of the nodes. In Section 3.5, we demonstrate the oscillatory behavior of AIMD-MAC and the relationship between the update cycle (UC) and the distribution of the APs, which explains why AIMD-MAC can achieve the optimal performance in dynamic environments without adjusting the parameters. Since the AP plays a significant role in the overall performance of AIMD-MAC, we developed a Markov chain model to analyze it. Specifically, we divide the AP (a value between 0 and 1) into ten intervals: [0, 0.1), [0.1, 02), ..., [0.9, 1.0]. These ten intervals are the ten states of the Markov chain. For instance, the AP is in state i if it is in the sub-interval $[(i-1) \times 0.1,\ i \times 0.1)$ for $i = 1, 2, ..., 10$. We develop the transition probability matrix according to observations of how the AP develops at a single node. The Markov chain model approximately captures the distribution of the APs in many scenarios but is not precise enough to analyze the performance of AIMD-MAC. Extensive experiments based on Matlab tools indicate that Gamma distribution approximately fits the distribution of the APs but it is unclear how the parameters of the Gamma distribution depend on the network setup. Certainly, AP update interactions are more complex and the perspective of a single node does not sufficiently capture the joint dynamics of AP updates by a group of nodes, which, for example, would have required the development of a higher dimensional Markov chain. The extent to which a single node AP Markov chain is sufficient for modeling the performance of a network of nodes is currently unclear.

- **MPT/MPR Constraints**
  The maximal clique (MC) model captures competing arcs and the eight pairwise interference patterns illustrate how two arcs interfere with each other in an MPT/MPR setting. With this information we can restrict the arcs with MPT/MPR capacity limits and determine the saturation status of a MC. Based on the saturation status of a MC, we developed the WF-ASYM algorithm to approximate the max-min allocation. Also, with these interference patterns, we extend the DSATUR algorithm to the MDSATUR algorithm, which computes a

feasible schedule for MPT/MPR networks. However, the MC and the eight pairwise interference patterns were not the first method neither can it be assumed that it is the ultimate one.

Wang *et al.* [97] uses the circle packing method [56] to determine the maximum number of interfering links which can coexist in the interference range of the considered link. This method does not consider the relative positions among the links and hence can only provide an approximate upper-bound constraint with respect to the MPT/MPR capacity limits. We tried a more accurate method to take into account the location of the interfering arcs. Specifically, we constructed receiver-based constraints to limit the inbound and outbound traffic of the nodes according to the MPT/MPR limits. However, the receiver-based constraints do not tell which arcs are mutually interfering with each other (as MC does) because arcs are not only restricted by the receivers but also the transmitters. Additionally, it is unclear how to construct the necessary and sufficient conditions for a receiver when the receiver is also a transmitter for another arc.

- **Successive Interference Cancellation Receiver**
  While the interference outage receiving model (*i.e.,* K-MPR) converts the MPR capability into an integer which upper bounds the number of concurrent transmissions, there are other receiving models which closely reflect the characteristics of certain multi-user detection (MUD) techniques.

  The successive interference cancellation (SIC) technique is an effective MUD technique that has received intense research attention lately because of its promising performance improvements [95, 99] and more tolerable implementation complexity compared to other MUD techniques [5]. An SIC receiver iteratively decodes the received signals one by one. After each step, the original signal of the decoded message is estimated, reconstructed and removed from the composite signal. Thus, the next decoding iteration will suffer from less interference and have a greater chance to success. This procedure continues until either all the signals have been processed or the remaining signals cannot be decoded further. The most widely applied SIC model assumes synchronous packet transmissions where packets are sorted by the received signal powers in decreasing order [6, 39, 64, 98, 99]. The system applies a unique targeting SINR assuming all users transmit at the same data rate. It is widely accepted that the strongest signal should be decoded first because it is the one with the greatest SINR when all other signals are considered as interference. The signal is

considered decodable if its SINR is greater than or equal to the required SINR threshold.

We simulated the SIC technique in the PHY model of ns-3 [1]. In the SIC model, we consider asynchronous system where the packets can have different length and can be transmitted at any time instant. A SIC receiver can receive multiple packets simultaneously. Assume the length of a packet is contained in the preamble such that the receiver knows the ending time of each packet. At the end of each packet, the receiver performs the successive interference cancellation to decode the packet. A packet is considered successfully decoded if its SINR is greater than the threshold; otherwise it fails. Consider decoding packet i, the interference of packet i includes the packets whose duration overlaps with the duration of packet i. However, due to the SIC technique, those packets which have been successfully decoded before the ending time of packet i are removed from the composite signal and are not considered as interference. Experiments show that with the SIC receiver, the performance of ALOHA is about twice that of the basic receiver with respect to the overall throughput. However, since the SINR determines whether a packet can be decoded or not, the near-far problem significantly impacts the fairness of the system. Thus, the transmitting power of the nodes should be adjusted according to their relative distances to the receiver. Hence, it appears that considering SIC inescapably involves solving power control as a component of the overall transmission scheduling problem.

- **Deterministic MAC**

  AB-MAC regulates multi-hop flows in distributed MPT/MPR wireless networks. It uses the AIMD method to update the node's access probability and the backpressure approach to alleviate intra-flow contentions. Thus, AB-MAC is a probabilistic method which determines the transmitting packets limit by the value of the AP. Like most probabilistic algorithms, because transmissions are controlled random events, collisions can occur even if optimal APs are known to all nodes. So, in addition to AB-MAC, we also experimented with a deterministic method.

  We tried a receiver-driven control method since it is K_R at the receiver that determines the outcome of the reception of a bundle of packets, regardless of the transmitters of the packets. However, due to the broadcast nature of wireless networks, any transmission can be heard by any neighbors in interference range, which implies that any transmitter could have multiple receivers. In distributed environments, those receivers could send different commands. *i.e.,* one

receiver (*e.g.,* n1) may request the transmitter to send K_T packets while another receiver (*e.g.,* n2) may request the transmitter to send 0 packets. If this transmitter follows the command of n1, it can cause collision at $n2$; if it follows the command of n2, it wastes the transmission opportunity to n1. These problems dramatically worsen as the values of K_T and K_R increase. When the capacity limits are large, the potential difference of the commands is more severe. Thus, it is likely to cause more collisions or waste more resource.

On the other hand, in the AB-MAC protocol, every node has only one AP to decide the transmitting packets limit, regardless of the flows crossing the node or the potential receivers. We use the ARC-AIMD method to update the AP according to all the neighbors' recent reception status. A transmitter cautiously decreases its AP if any neighbor experienced a failed reception. AB-MAC greatly outperforms the deterministic method in extensive simulations. This point emphasizes that, in order to control deterministically the access in an MPR/MPT channel, a significant control and coordination protocol is required amongst competing nodes. Schemes such as AB-MAC, trade some conceivable loss of performance, with a probabilistic means of access which requires less control information to be exchanged. The extent to which coordination (in particular receiver-driven coordination) and random choices need to be synthesized, is currently unclear.

7.3 Future Directions

In this section, we discuss potential directions to extend our work.

- **Alternative reception models**
  In the previous section, we discussed the SIC technique. The challenges to form a feasible link set in an ad hoc network with SIC receivers are studied by Lv *et al.* [63]. From the receiver's point of view, potential interfering links should be organized such that the signals simultaneously transferred by these links can be successfully decoded at the SIC receiver. Adding a new link to the current feasible link set is complicated because the new link will not only affect the decoding of the desired signal (*i.e.,* direct interference) but also impact the detection of the interfering signals (the signal interfering the desired signal) which should be removed before processing the desired signal (*i.e.,* indirect interference). For instance, link L_1 is the desired link and L_2 is an interfering link which needs to be decoded and removed before processing L_1, a new link will not be admitted if

it either interferes with L_1 or L_2. Hence, novel MAC protocols based on the SIC receivers should involve higher level of coordination among not only neighboring nodes, but also nodes that are two-hop away. The exchanged information needs to include, for instance, the transmitting power and the distance between the transmitter and the receiver which together determine the receiving signal strength. With the knowledge of the receiving power from the potential concurrent transmissions, a MAC protocol can schedule multiple conversations while minimizing collisions. With the broadcast nature of the wireless networks, the task can be challenging.

- **Transmission power control**
  It has been shown in [95] that if the received signal powers fit an exponential profile, the sum capacity of all users can reach the Shannon limit assuming perfect interference cancellation. According to this principle, the primary goal of the protocol should be assigning users different transmit powers such that the signal strength at the receiver presents an exponential diversity. This observation implies that, in order to be canceled from the composite signal, the current signal must be significantly stronger than the successive signals. Noticing that the number of the top K packets that can be successfully decoded by SIC and thus the system performance are strictly constrained by the transmit power limit and the required SINR threshold, Weber *et al.* [99] develop an upper bound and lower bound for the capacity and suggested that SIC should be used with direct sequence spread spectrum. With a large spreading factor, the SINR requirements of SIC will be reduced and therefore the system capacity can be increased.

  It will be interesting to incorporate the characteristics of the SIC receiver in the MPT/MPR networks and take into consideration the receiving power level of each transmission at the receiver when constructing a feasible schedule. Additionally, the distance from the transmitter to the receiver plays a significant role in determining the transmission power. The path-loss attenuation is one of the primary causes to the signal power loss from the transmitter to the receiver and it is generally unknown in an ad hoc network system. Therefore, it will be a challenging problem to determining the optimal transmitting power for every transmission in ad hoc networks to fairly and efficiently utilize the MPT/MPR capability.

- **Full-duplex radio**
  Throughput this thesis, we assume half-duplex radio are employed such that a

node cannot transmit and receive at the same time. According to this principle, we build the contention graph in which adjacent links are dependent because the joint node cannot transmit and receive simultaneously. The constraint caused by the half-duplex radio limits the capacity of the network. Naturally, if the half-duplex radio is replaced by the full-duplex radio, the network is expected to achieve a higher overall performance since more links can transmit concurrently.

When designing AB-MAC, we argue that in order to efficiently exchange local information, a node should not transmit in two successive slots. After a transmitting slot, a node is forced to listen such that it can learn the reception status from its neighbors. While this constraint will not significantly affect the performance of multiple-hop flows, because the relays on the path of a flow should switch between transmitting and receiving states when forwarding the packets from the source to the destination, it can nevertheless cause channel waste for single-hop flows. However, with the full-duplex radios, this constraint can be removed because nodes can listen to neighbors while transmitting and hence further improve the overall performance.

- **Practical environments**

  Another direction to extend the current work is to consider more practical environments. In the last section, we discussed the SIC receiver we simulated. This SIC receiver assumes that the system is asynchronous to approximate real environments. Packets may have different length and can be transmitted at any time instant. We mentioned the potential impact of the near-far problem because the reception result will be significantly affected by the arrival signal strength. Compared to synchronous systems, a major difference in asynchronous networks is that the starting time and the ending time of the packet transmissions are not aligned. Therefore, similarly to the difference between pure ALOHA and slotted ALOHA [4], in asynchronous systems, because a packet can be interfered by any overlapping packets, the overall performance will degrade since more collisions will occur compared to the synchronous counterpart. However, properly designed MAC protocol can alleviate the collisions and improve the system performance in asynchronous networks.

  AB-MAC uses the ARC-AIMD to adjust the AP according to the end-of-slot acknowledgments. In synchronous systems, because all the transmitters complete their transmissions at the same time, those acknowledgments will be heard by any neighbor which is not broadcasting acknowledgments themselves. In asynchronous networks, the end-of-slot acknowledgments may be lost by the

neighbors if they are transmitting either a data packet or an acknowledgment. Depending on how sensitive the algorithm will be to the acknowledgments, a separate control channel may be necessary to convey those short messages. If the full-duplex radio is employed, it can also help to solve the immediate feedback communication problem in asynchronous systems. With the help, either by a separate control channel of the full-duplex radio, once a node determined the recent reception status of its neighbors, the ARC-AIMD method can still serve to regulate the AP before the next transmission.

- **Extension of contention graphs into MPT/MPR networks**
  It is also interesting to extend the definition of contention graphs into MPT/MPR. We have seen in traditional contention graphs that we need the notion of a clique to capture the spatial reuse restrictions. Specifically, we developed the eight pair-wise interference patterns within cliques for capturing all pairwise combinations that are allowed (subject to some corresponding "rate" constraints on the receiver and transmitter side). Extending on this line of logic, there are also n-way ($n = 3, 4, ...$) constraints, as far as the size of the clique could allow. Each set of combinations thought becomes combinatorially more challenging to describe and to map the corresponding set of linear constraints. A possible direction would be to see how far into this set of n-way constraints one can describe and if there are points of diminishing returns that could be ignored, e.g., after n becomes large n-way interactions are so constraint (in terms of the corresponding rate constraints) that the possibility that n arcs are simultaneously active will be so low such that it could be ignored – but this n would depend on the particular configuration.

Bibliography

[1] http://code.nsnam.org/ns-3-dev/.

[2] IEEE 802.11e/d13.0. *Draft Supplement to Part 11: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Medium Access Control (MAC) Enhancements for Quality of Service (QoS)*. IEEE, Jan 2005.

[3] N. Abramson. The ALOHA system: another alternative for computer communications. In *Proc. AFIPS Conf.*, volume 37, pages 281–285, Nov 1970.

[4] N. Abramson. The throughput of packet broadcasting channels. *IEEE transactions on communications*, 25(1):128, 1977.

[5] J.G. Andrews. Interference cancellation for cellular systems: a contemporary overview. *Wireless Communications, IEEE*, 12(2):19 – 29, Apr 2005.

[6] J.G. Andrews and T.H. Meng. Optimum power control for successive interference cancellation with imperfect channel estimation. *Wireless Communications, IEEE Transactions on*, 2(2):375 – 383, Mar 2003.

[7] A. Bar-Noy, A. Mayer, B. Schieber, and M. Sudan. Guaranteeing fair service to persistent dependent tasks. In *Proceedings of the Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '95, pages 243–252, Philadelphia, PA, USA, 1995. Society for Industrial and Applied Mathematics.

[8] B. Bellalta, A. Faridi, D. Staehle, J. Barcelo, A. Vinel, and M. Oliver. Performance analysis of CSMA/CA protocols with multi-packet transmission. *Computer Networks*, 57(14):2675 – 2688, 2013.

[9] D. Bertsekas and R. Gallager. *Data networks (2nd ed.)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1992.

[10] V. Bharghavan. Performance evaluation of algorithms for wireless medium access. In *Computer Performance and Dependability Symposium, 1998. IPDS '98. Proceedings. IEEE International*, pages 86–95, Sep 1998.

[11] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang. MACAW: A media access protocol for wireless LAN's. *SIGCOMM Comput. Commun. Rev.*, 24(4):212–225, October 1994.

[12] A. Bhatji. *TDMA and CDMA in Mobile Communications*. PhD Thesis. Florida State University, 2004.

[13] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Gossip algorithms: design, analysis and applications. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, volume 3, pages 1653–1664 vol. 3, March 2005.

[14] J. Bradley, J. Barbier, and D. Handler. Embracing the internet of everything to capture your share of $14.4 trillion, 2013.

[15] D. Brélaz. New methods to color the vertices of a graph. *Commun. ACM*, 22(4):251–256, April 1979.

[16] Z Cao and E.W. Zegura. Utility max-min: an application-oriented bandwidth allocation scheme. In *INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 2, pages 793–801 vol.2, Mar 1999.

[17] J. Capetanakis. Tree algorithms for packet broadcast channels. *Information Theory, IEEE Transactions on*, 25(5):505 – 515, Sep 1979.

[18] P. Casari, M. Levorato, and M. Zorzi. On the implications of layered space-time multiuser detection on the design of MAC protocols for ad hoc networks. In *Personal, Indoor and Mobile Radio Communications, 2005. PIMRC 2005. IEEE 16th International Symposium on*, volume 2, pages 1354 –1360, Sep 2005.

[19] G.D. Celik, G. Zussman, W.F. Khan, and E. Modiano. MAC for networks with multipacket reception capability and spatially distributed nodes. *Mobile Computing, IEEE Transactions on*, 9(2):226–240, 2010.

[20] D.S. Chan and T. Berger. Performance and cross-layer design of CSMA for wireless networks with multipacket reception. In *Signals, Systems and Computers, 2004. Conference Record of the Thirty-Eighth Asilomar Conference on*, volume 2, pages 1917 – 1921, Nov 2004.

[21] A. Charny. *An Algorithm for Rate Allocation in a Packet-Switching Network With Feedback*. Massachusetts Institute of Technology, Cambridge, MA, USA, 1994.

[22] D. Chiu and R. Jain. Analysis of the increase and decrease algorithms for congestion avoidance in computer networks. *Comput. Netw. ISDN Syst.*, 17:1–14, June 1989.

[23] IEEE Committee. IEEE standard for information technology-telecommunications and information exchange between systems-local and metropolitan area networks-specific requirements - part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications. *IEEE Std 802.11-2007 (Revision of IEEE Std 802.11-1999)*, pages C1–1184, 12 2007.

[24] A. Eryilmaz, A. Ozdaglar, and E. Modiano. Polynomial complexity algorithms for full utilization of multi-hop wireless networks. In *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pages 499–507, May 2007.

[25] Z. Fang and B. Bensaou. Fair bandwidth sharing algorithms based on game theory frameworks for wireless ad-hoc networks. In *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, volume 2, pages 1284–1295 vol.2, 2004.

[26] G. Gallager. Conflict resolution in random access broadcast networks. *Proc. AFOSR Workshop Communications Theory and Applications*, pages 74 – 76, Sep 1978.

[27] V. Gambiroza, B. Sadeghi, and E.W. Knightly. End-to-end performance and fairness in multihop wireless backhaul networks. In *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking*, MobiCom '04, pages 287–301, New York, NY, USA, 2004. ACM.

[28] J. J. Garcia-Luna-Aceves, Hamid R. Sadjadpour, and Z. Wang. Challenges: towards truly scalable ad hoc networks. In *Proceedings of the 13th annual ACM international conference on Mobile computing and networking*, MobiCom '07, pages 207–214, Montreal, Quebec, Canada, 2007. ACM.

[29] L. Georgiadis, M.J. Neely, and L. Tassiulas. Resource allocation and cross-layer control in wireless networks. *Found. Trends Netw.*, 1(1):1–144, April 2006.

[30] M. Ghanbarinejad, C. Schlegel, and P. Gburzynski. Adaptive probabilistic medium access in MPR-capable ad-hoc wireless networks. In *In IEEE Globecom*, GLOBECOM'09, pages 4424–4428, Piscataway, NJ, USA, 2009. IEEE Press.

[31] S. Ghez, S. Verdu, and S. C. Schwartz. Stability properties of slotted ALOHA with multipacket reception capability. *IEEE Transactions on Automatic Control*, 33:640–649, 1988.

[32] P. Giaccone, B. Prabhakar, and D. Shah. Randomized scheduling algorithms for high-aggregate bandwidth switches. *Selected Areas in Communications, IEEE Journal on*, 21(4):546–559, May 2003.

[33] K.S. Gilhousen, I.M. Jacobs, R. Padovani, A.J. Viterbi, Jr. Weaver, L.A., and III Wheatley, C.E. On the capacity of a cellular CDMA system. *Vehicular Technology, IEEE Transactions on*, 40(2):303 –312, May 1991.

[34] C. Godsil and G. Royle. *Algebraic graph theory*, volume 207 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 2001.

[35] M. Guo, X. Wang, and M. Wu. On the capacity of k-MPR wireless networks. *Wireless Communications*, 8(7):3878 –3886, Jul 2009.

[36] Z. Haas. On the performance of a medium access control scheme for the reconfigurable wireless networks. In J. Holtzman and M. Zorzi, editors, *Advances in Wireless Communications*, volume 435 of *The International Series in Engineering and Computer Science*, pages 269–284. Springer US, 2002.

[37] M. Heusse, F. Rousseau, R. Guillier, and A. 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*, SIGCOMM '05, pages 121–132, Philadelphia, Pennsylvania, USA, 2005. ACM.

[38] G. Holland, N. Vaidya, and P. Bahl. A rate-adaptive mac protocol for multihop wireless networks. *Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 236–251, 2001.

[39] J. Hou, J.E. Smee, H.D. Pfister, and S. Tomasin. Implementing interference cancellation to increase the EV-DO rev a reverse link capacity. *Communications Magazine, IEEE*, 44(2):58 – 64, Feb 2006.

[40] T. Hou and D. M. Lucantoni. Buffer sizing for synchronous self-routeing broadband packet switches with bursty traffic. *International Journal of Digital & Analog Cabled Systems*, 2(4):253–260, 1989.

[41] C. Hu and J.C. Hou. A novel approach to contention control in IEEE 802.11e-Operated WLANs. In *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pages 1190 –1198, May 2007.

[42] X. Huang and B. Bensaou. On max-min fairness and scheduling in wireless ad-hoc networks: analytical framework and implementation. In *Proc. of MobiHoc*, MobiHoc '01, pages 221–231, Long Beach, CA, USA, 2001.

[43] ITU. Wi-Fi and mobile Internet (2G, 3G, 3G+ and 4G).

[44] ITU. Requirements related to technical performance for IMT-advanced radio interface(s), 2008.

[45] ITU. Cellular standards for the third generation: The ITU's IMT 2000 family, 2013.

[46] V. Jacobson. Congestion avoidance and control. In *Symposium proceedings on Communications architectures and protocols*, SIGCOMM '88, pages 314–329, New York, NY, USA, 1988. ACM.

[47] R. Jain, D. Chiu, and W. Hawe. A quantitative measure of fairness and discrimination for resource allocation in shared computer systems. *Computing Research Repository*, cs.NI/9809, 1998.

[48] L Jiang and J Walrand. A distributed CSMA algorithm for throughput and utility maximization in wireless networks. *IEEE/ACM Trans. Netw.*, 18(3):960–972, June 2010.

[49] J. Ju and V.O.K. Li. TDMA scheduling design of multihop packet radio networks based on latin squares. In *INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 1, pages 187–193 vol.1, Mar 1999.

[50] E. Jung and N. Vaidya. A power control MAC protocol for ad hoc networks. In *Wireless Networks*, volume 11, pages 55–66. Kluwer Academic Publishers, 2005.

[51] V. Kanodia, A. Sabharwal, B. Sadeghi, and E. Knightly. Ordered packet scheduling in wireless ad hoc networks: Mechanisms and performance analysis. In *Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking &Amp; Computing*, MobiHoc '02, pages 58–70, New York, NY, USA, 2002. ACM.

[52] S. Karande, Zheng Wang, H.R. Sadjadpour, and J.J. Garcia-Luna-Aceves. Capacity of wireless ad-hoc networks under multipacket transmission and reception. In *Signals, Systems and Computers, 2008 42nd Asilomar Conference on*, pages 2115 –2119, oct. 2008.

[53] P. KARN. MACA-A new channel access method for packet radio. *ARRL/CRRL Amateur Radio 9th, Computer Networking Conference, 1990*, pages 134–140, 1990.

[54] R. Karp, C. Schindelhauer, S. Shenker, and B. Vocking. Randomized rumor spreading. In *Foundations of Computer Science, 2000. Proceedings. 41st Annual Symposium on*, pages 565–574, 2000.

[55] L. Kleinrock and F. Tobagi. Packet switching in radio channels: Part i–carrier sense multiple-access modes and their throughput-delay characteristics. *Communications, IEEE Transactions on*, 23(12):1400 – 1416, Dec 1975.

[56] S. Kravitz. Packing cylinders into cylindrical containers. *Math. Mag.*, 40:65–71, 1967.

[57] J. Lee and W. Kuo. Review: Fairness provisioning in multi-hop wireless backhaul networks: Challenges and solutions. *Comput. Commun.*, 33(15):1767–1772, September 2010.

[58] B. Li. End-to-end fair bandwidth allocation in multi-hop wireless ad hoc networks. In *Distributed Computing Systems, 2005. ICDCS 2005. Proceedings. 25th IEEE International Conference on*, pages 471–480, 2005.

[59] J. Li, C. Blake, D. S.J. De Couto, H. Lee, and R. Morris. Capacity of ad hoc wireless networks. In *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, MobiCom '01, pages 61–69, New York, NY, USA, 2001. ACM.

[60] K. Li, M. Ghanbarinejad, I. Nikolaidis, and C. Schlegel. Additive-increase multiplicative-decrease mac protocol with multi-packet reception. In Vassilis Tsaoussidis, AndreasJ. Kassler, Yevgeni Koucheryavy, and Abdelhamid Mellouk, editors, *Wired/Wireless Internet Communication*, volume 7889 of *Lecture Notes in Computer Science*, pages 15–28. Springer Berlin Heidelberg, 2013.

[61] B Liu, N. Bulusu, H Pham, and S Jha. Csmac: A novel ds-cdma based mac protocol for wireless sensor networks. In *Global Telecommunications Conference Workshops, 2004. GlobeCom Workshops 2004. IEEE*, pages 33–38, Nov 2004.

[62] H. Luo, S. Lu, and V. Bharghavan. A new model for packet scheduling in multihop wireless networks. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, MobiCom '00, pages 76–86, New York, NY, USA, 2000. ACM.

[63] S. Lv, W. Zhuang, W. Wang, and X. Zhou. Scheduling in wireless ad hoc networks with successive interference cancellation. In *INFOCOM, 2011 Proceedings IEEE*, pages 1287–1295, 2011.

[64] S. Lv, W. Zhuang, X. Wang, and X. Zhou. Context-aware scheduling in wireless networks with successive interference cancellation. In *ICC*, pages 1–5. IEEE, 2011.

[65] A.B. MacKenzie and S.B. Wicker. Stability of multipacket slotted aloha with selfish users and perfect information. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 3, pages 1583 – 1590, Mar 2003.

[66] E. Modiano, D. Shah, and G. Zussman. Maximizing throughput in wireless networks via gossiping. In *Proceedings of the Joint International Conference on Measurement and Modeling of Computer Systems*, SIGMETRICS '06/Performance '06, pages 27–38, New York, NY, USA, 2006. ACM.

[67] J. Morrish. The connected life 25 billion connections by 2020, 2013.

[68] T. Mortimer and J. Harms. A MAC protocol for multihop RP-CDMA ad hoc wireless networks. In *Communications (ICC), 2012 IEEE International Conference on*, pages 424–429, 2012.

[69] J.C. Mundarath, P. Ramanathan, and B.D. Van Veen. NULLHOC : a MAC protocol for adaptive antenna array based wireless ad hoc networks in multipath environments. In *Global Telecommunications Conference, 2004. GLOBECOM '04. IEEE*, volume 5, pages 2765 – 2769, Nov 2004.

[70] A. Muqattash and M. Krunz. Cdma-based mac protocol for wireless ad hoc networks. In *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking &Amp; Computing*, MobiHoc '03, pages 153–164, New York, NY, USA, 2003. ACM.

[71] T. Nandagopal, T. Kim, X. Gao, and V. Bharghavan. Achieving MAC layer fairness in wireless packet networks. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, MobiCom '00, pages 87–98, New York, NY, USA, 2000. ACM.

[72] A. Nasipuri and S.R. Das. Multichannel CSMA with signal power-based channel selection for multihop wireless networks. In *Vehicular Technology Conference, 2000. IEEE-VTS Fall VTC 2000. 52nd*, volume 1, pages 211–218 vol.1, 2000.

[73] A. Nasipuri, J. Zhuang, and S.R. Das. A multichannel CSMA MAC protocol for multihop wireless networks. In *Wireless Communications and Networking Conference, 1999. WCNC. 1999 IEEE*, pages 1402–1406 vol.3, 1999.

[74] V. Naware, G. Mergen, and L. Tong. Stability and delay of finite-user slotted ALOHA with multipacket reception. *Information Theory, IEEE Transactions on*, 51(7):2636–2656, 2005.

[75] M.J. Neely, E. Modiano, and C. Li. Fairness and optimal stochastic control for heterogeneous networks. *Networking, IEEE/ACM Transactions on*, 16(2):396–409, April 2008.

[76] M.J. Neely, E. Modiano, and C.E. Rohrs. Dynamic power allocation and routing for time varying wireless networks. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 1, pages 745–755 vol.1, March 2003.

[77] M.J. Neely and R. Urgaonkar. Optimal backpressure routing for wireless networks with multi-receiver diversity. *Ad Hoc Netw.*, 7(5):862–881, July 2009.

[78] Joon-Sang Park, A. Nandan, M. Gerla, and Heechoon Lee. SPACE-MAC: enabling spatial reuse using MIMO channel-aware MAC. In *Communications, 2005. ICC 2005. 2005 IEEE International Conference on*, volume 5, pages 3642 – 3646, May 2005.

[79] S. Rangwala, R. Gummadi, R. Govindan, and K. Psounis. Interference-aware fair rate control in wireless sensor networks. *SIGCOMM Comput. Commun. Rev.*, 36(4):63–74, Aug 2006.

[80] A. Raniwala, D. Pradipta, and S. Sharma. End-to-end flow fairness over IEEE 802.11-based wireless mesh networks. In *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pages 2361–2365, May 2007.

[81] H.R. Sadjadpour, Z. Wang, and J.J. Garcia-Luna-Aceves. The capacity of wireless ad hoc networks with multi-packet reception. *Communications, IEEE Transactions on*, 58(2):600–610, February 2010.

[82] S. Sarkar and L. Tassiulas. Fair allocation of discrete bandwidth layers in multicast networks. In *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 3, pages 1491–1500 vol.3, 2000.

[83] C. Schlegel and A. Grant. *Coordinated multiuser communications*. Springer, New York, 2006.

[84] C. Schlegel, R. Kempter, and P. Kota. A novel random wireless packet multiple access method using CDMA. *Wireless Communications, IEEE Transactions on*, 5(6):1362–1370, 2006.

[85] J. Shi, O. Gurewitz, V. Mancuso, J. Camp, and E.W. Knightly. Measurement and modeling of the origins of starvation in congestion controlled mesh networks. In *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, pages –, April 2008.

[86] B. Sklar. *Digital communications: fundamentals and applications*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2000.

[87] J. So. *Design and Evaluation of Multi-channel Multi-hop Wireless Networks*. University of Illinois at Urbana-Champaign, 2006.

[88] J. So and N. H. Vaidya. Multi-channel Mac for ad hoc networks: Handling multi-channel hidden terminals using a single transceiver. In *Proceedings of the 5th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc '04, pages 222–233, New York, NY, USA, 2004. ACM.

[89] W. Szpankowski. Stability conditions for some multiqueue distributed systems: Buffered random access systems. *Adv. Appl. Probab.*, 26:498–515, 1994.

[90] L. Tassiulas. Linear complexity algorithms for maximum throughput in radio networks and input queued switches. In *INFOCOM '98. Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 2, pages 533–539 vol.2, Mar 1998.

[91] L. Tassiulas and A. Ephremides. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. *Automatic Control, IEEE Transactions on*, 37(12):1936–1948, Dec 1992.

[92] F.A. Tobagi and L. Kleinrock. Packet switching in radio channels: Part ii–the hidden terminal problem in carrier sense multiple-access and the busy-tone solution. *Communications, IEEE Transactions on*, 23(12):1417–1433, Dec 1975.

[93] A. Tzamaloukas and J.J. Garcia-Luna-Aceves. A receiver-initiated collision-avoidance protocol for multi-channel networks. In *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 1, pages 189–198 vol.1, 2001.

[94] Tijs van Dam and Koen Langendoen. An adaptive energy-efficient MAC protocol for wireless sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems*, SenSys '03, pages 171–180, Los Angeles, California, USA, 2003. ACM.

[95] A.J. Viterbi. Very low rate convolution codes for maximum theoretical performance of spread-spectrum multiple-access channels. *Selected Areas in Communications, IEEE Journal on*, 8(4):641 –649, May 1990.

[96] A.J. Viterbi. *CDMA: principles of spread spectrum communication*. Addison-Wesley wireless communications series. Addison-Wesley Pub. Co., 1995.

[97] X. Wang and J.J. Garcia-Luna-Aceves. Embracing interference in ad hoc networks using joint routing and scheduling with multiple packet reception. *Ad Hoc Networks*, 7(2):460 – 471, 2009.

[98] D. Warrier and U. Madhow. On the capacity of cellular CDMA with successive decoding and controlled power disparities. In *Vehicular Technology Conference, 1998. VTC 98. 48th IEEE*, volume 3, pages 1873 –1877, May 1998.

[99] S.P. Weber, J.G. Andrews, X. Yang, and G. de Veciana. Transmission capacity of wireless ad hoc networks with successive interference cancellation. *Information Theory, IEEE Transactions on*, 53(8):2799 –2814, Aug 2007.

[100] S. Wu, C. Lin, Y. Tseng, and J. Sheu. A new multi-channel MAC protocol with on-demand channel assignment for multi-hop mobile ad hoc networks. In *Parallel Architectures, Algorithms and Networks, 2000. I-SPAN 2000. Proceedings. International Symposium on*, pages 232–237, 2000.

[101] H. Xu, J.J. Garcia-Luna-Aceves, and HamidR. Sadjadpour. Enabling multi-packet transmission and reception: An adaptive MAC protocol for MANETs. In X. Zhang and D. Qiao, editors, *Quality, Reliability, Security and Robustness in Heterogeneous Networks*, volume 74 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 188–203. Springer Berlin Heidelberg, 2012.

[102] K. Xu, S. Bae, S. Lee, and M. Gerla. TCP behavior across multihop wireless networks and the wired internet. In *Proceedings of the 5th ACM International Workshop on Wireless Mobile Multimedia*, WOWMOM '02, pages 41–48, New York, NY, USA, 2002. ACM.

[103] K. Xu, M. Gerla, and S. Bae. How effective is the IEEE 802.11 RTS/CTS handshake in ad hoc networks. In *Global Telecommunications Conference, 2002. GLOBECOM '02. IEEE*, volume 1, pages 72–76 vol.1, Nov 2002.

[104] X. Yang and N. Vaidya. On physical carrier sensing in wireless ad hoc networks. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, volume 4, pages 2525–2535 vol. 4, March 2005.

[105] Y.R. Yang and S.S. Lam. General AIMD congestion control. In *Network Protocols, 2000. Proceedings. 2000 International Conference on*, pages 187 –198, 2000.

[106] Z. Yang and J.J. Garcia-Luna-Aceves. Hop-reservation multiple access (HRMA) for ad-hoc networks. In *INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 1, pages 194–201 vol.1, Mar 1999.

[107] W. Ye, J. Heidemann, and D. Estrin. An energy-efficient MAC protocol for wireless sensor networks. In *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 3, pages 1567–1576 vol.3, 2002.

[108] L. Ying, S. Shakkottai, A. Reddy, and S. Liu. On combining shortest-path and back-pressure routing over multihop wireless networks. *Networking, IEEE/ACM Transactions on*, 19(3):841–854, June 2011.

[109] H. Zhai, X. Chen, and Y. Fang. Alleviating intra-flow and inter-flow contentions for reliable service in mobile ad hoc networks. In *Military Communications Conference, 2004. MILCOM 2004. 2004 IEEE*, volume 3, pages 1640–1646 Vol. 3, Oct 2004.

[110] H Zhai and Y. Fang. Physical carrier sensing and spatial reuse in multirate and multihop wireless ad hoc networks. In *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, pages 1–12, April 2006.

[111] Q. Zhao and L. Tong. A multiqueue service room MAC protocol for wireless networks with multipacket reception. *IEEE/ACM Trans. Networking*, 11:125–137, 2003.

[112] J. Zhu, B. Metzler, X. Guo, and Y. Liu. Adaptive CSMA for scalable network capacity in high-density WLAN: A hardware prototyping approach. In *in Proc. of IEEE Infocom*, 2006.

[113] M. Zorzi, J. Zeidler, A. Anderson, B. Rao, J. Proakis, A.L. Swindlehurst, M. Jensen, and S. Krishnamurthy. Cross-layer issues in MAC protocol design for MIMO ad hoc networks. *Wireless Communications, IEEE*, 13(4):62 –76, Aug 2006.