FAIR PACKET SCHEDULING AND BANDWIDTH MANAGEMENT IN WIRELESS NETWORKS

By

YING JIAN

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

2008

To my family

ACKNOWLEDGMENTS

I would like to thank my advisor, Prof. Shigang Chen, for his constant guidance, support, and encouragement. I am privileged to have such a wonderful advisor, who is at all times enthusiastic, optimistic, patient, helpful, and encouraging. He gave me extensive advice and insight during the course of my research work.

My special thanks go to Prof. Sartaj Sahni, Prof. Randy Chow, Prof. Ye Xia, Prof. Jose Fortes, and Prof. Tan Wong, for their instructive comments and support during my work. I would also like to thank all my colleagues in Prof. Chen's research group, including Zhan Zhang, Liang Zhang, MyungKeun Yoon, Ming Zhang and Tao Li, for providing a high level of research support.

I want to express my deepest love to my darling wife Qianqian, my parents, and my brother. Their love, understanding, and encouragement have always been the strongest support to me.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

FAIR PACKET SCHEDULING AND BANDWIDTH MANAGEMENT
IN WIRELESS NETWORKS

By

Ying Jian

December 2008

Chair: Shigang Chen
Major: Computer Engineering

Our study focused on fair packet scheduling and bandwidth management in CSMA/CA based wireless networks. We address the fairness problem for MAC-layer links and study end-to-end service differentiation and rate assurance for multihop flows.

Fine-level rate control, particularly meeting rate requirements and differentiating various types of end-to-end traffic, remains an open problem for multihop wireless networks. Traditionally, rate assurance in wired networks is achieved through resource reservation and admission control, which can be efficiently implemented since the bandwidth capacity of each communication link is known and the sender of a link has the information of all flows that compete for the bandwidth of the link. In a wireless network, however, the capacity of each wireless link can change unpredictably over time due to contention from nearby links and dynamic channel conditions. An end-to-end flow consumes available bandwidth not only at links on its route but also at all nearby contending links, which makes resource reservation extremely complicated.

We propose a new adaptive rate control function based on two novel techniques, called *proportional packet scheduling* (PPS) and *dynamic weight adaptation with floor and ceiling* (DWA). PPS distributes channel bandwidth among MAC (one-hop) flows in proportion to their weights. DWA adapts flows' weight according to their rate requirements and priorities. End-to-end traffic is classified into two categories: best-effort flows and QoS flows with rate requirements. The QoS flows are assigned with different

priorities. PPS and DWA together achieve three important objectives without resource reservation and admission control. First, when bandwidth contention arises, the rate requirements of the QoS flows are satisfied in the order of priorities. Second, beyond the rate requirements, the rest bandwidth is allocated to the flows in a differentiated manner, taking both bandwidth demand and priority into consideration. Third, no flow is starved and all bandwidth is effectively utilized.

Another important problem is on how to achieve fairness for the MAC-layer links in multiple contending WLANs or multihop networks, where the carrier sensing range and the interference range are much larger than the transmission range. We demonstrate that CSMA/CA networks, including IEEE 802.11 networks, exhibit severe fairness problem in many scenarios. Most existing solutions require nodes to overhear transmissions made by contending nodes and, based on the overheard information, adjust local rates to achieve fairness among all contending links. Their underlying assumption is that transmissions made by contending nodes can be overheard. However, this assumption holds only when the transmission range is equal to the carrier sensing range, which is not true in most real networks. As our study reveals, the overhearing-based solutions, as well as several non-overhearing AIMD solutions, cannot achieve MAC-layer fairness in various settings.

We propose a new rate control protocol, called PISD (Proportional Increase Synchronized multiplicative Decrease). Without relying on overhearing, it provides fairness in CSMA/CA networks, particularly IEEE 802.11 networks, by using only local information and performing localized operations. It combines several novel rate control mechanisms, including synchronized multiplicative decrease, proportional increase, and background transmission. PISD works precisely for scenarios where all MAC flows mutually contend but has limitations when applied to networks consisting multiple contention groups. We develop PISD further and propose two new schemes to overcome the limitations. We prove that flows' rates attained under the two new schemes approximate proportional fairness.

CHAPTER 1
INTRODUCTION

Wireless technologies have come a long way, from the radio technologies that allow data bits to be exchanged between two physically-disconnected devices, to the multiple access technologies that allow a group of devices to share a common wireless communication channel, and to the routing technologies that allow out-of-range devices to communicate via multihop wireless paths. Following the enormous success of WLAN, multihop wireless networks, including mesh networks, ad-hoc networks, sensor networks, are expected to lead in the next wave of deployment. To improve their applicability in practice, not only must these networks provide a robust and efficient communication service, but also they should provide flexible tools for traffic engineering in order to support diverse user applications.

1.1 Service Differentiation and Rate Assurance Based on Weighted Fair Packet Scheduling

In multihop wireless networks, rate control is an important function for meeting rate requirements and differentiating various types of end-to-end data flows. *Fine-level* rate control remains an open problem in multihop wireless networks, particularly those based on the popular CSMA/CA protocols. The large body of literature for wired networks cannot be applied to wireless networks due to their fundamental differences. Meeting rate requirements is traditionally implemented through resource reservation and admission control in wired networks, which can be efficiently done since the bandwidth capacity of each communication link is known and the sender of a link has the information of all flows that compete for the bandwidth of the link. However, in a wireless network based on CSMA/CA, the capacity of a wireless link is undefined and may change drastically over time, depending on the load of the contending links, the relative positions of the links, and the channel conditions. Even the channel capacity varies from place to place and from time to time due to environmental noise, obstacles, multipath fading, and multi-rate links. For instance, when IEEE 802.11b links select different rates (11Mbps, 5Mbps, 2Mbps and 1Mbps) based on their levels of signal strength, the channel capacity is a variable dependent upon how much time each link occupies the channel. Other unpredictable factors can also come into play. For example, when two wireless networks with overlapping channels are deployed in the same area, the channel capacity perceived by one network will depend on the activities of the other. Admission control cannot be performed if the link/channel capacity is dynamic.

Resource reservation in multihop wireless networks also has problems. An end-to-end flow consumes bandwidth not only at links on its route but also at all nearby contending links, which makes resource reservation extremely complicated. With spatial channel reuse, the local channel perceived by each wireless link is different because each link has a different set of contending links. Two contending links will consume bandwidth in each other's perceived channel. Consider a new flow whose rate requirement is r and routing path is $a \rightarrow b \rightarrow c \rightarrow d$. In order to support the flow, the channel where link (a, b) resides should have $3r$ residual (unused) bandwidth because (a, b), (b, c) and (c, d) mutually contend and they will each consume r bandwidth in the channel when carrying the flow (assuming IEEE 802.11 DCF). Similarly, the channels where other links of the routing path reside also need more than r residual bandwidth for the flow. Even links outside of the path need residual bandwidth to support the flow. Consider a nearby link (x, y) that contends with (a, b). Suppose its channel is already *saturated* due to heavy traffic on some other contending links. Now if we add the new flow, as the rate on (a, b) is increased, the rate on (x, y) will be driven down, causing the violation of the previous resource reservation made on (x, y). Determining how much bandwidth (x, y) needs in order to support the new flow is not an easy task. It depends on how much channel spatial reuse can be done between (a, b) and other links contending with (x, y). Therefore, resource reservation requires coordination among links on the route and all other links that contend with them.

Facing the above challenges, the past research has followed three directions. The first direction is to restrict the study on wireless LANs [1–4]. When every link sees the same channel with the same set of contending links, many of the above problems are either avoided or much simplified. A different restriction can be assuming each node transmits at a different frequency [5]. The second direction is to work on coarse-level service differentiation that does not provide rate assurance [6–9]. For example, different backoff policies [7] or different contention window sizes [6] are assigned to packets of different classes to provide qualitative scheduling preferences. IEEE 802.11e [10] belongs to this category when it is applied in a multihop wireless network. The third direction is to design heuristics to address the hard problems in resource reservation and admission control. Most work focuses on establishing a heuristic approach for each node to estimate its channel's residual bandwidth, which will be used to guide admission control. The bandwidth estimation is made based on channel idle time [6; 11], average packet transmission delay [12; 8; 2], or channel-access probabilistic models [13–16]. As detailed analysis in [16] points out, none of them considers the impact of hidden terminals in multihop wireless networks, and each will perform poorly under certain scenarios. Moreover, the residual bandwidth measured may continuously change due to dynamic channel conditions, and estimating bandwidth does not solve the complicated resource reservation problem discussed previously.

Instead of taking a head-on approach to address the difficult problems of resource reservation and admission control, we want to take a step back and ask whether resource reservation and admission control, legacy from wired networks, are suitable for multihop wireless networks. We want to find an alternative solution for wireless networks that can not only solve the problems but also have a much simpler design. In this study, we propose to replace admission control and resource reservation with a simple yet effective adaptive rate control function suited for handling network/traffic dynamics. It automatically adapts the bandwidth distribution to satisfy the rate requirements of as many flows as possible *in the order of their priorities*. This *global* objective for all *end-to-end* flows in the network should be implemented solely based on *localized* operations. The adaptive rate control should not require the exchange of any topological or per-flow information among contending nodes, nor should it rely on the accurate measurement of link or channel capacities.

We classify end-to-end traffic into two categories: best-effort flows and QoS flows with minimum rate requirements. The QoS flows are assigned to service classes of different priorities. We have the following three objectives for rate control. The *rate assurance objective* requires QoS flows to be supported in the order of priorities. A higher-priority flow can preempt the bandwidth of a lower-priority flow. Following the priority order, the network should support as many QoS flows as possible. Beyond meeting the minimum rate requirements, the *bandwidth differentiation objective* requires the remaining bandwidth to be allocated to end-to-end flows based on their priorities as well as bandwidth demand. A flow with a higher minimum rate requirement and a higher priority should receive a larger amount of extra bandwidth. The *no-starvation/maximum-utilization objective* requires that no flow is starved and all network bandwidth is utilized when possible.

To achieve the three objectives, we design our adaptive rate control function based on two novel techniques [17]. First, we enhance CSMA/CA protocols for weighted bandwidth allocation through a new technique, called *proportional packet scheduling* (PPS), which distributes channel bandwidth among MAC (one-hop) flows in proportion to their weights. Comparing with the existing schemes, PPS realizes weighted bandwidth allocation in a fine granularity and achieves better throughput due to reduced radio collision. It is also the first fully distributed solution that achieves provable weighted maxmin fairness in CSMA/CA networks of dynamic channel conditions, with a bounded error that can be made arbitrarily small.

Second, working on top of PPS, a new technique called *dynamic weight adaptation with floor and ceiling* (DWA) is proposed, which allows each MAC flow to independently adapt its weight based on local information and acquire an appropriate fraction of channel bandwidth. We show that, when the weights of the MAC flows are adapted between certain upper bounds (ceilings) and lower bound (floors), the above introduced three objectives can be met. Adaptation at the MAC layer is common, but such adaptation designed for end-to-end objectives is not. The proposed technique demonstrates great flexibility in bandwidth distribution, yet it is simple to implement, which is important for practical wireless systems.

1.2 Fair Bandwidth Allocation in CSMA/CA Networks

Another part of our study focused on solving the fairness problem in the MAC layer of CSMA/CA networks, in particular IEEE 802.11 networks, where the carrier sensing range of each node is considerably larger than their transmission range.

When wireless hosts share the same communication channel, they should be given a fair chance of accessing the wireless medium. *Fairness* is one of the core problems that any MAC protocol must address. It prevents the situation that some hosts obtain most of the channel's bandwidth while others starve. A more general problem is that of *weighted fairness*, where the channel's bandwidth obtained by a host is proportional to its weight, which is assigned by the user based on application requirements. For example, when a web server and a client host share the same local channel (e.g. in a WLAN), the server may be given a higher weight because it may have to upload content to multiple users on the Internet simultaneously.

Random backoff in the IEEE 802.11 DCF achieves fairness in a WLAN where all hosts are downloading content from the Internet via the same access point. However, as observed in our study, it cannot achieve fairness (let alone weighted fairness) in many other scenarios. For example, when a server that uploads content to the Internet shares the access point of a WLAN with a client host that downloads, the client may obtain most of the channel's bandwidth while the server is slowed to crawl. When the access points at two nearby homes choose the same channel,[1] the hosts in one home may obtain most of the channel's bandwidth at the expense of the hosts in the other home. Furthermore, in any ad hoc deployment of 802.11 DCF links, bandwidth distribution is likely to be very skewed among those sharing a channel. We will use simulations in ns-2 to show unfairness in the above cases. IEEE 802.11e provides *qualitative* service differentiation among different categories of traffic. It does not solve the fairness problem for flows within the same category, nor does it provide quantitative service differentiation (such as weighted fairness) for flows in different categories.

The fairness problem in IEEE 802.11 networks is mostly due to the fundamental limitation of CSMA/CA, which gives preference in media access to some links over others, depending on their spatial locations. As this problem is well recognized, many fairness solutions have been proposed in the past decade [18–25]. They fall in two categories: overhearing-based solutions and non-overhearing solutions.

The overhearing-based solutions require each node to monitor the activity of all contending nodes and collect their links' information (such as rate, scheduling tag, or buffer status). Based on the collected information, a node decides its own media contention policy: i) increase/decrease minimum contention window if the local rate is above/below the average rate of all contending links [18; 19; 22], ii) serialize transmissions among contending links based on their scheduling tags [20; 21], or iii) emulate TDMA by computing a contention-free slotted schedule among the links [23]. The key question is how to collect information for the contending links, which may be multiple hops away. One naive approach is for each node to flood the information describing its adjacent links to all nodes within a certain number of hops. This approach is not only costly but also flawed because, as is observed in [23; 16], hop count is not a reliable means to identify

[1] This can happen when there are more neighboring access points than the number of non-overlapping channels.

a contending relationship. Hence, in virtually all existing solutions, a node learns the information about others by overhearing. However, the overhearing approach faces another serious problem: Contention is defined by the carrier sensing range and the interference range, whereas overhearing is limited to the transmission range, which is much shorter. Consequently, transmissions on many contending links (often the majority of them) cannot be overheard for information gathering, which severely limits the effectiveness of overhearing-based solutions.

Most non-overhearing solutions use the classical AIMD (Additive Increase Multiplicative Decrease) for rate control. On one hand, a node cannot overhear the exact information in transmissions made on contending links whose radio signal is strong enough to cause interference but too weak to decode. On the other hand, without overhearing, the node can still sense the aggregate impact of interference from those links by monitoring how busy the channel is, how frequently its own transmissions fail [24; 26], or how fast its local buffer is filled up. Based on such information, emulating the behavior of TCP in some sense, each node may set a threshold to decide when the channel is congested such that multiplicative decrease should be performed. This direction looks reasonable. However, our simulations in ns-2 show that AIMD fails to achieve fairness, too, not because the rationale behind AIMD is flawed, but because the interaction between AIMD and CSMA/CA neutralizes the effectiveness of AIMD. AIMD may also be applied to the contention window based on the number of idle slots between two transmissions [25; 27] (which can be measured through carrier sense instead of overhearing). We will show later that this approach also has limitations.

We propose a new rate control protocol, PISD (Proportional Increase Synchronized multiplicative Decrease) [28], that provides fairness in CSMA/CA networks, particularly in IEEE 802.11 networks. Within the design of PISD, we make three contributions. First, our study reveals the fundamental reasons exactly why the existing fairness solutions, as well as AIMD, do not work under realistic contention conditions. Particularly, for AIMD, we demonstrate that when the channel is saturated, nodes will see different channel occupancy levels, experience different frequencies of transmission failure, and encounter different buffer lengths. Unsynchronized multiplicative decrease is the reason for the failure of AIMD in CSMA/CA networks. Second, we introduce a number of novel rate control mechanisms, on which PISD is designed. The first mechanism relies on localized operations to ensure *synchronized multiplicative decrease*. The second mechanism extends PISD for weighted fairness through *proportional increase*. The third mechanism uses *background transmission* to ameliorate throughput degradation due to multiplicative decrease. Efforts are made to simplify the implementation of the rate control mechanisms, which we believe will benefit practical systems that adopt them. Third, we perform detailed analysis on PISD and prove that it will converge and achieve (weighted) fairness.

We also enhance PISD further. PISD works precisely for scenarios where all MAC flows mutually contend, but when it is applied to a network consisting of multiple contention groups, the network's throughput can be degraded. We develop PISD further and propose two new schemes, PISD-RS and PFS, to overcome the limitation. PISD-RS is a simple solution, while PFS is a better yet more sophisticated solution. Both of them improve PISD by revising the way they jam the wireless channel, where channel jamming is the key technique that PISD exploits to achieve synchronized multiplicative decrease. The two new proposed schemes are able to provide fairness in CSMA/CA networks with multiple contention groups. We conduct both theoretical analysis and simulations to study their effectiveness. By using convex optimization theory, we prove that flows' rates achieved by PFS (and PISD-RS) approximate proportional fairness.

The rest of this study is organized as follows. Chapter 2 proposes PPS, the technique for weighted bandwidth allocation among MAC flows in multihop wireless networks. Chapter 3 proposes DWA, which works on top of PPS and achieves end-to-end service differentiation and flow rate assurance. Chapter 4 proposes PISD and its enhanced versions, PISD-RS and PFS, all of which are designed to provide fairness in CSMA/CA networks without relying on overhearing. Chapter 5 concludes our study.

CHAPTER 2
PROPORTIONAL PACKET SCHEDULING AMONG MAC FLOWS

In this chapter, we present a new MAC-layer scheduling technique called *proportional packet scheduling* (PPS) that achieves weighted bandwidth allocation among single-hop MAC-layer flows in a multihop wireless network. We design this technique to serve as a basis for the implementation of DWA, an upper layer QoS protocol that will be introduced in Chapter 3.

DWA in Chapter 3 needs to work on top of a MAC-layer protocol that supports weighted bandwidth allocation, and the flow weights should be able to be dynamically changed. With added complexity, some MAC protocols in the current literature can potentially be modified to serve for this purpose. Protocols [29; 18; 19; 21] proposed to achieve fairness among contending MAC flows may be enhanced to support weighted bandwidth allocation, but in such protocols knowledge on local network topology or an additional fair share computation phase is needed, which may lead to difficulties on achieving dynamic weight change. The overlay MAC (OML) [23] and the Regulated Contention MAC (RCMAC) [4] can directly support weighted bandwidth allocation. However, RCMAC was designed to work under single-hop contentions and OML cannot be easily modified to support fast weight adjustment. In addition, most of these works do not provide weighted bandwidth allocation in a very fine granularity.

PPS, which is introduced in this chapter, achieves weighted bandwidth allocation in a fine granularity and is particularly suitable for dynamic weight adaptation. Allowing channel capacity to evolve spatially and temporally, this is the first fully distributed solution that achieves *provable* weighted maxmin fairness in CSMA/CA networks with a bounded error that can be made arbitrarily small. We believe it is a strong result. The max-min fairness achieved in [30] uses a multi-channel contention model that is different from the CSMA/CA model in our study. The work in [19] requires the computation of all contention cliques (which is difficult to implement distributedly and must be redone each time when the set of backlogged MAC flows change), and it assumes all cliques have equal, fixed capacity. Comparing with the existing protocols, the new scheduling technique also has other advantages: It is much simpler, easy to implement and analyze, and reduces radio collision. It does not require modifying the backoff algorithm of the existing channel access protocols.

The rest of this chapter is organized as follows. Section 2.1 discusses the related work. Section 2.2 studies the five types of contentions in the network. Section 2.3 describes PPS in details. Sections 2.4 analyzes the properties of PPS. Sectoin 2.5 introduces RRTS, which is an enhancement to PPS. Section 2.6 points out the state of the art and our contribution. Section 2.7 evaluates the performance of PPS through simulations. Section 2.8 summarizes the chapter.

2.1 Related Work

The area of research related to PPS is on solving the fairness problem in the link-layer of wireless networks. Many works have been dedicated to this area. Luo et al. [29] studied the problem of fair distribution of bandwidth and maximization of resource utilization. They first assure each flow in a network with a minimum channel share, then maximize aggregate channel utilization by spatial channel reuse. The distributed implementation of this algorithm needs topology information to be propagated along a conflict-free spanning tree. In their follow-up work [31; 21], SFQ (Start-time Fair Queueing) is applied to multihop wireless networks in a distributed way. In the approach, service tags (representing transmission deadlines) are piggybacked in packets, and each node maintains the status of all its contending flows. Locally, the flow with the minimum service tag is scheduled first. In addition, spatial channel reuse is exploited through simultaneous transmissions. This approach needs each node to maintain the status of all its contending flows, and the way it deals with the hidden terminal problem is only heuristic, which can only alleviates but not solve the problem. In [18; 19; 32], to achieve fair bandwidth distribution among contending wireless links in a multihop wireless network, every node is required to measure the rates on contending links through overhearing and then adjust its own rate by adjusting either the minimum contention window or the contention window directly. This approach requires the fair share of each link to be properly calculated (often based on the contention cliques) first. In [33], Li exploited a similar approach to ensure a conservative, small fair share of bandwidth for each end-to-end flow. In [34], Nandagopal et al. proposed a general analytical fairness model and a MAC protocol to approximate proportional fairness, which adjusts the contention window size based on the occurrence of retransmissions. In [20], Vaidya et al. achieved fairness in a wireless LAN by adjusting the contention window using a fair queueing algorithm. In [30], Tassiulas and Sarkar addressed the max-min fairness with a multi-channel contention model that is different from the CSMA/CA model used in our study.

Many algorithms described above require additional procedures that calculate and exchange the fair shares of the wireless links, or require the maintenance of information about local network topology and flow status. These requirements diminish the flexibility of the algorithms. In addition, none of the existing algorithm promises to provide fairness in a very fine granularity, which is essentially caused by the hidden terminal problem. In this study, based on a thorough study on all different types of contentions, we design our weighted bandwidth allocation protocol that does not have the above defects, and it is able to provide a foundation for implementing service differentiation and rate assurance in the upper layer. In our protocol, a special control packet called RRTS (Request-for-RTS) is used to help provide fairness in a fine granularity. It should be mentioned that we are not the first one who introduce the RRTS control packet. RRTS is initially introduced in [35] to solve contention problems in multihop wireless networks. In this study, we improve it further and use it together with packet labeling to achieve weighted fair scheduling.

2.2 Contention among MAC Flows

We know that in CSMA/CA contention occurs between two MAC flows when either the sender or the receiver of one flow is within the transmission range of either the sender or the receiver of the other flow. The MAC flow on a wireless link (i, j) is denoted as $f_{i,j}$, whose contending flows can be classified into five types, as shown in Fig. 2-1. Operations in PPS are designed differently and specifically for them.

The first type of contending flows includes those whose senders are also i. An example is $f_{i,k}$ in the figure. $f_{i,k}$ contends with $f_{i,j}$ because node i cannot send two packets simultaneously.

The second type includes those MAC flows whose senders are in the transmission range of node i. Their receivers may or may not be in the transmission range of i. An example is $f_{a,b}$ in the figure. It contends with $f_{i,j}$ because RTS/DATA sent by a can reach i and cause radio collision when i is receiving CTS/ACK from j.

The third type includes those MAC flows whose senders are in the transmission range of node j. Their receivers may or may not be in the transmission range of j. An example is $f_{e,m}$ in the figure. It contends with $f_{i,j}$ because RTS/DATA sent by e can reach j and cause radio collision when j is receiving RTS/DATA from i.

The fourth type includes those MAC flows whose senders are outside the transmission ranges of both i and j, but receivers lie within the transmission range of i. An example is $f_{d,c}$ in the figure. It contends with $f_{i,j}$ because CTS/ACK packets from c can reach node i and cause radio collision when i is receiving CTS/ACK from j. It is well known that the most severe unfairness occurs under this scenario.

The fifth type includes those MAC flows whose senders are outside the transmission ranges of both i and j, but receivers lie within the transmission range of j. An example is $f_{h,g}$ in the figure. It contends with $f_{i,j}$ because CTS/ACK packets from g can reach node j and cause radio collision when j is receiving RTS/DATA from i. Because the senders of these two flows cannot overhear each other, it complicates any solution that relies on the senders to overhear information from their counterpart.

Finally, it is easy to see that any flow not belonging to the above five types does not contend with $f_{i,j}$.

2.3 Proportional Packet Scheduling

Our objective is to design an enhanced CSMA/CA protocol that allocates bandwidth to MAC flows in proportion to their weights. The new media contention technique, called *proportional packet scheduling* (PPS), tries to distribute bandwidth in such a way that, for any two contending MAC flows, the ratio of their rates are equal to the ratio of their weights.

Consider a MAC flow $f_{i,j}$ over a wireless link (i, j). Let $w_{i,j}$ be the weight of the flow. The *mean rate* of the flow is defined as its rate divided by its weight. The goal of PPS is to equalize the mean rates of contending flows. The sender of $f_{i,j}$ maintains a separate packet queue for the flow. It also keeps a counter, denoted as $c_{i,j}$, providing an indirect, discrete measurement for the mean rate of the flow, which we will discuss in details shortly.

The basic idea behind PPS is that, in order to equalize the mean rates of contending flows, we should always give the highest priority in media access to the flow whose current mean rate is the smallest. PPS tries to maximize the smallest flow rate in the network, then maximize the second smallest, and so on. In order to maximize spatial channel reuse, PPS never leaves local channel idle when there is a backlogged MAC flow. The design of PPS follows two rules below. Assume all MAC flows under discussion are backlogged.

1. A MAC flow should occupy the channel for transmission if it has the smallest mean rate among all contending flows or if the channel is idle.
2. A MAC flow should not compete for media access if its mean rate is not the smallest among its contending flows and the channel is busy.

First, we describe how the sender of a MAC flow maintains its counter. Assume the clocks at all nodes are loosely synchronized. Time is divided into periods. At the beginning of each period, all counters are initialized to zero. The counter $c_{i,j}$ is increased by one for every $w_{i,j} \cdot l$ bits of data transmitted in flow $f_{i,j}$ over link (i, j), where l is a system-wide parameter whose impact will be discussed later. Let Δt be the time passed since the beginning of the current period. The rate of the flow $f_{i,j}$ is about $\frac{c_{i,j} \cdot w_{i,j} \cdot l}{\Delta t}$. The mean rate is $\frac{c_{i,j} \cdot w_{i,j} \cdot l}{\Delta t \cdot w_{i,j}} = c_{i,j} \frac{l}{\Delta t}$. Hence, $c_{i,j}$ can serve as a discrete measurement of mean rate in units of $\frac{l}{\Delta t}$. The maximum rounding error [1] in this discrete measurement is $\frac{l}{\Delta t}$, which diminishes at the end of each period if the period is sufficiently long with respect to l. Therefore, if the counter values of contending flows are equalized to the end of each period, the mean rates of the flows are also equalized.

Next, we describe how the senders of contending flows coordinate the order of transmissions based on their counter values. Consider an arbitrary MAC flow $f_{i,j}$. Below we discuss what information the sender and the receiver will gather and what operations they will perform based on that information.

Let $n_{i,j}$ be the number of bits yet to be transmitted over (i, j) before $c_{i,j}$ is increased by one. When a data packet is sent from i to j, RTS/CTS/DATA/ACK piggyback both $c_{i,j}$ and $n_{i,j}$. For all different types of contending flows of $f_{i,j}$ shown in Fig. 2-1, neighbors of either i or j, such as a, c, e and g, will learn $c_{i,j}$ and $n_{i,j}$ through overhearing. Similarly, i will learn the counter values of $f_{a,b}$ and $f_{d,c}$, and j will learn the counter values of $f_{e,m}$ and $f_{g,h}$, as well as the number of bits yet to be transmitted before each counter is increased by one. Therefore, the sender i of flow $f_{i,j}$ only knows the information for some contending flows, and the receiver knows the information for the rest contending flows.

To help understand packet labels, Fig. 2-2 shows two examples. When scheduling packet transmissions the goal is to give packets with smaller labels higher priorities, and at the same time, the channel should be fully utilized. In Fig. 2-2(a), a is sending a packet

[1] The source of the rounding error is due to the fact that $c_{i,j}$ is increased by one only after $w_{i,j} \cdot l$ bits of data are transmitted.

with a label of 2, and c is sending a packet with a label of 3. Thus, node c should not use the channel, because node a is sending a packet with a smaller label, which means $f_{a,b}$ currently has a lower mean rate. A different situation is showed in Fig. 2-2(b), where flows 1-4 mutually contend with each other and flow 5 contends only with flow 4. Flow 4 cannot send its packet with a label of 4, because all the labels of its contenders are still 3. However, for flow 5, though its contender, flow 4, has a smaller label, it should ignore this and continue to send, otherwise, the channel would be locally idle and spatial channel reuse is not utilized. The operations of PPS are described as follows:

When i is the sender of multiple flows such as $f_{i,j}$ and $f_{i,k}$, i always schedules the flow with the minimum counter. Without losing generality, let this flow be $f_{i,j}$. Node i will refrain from accessing media if it overhears that a contending flow with a smaller or equal counter is transmitting. It will *attempt to access media* with RTS if $f_{i,j}$ has the smallest counter among the contending flows that it knows or if it senses an idle channel for a certain period of time. After RTS is successfully delivered to the receiver j through a CSMA/CA protocol, there are two possible cases.

Case 1: If $f_{i,j}$ has the smallest counter among the contending flows that j knows or if j also senses an idle channel for a certain period of time, j responds with CTS. After that, i and j exchange DATA/ACK. Node i will continue to send j a sequence of data packets (called *transmission burst*) until $c_{i,j}$ is increased by one.

Case 2: By overhearing, if j knows that a contending flow ($f_{e,m}$ or $f_{g,h}$ in Fig. 2-1) with a smaller or equal counter is currently in a transmission burst, j will reject i's RTS with a new control message REJ, carrying the contending flow's counter and the number of bits to be transmitted before that counter will be increased by one. Note that REJ should be sent after j's current NAV expires in order to avoid interfering with concurrent transmissions. Based on the information received in REJ, i sets an appropriate timer and will re-attempt transmission after timeout.

2.4 Properties

When the sender of a flow is idle, the flow is said to be *inactive*. When the sender of a flow is transmitting data packets, the flow is said to be *active* for a transmission burst. The transmission burst is said to be *preempted* if a contending flow stops the burst and starts its own transmission of data packets. Preemption changes the right of transmission from one flow to another. It does not mean that the preempted burst has wasted its effort; those packets in burst prior to preemption have been delivered. The length of each transmission burst is controlled by the system parameter l. PPS reduces radio collisions by serializing transmission bursts based on the flows' counter values. Radio collision mostly happens between transmission bursts, and its impact diminishes when the burst length (i.e., l) increases. Even when l is one packet, the overall throughput of PPS is still higher than that of CSMA/CA because radio collision is reduced as many flows with larger counters refrain from accessing media. The throughput of PPS improves when l is larger. However, the rate of improvement diminishes to zero once l is sufficiently large.

The above design of PPS has the following properties.

Property 1. When $f_{i,j}$ is active, if it has the smallest counter value $c_{i,j}$ among all contending flows, then its transmission burst will not be preempted.

Proof: We examine the five types of contending flows. The senders of $f_{i,k}$, $f_{a,b}$ and $f_{e,m}$ will refrain from accessing media because they know that $f_{i,j}$ has a smaller counter. In particular, a learns the information by overhearing RTS/DATA from i, and e learns the information by overhearing CTS/ACK from j.

Next, consider $f_{d,c}$ and $f_{h,g}$. Their senders, d and h, may attempt to access media because they cannot overhear the counter value of $f_{i,j}$ that is piggybacked in transmissions made by i and j. By a CSMA/CA protocol, they will eventually succeed in delivering RTS to their receivers, c and g, who know that $f_{i,j}$'s counter is smaller. The receivers will reject RTS by replying REJ back to the senders. Without overhearing a transmission carrying a smaller counter, i will continue its transmission burst via CSMA/CA. $\square$

Property 2: When $f_{i,j}$ is inactive, if it has the smallest counter value $c_{i,j}$ among all contending flows, then it will preempt the transmission burst of an active contending flow whose counter is larger.

Proof: Once i finds that $f_{i,j}$ has the smallest counter among the contending flows that it knows, it will attempt to access media by delivering RTS to j. When j also finds that $f_{i,j}$ has the smallest counter among the contending flows that it knows, it will reply CTS. Then i will send DATA, starting its transmission burst for flow $f_{i,j}$, which interrupts the current transmission burst of an active contending flow (whose counter is larger).

Now let's examine the interrupted flow. If it is $f_{a,b}$ or $f_{e,m}$, its sender will overhear the counter value of $f_{i,j}$, piggybacked in the transmissions made between i and j. Once the sender finds that its counter is larger, it will not try to resume its transmission burst. Next, if the interrupted flow is $f_{d,c}$ or $f_{h,g}$, its receiver will overhear the counter value of $f_{i,j}$, but its sender will not. The sender will try to resume the transmission burst by sending RTS to the receiver, which is d or h. Upon receiving RTS, the receiver will reply REJ, stopping the sender from further trying. □

Property 3: When $f_{i,j}$ is inactive, if none of the contending flows is active, then it will become active.

Proof: By the design of PPS, the sender will attempt to access media if the local channel is idle for a certain period of time. Upon receiving RTS, the receiver will reply CTS if it also senses an idle channel for a certain period of time. Consequently the sender will become active and transmit data packets. □

Property 1 and 2 ensure that the counter values of all MAC flows in the same bottleneck channel will not differ more than one. The reason is that, once a flow's counter is greater by one than another flow's counter, the latter will preempt the former to increase its counter. Property 3 ensures that the channel capacity is fully utilized. Together they show that PPS achieves weighted bandwidth allocation with an error corresponding to the difference in the counter values, which is at most one, representing a data rate of $\frac{l}{T}$, where l is the number of bits transmitted before a counter is increased by one and T is the PPS period. Both l and T are system parameters.

More significantly, when we increase the PPS period, the flow rates resulted from PPS indefinitely approach towards weighted max-min fairness: the mean rate $m_{i,j}$ of any MAC flow $f_{i,j}$ cannot be increased without decreasing the mean rate $m_{i',j'}$ of another MAC flow $f_{i',j'}$, for which $m_{i',j'} \leq m_{i,j}$.

Theorem 1. When increasing the PPS period, the flow rates by PPS approach weighted max-min fairness.

To prove the theorem, we need show that the mean rate of any MAC flow cannot be increased without decreasing the mean rate of any another MAC flow that has a lower mean rate. Let $c_{i,j}(t)$ denote the counter value of MAC flow $f_{i,j}$ at time t and let $mr_{i,j}(t)$ denote the mean rate of $f_{i,j}$ from time zero to t. By PPS, we have

$$\frac{c_{i,j}(t) \cdot l}{t} \leq mr_{i,j}(t) < \frac{(c_{i,j}(t)+1) \cdot l}{t} \tag{2–1}$$

Now we first prove the following lemma.

Lemma 1. At any arbitrary time t within a PPS period, for any MAC flow $f_{i,j}$, its mean rate $mr_{i,j}(t)$ cannot be increased without decreasing another flow $f_{i',j'}$ that has a mean rate $mr_{i',j'}(t) \leq mr_{i,j}(t) + \frac{2 \cdot l}{t}$.

proof: We prove it by contradiction. Suppose there exists a flow $f_{i,j}$ such that its mean rate $mr_{i,j}(t)$ can be increased without decreasing the mean rate of any other flow with a mean rate lower than or equal to $mr_{i,j}(t) + \frac{2 \cdot l}{t}$. By Property 3, under PPS, there is no extra unused bandwidth for $f_{i,j}$ to increase its rate. In other words, increasing $mr_{i,j}(t)$ means that the mean rate of anther flow $mr_{i',j'}(t)$ has to be decreased, and by our assumption, $mr_{i',j'}(t) > mr_{i,j}(t) + \frac{2 \cdot l}{t}$. Then, by inequality 2–1, we have $c_{i',j'}(t) \geq c_{i,j}(t) + 2$. However, by Property 2, $f_{i,j}$ will not let $f_{i',j'}$ has a counter value that is greater than its own counter value by two, unless there is another flow $f_{i'',j''}$ contending with $f_{i,j}$ and $c_{i'',j''}(t) \leq c_{i,j}(t)$. Therefore we have $mr_{i'',j''}(t) < mr_{i,j}(t) + \frac{l}{t} < mr_{i,j}(t) + \frac{2 \cdot l}{t}$. Thus, increasing $mr_{i,j}(t)$ will cause $mr_{i'',j''}(t)$ to decrease, which is contradicting with our assumption. $\square$

The proof of Theorem 1 can be easily drawn from Lemma 1. Let time t to be the end of each PPS period. As the PPS period is increasing, we have $t \to \infty$ and thus $\frac{2 \cdot l}{t} \to 0$. Then, the mean rate condition in Lemma 1 turns to be the same as weighted max-min fairness.

2.5 Enhancing PPS by Request-for-RTS Packets

In Fig. 2-3 (a), suppose $f_{d,c}$ contends with both $f_{i,j}$ and $f_{x,y}$, but $f_{i,j}$ and $f_{x,y}$ do not contend with each other. Suppose the counter of $f_{x,y}$ is smaller than that of $f_{d,c}$ and the counter of $f_{d,c}$ is smaller than that of $f_{i,j}$. Flow $f_{x,y}$ transmits because it has the smallest counter. $f_{i,j}$ also transmits because it senses an idle channel. $f_{x,y}$ stops when its counter becomes larger than that of $f_{d,c}$. At this time, $f_{d,c}$ needs to preempt $f_{i,j}$. However, if d cannot overhear the transmissions by i, its RTS packets are very likely to collide with the DATA packets from i [21]. The proper time for d to send RTS is after the current RTS/CTS/DATA/ACK exchange between i and j completes and before the next exchange begins. This is the time when the NAV at c expires, assuming PPS is implemented on top of IEEE 802.11 DCF. As an enhancement to PPS, when the NAV at c expires, c sends d a RRTS control packet [2] to solicit RTS. In general, when a node's NTV expires, if the node is the receiver of a flow whose counter is smaller than the counter carried in the previous RTS/CTS/DATA/ACK exchange in the channel, the node sends RRTS after a small random delay (to prevent collision with another RRTS).

[2] RRTS was originally proposed in [35] to address the hidden terminal problem. We use it for a different purpose in our study.

Fig. 2-3 (b) shows a different scenario where $f_{h,g}$ tries to preempt $f_{i,j}$. If g is not in the transmission range of i, even when DATA is transmitting from i to j, node g is able to receive RTS from h. However, g cannot reply ACK because it is waiting on NAV. To solve this problem, after the NAV expires, g sends a RRTS packet to solicit a new RTS from h.

2.6 State of the Art and Our Contribution

PPS achieves weighted bandwidth allocation, and it is worth emphasizing that PPS is particularly suitable for dynamic weight adaptation, which is introduced in the next chapter. First, the operations in PPS are fully localized; a node only uses its local information and the information it currently overhears. It does not maintain the state information of its contending flows (which can cause problem if the information becomes stale). Second, it achieves *provable* weighted maxmin fairness in CSMA/CA networks with a bounded error that can be made arbitrarily small. We believe this is a strong result. Third, it does not assume a fixed channel capacity, and does not assume a static set of flows. It can work in a wireless environment where channel capacity evolves spatially/temporally and flows join and depart.

While a number of MAC protocols in the literature [29; 18; 19; 21–23] were designed to achieve fairness among contending MAC flows, to the best of our knowledge, our protocol is the first to achieve all three properties discussed above. In comparison, EMLM-FQ [21] requires each node to keep track of certain state information of contending flows and does not guarantee a tight bound on its approximate fairness. The max-min fairness achieved in [30] uses a multi-channel contention model that is different from the CSMA/CA model used in this study. The work in [19] requires each node to compute fair bandwidth shares for its own links and the nearby contending links, which in turn relies on the knowledge of neighborhood topology. To calculate fair shares, a node must know the set of contending flows that are currently backlogged, from which the local contention cliques can be computed. In addition, it assumes all cliques have equal, fixed capacity.

These requirements make the protocol less suitable for a dynamic wireless environment where the set of backlogged flows, as well as the channel capacity, may constantly change.

PPS also has other advantages: It is much simpler, easy to implement and analyze, and reduces radio collision. It does not require modifying the backoff algorithm of the existing channel access protocols.

2.7 Performance Evaluation

In this section, we study the performance of PPS (proportional packet scheduling) through simulations. We implemented PPS in ns2 v2.32 [36], on top of IEEE 802.11 DCF. For comparison, we also implemented an influential packet scheduling protocol [19] (referred as AdjConWin) and IEEE 802.11e EDCA [10]. AdjConWin achieves fairness among MAC flows by dynamically adjusting their minimum contention windows. IEEE 802.11e EDCA has four access categories: background, best effort, video, and voice.

If not specified otherwise, the default simulation parameters are given as follows: The transmission rate is set to be 11Mbps based on IEEE 802.11b, and each packet is 1000 bytes long. The PPS period is 1 second. The parameter l is set to be the length of five packets. Besides what have been stated above, other parameters (such that those for 802.11 DCF) use the default values set by ns2 according to the protocol standards.

Since AdjConWin is designed to achieve fairness, in order to make the simulation results comparable, we set the weights of all flows in PPS to be one. We first perform simulations on the simple topology in Fig. 2-4, where a is not in the transmission range of c, but b is. Suppose two MAC flows, $f_{a,b}$ and $f_{c,d}$, are both backlogged. Under IEEE 802.11 DCF, it is well known that severe unfairness can happen in this scenario because $f_{a,b}$ can hardly acquire the channel [19; 23]. We run simulations for fifty seconds under DCF, EDCA, AdjConWin, and PPS, respectively. The flows' rates are still measured in packets per second. The results are shown in Table 2-1. Clearly, DCF causes severe unfairness, with $f_{c,d}$ acquiring most of the channel capacity and $f_{a,b}$ receiving little. For EDCA, we assign $f_{a,b}$ to the video access category and $f_{c,d}$ to the best-effort category in order to give $f_{a,b}$ more bandwidth, which however causes unfairness the other way around. Other category assignments will also lead to unfairness, which is understandable because EDCA is not designed for fairness or weighted bandwidth allocation. Both AdjConWin and PPS can achieve fairness between the two flows.

There are some important differences between AdjConWin and PPS, which are elaborated in Section 2.6. Because of its fully localized operations, PPS is equally effective under dynamic setting where the set of MAC flows, as well as the weights of the flows, change over time. This is critical for DWA, the technique introduced in the next chapter based on PPS, because a MAC flow for a service class on a wireless link will be inactive when its queue is empty and become active again when its queue is backlogged, which happens when end-to-end flows join or depart from the network. Unlike PPS, AdjConWin is less effective with a dynamic set of MAC flows because it requires each node to centrally compute the local contention cliques and the fair bandwidth shares for its own flows as well as nearby contending flows, under the assumption that each clique has an equal, fixed capacity. The senders of contending flows can be three hops away. In a dynamic environment, the overhead will be very high if all nodes constantly exchange their current flow information in order to update the correct values for fair bandwidth shares. If such update is not done, the network performance suffers. We perform simulations on the five-flow topology in Fig. 2-5, where each dashed ellipse contains nodes can hear each other's transmission. Suppose flow 1's queue is empty from time 10 to 20, flow 2's queue is empty from time 20 to 30, flow 3's queue is empty from time 30 to 40, flow 5's queue is empty from time 40 to 50, and all queues are otherwise backlogged. Fig. 2-6 and 2-7 show the flow rates under AdjConWin and PPS, respectively. The average flow rates under PPS are much higher because AdjConWin requires close coordination among contending nodes and such coordination breaks down with a dynamic set of flows, while PPS relies on fully localized operations.

To support DWA, each MAC flow must be allowed to *independently and locally* change its weight. Yet, without explicit coordination among contending flows, the bandwidth allocation must follow such changes to maintain the invariant that the flow rates are proportional to their current weights. We perform a simulation on the topology of Fig. 2-5, where all flows begin with weight one and flow 1 changes its weight to 2 at time 15 and then to 4 at time 30. The result in Fig. 2-8 shows that PPS maintains weighted bandwidth allocation under dynamic weight. AdjConWin does not consider flow weights, let alone dynamic weight (which requires fully localized operations).

2.8 Summary

In this chapter we proposed a new technique called Proportional Packet Scheduling (PPS) to solve the weighted fair bandwidth allocation problem in the MAC layer of multihop CSMA/CA-like wireless networks. PPS schedules packets by labels such that the flow with the lowest mean rate in a contending region is guaranteed to transmit first. The advantages of this technique include: 1) Weighted max-min fairness among MAC flows is achieved without the need to exchange knowledge of flow status and network topology; 2) Weighted bandwidth allocation is realized in a very fine granularity; 3)Dynamic weight change is well supported. The performance of PPS is studied through simulations. In Chapter 3, we will introduce a new technique established on top of PPS to provide service differentiation and rate assurance among end-to-end flows in multihop wireless networks.

Table 2-1. Flow rates (in packets per second) on the two-flow topology

	$f_{a,b}$	$f_{c,d}$
802.11 DCF	64.6	381.0
802.11e EDCA	347.8	193.1
AdjConWin	232.0	229.5
PPS	227.9	228.8

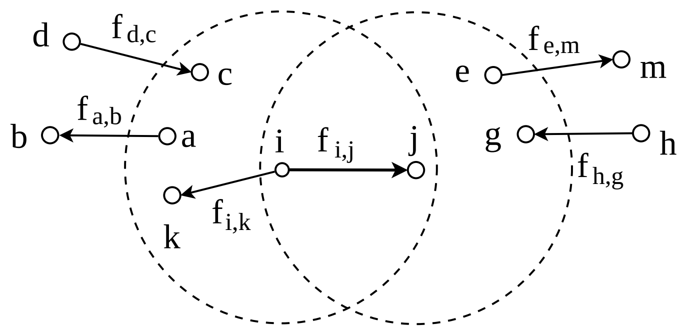


Figure 2-1. Five types of contending flows of $f_{i,j}$.

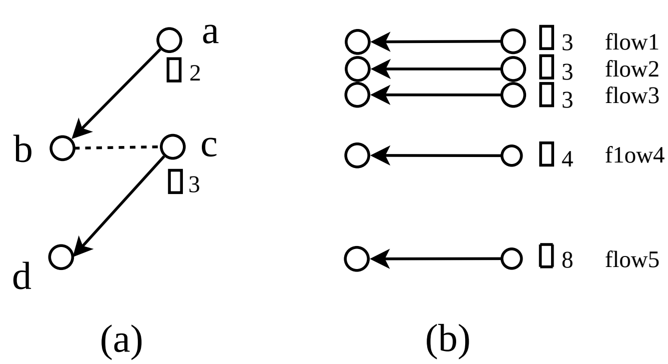


Figure 2-2. Packet label priority

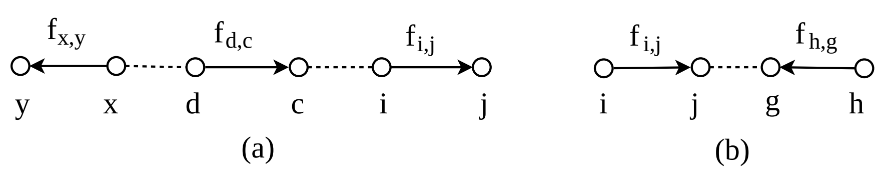


Figure 2-3. Request-for-RTS helps preemption

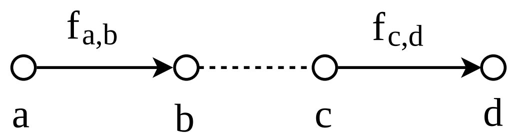


Figure 2-4. Two-flow topology

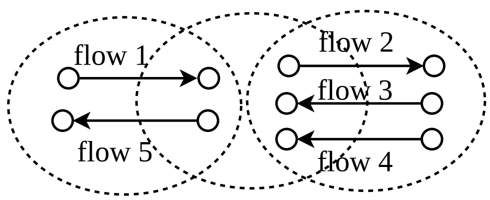


Figure 2-5. Five-flow topology

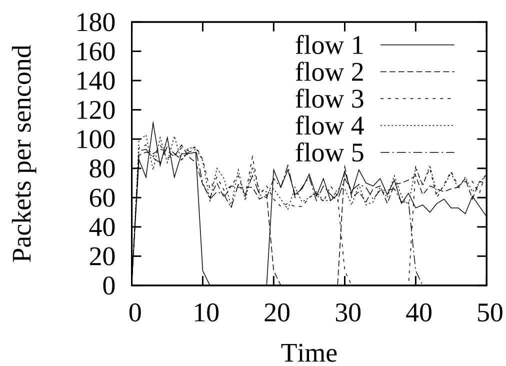


Figure 2-6. Protocol of AdjConWin

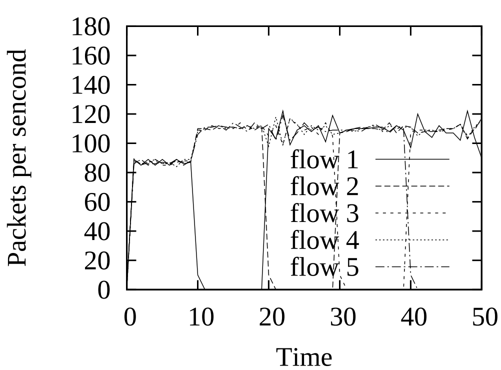


Figure 2-7. Proportional Packet Scheduling

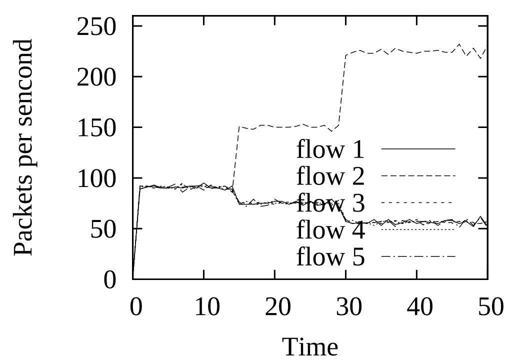


Figure 2-8. Proportional Packet Scheduling supports dynamic weight

CHAPTER 3
END-TO-END SERVICE DIFFERENTIATION AND RATE ASSURANCE

Based on PPS, the weighted bandwidth allocation technique introduced in the Chapter 2, we design a novel technique called *dynamic weight adaption with floor and ceiling* (DWA) to provide service differentiation and rate assurance among a dynamic set of end-to-end flows in multihop wireless networks.

3.1 Related Work

Providing QoS features in multihop wireless networks is a relatively less explored subject. Kang and Mutka [7] proposed a simple approach for service differentiation, in which different classes of packets are assigned with different backoff policies. Ahn et al. proposed SWAN [8], which provides service differentiation between best-effort TCP and real-time UDP. In SWAN, a local AIMD rate control scheme is utilized to shape best-effort traffic. For real-time traffic, probe packets are used to achieve admission control and explicit congestion notifications are exploited to dynamically regulate traffic changes. In their other works [37; 6], different contention window sizes are applied to packets from real-time and best-effort traffic respectively in order to achieve service differentiation. A virtual MAC algorithm is introduced to estimate the current channel condition, and admission control is made accordingly. However, these mechanisms are not designed for multihop networks. Karenos et al. [9] designed a rate control framework in sensor networks. None of these works achieves the rate assurance objective and the bandwidth differentiation objective as defined in our study.

Some research has been dedicated to support bandwidth guarantee at either the link layer or the network layer. Lin and Gerla [38] proposed a real-time protocol called MACA/PR to provide bandwidth guarantee over single-hop links, which emulates TDMA in CSMA and requires each node to maintain a reservation table that keeps track of dynamic time reservations of real-time flows. This emulation may lead to inefficient transmission schedules. Shah et al. [2] proposed an admission control and dynamic bandwidth management scheme to provide fairness and soft rate guarantee in wireless networks. However, the algorithm was designed for a single-hop network model where all nodes can hear each other. Another approach that addresses dynamic bandwidth allocation in single-hop networks was proposed by Chou and Shin in [3]. Banchs et al. [1] proposed a mechanism that provides rate assurance to wireless LANs on a per staion basis, where desired bandwidth is allocated by adaptively adjusting contention window size. However, this mechanism cannot be applied to multihop wireless networks. Sarkar and Tassiulas [5] designed a distributed algorithm for end-to-end fairness under a multi-channel model where each node transmits at a different frequency. In the proposed mechanism, session rates are controlled by a token bucket algorithm, and tokens are generated according to one-hop neighbors in such a way that the rate limit in bottlenecks can spread out.

The major differences of our proposed QoS mechanism from the above mechanisms are as follows. First, our mechanism works for multihop end-to-end flows in ad hoc wireless networks. Second, we combine the concepts of service differentiation and rate assurance together, where flows are categorized into arbitrary multiple priorities, and flow rate requirements are assured based on their priorities. Thirdly, and the most important, we emphasize the fairness issue when implementing QoS features. We argue that, without fairness problems solved, service differentiation itself is less meaningful. It is not acceptable that a flow in a lower priority gets a better quality of service than another flow in a higher priority just because their link-layer contentions are unfair. That is why we build our QoS mechanism based on our weighted fair scheduling protocol.

3.2 Objectives

We classify end-to-end traffic in a multihop wireless network into two broad categories: *best-effort flows* and *QoS* (quality of service) *flows*. Each QoS flow has a minimum rate requirement, but it may send data at a higher rate if extra bandwidth is available. The rate requirement may be *soft* or *hard*. With a soft requirement, we assume the application is able to adapt to live with a lower-than-expected rate, for example, by compressing data before sending. With a hard requirement, we assume the application will terminate the flow when the rate is too low, and it may attempt to re-establish the flow after a timeout period, which may be doubled for each failed attempt until the application gives up. If the network cannot satisfy a flow's minimum rate requirement, it will continue serving the flow to the best it can. It is up to the application to decide whether adaptation should be performed or the flow should be terminated. We assume each flow has a routing path established by a routing protocol; the subject of optimal routing is beyond the scope of this study.

The network provides a number of differentiated service classes, each having a different priority. Best-effort flows are assigned to the *best-effort service class* that has the lowest priority. QoS flows are assigned to other classes. The priority of a QoS flow is equal to the priority of the service class to which the flow is assigned. When the minimum rate requirement of a QoS flow is satisfied, we say the network *supports* the flow. We have three objectives.

- **Rate Assurance Objective**: When the bandwidth available in the network cannot support all end-to-end QoS flows, we first try to support all flows with the highest priority. When that is done, we then try to support all flows with the second highest priority. This policy repeats until there is no longer enough bandwidth to support flows of a certain priority. In other words, when two end-to-end flows contend for bandwidth at a common bottleneck location (channel is spatially reused), the lower-priority flow will be supported only after the higher-priority is supported.

- **Bandwidth Differentiation Objective**: After the minimum rate requirements of all QoS flows are satisfied, if there is extra bandwidth left, the remaining bandwidth should be distributed to flows in proportion to their bandwidth demands.[1] However, if there is not enough bandwidth to support QoS flows of certain low priorities, after QoS flows of higher priorities are supported for their minimum requirements, the remaining bandwidth should be distributed to the unsupported flows in proportion to the product of *minimum rate requirement* and *differentiating factor*. Each service class is pre-assigned a differentiating factor, which is larger for a higher-priority class; a flow's differentiating factor is equal to that of its class. Hence, the above design not only considers each flow's bandwidth demand but also takes the priority into consideration.

- **No Starvation and Maximum Utilization Objective**: While QoS flows are supported in the order of their priorities, low-priority flows (including best-effort flows) should not be starved. Hence, the bandwidth consumed by each service class should be limited to a certain fraction of the available bandwidth in the network. This fraction is proportional to both the differentiating factor of the class and the combined rate requirement of all flows in the class. Consequently, the fraction of bandwidth available to a service class is not only dynamic (due to flow join/departure) but also easily configurable (by changing the differentiating factor). No bandwidth should be wasted; bandwidth assigned to but not used by any flow should be automatically picked up by other flows based on the distribution policies specified in the previous objectives.

It should be noted that there are other ways of defining the objectives. For example, instead of distributing the remaining bandwidth (after rate assurance) among end-to-end flows based on bandwidth demand and differentiating factor, one may prefer to *optimize the aggregate throughput of the network* under the constraint of rate assurance. Optimizing the aggregate throughput will naturally prefer short flows over long flows and disregard the priorities. Proportional fairness [39] can be used to address the

[1] It is reasonable that a video stream is assigned more *extra* bandwidth than an audio stream.

problem of short-flow preference. However, integrating proportional fairness with multiple prioritized service classes will complicate our system design. Focusing on rate assurance and service differentiation, we shall leave other design choices to future work and, in this study, introduce an end-to-end weight adaptation approach that distributes the network bandwidth based on the above three objectives.

3.3 Challenge

First, we examine how to satisfy the rate requirements of end-to-end flows in a wired network, where each link has a fixed capacity. While DiffServ [40; 41] does not provide per-flow bandwidth assurance, under the IntServ model [42] we can reserve a certain amount of bandwidth for a flow along its routing path by using per-flow weighted fair queueing [43]. At each link (i, j) on the path, node i assigns each passing flow a weight that is proportional to the flow's rate requirement. The rate requirements of the passing flows can be satisfied as long as their sum does not exceed the link capacity.

Next, we try to map the above solution to a multihop wireless network. We begin by considering only one QoS service class. While we still perform weighted fair queueing at each link and assign every passing flow a weight that is proportional to the flow's rate requirement, there is a fundamental difference between wireless networks and wired networks. A wireless link does not have a fixed capacity. It shares a common channel with nearby contending links. Even the capacity of the wireless channel may be dynamic due to interference and multi-rate links. The key for rate assurance is to distribute bandwidth to wireless links in such a way that each link receives a portion that is equal to or above the total rate requirement of all flows passing the link.

The problem becomes much more complicated if there are multiple service classes and the combined bandwidth demand exceeds the available bandwidth. Moreover, as end-to-end flows come and go and channel conditions change over time, we must adapt the bandwidth allocated to each wireless link, as well as the bandwidth allocated to each service class of the link, in order to contiguously support rate assurance and service differentiation in a changing environment. And this has to be done in a fully distributed fashion without reliance on any centralized entity.

3.4 Design Overview

We model a multihop wireless network as a set of MAC (one-hop) flows. Each wireless link carries one MAC flow for each service class. Two MAC flows contend with each other if they belong to the same link or two contending links. An end-to-end flow of a given priority is mapped to a sequence of MAC flows of the same priority along its routing path. A MAC flow of a given priority carries all end-to-end flows of the same priority that pass the link. The rate requirement of a MAC flow is the summation of the rate requirements of the end-to-end flows that are carried by the MAC flow.

End-to-end service differentiation and rate assurance require MAC-layer support. In particular, we need a protocol that allocates bandwidth to MAC flows in proportion to their weights. Consider two contending links, (i, j) and (a, b), each receiving a fair share of the channel capacity via a CSMA/CA protocol such as IEEE 802.11 DCF. Suppose two end-to-end flows of high priority pass through (i, j) and one end-to-end flow of low priority passes through (a, b). IEEE 802.11 DCF will limit the rate of either high-priority flow to half that of the low-priority flow.

For the high-priority flows to receive more bandwidth than the low-priority one, we need a MAC protocol that can flexibly redistribute bandwidth among MAC flows. In particular, we are interested in a protocol that allocates bandwidth to MAC flows in proportion to their weights. PPS introduced in Chapter 2 serves for this purpose perfectly. In the above example, with PPS, if we want the rate of a high-priority end-to-end flow to be twice that of a low-priority flow, we simply assign 4 as the weight of the MAC flow on (i, j) and 1 as the weight of the MAC flow on (a, b).

There are two levels of bandwidth distribution. At the first level, we perform weighted bandwidth allocation by PPS that is introduced in Chapter 2 to distribute the channel capacity among contending MAC flows. Whenever possible, each MAC flow should acquire enough bandwidth to support the end-to-end flows it carries, but not too much bandwidth that causes shortage for other MAC flows to support end-to-end flows they carry. Note that PPS can be easily applied to the case where each wireless link has multiple MAC flows. At the second level, we distribute the bandwidth acquired by each MAC flow to the end-to-end flows (that it carries) by weighted fair queueing. Both levels use weights, but they are independent of each other. The challenge is on the first level, because once enough bandwidth is acquired by a MAC flow, at the second level, we can simply use the end-to-end flows' rate requirements as weights and perform any classical weighted fair queueing algorithm [43] to assure every end-to-end flow's rate requirement is met (locally at this wireless link). Therefore, we will focus on the first level in the rest of this chapter.

While the above two-level bandwidth distribution architecture may appear to be a routine design, our novelty is in solving the problem of how to assign appropriate weights to MAC flows (at the first level) such that the objectives in Section 3.2 can be achieved in a dynamic, fully-distributed environment, where both flows and channel conditions may change and there is not an entity that has global network/traffic information.

To solve this problem, we propose a new technique called *dynamic weight adaptation with floor and ceiling*. The weight of each MAC flow adapts between a lower bound (called *floor*) and an upper bound (called *ceiling*). The floor is proportional to the rate requirement of the MAC flow, and the ceiling is proportional to the product of the rate requirement and the differentiating factor (which is larger for a MAC flow of higher priority, giving such a flow a higher ceiling). Because the rate requirement of a MAC flow may change as end-to-end flows come and go, the floor and the ceiling may change, too. Each MAC flow periodically adapts its weight between the floor and the ceiling as follow: The sender of the MAC flow measures its rate over each weight-adaptation period, which may be set the same as or different from the PPS period. If the measured rate is below the requirement of the MAC flow, the sender increases the weight of the flow at the end of each period until the ceiling is reached or the requirement is satisfied. If the measured rate is above the requirement, the sender decreases the weight at the end of each period until the floor is reached or the rate is equal to or lower than the requirement. By setting the ceiling and the floor appropriately, we can show that, when the weights of all MAC flows stabilize, the three objectives will be met.

3.5 DWA: Dynamic Weight Adaptation with Floor and Ceiling

We first define some notations. Let $f_{i,j}^k$ be the MAC flow on link (i, j) that carries the service class of priority k. For the best-effort service class, $k = 0$. For the QoS service classes, $k = 1, 2, 3, ...$ Let $w_{i,j}^k$ be the weight of flow $f_{i,j}^k$. Let $L_{i,j}^k$ and $H_{i,j}^k$ be the floor and the ceiling for $w_{i,j}^k$, respectively. Let $c_{i,j}^k$ be the counter (used in PPS), $r_{i,j}^k$ be the actual data rate, and $q_{i,j}^k$ be the rate requirement of the MAC flow $f_{i,j}^k$, respectively. Let d_k be the differentiating factor for priority k. $d_k > d_{k'}$ if $k > k'$. Let T be the duration of each weight-adaptation period.

$c_{i,j}^k$ is a measurement of the mean rate of flow $f_{i,j}^k$. From it, we can estimate $r_{i,j}^k$ if the weight-adaptation period is equal to the PPS period. At the end of each period, we calculate

$$r_{i,j}^k \approx \frac{c_{i,j}^k \times w_{i,j}^k \times l}{T}$$

because $c_{i,j}^k$ is increased by one for every $w_{i,j}^k \times l$ bits sent in flow $f_{i,j}^k$. If the weight-adaptation period is different from the PPS period, $r_{i,j}^k$ has to be measured separately by counting the actual number of bytes transmitted over the link in each weight-adaptation period.

To compute $q_{i,j}^k$, we need to examine two cases. First, if an end-to-end flow carried by $f_{i,j}^k$ has a backlogged queue at i, we should allocate sufficient bandwidth for $f_{i,j}^k$ to support the minimum rate requirement. Second, if the end-to-end flow does not have a backlogged queue and the arrival rate to the queue is smaller than its rate requirement, the flow must have an upstream bottleneck and we only need to allocate enough bandwidth to cover the arrival rate. The *effective rate requirement* of an end-to-end flow is equal to the minimum rate requirement in the first case and the arrival rate in the second case. We define $q_{i,j}^k$ as the summation of the effective rate requirements of all end-to-end flows carried by $f_{i,j}^k$.

We define the floor and the ceiling for the weight $w^k_{i,j}$ of a MAC flow $f^k_{i,j}$ as follows. When $k = 0$, the flow is best-effort and we set the floor and the ceiling to be a fixed small value. Namely, the weight of a best-effort MAC flow is fixed. When $k > 0$, its floor and ceiling are

$$L^k_{i,j} = \alpha \cdot q^k_{i,j}$$

$$H^k_{i,j} = \alpha \cdot d_k \cdot q^k_{i,j}$$

where α is a normalizing factor whose value can be set arbitrarily.

Weight $w^k_{i,j}$ is initialized to be $L^k_{i,j}$ and iteratively adjusted. At the end of each weight-adaptation period, if $r^k_{i,j} < q^k_{i,j}$, we increase $w^k_{i,j}$ by a percentage of β if it has not reached the ceiling yet.

$$w^k_{i,j} \leftarrow \min\{w^k_{i,j} \cdot (1+\beta), H^k_{i,j}\}$$

If $r^k_{i,j} > q^k_{i,j}$, we decrease $w^k_{i,j}$ by a percentage of β if it has not reached the floor yet.

$$w^k_{i,j} \leftarrow \max\{w^k_{i,j} \cdot (1-\beta), L^k_{i,j}\}$$

It is possible to adapt the value of β based on the gap between $r^k_{i,j}$ and $q^k_{i,j}$. But we found a constant value for β already worked very well in our simulations.

3.6 Properties

We show that the above design of dynamic weight adaptation is able to satisfy the three objectives. To facilitate the discussion, we introduce the concept of *normalized weight* for a MAC flow $f^k_{i,j}$, which is defined as

$$\overline{w}^k_{i,j} = \frac{w^k_{i,j}}{q^k_{i,j}}$$

It is the average weight per unit of rate requirement. Because $\frac{L^k_{i,j}}{q^k_{i,j}} \le \overline{w}^k_{i,j} \le \frac{H^k_{i,j}}{q^k_{i,j}}$, the lowest normalized rate for any MAC flow is α, and the highest normalized rate is $\alpha \cdot d_k$, which varies for different priorities. If a MAC flow has a higher normalized weight than another flow under the same contention condition, it will acquire a larger amount of bandwidth for each unit of its rate requirement and therefore has a better chance to satisfy its requirement.

First, consider two end-to-end flows contend for bandwidth at a common bottleneck. Suppose they pass two contending MAC flows, $f^{k}_{i,j}$ and $f^{k'}_{i',j'}$, in the same bottleneck channel, and $k > k'$. If the rates of the MAC flows are below the minimum requirements, they will both increase weights unless the ceilings are reached. Their ceilings are different. The largest normalized weight for $f^{k}_{i,j}$ (which is $\alpha \cdot d_k$) is larger than the largest normalized weight for $f^{k'}_{i',j'}$ (which is $\alpha \cdot d_{k'}$). Hence, $f^{k}_{i,j}$ is able to increase its weight further to acquire more bandwidth per unit of rate requirement than $f^{k'}_{i',j'}$. Consequently, if $f^{k'}_{i',j'}$ is supported, $f^{k}_{i,j}$ must also be supported, but if $f^{k}_{i,j}$ is supported, $f^{k'}_{i',j'}$ may or may not be supported (due to the constraint of lower ceiling). The rate assurance objective is met. Now, suppose neither $f^{k}_{i,j}$ nor $f^{k'}_{i',j'}$ can be supported. Their weights will both reach the ceilings, and the bandwidth allocation between them will be proportional to the product of the rate requirement and the differentiating factor. Due to the two-level bandwidth distribution architecture, the ratio of bandwidth allocations among MAC flows will be inherited by the end-to-end flows that the MAC flows carry. Hence, the second half of the bandwidth differentiation objective is satisfied.

Next, we study the case that the network has enough bandwidth to support the minimum rate requirements of all end-to-end flows. Consider contending MAC flows at an arbitrary location in the network. We first show that all flows will be supported. Any flow whose rate is larger than its minimum requirement will decrease its weight, giving up some bandwidth. If no flow uses more bandwidth than its minimum requirement, certainly every one will be supported. What happens if a flow has a higher rate than the minimum requirement even when its weight is reduced to the floor? Since the flow's normalized weight, now α, is the lowest among all, other flows must be receiving the same or more bandwidth for each unit of rate requirement. Hence, their rate requirements must have been satisfied as well. Now, if there is still extra bandwidth left, because all flows will reduce their weights to the floors in an effort to avoid taking more-than-minimally-required bandwidth, the extra bandwidth will be distributed based on the ratio of the floors, which are proportional to the rate requirements. Therefore, the first half of the bandwidth differentiation objective is met.

Finally, because the weight of each MAC flow has a ceiling ($\alpha \cdot d_k \cdot q^k_{i,j}$), it cannot indefinitely increase the fraction of channel bandwidth that it consumes. In other words, the bandwidth consumed by end-to-end flows in a certain priority class is limited at any location in the network; the maximum fraction of bandwidth they can consume is proportional to the differentiating factor d_k, which is configurable. Therefore, the no-starvation objective is also met. To maximally utilize the available bandwidth, the underlying MAC-layer scheduling protocol that implements weighted bandwidth allocation must be work-conserving, i.e., it must allow MAC flows to consume bandwidth left unused by other MAC flows.

3.7 Avoiding Packet Drops

An end-to-end flow may receive different amount of bandwidth from the links on its path. Let (i, j) be the bottleneck link and (k, i) be the link preceding the bottleneck. Because there is more bandwidth available upstream, i will receive more packets from k than it can forward to j. Its queue for the flow will be filled up and eventually overflowed, causing packet drops. We adopt the congestion avoidance scheme in [44], which allows the upstream node k to send a packet to i only when i has enough free space in the queue to hold the packet. Suppose the buffer space for the queue is slotted with each slot storing one packet. The residual buffer at node i changes when i receives or sends a packet. To keep the upstream node updated with i's buffer state, whenever i transmits a packet (RTS/CTS/DATA/ACK), it piggybacks its current buffer state in the frame header, for example, using one bit to indicate whether there is at least one free buffer slot. When the upstream node k overhears a packet from i, it caches the buffer state of i. If i's buffer is full, k will hold its packets and wait until overhearing new buffer state from i. Readers are referred to [44] for discussion on various issues such as failed overhearing.

3.8 Flow Dynamics and Channel Dynamics

The adaptive nature of the proposed bandwidth distribution scheme makes it suitable for a dynamic environment with a changing set of flows and evolving channel conditions. We illustrate the adaptation process by the following example. Consider a network with two QoS service classes and all end-to-end QoS flows having the same rate requirement. Suppose at one time each class has one end-to-end flow and the network is able to support both flows. The weights of all MAC flows are at their floors.

First, let another high-priority end-to-end flow join in the network. The flow source signals along the routing path to add its rate requirement to the high-priority MAC flows. The ceilings of the high-priority MAC flows are increased accordingly. Suppose at one location x the channel capacity is not sufficient to support all QoS flows. With their weights at the floors, the MAC flows of two priorities both find that they are not getting enough bandwidth. They will increase their weights. The low-priority flow has a lower ceiling and will stop increasing first. The high-priority flow will be able to get more bandwidth to satisfy the requirement.

Second, let the newly-joined end-to-end flow depart from the network. At location x the bandwidth for the departed flow will be inherited by the other high-priority end-to-end flow sharing the same MAC flow. Since it acquires more bandwidth than the rate requirement, the high-priority MAC flow will reduce its weight to the floor, giving away bandwidth to the low-priority flow.

Third, suppose the channel capacity at location x is decreased due to environmental noise, causing the actual data rates of both MAC flows to decrease below the rate requirements. Similar to the first scenario, their weights will adapt individually and independently, but because the high-priority flow has a higher ceiling, it will receive more bandwidth to meet its rate requirement.

3.9 Intra-flow Contention and Inter-flow Contention

One may question why we have not discussed intra-flow and inter-flow contentions [33] in bandwidth distribution among end-to-end flows. Consider an end-to-end flow that follows a path $k \rightarrow i \rightarrow j$ where the intra-flow contention between sub-flow $k \rightarrow i$ and sub-flow $i \rightarrow j$ may lead to one sub-flow grabbing more bandwidth than the other, while they should each have the same bandwidth. Consider another end-to-end flow that follows the same path. The inter-flow contention may assign the same bandwidth share to each end-to-end flow, while they should be assigned shares based on their rate requirements and priorities. Intra-flow and inter-flow contentions are a problem for random-access wireless networks. However, when we have a MAC-layer scheduling protocol (PPS, in Chapter 2) that achieves weighted bandwidth allocation and a congestion avoidance scheme [44] that prevents packet drops, these contentions can be solved by assigning appropriate weights to MAC flows, as we did in this section.

3.10 Performance Evaluation

We study the performance of our new technique through simulations. We implemented PPS (proportional packet scheduling) and DWA (dynamic weight adaptation) in ns2 v2.32 [36], on top of IEEE 802.11 DCF.

We show how well DWA can achieve rate assurance and bandwidth differentiation. If not specified otherwise, the default simulation parameters are given as follows: The transmission rate is set to be 11Mbps based on IEEE 802.11b, and each packet is 1000 bytes long. The PPS period is 2 seconds. The parameter l is set to be the length of five packets. The weight-adaptation period is 2 seconds. There are two QoS service classes for the lower priority 1 and the higher priority 2. Their differentiating factors are $d_1 = 2$ and $d_2 = 4$, respectively. β is 10%. Besides what have been stated above, other parameters (such that those for 802.11 DCF) use the default values set by ns2 according to the protocol standards. We also compare DWA with IEEE 802.11e EDCA [10], which has four access categories: background, best effort, video, and voice.

We evaluate DWA (which is implemented on top of PPS) using the network topology in Fig. 3-1 with a dynamic set of flows, whose rate requirements and priorities change over time. The network consists of 30 nodes that are randomly deployed in an area. The wireless links formed between nodes are shown in the figure, where the average degree of a node is 4.4. The diameter of the network is 7 hops. There are 24 multihop end-to-end flows, which cannot be shown in the figure. The source/destination nodes of each flow are randomly chosen. We perform simulations under four different settings. The results are shown in Fig. 3-2-3-8.

Setting A: Flows 0-7 are assigned to the best-effort service class, flows 8-15 to the QoS service class of priority 1, and flows 16-23 to the QoS service class of priority 2. All QoS flows have the same rate requirement of 10 pps (packets per second).

We first turn off DWA. The flow rates under IEEE 802.11 DCF are shown in Fig. 3-2. Flow rate is measured in the number of packets successfully transmitted per second (pps). The contention levels experienced by the randomly-generated flows are vastly different. Without additional mechanisms to compensate such difference, the flow rates achieved under 802.11 are quite unpredictable with some much higher than others.

We then turn on DWA. The simulation result is shown in Fig. 3-3. The network is able to satisfy the rate requirements of all QoS flows. Flows 4, 6 and 16 have much higher rates than others because their routing paths happen to have less contention with other flows. The rates of all QoS flows vary from one to another also because the flows experience different levels of contention on their paths. The variation among the rates of best-effort flows are due to the same reason.

Next we create 8 new flows in each service class. For new QoS flows, their rate requirements are still 10 pps. The result is shown in Fig. 3-4. For easy comparison, we reassigned flow ids such that the rates of best-effort flows are shown under ids 0-15, the rates of priority-1 flows are shown under 16-31, and the rates of priority-2 flows are shown under 32-47. After doubling the number of flows, the rate requirements of priority-2 flows can still be met, but those of priority-1 flows can no longer be met. The rates of priority-1 flows are not identical because their routing paths contend with different sets of other flows.

Finally, we let the new flows depart from the network, and the flow rates go back to Fig. 3-3.

Setting B: It is the same as Setting A except that we increase the rate requirements of all QoS flows to 20 pps, so that not all of them can be satisfied.

When DWA is turned on, the simulation result is shown in Fig. 3-5. The rate requirements of flows 16-23 (priority 2) are satisfied or nearly satisfied. Some of them receive slightly lower rates due to more intense contentions. The network cannot support the rate requirements of flows 8-15 (priority 1) even when their weights are adapted to the ceilings. Due to lower priority, their ceilings are lower than those of flows 16-23. But they have higher rates with DWA than without it (Fig. 3-2). The best-effort flows receive the remaining network bandwidth. By design, no best-effort flow is starved.

For the purpose of comparison, we perform the same simulation under IEEE 802.11e EDCA, with priority-2 flows assigned to the video access category, priority-1 flows assigned to the best-effort access category, and prority-0 flows assigned to the background category. The results are shown in Fig. 3-6, which shows that flows 16-23 (priority 2) takes network bandwidth aggressively, starving the other flows. EDCA gives fixed preference to higher-priority flows and lacks a fine-level rate control mechanism that not only allocates bandwidth based on priorities but also balance the minimum needs among all QoS flows.

The next two settings are designed to demonstrate the great flexibility of DWA in controlling the bandwidth distribution among end-to-end flows.

Setting C: Keeping the rate requirements to be 20 pps, we now reassign the flow priorities. Let flows 8-15 be priority 2, flows 16-23 be priority 1, and flows 0-7 be best-effort. The simulation result is shown in Fig. 3-7. Now the rate requirements of flows 8-15 (priority 2) are satisfied.

Setting D: We again reassign the flow priorities. Let flows 0-7 be priority 2, flows 8-15 be priority 1, and flows 16-23 be best-effort. The simulation result is shown in Fig. 3-8. The rate requirements of the new priority-2 flows, whose ids are 0-7, are satisfied.

3.11 Summary

In this chapter a novel technique is introduced for providing service differentiation and rate assurance to end-to-end flows in multihop wireless networks. We have three objectives: (a) the rate assurance objective, which requires that QoS flows are supported in the order of priorities; (b) the bandwidth differentiation objective, which requires that, beyond meeting the rate requirements, bandwidth is allocated to the flows in a differentiated manner, taking both bandwidth demand and priority into consideration; (c) the no-starvation objective, which requires that the bandwidth allocated to each service class is limited and configurable. Our major technique proposed to realize the above objectives is dynamic weight adaptation with floor and ceiling, which works in conjunction with proportional packet scheduling, a MAC-layer technique for weighted bandwidth allocation introduced in Chapter 2. The new technique is effective yet simple to implement, which is important for practical wireless systems. The performance is extensively evaluated by simulations, demonstrating the new capabilities that can be made available in future wireless networks.

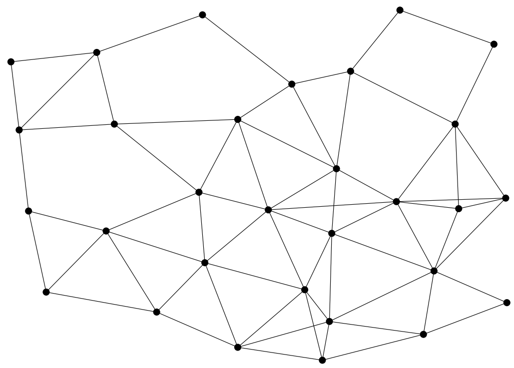


Figure 3-1. Network topology

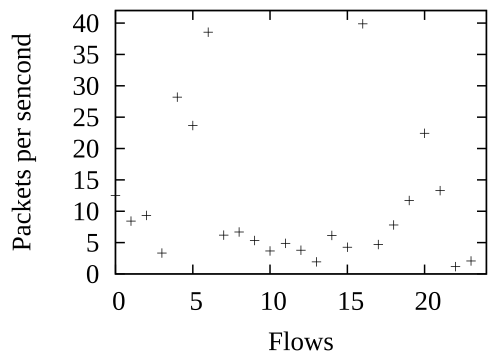


Figure 3-2. Setting A, IEEE 802.11 DCF

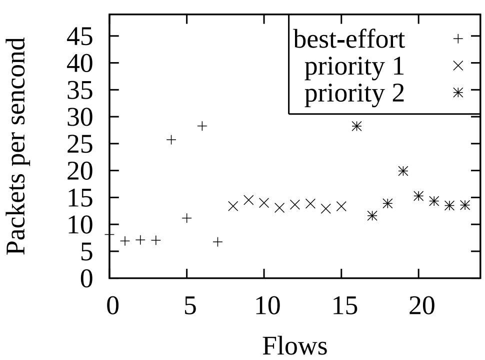


Figure 3-3. Setting A, DWA

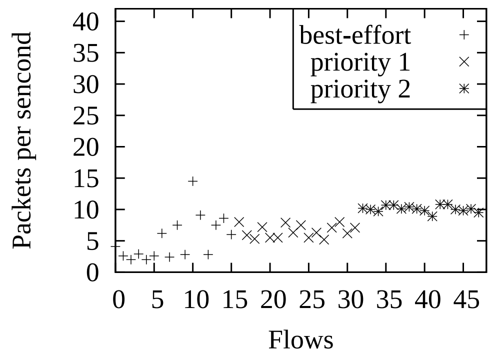


Figure 3-4. Setting A, DWA, doubling the number of flows

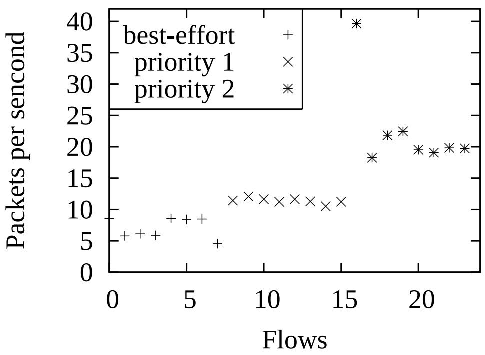


Figure 3-5. Setting B, DWA

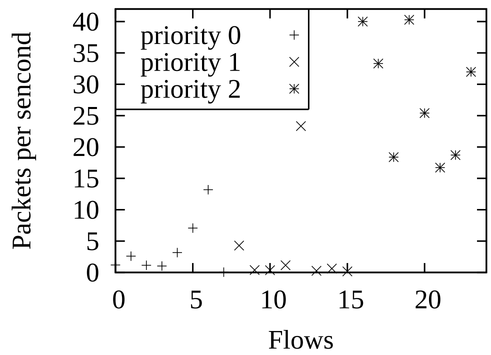


Figure 3-6. Setting B, EDCA

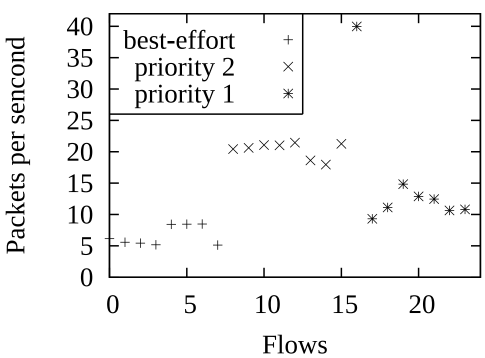


Figure 3-7. Setting C, DWA

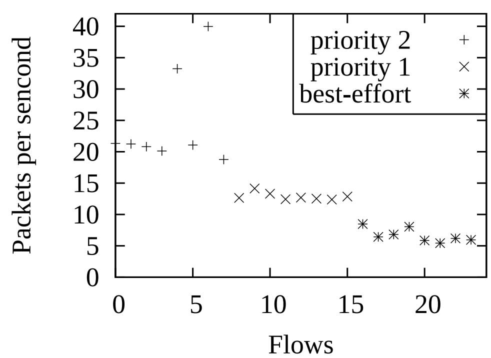


Figure 3-8. Setting D, DWA

CHAPTER 4
FAIR BANDWIDTH ALLOCATION IN CSMA/CA NETWORKS

In this chapter, we demonstrate that CSMA/CA networks, including IEEE 802.11 networks, exhibit severe fairness problem in many scenarios, where some hosts obtain most of the channel's bandwidth while others starve. Most existing solutions require nodes to overhear transmissions made by contending nodes and, based on the overheard information, adjust local rates to achieve fairness among all contending links. Their underlying assumption is that transmissions made by contending nodes can be overheard. However, this assumption holds only when the transmission range is equal to the carrier sensing range, which is not true in most real networks. As our study reveals, the overhearing-based solutions, as well as several non-overhearing AIMD solutions, cannot achieve MAC-layer fairness in various settings. We propose a new rate control protocol, called PISD (Proportional Increase Synchronized multiplicative Decrease). Without relying on overhearing, it provides fairness in CSMA/CA networks, particularly IEEE 802.11 networks, by using only local information and performing localized operations. It combines several novel rate control mechanisms, including synchronized multiplicative decrease, proportional increase, and background transmission.

We also improve PISD further. PISD works precisely for scenarios where all MAC flows mutually contend, but when it is applied to a network consisting multiple contention groups, the network's throughput can be degraded. We develop PISD further and propose two new schemes, PISD-RS and PFS, to overcome the limitation. We prove that flows' rates attained under the two new schemes approximate proportional fairness.

The rest of the chapter is organized as follows. Section 4.1 discusses the related work. Section 4.2 gives the network model. Section 4.3 describes the fairness problem. Section 4.4 proposes our PISD solution. Section 4.5 analyzes the performance of PISD. Section 4.6 presents additional simulation results on PISD. Section 4.7 discovers a limitation of PISD. Section 4.8 introduces PISD-RS and PFS as solutions to the limitation. Section 4.9 proves that proportional fairness is achieved by PFS (and PISD-RS). Section 4.10 conducts simulations on PISD-RS and PFS. Section 4.11 summarizes the chapter.

4.1 Related Work

In [18; 19], to achieve fair bandwidth distribution among contending wireless links in a multihop wireless network, every node is required to measure the rates of contending links through overhearing and then change its own rate by adjusting either the minimum contention window or the contention window directly. In [22], Chen and Zhang also rely on overhearing among contending nodes for appropriate distribution of channel capacity in order to achieve aggregate fairness.

Luo et al.'s approach [29] assigns each MAC flow a basic fair share of bandwidth and then maximizes aggregate channel utilization through spatial channel reuse. The distributed implementation requires each sender to know all contending flows (through piggybacking and overhearing), and also requires topology information to be propagated through a conflict-free spanning tree. In follow-up work [21] they propose MLM-FQ (Maximize-Local-Minimum Fair Queueing), which requires contending nodes to transmit in the order of packet service tags (representing transmission deadlines). It relies on each node keeping track of service tags at other nodes through overhearing. In Vaidya et al.'s earlier DFS (Distributed Fair Scheduling Protocol) [20], a node sets a backoff timer based on the finish tag of its next packet to be transmitted. DFS depends on overhearing to correctly update the local virtual clock, based on which the final tag is computed.

OML [23] emulates TDMA on top of CSMA/CA to implement distributed weighted fair queueing. For each of its packets, the sender must inform the contending nodes that it will participate in the timeslot competition. This information is piggybacked in the packet header and overheard by other nodes. (Alternatively, one can use control messages to flood this information to contending nodes a few hops away, which, however, causes significant overhead.)

Nandagopal et al. [34] propose a general analytical fairness model and a MAC protocol to approximate proportional fairness. This work assumes that links in the same contention region will experience the same loss probability, which is however not always true. As we describe in Section 4.3.1, the loss probabilities of two links can be very different — the actual values depend on the relative spatial locations of the links. In [30], Tassiulas and Sarkar address the max-min fairness problem using a multi-channel MAC model that is different from the CSMA/CA model used in this study. Their later work [45] for end-to-end bandwidth guarantees is also based on the multi-channel model.

AIMD has been extensively studied in the past [46–48; 39], mostly in the context of TCP. Crowcroft and Oechslin [49] modify AIMD to achieve weighted proportional fairness in TCP. As we demonstrate later, the AIMD protocols designed for CSMA/CA networks by Cai et al. [24], by Xue et al. [26] and by Heusse et al. [25; 27] can only provide fairness under certain situations. AIMD has also been used in wireless networks for congestion control [50; 51].

4.2 Network Model

We study the fairness problem of CSMA/CA in one or more contending WLANs, or alternatively, among single-hop, ad hoc wireless links. Throughout the entire chapter, we consider CSMA/CA to mean the full RTS-CTS-DATA-ACK exchange. A network is modeled as a set of nodes (access points or hosts) and a set of wireless links. Each node has a transceiver. Each link supports two-way communication (for data/ACK exchange) between two nodes that can reliably decode each other's signal when radio interference is not present. All links transmit in the same frequency band. We also model a physical network where links transmit at different frequencies as multiple orthogonal networks, each using one frequency band, and then we deal with each network separately. A link whose sender is node a and receiver is b is referred to as (a, b). Link (a, b) has a contending link (c, d) if the transmission made by c (or d) can be carrier-sensed by the sender a, or causes interference at the receiver b. In this case, we also say that node a has a contending node

c. Generally speaking, the carrier sensing range is greater than the interference range, which is in turn greater than the transmission range. Two nodes within transmission range may be able to decode (overhear) each other's transmissions. A physical wireless link may carry zero, one or more MAC flows. If it carries more than one MAC flow, we will model the physical link as multiple logical links, each carrying one flow. The MAC flow carried by (logical) link (a, b) is referred to as flow (a, b). Note that this chapter studies single-hop flows in a WLAN or ad hoc deployment setting. We do not consider multi-hop flows.

4.3 Fairness Problem Remains Open in CSMA/CA Networks

In this section, we use a simple example to illustrate the fairness problem in CSMA/CA networks and explain why this problem remains unsolved. The existing solutions based on overhearing would work if the transmission range, the interference range and the carrier sensing range were all identical. However, in reality, the transmission range is much shorter than the other two. Consequently a node will not be able to gather, through overhearing, the necessary information from all contending flows. We will show that the classic fairness approach of AIMD (which is widely used on wired and wireless networks) cannot solve the problem in CSMA/CA networks, either.

4.3.1 Fairness Problem

Fairness is one of the core problems that must be addressed in any MAC design that allows contending nodes to share the same wireless medium. It requires that all wireless links of the same class have an equal right to access the communication channel and no link is starved. The random backoff algorithm in the IEEE 802.11 DCF is designed to give each host a fair chance of obtaining the channel during contention. Random backoff works fine in a symmetric environment where all hosts communicate with the same access point. However, it does not work well in asymmetric settings.

An example is shown in Fig. 4-1, where each of the two 802.11 DCF wireless links carries a MAC-layer flow. The figure shows an ad hoc network or two nearby WLANs whose access points (a and c) each support a wireless host (b and d). When the distance between b and c is zero such that the two merge into one, this example represents one WLAN whose access point b/c supports two hosts — node a is a server that is uploading to the Internet, and node d is a client that is downloading from the Internet. The fairness problem in the above network topology is first documented and analyzed in [35], which, however, does not consider the situations where the carrier-sensing range and interference range are greater than the transmission range.

We run simulations with ns-2 v2.32 [36] to study the rates of the two flows. The simulation parameters are given as follows: The transmission range of the nodes is $250m$, the carrier sensing range is $550m$, and the lengths of both links are $150m$. The transmission rate is 11 Mbps, and the packet length is 1,000 bytes. The parameters for the IEEE 802.11 DCF are the default values set by ns-2 according to the protocol standards.

Fig. 4-2 shows the average numbers of packets per second sent over the two links with respect to the distance between node b and node c. When the distance is below $250m$, flow (c, d) obtains most of the channel's bandwidth. When the distance is between $250m$ and $400m$, flow (a, b) obtains most of the channel's bandwidth. When the distance is between $400m$ and $550m$, flow (c, d) regains the upper hand. When the distance is greater $550m$, the two links are out of each other's carrier sensing range and they will both obtain high bandwidth. The explanation on why such unfairness happens is given in Appendix A.

Higher-layer rate control such as TCP cannot substitute for a MAC-layer fairness solution. Suppose a TCP connection C_1 traverses link (a, b) while another connection C_2 passes (c, d). These two TCP connections compete for the same resource — the wireless channel shared by (a, b) and (c, d). Consider the scenario where the length of the wireless links is $150m$ and the distance between b and c is $100m$. The simulation in ns-2 shows that C_1 is almost starved while the rate of C_2 is around 280 packets per second. The reason is that (c, d) is far more capable of obtaining the channel than (a, b) under CSMA/CA, which makes packets from C_1 prone to more drops and larger delay. For the two TCP connections to receive fair bandwidth, a MAC-layer solution must exist to distribute the channel's bandwidth fairly between (a, b) and (c, d).

4.3.2 Limitation of Overhearing-Based Solutions

Realizing the fairness problem in CSMA/CA networks, researchers have proposed numerous solutions [18–23], most relying on traffic information that each node collects by overhearing transmissions made by contending nodes. The most prevalent rate control scheme is to modify the random backoff algorithm such that a MAC flow that has a smaller rate than others will set a smaller backoff window and thus acquire more bandwidth [18; 19; 22]. How does a flow learn that its rate is smaller than the rates of its contending flows? The common approach requires the sender of the flow to estimate the rates of other flows by overhearing. Other rate control schemes, such as OML [23], DFS [20] and EMLM-FQ [21], also depend on overhearing (see Section 4.1).

The problem is that overhearing is limited within the transmission range but contention is defined by the interference range and the carrier sensing range. Consider a wireless link (i, j) in Fig. 4-3, where the transmission range of the sender i is shown by the solid circle, the carrier sensing range of i is shown by the dotted circle, and the interference range of the receiver j is shown by the dashed circle. When any node in the carrier sensing range of i makes a transmission, i will sense a busy channel and withhold its own transmission. When any node in the interference range of the receiver j makes a transmission, it will interfere with the signal from i. In the 802.11 DCF, if j senses a busy channel before receiving an RTS, it will not return CTS. In this case, any node in the carrier sensing range of j will interfere with the communication on (i, j). Clearly, the interference range is determined by the signal strength at the receiver, which is related to the distance between the sender i and the receiver j. The carrier sensing range is typically set to be no less than the maximum interference range, which can be 1.78 times the transmission range as suggested in [52] (also the default value used in ns-2).

Under CSMA/CA, on one hand, (i, j) contends with any wireless link whose sender or receiver is located within the carrier sensing range of i or the interference range of j, including (a, b), (c, d), (e, f), (g, h), and (x, y). On the other hand, the sender i can only overhear the CTS/ACK packets sent by y in the figure. Comparing the area in which contending nodes may reside (within the dotted and dashed circles or beyond such as a and e) with the shaded area from which s can overhear, it is clear that the number of contending nodes that cannot be overheard can be greater than the number of nodes that can be overheard. This seriously limits the effectiveness of any solution based on overhearing.

We implement the Huang-Bensaou protocol [19], where fairness is achieved by each node adjusting its contention window based on the overheard information of the contending flows. The simulation result for the network of Fig. 4-1 is shown in Fig. 4-4. The Huang-Bensaou protocol achieves almost perfect fairness when b and c are within the transmission range of each other (such that c can overhear b's CTS/ACK). However, when the distance between b and c is beyond $250m$, the Huang-Bensaou protocol is totally ineffective. The same is true for all other schemes relying on overhearing.

4.3.3 AIMD Does Not Work Either

Is there a fairness solution that allows a wireless link to adapt its rate without knowing the rates of its contending links? The classical fairness control scheme of AIMD (Additive Increase Multiplicative Decrease) may be first to come into mind. TCP uses AIMD (together with slow start) to achieve approximately proportional fairness among end-to-end flows without requiring each flow to know the rates of its contending flows. AIMD has been used in multihop wireless networks for congestion control [50; 51], but there is very limited research on applying AIMD to achieve fairness among MAC flows. In the following, we will show that AIMD is ill-fitted to this purpose.

For AIMD to work, the sender of a flow must be able to detect when the channel is saturated (congested), which is the time for multiplicative decrease. There are a number of possible approaches. First, the sender may measure how busy the channel is. The channel is considered to be saturated if the fraction of time for which it is busy exceeds a certain threshold. This straightforward approach however does not work. Consider the network in Fig. 4-1 and assume that node c is within the interference range of b but outside the carrier sensing range of a. In this case, even when the channel is saturated by transmissions on (c, d), node a will sense an idle channel.

Second, the sender may treat every failed transmission as a signal of channel saturation and perform multiplicative decrease [24; 26]. We simulate the AIMD protocol in [24] (the one in [26] is similar) on the network in Fig. 4-1, and the result is shown in Fig. 4-5. The protocol works fine in a WLAN environment where the links are all from a common access point to different hosts, which corresponds to the data points for distance being $-150m$ (such that a and c overlap to serve as the access point while b and d are hosts.) However, it performs poorly in asymmetric settings when the distance between b and c is greater than zero.

Third, the sender may monitor its buffer occupancy. Each sender generates packets for transmission at a certain rate, which is controlled by AIMD. It signals congested channel when the buffer length exceeds a threshold. The simulation result on the network of Fig. 4-1 is shown in Fig. 4-6. Again, fairness is not achieved.

AIMD may also be used to indirectly control the flow rates. Idle Sense [25; 27] replaces DCF's random backoff by adaptively setting the same optimal size for the contention window at all hosts. It was shown in [25] that, if the mean number of idle slots between two transmissions in the channel is controlled to a certain desirable value, e.g., 5.6 for 802.11b, the contention window size will be near-optimal for traffic throughput and fairness. The algorithm of Idle Sense is for each host to measure the mean number $\hat{n}_i$ of idle slots between two transmissions in the channel and to gradually increase its contention window when $\hat{n}_i$ is below the desirable value or multiplicatively decrease its window when $\hat{n}_i$ is above the desirable value. Idle Sense makes the assumption that, when AIMD converges, the contention windows at all hosts will reach the same size, and thus, the hosts will send at the same rate. This assumption is true in a WLAN where all hosts can symmetrically sense one another's carriers. It is however not true in general for multiple contending WLANs. Consider the network of Fig. 4-1, where the distance between b and c is 150m. Suppose the contention windows at the senders are set to the same size. Fig. 4-7 shows that the rate of flow (c, d) is far greater than that of (a, b) because the former's spatial location gives it a better chance to obtain the channel even when its contention window is the same.

4.4 Proportional Increase Synchronized multiplicative Decrease

In this section, we analyze why AIMD does not work in CSMA/CA networks and propose our solution, PISD, which consists of three rate control mechanisms: synchronized multiplicative decrease, proportional increase, and background transmission. We have extensively explored alternative ways for realizing the objectives of these mechanisms, and used simplicity and effectiveness as guiding selection criteria.

4.4.1 Synchronized Multiplicative Decrease

Why does AIMD work for TCP but not for CSMA/CA? The reason is that AIMD achieves fairness only with synchronized multiplicative decrease. Contending TCP connections always perform multiplicative decrease simultaneously but that is not true for MAC flows in CSMA/CA networks.

When a router becomes congested, packet loss is felt by all TCP connections that pass the router. Hence, synchronized multiplicative decrease will be performed at the senders. We illustrate the rates of two TCP connections over time in Fig. 4-8. The rates are normalized such that the congestion happens when their sum is equal to 1. Initially, the rates are different. At each multiplicative decrease, the two rates are reduced by the same percentage and consequently the larger rate will be reduced by a larger amount, closing the gap between the two, which will eventually converge to the same value.

In a CSMA/CA network such as Fig. 4-1, the wireless links have different opportunities to obtain the wireless medium for transmission, depending on their spatial locations. From Fig. 4-2, we know that (c, d) is more capable of obtaining the medium than (a, b) when the distance between b and c is shorter than $250m$. At time 6 in Fig. 4-9, when the combined rate of flow (a, b) and flow (c, d) reaches the channel capacity, *because* (c, d) *is able to obtain the bandwidth it needs*, node c sends all its packets out but node a observes buffer buildup and transmission failure (with a much larger likelihood). Consequently, a detects channel congestion and performs multiplicative decrease, while c does not. Since the rate of (a, b) experiences multiplicative decrease more frequently, it will be smaller than the rate of (c, d).

4.4.2 AISD: Additive Increase Synchronized Multiplicative Decrease

In order to achieve fairness in CSMA/CA networks, we must ensure that multiplicative decrease is performed at contending senders simultaneously. We design a new protocol, called AISD, for this purpose. There are two major problems to be solved.

The first problem is how to detect channel congestion. For each flow (i, j), the sender i stores all arrival packets in a *repository buffer* above the MAC layer. It locally maintains a time-dependent *target rate* $r_{i,j}(t)$ at which packets from the repository buffer are released to the MAC layer for transmission to the receiver j. The flow is *backlogged* if the packet arrival rate is greater than the target rate such that the repository is not empty. The target rate of a backlogged flow is additively increased over time. The actual rate at which the MAC layer sends out packets is called the *sending rate*, which is bounded by the target rate.

When the sum of the target rates of all contending flows in the channel is smaller than the capacity of the channel, all (or most) packets released by the senders to the MAC layer can be transmitted. Consequently the senders will not observe persistently growing packet queues at their MAC layer. However, additive increase will eventually improve the target rates such that their sum exceeds the channel capacity. When this happens, the flow that is least capable of competing for media access will see its packet queue growing. When the queue length passes a threshold, the sender claims that the channel is congested. Other flows that are more capable of obtaining the wireless medium may still find their queues empty. When we refer to *packet queues*, we always mean *the queues storing packets released to the MAC layer*, not the repository above the MAC layer.

The second problem is how the sender that detects channel congestion informs the contending nodes such that they can perform synchronized multiplicative decrease. One solution is for the sender and its receiver to jam the channel with a radio signal for an extended period of time. Before jamming, the contending nodes are able to transmit at decent rates (because the sum of all target rates has just passed the channel capacity for a small amount after the most recent additive increase). During jamming, they can hardly send out any packets, which gives them a clear indication that someone is jamming, and the only reason for jamming is that channel congestion has been detected. As their queue lengths exceed the threshold, they will join jamming, which provides additional assurance that all contending nodes in the channel will learn that the channel is congested. Although the jamming approach works, it wastes bandwidth. Instead of using a dedicated radio signal, a node can jam the channel with its own packets. During jamming, to ensure that the node is able to occupy the channel, we reduce its minimum congestion window to a small fraction of the default size. Besides window reduction, the jamming packets are expected to follow the same collision avoidance/resolution protocol (such as DCF) as other packets do. (This is what we do in all our simulations.)

The AISD protocol is summarized as follows. After each unit of time, the sender of a backlogged flow (i, j) increases its target rate by

$$r_{i,j}(t) = r_{i,j}(t-1) + \alpha \tag{4–1}$$

At this rate, the sender releases packets to a queue, from which the MAC layer picks up packets for transmission. In one time unit, packets of total size $r_{i,j}(t)$ will be released.

The *quota* is defined as the number of bytes that remain available for transmission in the current time unit, which is equal to $r_{i,j}(t)$ less the number of bytes that have been transmitted during the current time unit. (Note that it includes both packets to be released and packets already released to the queue but not transmitted yet.)

When the packet queue at node i for link (i, j) exceeds a threshold length, i claims that the channel is congested and jams the channel immediately. If the quota is sufficiently large, it jams for the rest of the current time unit; otherwise, it jams for one more time unit. The jamming is performed by releasing all packets within the quota to the queue and reducing the minimum contention window to a small value. Multiplicative decrease is performed at the end of the time unit during which jamming is performed.

$$r_{i,j}(t) = r_{i,j}(t-1) \times (1-\beta) \tag{4–2}$$

As a safeguard, multiplicative decrease should not be performed for two consecutive time units. The protocol does not require the clocks of the nodes to be synchronized. If a node is the sender for multiple flows, it performs media access and random backoff independently for each flow. Packets for different flows are queued separately. Consider flows (a, b) and (a, c). Suppose (a, b) contends with (i, j) while (a, c) does not. When (i, j) is transmitting, node a performs independent media access for its two flows. For example, it may send an RTS to b and then set the backoff timer for (a, b) due to an RTS collision at b. While waiting on the timer for (a, b), it sends an RTS to c and then delivers a packet on (a, c).

We simulate AISD on the network of Fig. 4-1 with the following additional parameters: α is 5 kBps, β is 25%, the time unit is one second, the queue-length threshold that triggers jamming is 10 packets, and the minimum contention window for jamming is one tenth of the default size. In the simulation, each packet is 1 kB long. We find AISD can robustly ensure synchronized multiplicative decrease. Fig. 4-10 shows that, using AISD, the rates of the two flows are about the same for any distance between b and c. Fig. 4-11 shows AISD in action over time when the distance between b and c is $100m$. At time 0, the rate of flow (c, d) is much larger. Then, AISD kicks in to equalize the two rates.

4.4.3 PISD: Proportional Increase Synchronized Multiplicative Decrease

Next we extend AISD for weighted fairness by replacing additive increase with proportional increase. The resulting protocol is called PISD. Suppose the network administrator assigns a weight $w_{i,j}$ to each MAC flow (i, j) based on application requirements. For example, a MAC flow serving an important server should be given a higher weight than a MAC flow serving a regular client host. The problem is for the MAC layer to allocate the channel's bandwidth among contending MAC flows in proportion to their weights. This is called *weighted fairness*. Fairness as discussed previously is a special case in which all weights are equal.

The PISD protocol is similar to AISD except for how the target rates are increased: After each unit of time, the sender of flow (i, j) increases its target rate by

$$r_{i,j}(t) = r_{i,j}(t-1) + \alpha w_{i,j} \tag{4–3}$$

The rest of the protocol is the same as AISD. We prove in the next section that PISD achieves weighted fairness.

We again simulate PISD on the network in Fig. 4-1. We assign $w_{a,b} = 3$ and $w_{c,d} = 1$, and the result is shown in Fig. 4-12. Weighted fairness is achieved. Fig. 4-13 shows the rates of the two flows with respect to time when the distance between b and c is $100m$. Clearly, flow (a, b) achieves three times the rate of flow (c, d) because it increases the rate at three times the speed of the latter.

4.4.4 PISD with Background Transmission

Using AIMD, TCP will not utilize the bottleneck router's full capacity at all times due to multiplicative decrease, which is evident from Fig. 4-8. Similarly, using PISD, CSMA/CA will not fully utilize the channel capacity right after multiplicative decrease. It can be easily shown that, in theory, the average rate of a flow is smaller than the optimal value by a fraction no more than $\frac{\beta}{2}$ (see the next section). If $\beta = 25\%$, then the fraction is 12.5%. However, in our simulations, the degradation is mostly around 5% and sometimes up to 10%. We believe this is due to the interaction of PISD with the protocol details of CSMA/CA, particularly the IEEE 802.11 DCF, many of whose details concerning RTS, CTS, DIFS, EIFS, minimum/maximum contention windows, and backoff algorithm can impact flow rate. No matter what the degradation may be, we augment PISD with a new technique, *background transmission*, which will utilize the unused channel bandwidth for packet transmission.

Right before each multiplicative decrease, the sum of the target rates at all contending nodes exceeds the channel capacity by a small amount. After the target rates are multiplicatively decreased, their sum is below the channel capacity by a fraction of β at most. For each flow (i, j), the sender i remembers its target rate right before the most recent multiplicative decrease. This rate is called the *background rate*, which stays the same until the next multiplicative decrease. Our basic idea is that we want to ensure that all senders are able to transmit at their target rates and, if there is extra channel bandwidth, we allow the senders to compete for additional transmissions up to their background rates. When a node's sending rate is above its target rate, its transmission is called a *background transmission*. When the sending rate is below the target rate, its transmission is called a *regular transmission*. When a regular transmission of one node contends with a background transmission of another, the former should be given priority. To achieve such differentiation, we increase the minimum contention window for background transmission.

The PISD protocol with background transmission is as follows. Proportional increase synchronized multiplicative decrease is performed on the target rate as usual. But a sender i releases packets to the MAC layer at the background rate (the rate before the last multiplicative decrease). The node also keeps track of the number n_t of bytes that would have been released at the target rate. Let δ be the time that has elapsed in the current time unit. $n_t = r_{i,j}(t) \times \delta$. Let n_s be the number of bytes that has been delivered to the receiver in the current time unit. When n_s is equal to or greater than n_t, the sender knows that it is now making background transmissions and therefore it increases the minimum contention window. When n_s becomes smaller than n_t, the sender changes its minimum contention window back.

We simulate PISD with background transmission on the network in Fig. 4-1 with a minimum contention window for background transmission twice the size of the default minimum contention window for regular transmission. (Note that the default value for regular transmission is set by ns-2 based on the standard of 802.11 DCF.) Fig. 4-14 shows that the flows pick up the extra bandwidth left by PISD for additional packet transmission. This extra bandwidth, representing only a small fraction of channel capacity, is not regulated by proportional increase multiplicative decrease, and consequently it is unevenly distributed between the flows based on the IEEE 802.11 DCF.

4.4.5 Discussion

The fairness problem becomes tricky when wireless links have different transmission (modulation) rates. The operations of PISD are independent of transmission rate. It achieves fairness regardless of whether the transmission rates of the links are the same or different. However, under non-uniform transmission rates, it is well known that ensuring each flow a fair share of bandwidth may cause significant reduction in a WLAN's overall throughput [53]. One solution to this problem is to change the definition of fairness. Instead of ensuring a fair bandwidth share, we allocate each flow a fair share of channel occupation time. PISD can be adapted to serve this purpose. Consider three contending wireless links whose transmission rates are 11 Mbps, 5 Mbps and 2 Mbps, respectively. If we let the weight of the 11 Mbps flow be one, we shall assign the weights of other two flows to be $\frac{5}{11}$ and $\frac{2}{11}$, respectively. While the two flows will send at lower rates, their transmissions take inversely proportionally longer time, resulting in the same channel occupation time for the three flows.

For implementation, PISD performs all rate adaptation operations on top of the MAC layer. It requires the MAC to support multiple queues (one for each adjacent link) and provide an API that allows MAC parameters to be changed. For example, jamming is implemented by reducing the minimum contention window.

4.5 Analysis

It is well known that AIMD converges [46]. Much work about AIMD has been performed in the context of TCP [47; 48; 39]. In this section, we analyze PISD and show that it achieves weighted fairness after convergence. More importantly, we derive the convergence time, the channel coverage, and the convergence accuracy with respect to α and β, and reveal the performance tradeoff that can be made by changing these two parameters.

4.5.1 Weighted Fairness and Convergence Time

Consider a set L of MAC flows that contends in the same wireless channel whose effective capacity is C. When the sum of the target rates of all flows is below the channel capacity, the channel will be able to deliver the packets of the flows and the senders will proportionally increase their target rates. Once the sum exceeds the channel capacity, the senders will immediately decrease their target rates multiplicatively. PISD performs the following rate control.

$$r_{i,j}(t+1) = \begin{cases} r_{i,j}(t) + \alpha w_{i,j}, & \text{if } \sum_{(i,j)\in L} r_{i,j}(t) \leq C \\ r_{i,j}(t)(1-\beta), & \text{if } \sum_{(i,j)\in L} r_{i,j}(t) > C \end{cases}$$

We derive how much time it takes PISD to converge such that the rates of the flows are stabilized and proportional to their weights. Our results show that the convergence time is a decreasing function of both α and β.

When multiplicative decrease happens, even if the combined target rate of all flows may be greater than the channel capacity, it will be greater only by a small amount due to the nature of additive increase. To simplify the analysis, we treat them as equal. A *PISD period*, denoted as P, is defined as the time between two consecutive multiplicative decreases. We derive the value of P as follows: Consider an arbitrary multiplicative decrease, which is triggered when $\sum_{(i,j)\in L} r(i,j)(t) = C$. It reduces all target rates by a fraction of β and hence leaves βC of channel capacity unused. The proportional increase improves the combined rate of all flows at a speed of αW, where $W = \sum_{(i,j)\in L} w_{i,j}$. After a period P, the combined rate should be increased by βC in order to make the channel saturated again and cause the next multiplicative decrease. Since $P \times \alpha W = \beta C$, we have

$$P = \frac{\beta C}{\alpha W}$$

Without losing generality, for $l = 0, 1, 2, \ldots$, let $t = lP$ be the time units right before multiplicative decrease, and $t = lP + 1$ be the time units after multiplicative decrease. Multiplicative decrease occurs at the time instant between lP and $lP + 1$. Given arbitrary values for $r_{i,j}(0), \forall(i,j) \in L$, we show that $r_{i,j}(t)$ will converge towards a value that is proportional to $w_{i,j}$ as t increases.

First, we determine the value of $r_{i,j}(lP)$. During each PISD period, the target rate is first multiplicatively decreased and then proportionally increased. Hence, for $l > 0$,

$$r_{i,j}(lP) = r_{i,j}((l-1)P) \times (1-\beta) + \alpha w_{i,j} \times P$$

By induction over the above iterative formula, we have

$$r_{i,j}(lP) = C\frac{w_{i,j}}{W} + (r_{i,j}(0) - C\frac{w_{i,j}}{W})(1-\beta)^l$$

The second term on the right side diminishes to zero when l becomes large. Hence, $r_{i,j}(lP)$ converges to

$$r^*_{i,j} = C\frac{w_{i,j}}{W}.$$

Next, we determine the value of $r_{i,j}(t)$, $t \leq lP$, for $l = 0, 1, 2, \ldots$. The rate is multiplicatively decreased immediately after time $\lfloor t/P \rfloor P$ and then proportionally increased. We have

$$\begin{aligned}
r_{i,j}(t) =& r_{i,j}(\lfloor t/P \rfloor P) \times (1-\beta) + \alpha w_{i,j} \times (t \mod P) \\
=& C\frac{w_{i,j}}{W}(1-\beta) + (r_{i,j}(0) - C\frac{w_{i,j}}{W})(1-\beta)^{\lfloor t/P \rfloor + 1} \\
&+ \alpha w_{i,j} \times (t \mod P)
\end{aligned}$$

As t increases, the second term on the right side diminishes to zero. Hence, $r_{i,j}(t)$ converges to the following curve,

$$r^*_{i,j}(t) = C\frac{w_{i,j}}{W}(1-\beta) + \alpha w_{i,j} \times (t \mod P),$$

which is independent of the initial value $r_{i,j}(0)$. The average of $r_{i,j}(t)$ over the l period, denoted as $A_{i,j}(l)$, is given below

$$\begin{aligned}
A_{i,j}(l) &= \frac{(\sum_{(l-1)P<t<lP} r_{i,j}(t)) + r(lP)}{P} \\
&= C\frac{w_{i,j}}{W}(1-\frac{\beta}{2}) + \frac{\alpha w_{i,j}}{2} + (r_{i,j}(0) - C\frac{w_{i,j}}{W})(1-\beta)^l
\end{aligned}$$

The third term on the right side diminishes to zero when l increases. Hence, the average rate converges to

$$A^*_{i,j} = C\frac{w_{i,j}}{W}(1-\frac{\beta}{2}) + \frac{\alpha w_{i,j}}{2},$$

which is proportional to the weight $w_{i,j}$.

We define the *convergence time* as the time it takes for $A_{i,j}(l)$ to be ε-*close* to its target $A^*_{i,j}$. The ε-*closeness* is defined as follows:

$$\frac{|A_{i,j}(l) - A^*_{i,j}|}{A^*_{i,j}} \le \varepsilon$$

We derive the lower bound of l that can satisfy the above inequality. Note that $r_{i,j}(0) \le C$.

$$\begin{aligned}
l &\ge \log_{1-\beta} \frac{\varepsilon(C\frac{w_{i,j}}{W}(1-\frac{\beta}{2}) + \frac{\alpha w_{i,j}}{2})}{|r_{i,j}(0) - C\frac{w_{i,j}}{W}|} \\
&\ge \log_{1-\beta} \frac{\varepsilon(C\frac{w_{i,j}}{W}(1-\frac{\beta}{2}) + \frac{\alpha w_{i,j}}{2})}{|C - C\frac{w_{i,j}}{W}|}
\end{aligned}$$

The time for l periods is $t = lP$. Hence, the convergence time is

$$t \geq \frac{\beta C}{\alpha W} \log_{1-\beta} \frac{\varepsilon(C\frac{w_{i,j}}{W}(1-\frac{\beta}{2}) + \frac{\alpha w_{i,j}}{2})}{|C - C\frac{w_{i,j}}{W}|}$$

From the above formula we see that, the convergence time is a decreasing function for both α ($\alpha > 0$) and β ($\beta \in (0,1)$). In other words, the larger the value of α (or β) is, the faster the convergence is.

4.5.2 Channel Coverage

We study how much bandwidth is regulated (or *covered*) by PISD. The channel bandwidth covered by PISD is distributed to the flows in proportion to their weights. The channel bandwidth not covered by PISD is arbitrarily distributed to flows through background transmission. Formally, the *channel coverage*, denoted as Cov, is defined as the sum of the average target rates of the flows after PISD fully converges divided by the channel capacity.

$$Cov = \frac{\sum_{(i,j)\in L} A^*_{i,j}}{C} = 1 - \frac{\beta}{2} + \frac{\alpha W}{2C}$$

We know that $\frac{\alpha W}{2C} = \frac{\beta}{2P}$, where P is the PISD period that is greater than 1. Hence, the channel coverage is mainly controlled by β. The smaller the value of β is, the more the channel bandwidth PISD controls.

4.5.3 Convergence Accuracy

If every flow is able to deliver all packets released to the MAC layer in a timely fashion, then the sending rate will be equal to the target rate and therefore weighted fairness is accurately achieved once all target rates are fully converged. That is not the case if not all packets released based on the target rate can be delivered.

As the flows' target rates are proportionally increased in a PISD period, it is possible that, right before multiplicative decrease, the sum of all target rates is slightly greater than the channel capacity, in which case not all packets can be delivered. We study the impact of this case below.

In the worst case, a flow (i, j) cannot transmit any packet in the last time unit of a PISD period. Suppose the target rates of all flows are fully converged. The amount of data that flow (i, j) cannot deliver in the last time unit is bounded by $r^*_{i,j}$. The amount of data that is supposed to be delivered in the whole period is $A^*_{i,j}P$. The *convergence accuracy* of PISD, denoted as Acc, is defined as follows.

$$Acc = 1 - \frac{r^*_{i,j}}{A^*_{i,j}P} = 1 - \frac{1}{(1-\frac{\beta}{2})\frac{\beta C}{\alpha W} + \frac{\beta}{2}}$$

Hence, the convergence accuracy decreases as α increases.

Putting all of the above analysis together, we can see that choosing the values of α and β is actually making a tradeoff among three system properties: convergence time, channel coverage, and convergence accuracy.

4.6 Additional Simulations

We perform additional simulations with ns-2 under two scenarios to evaluate the effectiveness of the proposed PISD protocol. PISD is implemented on top of the IEEE 802.11 DCF. *If not specified otherwise, the simulation parameters are the same as those in Section 4.3.1 and Section 4.4.2.* The parameters for 802.11 DCF use the default values set by ns-2 according to the protocol standards. By default, $\alpha = 2$ kBps, and $\beta = 25\%$. We will study how different values for α and β affect PISD's performance. By default, background transmission is turned off.

Shown in Fig. 4-15, our first simulation scenario consists of two access points, a and b, located at two nearby buildings. Node a sends data to three client hosts, $h1$, $h2$ and $h3$. Node b also sends data to three client hosts, $h4$, $h5$ and $h6$. The clients are evenly spread around the access points. The length of each wireless link is $80m$, and the distance between a and b is $480m$. Fig. 4-16(a) shows the flow rates under the IEEE 802.11 DCF. When the rate curves of several flows overlap, we will explain which flows each curve represents in both text and figure caption. Under the 802.11 DCF, the rates of flows $(a, h1)$, $(a, h2)$ and $(a, h3)$ are the same (the lower curve in the left plot) because node a

schedules packets in round robin order among local flows. The rates of flows $(b, h4)$, $(b, h5)$ and $(b, h6)$ are also the same (the upper curve in the left plot). However, the rates of flows from a are much smaller than the rates of flows from b. Because the distances from a and its clients to b and its clients are greater than the transmission range $250m$ (see simulation parameters in Section 4.3.1), no overhearing-based solutions work here. The simulation result for the Huang-Bensaou protocol [19] is shown Fig. 4-16(b), which is comparable to what the 802.11 DCF produces.

PISD is able to achieve fairness among all flows, as shown in Fig. 4-17. Starting from different initial rates, all flows converge to the same fair rate. The total throughput is 428.1 packets per second, comparing with 444.6 under the 802.11 DCF with or without the Huang-Bensaou protocol. Next, we turn on background transmission, and the result is shown in Fig. 4-18. Some flows achieve higher average rates, and the total throughput becomes 455.8 packets per second. It is higher than the throughput under the 802.11 DCF because of reduced radio collisions, thanks to intermittent release of packets to the MAC layer at the background rate. Since the additional rate acquired through background transmission obscures the rate curve produced by PISD, for presentation clarity, we will turn it off in other simulations.

We now study how β and α affect the performance of PISD. The simulations confirm the analytical results in Section 4.5.1. First, we double the value of β while keeping α the same, and the simulation result in Fig. 4-19 shows that the flow rates converge quicker, when comparing with Fig. 4-17. It also shows that the average flow rate is smaller, indicating a smaller channel coverage by PISD (as predicted in Section 4.5.2), and therefore more bandwidth is allocated for background transmission. Second, we double the value of α while keeping β the same, and the simulation result in Fig. 4-20 shows that the flow rates converge quicker, when comparing with Fig. 4-17. It also shows that convergence accuracy decreases due to the sudden drop in the rates of some flows right before multiplicative decrease, as predicted in Section 4.5.3.

So far we have set all flow weights to 1. As shown in Fig. 4-21, we modify the topology by turning $h2$ and $h6$ into servers that upload data to access points a and b, respectively. We set the servers' weights to 3, and the simulation result in Fig. 4-22(a) shows that the rate of a server is about twice the rate of a client. One may notice the spikes in the rate adaptation curves in the figure. Such spikes are expected according to our analysis in Section 4.5.3. They only happen in the last time unit of a PISD period when the aggregated target rate of all contending flows exceed the channel capacity. When nodes that detect channel congestion jam the channel with their packets, the node that detects congestion last may send at a low rate, causing a downward spike. Since the spikes may only happen at the end of a PISD period, its impact on the average flow rate is limited, which is confirmed by the analytical result in Section 4.5.3 and the simulation result in the left plot, where the average rates of the server flows are 124.0 and 125.1 packets per second respectively, and the average rates of the client flows are 41.8, 42.3, 42.4 and 42.7 packets per second respectively. Moreover, Section 4.5.3 shows that decreasing α will improve convergence accuracy (i.e., reduce spikes), which is confirmed by the simulation result in Fig. 4-22(b), where α is reduced by two thirds.

Next we expand the network to have four access points and ten hosts. The access points are located at the corners of a 380m $\times$ 380m square. The distance from the hosts to their access points varies from 70m to 150m. Their relative positions are shown in Fig. 4-23. The rates of the flows under PISD, PISD with background transmission (PISD-b), DCF, and the Huang-Bensaou protocol (H.-B.) are shown in Table 4-1. PISD is able to achieve fairness while the DCF and the Huang-Bensaou protocol cannot in this scenario.

Our second simulation scenario is an ad hoc network shown in Fig. 4-24, where visitors to a commercial conference download information from exhibit booths to their laptops via direct wireless links that share the same channel. The size of the area is $400m$ by $600m$, and the nodes are plotted in the area based on their assigned coordinates. The simulation results are shown in Table 4-2. Each row contains one weight assignment and the corresponding flow rate achieved by PISD. The results demonstrate the great flexibility and *quantitative precision* that PISD is able to bring into CSMA/CA networks.

4.7 Limitation of PISD

In the previous sections, we have shown that PISD is able to provide fairness in CSMA/CA networks by using only localized operations, and it does not require hosts to overhear their neighbors. However, PISD still has a limitation, which is that it assumes that all flows are mutually contending. In this section, we show that when PISD is applied in a multihop network or a set of WLANs consisting of multiple contention groups, its performance may be degraded. Note that a contention group is defined as a group of maximal number of flows that mutually contend. Each contention group is corresponding to a maximal clique in the contention graph of a network.

In PISD, to insure synchronized multiplicative decrease, a flow jams its neighboring flows with full transmission capability after it detects a saturated channel. As a result, the neighboring flows being jammed will again detect a saturated channel and jam their neighbors. This process indeed leads to a synchronized multiplicative decrease in the whole network. In fact, under PISD, the transmission rates of the flows are bounded by the *bottleneck* of a multihop network, where we define the bottleneck of a multihop network as the contention group with the largest number of mutually contending flows. We can observe this problem from an example below.

As shown in Fig. 4-25, the network consists of six flows. The six flows form two contention groups. In the first contention group S_1, flow $(h1, h2)$ only contends with flow $(h3, h4)$. In the second contention group S_2, the five flows $(h3, h4)$, $(h5, h6)$, $(h7, h8)$, $(h9, h10)$, $(h11, h12)$, all contend with each other. We conduct simulations with ns2 to study the rates of the flows achieved by PISD. The lengths of the six links are $150m$, and other simulation parameters remain the same as previous sections.

Table 4-3 shows the average numbers of packets per second sent over the six flows. We see that PISD achieves fairness among all the flows. Contention group S_2 is the bottleneck of the network, in which six flows mutually contend. For these six flows, they all attain equal rates, and the summation of their rates is 368.9 PPS (packets per second), which is close to the channel capacity[1] . However, the interesting point is on the rate of flow $(h1, h2)$, or more specifically, the channel utilization of the contention group S_1. We notice that the combined rate of flow $(h1, h2)$ and $(h3, h4)$ is just 147.9 PPS, which is far lower than the channel capacity. There is room for flow $(h1, h2)$ to attain a much higher rate, but PISD does not make it. The reason is that $(h1, h2)$ is often obliged to perform multiplicative decreases by $(h3, h4)$, although the channel of contention group S_1 is not really saturated.

We have seen that the problem of PISD is that the low rates in the bottleneck of a network will propagate to other parts of the network. We expect a new scheduling scheme that solves this problem and works properly for networks with multiple contention groups. In the following sections, we extend PISD into new schemes that are able to approximate proportional fairness.

4.8 Proportional Fair Scheduling

Having known the limitation of PISD, in this section we develop PISD into new schemes that are suitable for networks with multiple contention groups. We first give a simple solution called PISD-RS (PISD with Reduced jamming Strength), which comes directly through simple modifications on PISD. We then give a more sophisticated solution called PFS (Proportional Fair Scheduling), which overcomes a problem that PISD-RS

[1] Channel capacity is about 450 PPS. The summation of the rates is lower than the channel capacity. This is due to the channel coverage degradation caused by the AIMD scheme employed in PISD, and it can be tuned by the value of β. In this simulation, β is 0.5.

has. In Section 4.9, through theoretical analysis we show that the essence behind PFS is proportional fairness.

4.8.1 Reducing Channel Jamming Strength

Before introducing our solution, we first need to take a closer observation on why flow $(h1, h2)$ in Fig. 4-25 get low rate under PISD. Suppose at a certain time unit, contention group S_2 is saturated but contention group S_1 is not. Not losing generality, suppose flow $(h5, h6)$ is the first one that detects channel saturation. By the design of PISD, node $h5$ will jam the channel by sending a number of packets with a small contention window size. Then, flow $(h3, h4)$ will feel the jamming through its increasing queue length. Note that there is no difference between detecting a jamming and detecting the channel to be saturated. Once the jamming is detected by $h3$, it also performs the same thing — jamming the channel. The jamming made by $h3$ will eventually cause $(h1, h2)$ to execute a multiplicative decrease, although at this time S_1 is not really saturated.

A first intuition on solving the above problem is to simply reduce the strength of each jamming. This is reasonable, although it still has problems as we will show later. The idea is that, when a contention group S_k is saturated, any flow (i, j) in S_k tries to jam the channel with a proper strength that is just enough to make the flows in S_k feel the jamming, but will not make other neighbors of (i, j) feel it. Thus, we change the jamming scheme of PISD to the following:

At time unit t, if a flow (i, j)'s queue length passes a threshold, to jam the channel, i temporarily increases its target rate by a factor of λ, from $r_{i,j}(t)$ to $r_{i,j}(t) \times (1 + \lambda)$, and at the same time, i reduces its minimum congestion window to a small fraction of the default size.

A typical value of λ used in our experiments is 0.2. Note that the above modification only applies to the jamming part in the protocol of PISD, and the AIMD part still remains unchanged, which can be summarized as:

$$r_{i,j}(t+1) = \begin{cases} r_{i,j}(t)(1-\beta), & \text{if queue length} > \text{threshold} \\ r_{i,j}(t) + \alpha w_{i,j}, & \text{otherwise} \end{cases}$$

where α and β are system parameters, and $w_{i,j}$ is the weight of flow (i, j). The default value of any $w_{i,j}$ is 1. It is worth of noting that, as a safeguard that is also in the original PISD, multiplicative decrease should not be performed for two consecutive time units.

We name the above revised protocol as PISD-RS (PISD with Reduced jamming Strength). Its effectiveness can be immediately seen from a simulation again on the network shown in Fig. 4-25. In the simulation, $\alpha = 10$ kBps (equivalent to 10 packets per second), $\beta = 0.5$, and $\lambda = 0.2$. Results are shown in Table 4-4. Compared with Table 4-3, the rate attained by flow $(h1, h2)$ is much increased, from 74 PPS to 275.5 PPS. The combined rates of the flows in each of the contention groups (S_1 and S_2) are also increased, which shows that the channel bandwidth is now better utilized then the original PISD. In fact, as we will prove later, PISD-RS approximates proportional fairness. A set of theoretical values given by proportional fairness are shown in the third column of the table, assuming the capacity of each contention group to be 450 PPS. Note that the theoretical values are just to give us a sense on proportional fairness. For a purely distributed algorithm like PISD-RS in which there is merely no communication or collaborations among the hosts, it is very hard to strictly achieve the theoretical values. Although the rates attained by PISD-RS are lower then the theoretical values for various extents due to interferences in real protocol execution, they still show a consistency to the proportional fairness. As a baseline, in the forth column of the table, we show the rates of the flows attained under ordinary IEEE 802.11 DCF. We see that very severe unfairness exists. In fact, $(h3, h4)$ can hardly transmit any packets.

4.8.2 Problem of PISD-RS

As a simple solution, although PISD-RS moves a step towards proportional fairness, it is still problematic. The concern is on whether the new jamming scheme with reduced strength is always able to guarantee each jamming to be successful. We know that jamming is introduced to assure synchronized multiplicative decreases. An unsuccessful jamming will cause an asynchronous multiplicative decrease, which should be avoided. Indeed, unsuccessful jammings do occur in PISD-RS.

We still use a simulation to show the problem. We take away flow $(h1, h2)$ from the network shown in Fig. 4-25, and hence all the remaining flows are in a common contention group, through which we can better see the problem. The new network is shown in Fig. 4-26. Clearly, if synchronized multiplicative decrease is always guaranteed, the five flows should gain equal rates. However, the simulation result shows that the rates of the flows $(h3, h4)$, $(h5, h6)$, $(h7, h8)$, $(h9, h10)$, and $(h11, h12)$ attained by PISD-RS are 47.3, 86.6, 84.2, 89.3, and 90.9 packets per second, respectively. The result indicates that $(h3, h4)$ sometimes makes unsuccessful jammings and performs multiplicative decreases on its own. The problem is resulted from an insufficient jamming strength.

Can we remedy the problem simply by increasing the value of λ so that the jamming strength is increased? Unfortunately, the answer is no. Because under PISD-RS, during a synchronized multiplicative decrease in a contention group, every flow indeed performs a jamming. Assuming that the combined rate of the flows before the multiplicative decrease is approximately equal to the channel capacity C, the total jamming strength is about λC. Increasing the value of λ directly leads to a larger total jamming strength and will likely propagate the jamming to neighboring contention groups, which is just the same problem that the original PISD suffers from.

On the other hand, as we have seen, keeping a small λ value may result an insufficient strength for a single jamming. Back to the example of Fig. 4-26, assuming that the target rate of flow $(h3, h4)$ before performing a jamming is $\frac{1}{5}C$, the jamming strength is $\frac{1}{25}C$, if

$\lambda = 0.2$. With such a small strength, it is hard to guarantee other neighbors to detect the jamming.

4.8.3 PFS (Proportional Fair Scheduling)

To extricate from the dilemma of PISD-RS, a more delicate scheme should be introduced. We want to assure a sufficient strength for each single jamming, while at the same time we also want to keep the total jamming strength performed by all flows in a contention group as small as possible. The solution is that, only one flow should jam the channel for each synchronized multiplicative decrease in a contention group. This is the idea that inspires the design of PFS (Proportional Fair Scheduling).

Though the intuition is straightforward, the implementation is not. Suppose flow (i, j) is the first one that detects a saturated channel and wants to initiate a synchronized multiplicative decrease in a saturated contention group. (i, j) must by some means notify the other flows in the contention group. For those flows being notified, we want them to only perform multiplicative decreases but not any additional jammings. To meet this requirement, we introduce a two-phase jamming scheme. In the first phase, the first flow that detects channel saturation and wants to jam the channel claims itself as the *leader* to all its neighboring flows. In the second phase, the leader jams the channel, while its neighbors try to detect the jamming but they will not jam the channel again. Detailed operations are depicted below.

In the first phase, if a flow (i, j) detects that its queue length passes a constant threshold Φ_1, it starts to claim itself as the jam leader in the current time unit. To claim to be the leader, (i, j) conducts a *noise jamming*, which is different from a regular jamming that we have seen. For regular jammings, nodes just increase their target rates and decrease the contention window size, and all packet transmissions still follow the CSMA/CA protocol. In a noise jamming, node i first signals node j with a noise jamming bit carried in a control packet that is transmitted with a small contention window. After the signal is received by j, both i and j jams the channel with noise for a certain period of time T_{noise}. Note that, noise jamming does not follow the carrier sensing backoff or NAV stage, hence the noise is guaranteed to be sent out immediately. If a node senses the channel to be consecutively busy for T_{noise}, the node determines that it has been noise jammed, and the node decides to become a *MD candidate* (a candidate to perform a multiplicative decrease). In our simulations, T_{noise} is set to 0.016 seconds, which is equivalent to transmit 2000 bytes in broadcast mode.

In the second phase, the leader jams the channel with a strength of λC, by which we mean the leader temporarily increase its target rate from $r_{i,j}(t)$ to $r_{i,j}(t) + \lambda C$. Still, λ is set to 0.2 in our simulations. Note that, being different from PISD-RS, here the jam strength is λ times the channel capacity C instead of the target rate of the flow itself. After jamming the channel, the leader makes a multiplicative decrease at the end of the time unit. For the MD candidates, they use the same jamming detection scheme that is used in PISD-RS. If the queue length of a MD candidate passes a constant threshold Φ_2 (normally, Φ_2 is smaller than Φ_1), it makes a multiplicative decrease at the end of the time unit. Otherwise, the node does nothing. For all nodes, their roles of being leaders or MD candidates are cleared at the next time unit.

We name the above redesigned solution with a two-phase jamming scheme PFS (Proportional Fair Scheduling). Except the jamming scheme, the remaining AIMD rate adaptation part of PFS is the same as PISD-RS. We can understand how PFS works through an example again on the network shown in Fig. 4-25. Suppose at the beginning of time unit t, contention group S_2 is saturated (the total rate of the five flows in S_2 is almost C), while at the same time contention group S_1 is not saturated. Also suppose flow $(h3, h4)$ first detects that its queue length passes Φ_1. Note that $(h3, h4)$ is more likely to detect channel saturation than other flows because of two reasons. First, $(h3, h4)$ is the only flow in the network that belongs to both of the two contention groups. Second, due to location dependant contentions and the hidden terminal problem, all the four flows to the right of $(h3, h4)$ have priority on accessing the channel than $(h3, h4)$ [35; 19]. By the design of PFS, both nodes $h3$ and $h4$ immediately make a noise jamming, and $(h3, h4)$ becomes a jam leader. All nodes in the carrier sensing ranges of $h3$ and $h4$ will detect the noise jamming and then become MD candidates. In this case, all nodes in the network except $h3$ and $h4$ are now MD candidates. Next, $(h3, h4)$ decreases its contention window and increases its target rate for λC, about 90 PPS, to jam the MD candidates. For flow $(h1, h2)$, because contention group S_1 is not saturated, it will be unlikely to feel the jamming, and hence it will not perform multiplicative decrease. For the four flows to the right of $(h3, h4)$, because S_2 is already saturated, they will detect the jamming. Finally, at the end of the time unit, all flows in S_2 will perform multiplicative decreases. Simulation results are shown in Table 4-5. In the simulation, $T_{noise} = 0.016$ seconds, $\lambda = 0.2$, $\Phi_1 = 30$ packets, $\Phi_2 = 10$ packets. Compared with PISD-RS, the rate of the weakest flow: $(h3, h4)$ is increased. In fact, the flows' rates attained by PFS better approximate the theoretical values of proportional fairness shown in Table 4-4.

Next, we show that the problem of PISD-RS is avoided by PFS. We run a simulation on the network of Fig. 4-26 under PFS. The resulted rates of the flows $(h3, h4)$, $(h5, h6)$, $(h7, h8)$, $(h9, h10)$, and $(h11, h12)$ are 72.4, 74.8, 87.5, 88.3, and 75.0 packets per second, respectively. Unlike PISD-RS, all the five flows gain similar rates now under PFS.

We then test the weighted fairness achieved by PFS. We change the weight of flow $(h3, h4)$ to 2, and the weights of all other flows still remain the default value: 1. Simulation shows that the rates of the six flows are 122.8, 61.8, 61.7, 61.4, and 73.7 packets per second. We see that their rates attained under PFS are in proportion to their weights.

We stress that PFS is suitable for multihop wireless networks or multiple WLANs with multiple contention groups, and the rates attained by PFS approximate proportional fairness. In the following sections, we prove the above facts through both theoretical analysis and additional simulations.

4.9 Analysis on Proportional Fairness

In this section, we prove that flows' rates attained by PISD-RS and PFS approximate proportional fairness [2] . In the prove, we consider the ideal case, where we assume that flows in a contention group make a synchronized multiplicative decrease only when their combined rate gets to the channel capacity C. This assumption does not always hold in real protocol execution, because a jamming with a strength of λC may cause a contention group whose total rate is $(1-\lambda)C$ to make a multiplicative decrease. As λ is small, we claim that the protocols approximate proportional fairness. The prove exploits the theory on convex optimization, and it is inspired by the work in [54].

We formulate the problem as follows. A network is modeled as $G_N = (V_N, E_N)$, where V_N represents the set of all nodes in the network, and E_N represents the set of all single-hop MAC flows. The contention relations among the flows can be summarized using a contention graph $G_C = (V_C, E_C)$ based on the network graph G_N. Here every flow $e_n \in E_N$ has a corresponding vertex $v_i \in V_C$, hence $|E_N| = |V_C|$. Any pair of vertices in G_C has an edge connecting them if and only if the corresponding two flows in G_N contend with each other. A subgraph is called a *clique* if every pair of vertices in the subgraph has an edge between them. A *maximal clique* is a clique that is not contained in any other clique. For simplicity, in the rest of this section we use the term clique to denote maximal clique. We consider the contention graph G_C and assume that the resource is designated to every clique.

We can use a convex optimization model to describe the network. The constraints are determined by the network's topology, while different fairness objectives can be achieved by different utility functions. According to the optimization problem, we have a rate update function using the gradient decent method. By comparing the update function

[2] The analysis in this section is finished by Bo Li, a coauthor of one part of our work, and it is included here for completeness.

with the rate adaptation scheme employed in the protocol of PFS (and PISD-RS), we can derive the utility function in the optimization problem corresponding to the particular protocol. From the utility function we can get to know what kind of fairness the protocol actually achieves.

Suppose a concave function $U_i(x_i)$ is the utility function, where $x_i \geq 0$ denotes the rate of a flow. The optimization problem can be formalized as follows:

$$\max_{x_i} \sum U_i(x_i) \tag{4–4}$$

$$\text{subject to } \sum_{i \in K_j} x_i \leq C \tag{4–5}$$

$$x_i \geq 0. \tag{4–6}$$

K_j in the constraints (4–5) represents the jth clique. These constraints mean that, for each clique, the summation rate of all the flows in that clique can not exceed the capacity C. Due to the continuous and the concave properties of the objective function and the compactness of the constraints, an optimal solution exists. Letting $x(t)$ to be the primal variable for each flow and $p(t)$ to be the dual variable for each clique, we have the *Lagrangian dual*:

$$L(x,p) = \sum U_i(x_i) + \sum_j p_j(C - \sum_{i \in K_j} x_i). \tag{4–7}$$

Since the objective function is concave and the constrains are linear, strong duality holds. According to the strong duality theory, we have the following:

$$x^* = arg \max_{x \geq 0} L(x, p^*) \tag{4–8}$$

Here x^* is the optimal solution for the primal problem and p^* is the optimal solution for the dual problem. Note that, here the dual variable p can be interpreted as the shadow prices of the constraints. Each p_j indicates the tension of each clique. When the clique's capacity has been almost used out, the price p_j is high; on the opposite, when the clique's capacity has a lot of free space, the price p_j is low. Since every term containing x_i in the *Lagrangian* function is independent to $x_j, (j \neq i)$, the gradient of x can be decomposed to every component. Suppose we use a gradient descent algorithm for the primal to achieve the optimal, for every component, we have

$$x_i(t+1) = x_i(t) + s_i(x_i)L'(x_i). \tag{4–9}$$

Here $s_i(x_i)$ is a step size. It can be a constant or a function of x_i. Usually, this function is a linear function of x_i.

In the AIMD algorithm used in PFS (and PISD-RS) we have

$$x_i(t+1) = \begin{cases} x_i(t) + \alpha w_i & \text{if } \sum_{i \in K_j} < C \qquad \forall j \\ \beta x_i(t) & \text{if } \sum_{i \in K_j} = C \;\; for\ some\ j \end{cases}$$

Suppose at the equilibrium point, there are n_i multiple decreases between two consecutive additive increases (n_i is not necessarily an integer), we have

$$x_i(t+1) = x_i(t) + \alpha w_i - n_i(1-\beta)x_i(t). \tag{4–10}$$

Let $y_i(t) = x_i(t)/w_i$ and substitute it in Equations (4–9) and (4–10) correspondingly:

$$y_i(t+1) = y_i(t) + (s(y_i(t) * w_i)/w_i) * (U_i' - \sum_{i \in K_j} p_j) \tag{4–11}$$

$$y_i(t+1) = y_i(t) + \alpha - n_i(1-\beta)y_i(t). \tag{4–12}$$

When $y_i(t)$ is close to the equilibrium point, $\Delta y_i(t) \rightarrow 0$, Equations (4–11) and (4–12) should have the same derivative for y_i with respective to time t. By making the positive parts and negative parts equal to each other or just a constant factor of difference, we have:

$$\frac{s(y_i(t)w_i)}{w_i}U_i' = \gamma\alpha \tag{4–13}$$

$$\frac{s(y_i(t)w_i)}{w_i}\sum_{i\in K_j} p_j = \gamma n_i(1-\beta)y_i(t) \tag{4–14}$$

Equations (4–13) and (4–14) are for the positive and negative parts respectively. To satisfy Equation (4–14), we should let

$$\frac{s(y_i(t)w_i)}{w_i} = \gamma y_i(t) \tag{4–15}$$

then

$$s(y_i(t)w_i) = \gamma y_i(t)w_i \tag{4–16}$$

or

$$s(x) = \gamma x \tag{4–17}$$

Here γ is a constant factor. The equality of the negative parts shows that, when the constraints for some cliques become tighter and the dual prices become higher, the flows in these cliques perform multiplicative decreases more frequently.

By applying Equation (4–16) into Equation (4–13), we obtain that:

$$U = \alpha ln(x_i/w_i) + D \tag{4–18}$$

where D is a constant. Since the utility function is a logarithm function [39], proportional fairness is achieved.

4.10 Simulations on Proportional Fairness

We perform additional simulations to investigate the effectiveness of our new proposed solutions. Simulations have also been conducted in Section 4.7 and 4.8, but they are in rather small networks. In this section, we construct a relatively large network and perform simulations on it. We revised ns2 (v2.32) [36] to implement PISD-RS, and PFS on top of the IEEE 802.11b DCF. The parameters for DCF use the default values set by ns-2 according to the protocol standards. The transmission range is 250 meters, and the carrier sensing range is 550 meters. Network transmission rate is 11 Mbps. RTS/CTS control packet is turned on. For the parameters in the proposed algorithms, $\alpha = 10$ kBps, $\beta = 0.5$, $\lambda = 0.2$, $\Phi_1 = 30$ packets, $\Phi_2 = 10$ packets. Each packet is 1000 bytes long. Each run of the simulation lasts 100 seconds.

The network that we built is shown in Fig. 4-27. There is a node at each of the crossing points of a 9 by 9 grid. In the grid, the distance between any two adjacent (horizontal or vertical) nodes is 200 meters. The network consists of 23 MAC flows as shown in the figure. Contention is determined by the carrier sensing range. For examples, flow $f15$ contends with $f9$, $f10$, $f13$, $f14$, $f16$, $f17$, $f18$, and $f19$.

We run IEEE 802.11 DCF, PISD, PISD-RS, and PFS one by one on the network, and the simulation results are shown in Fig. 4-28. First, we see 802.11 DCF has severe fairness problem. Some flows gain extremely high rates, while some flows are starved. In the 23 flows, $f8$, $f15$, $f16$, $f17$, $f20$, and $f21$, are indeed getting a zero (or nearly zero) throughput, which shows that 802.11 DCF is totally unacceptable. We then run the original PISD protocol, under which all flows got similar rates around 60 packets per second. This result is consistent to our study on the problem of PISD in Section 4.7. Because under PISD multiplicative decreases are synchronized throughout the whole network, many flows can not fully utilize their local channel capacity. The results given by PISD-RS and PFS do not have this problem, where some flows gain much higher rates than those in PISD, resulting a much better channel capacity utilization. Basically, both PISD-RS and PFS work well in this network topology. PFS slightly outperforms PISD-RS, since some very weak flows, such as $f8$, $f11$, $f15$, and $f20$, gain higher rates under PFS. The reason is that, under PISD-RS flows with weak channel access ability are likely to fail in channel jammings.

Finally, we want to take a brief discussion on the issue of total throughput. The total rate of all the 23 flows in 802.11 DCF is higher than those in PISD-RS and PFS. There are two reasons. First, fairness is indeed a contrary objective to total throughput optimization. Realizing fairness has to lower the total throughput as a sacrifice. Proportional fairness is introduced to make a balance between fairness and throughput. Second, there is overhead caused by the protocols. Some are tunable (such as by the value of β), some are not. On one hand, we admit that there is still space to improve throughput. On the other hand, the design of PISD-RS and PFS emphasizes robustness and simpleness, which are also very important factors that make a protocol really workable and practical.

4.11 Summary

In this chapter, we have investigated the unfairness problem in CSMA/CA networks. We have shown that existing solutions based on overhearing are not effective when contending nodes are outside each other's transmission range. We have also shown that the existing non-overhearing AIMD solutions do not work either. We then propose our new fairness solution, PISD, which performs proportional increase synchronized multiplicative decrease with background transmission to support not only fairness but also weighted fairness in CSMA/CA networks, including IEEE 802.11 networks. We also enhance PISD by introducing two schemes, PISD-RS and PFS, which are able to achieve proportional fairness in networks consisting of multiple contention groups. To the best of our knowledge, our work in this chapter is the first one that is able to achieve provable fairness in CSMA/CA networks under realistic conditions where the carrier sensing range and the interference range can be much larger than the transmission range.

Table 4-1. Flow rates achieved by different protocols.

flow	f1	f2	f3	f4	f5	f6	f7	f8	f9	f10
PISD	42.7	43.2	43.7	43.7	43.4	42.8	43.7	43.7	43.7	43.6
PISD-b	43.0	45.6	48.1	43.0	48.8	46.2	45.9	43.4	46.1	46.0
DCF	74.7	55.1	99.7	51.4	51.4	15.3	15.3	15.3	40.8	40.8
H.-B.	18.6	17.2	36.8	35.5	35.5	47.5	47.5	47.5	84.0	84.0

Table 4-2. Under different weight assignments, flow rates are always proportional to flow weights.

flow	f1	f2	f3	f4	f5	f6	f7	f8
weight	1	1	1	1	1	1	1	1
rate	53.3	51.9	53.2	53.3	53.1	53.4	53.3	53.0
weight	1	2	1	1	1	1	2	1
rate	43.4	82.1	41.6	43.4	41.7	41.7	82.5	43.4
weight	1	2	1	1	2	1	2	1
rate	38.9	75.2	38.6	38.9	77.2	38.9	77.31	38.7
weight	1	1	1	2	1	2	1	1
rate	43.6	41.2	42.9	86.7	43.4	86.6	43.1	43.3
weight	2	1	3	1	1	1	1	2
rate	72.4	35.4	105.5	35.8	36.2	36.3	36.0	72.3
weight	1	2	1	4	1	4	1	1
rate	28.4	55.5	30.9	112.8	30.9	112.6	30.8	27.8

Table 4-3. Flow rates achieved by PISD.

Flow	Rate
$(h1, h2)$	74.0
$(h3, h4)$	73.9
$(h5, h6)$	73.5
$(h7, h8)$	73.6
$(h9, h10)$	73.9
$(h11, h12)$	74.0

Table 4-4. Flow rates achieved by PISD-RS, compared with proportional fairness and baseline 802.11 DCF.

Flow	PISD-RS	P-Fair	802.11
$(h1, h2)$	275.5	375.0	436.7
$(h3, h4)$	32.9	75.0	1.2
$(h5, h6)$	81.7	93.75	114.5
$(h7, h8)$	98.0	93.75	118.1
$(h9, h10)$	96.5	93.75	112.3
$(h11, h12)$	98.7	93.75	115.8

Table 4-5. Flow rates achieved by PFS.

Flow	Rate
$(h1, h2)$	223.5
$(h3, h4)$	49.7
$(h5, h6)$	90.3
$(h7, h8)$	92.0
$(h9, h10)$	86.2
$(h11, h12)$	92.3

Figure 4-1. Network of two flows, (a, b) and (c, d).

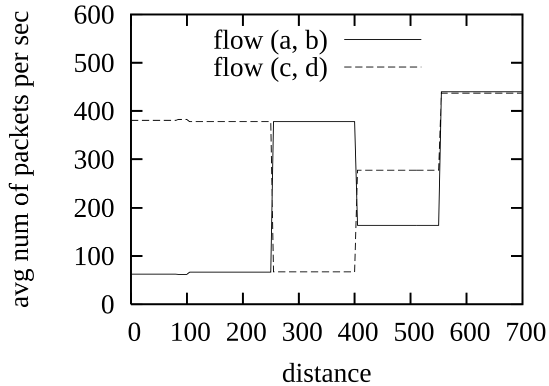


Figure 4-2. Rates of two flows with respect to the distance between b and c. The distance from a to b and that from c to d are both $150m$.

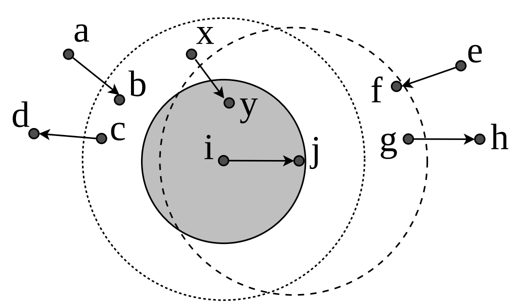


Figure 4-3. Many contending nodes cannot be overheard by i.

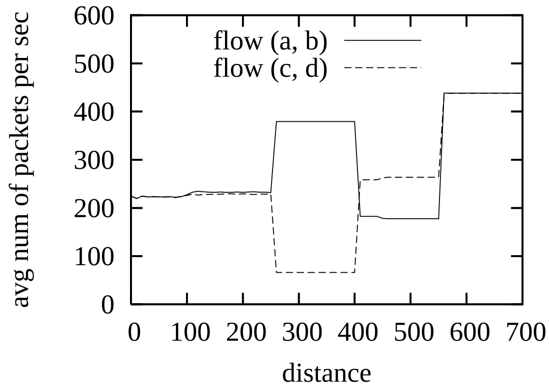


Figure 4-4. In general, Huang-Bensaou protocol does not work if any one of the contending links cannot be overheard.

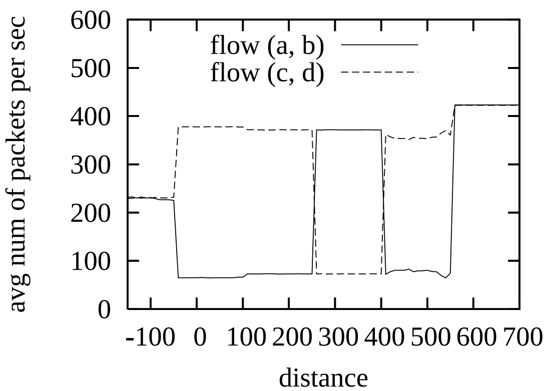


Figure 4-5. Additive Increase Multiplicative Decrease: multiplicative decrease occurs upon transmission failure.

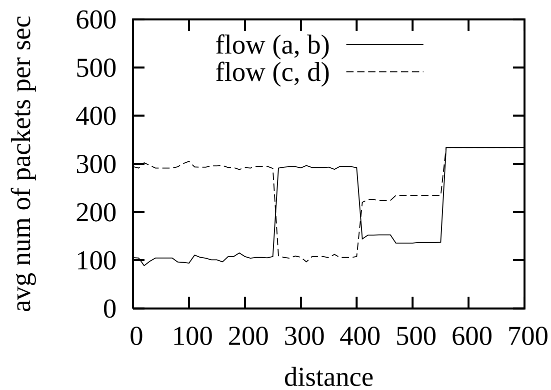


Figure 4-6. Additive Increase Multiplicative Decrease: multiplicative decrease occurs when buffer occupancy passes a certain threshold.

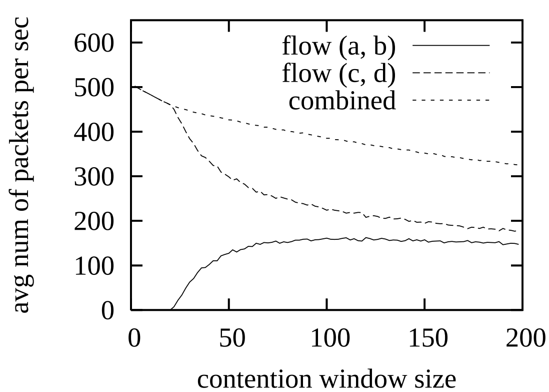


Figure 4-7. Idle sense: The same contention window size does not ensure fairness.

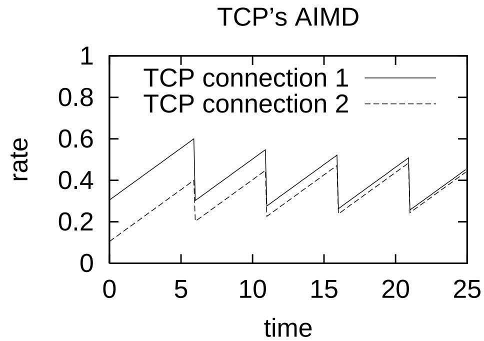


Figure 4-8. Synchronized multiplicative decrease in TCP achieves fairness.

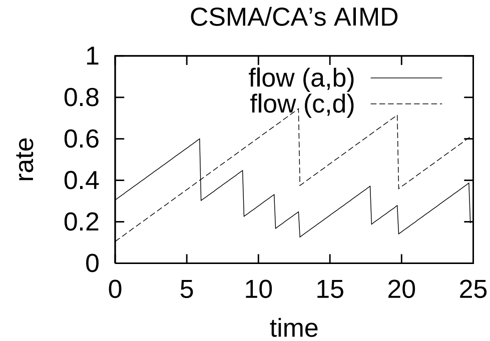


Figure 4-9. Unsynchronized multiplicative decrease in CSMA/CA cannot achieve fairness.

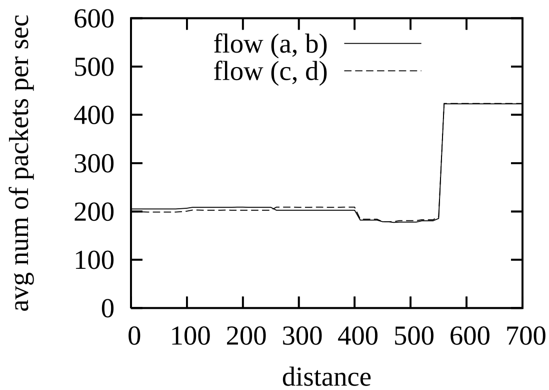


Figure 4-10. Synchronized multiplicative decrease equalizes the flow rates for the network in Fig. 4-1.

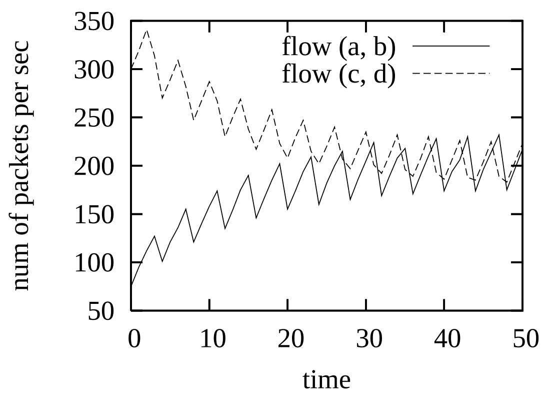


Figure 4-11. Rates of two contending flows under AISD with respect to time.

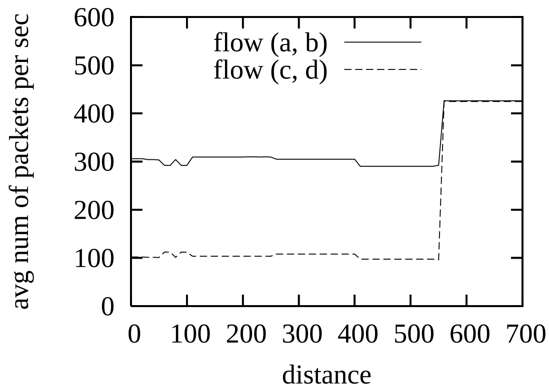


Figure 4-12. Given $w_{a,b} = 3$ and $w_{c,d} = 1$, under PISD, the rate of flow (a, b) is about three times that of flow (c, d).

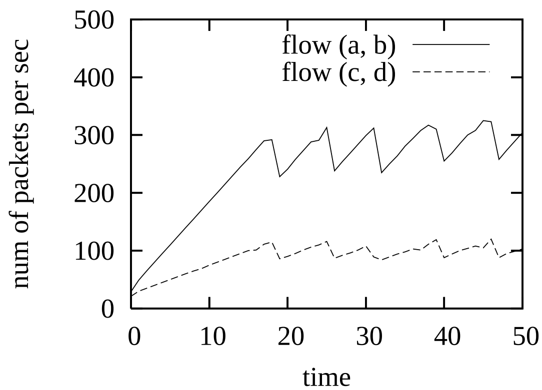


Figure 4-13. Rates of two contending flows under PISD with respect to time. $w_{a,b} = 3$, $w_{c,d} = 1$, and the distance from b to c is 100m.

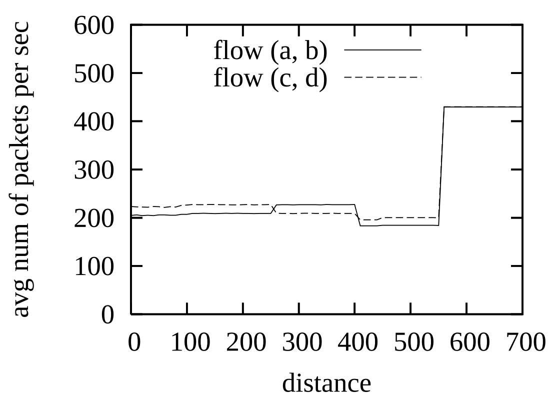


Figure 4-14. Background transmission will utilize some unused channel bandwidth for packet transmission.

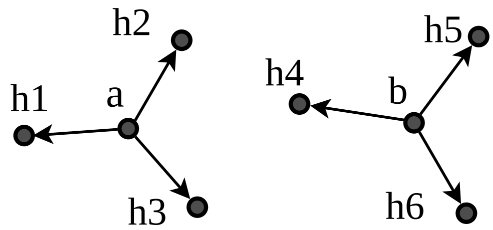


Figure 4-15. Network topology

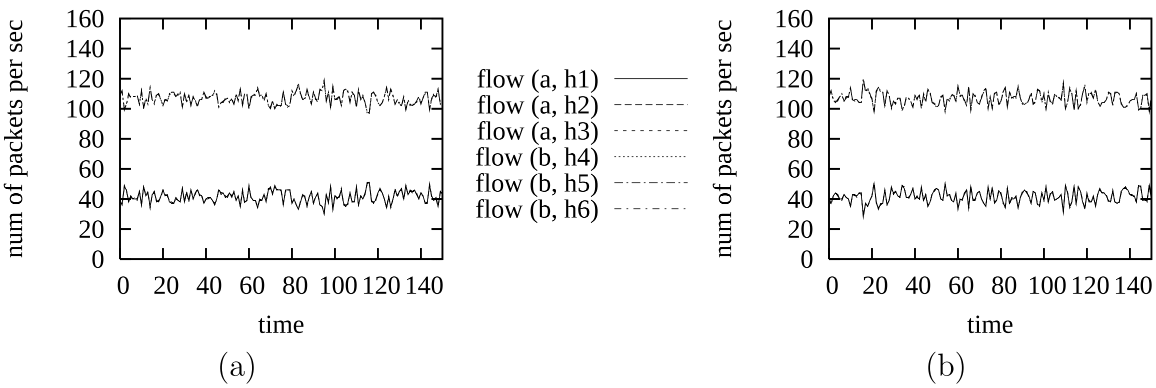


Figure 4-16. (a): Flow rates under IEEE 802.11 DCF. The lower curve shows the rates of flows $(a, h1)$, $(a, h2)$ and $(a, h3)$, which are the same. The upper curve shows the rates of flows $(b, h4)$, $(b, h5)$ and $(b, h6)$, which are the same; (b): Flow rates under Huang-Bensaou protocol. Similarly, the lower curve shows the rates of flows $(a, h1)$, $(a, h2)$ and $(a, h3)$. The upper curve shows the rates of the other three flows.

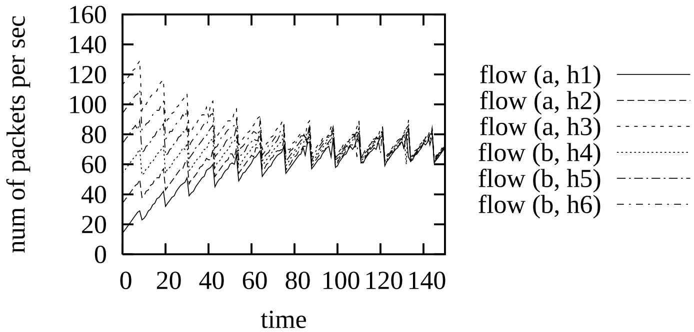


Figure 4-17. Proportional Increase Synchronized Multiplicative Decrease achieves fairness among all flows.

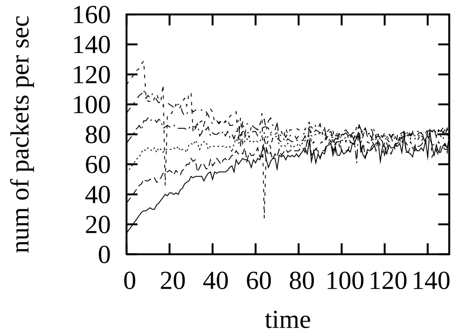


Figure 4-18. Flow rates are slightly improved with background transmission.
.

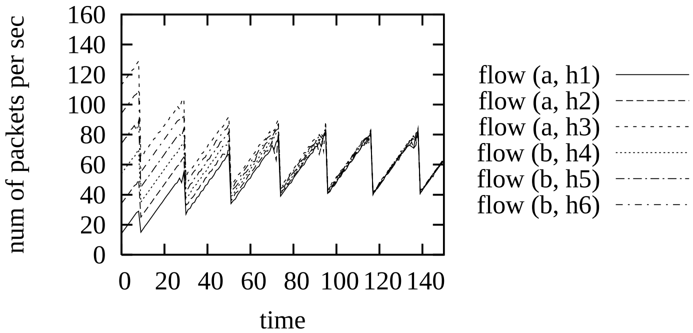


Figure 4-19. Increasing the value of β reduces both convergence time and channel coverage.
.

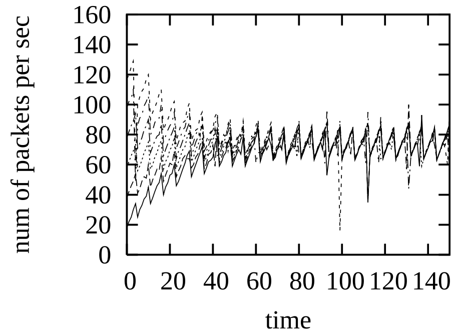


Figure 4-20. Increasing the value of α reduces both convergence time and convergence accuracy.

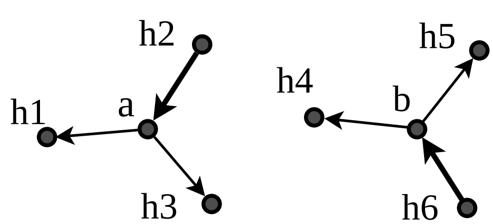
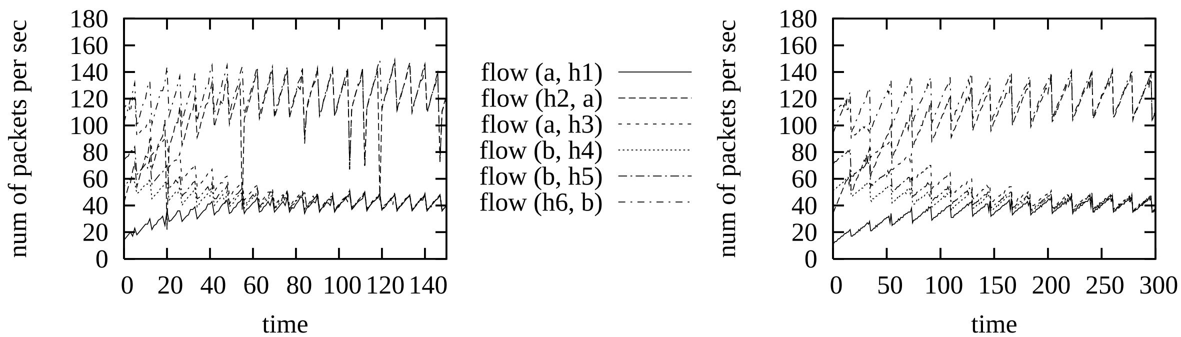


Figure 4-21. Hosts $h2$ and $h6$ are changed to servers.



(a) (b)

Figure 4-22. (a) When the servers each have weight 3 and the clients each have weight 1, the rate of a server is three times that of a client; (b) Downward spikes are reduced when α is decreased.

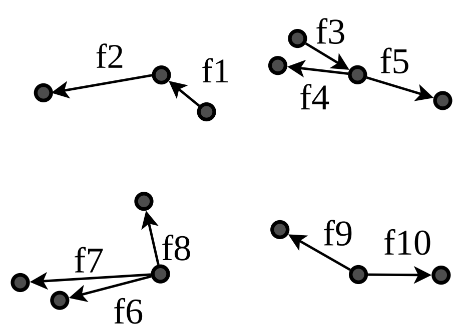


Figure 4-23. Four WLANs network topology

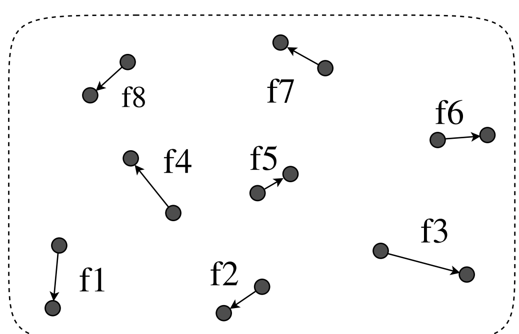


Figure 4-24. Ad hoc network topology

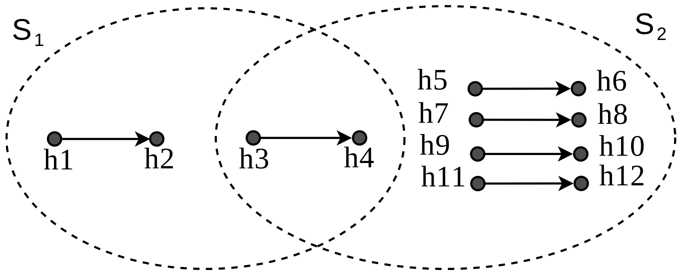


Figure 4-25. Multihop network with six flows.

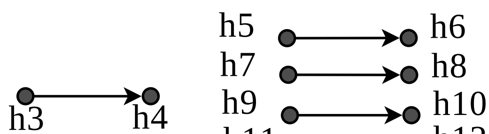


Figure 4-26. Multihop network with five flows.

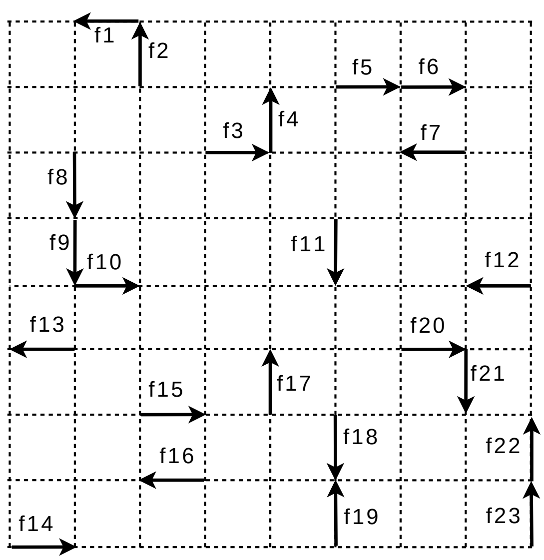


Figure 4-27. Multihop wireless network with 23 MAC flows.

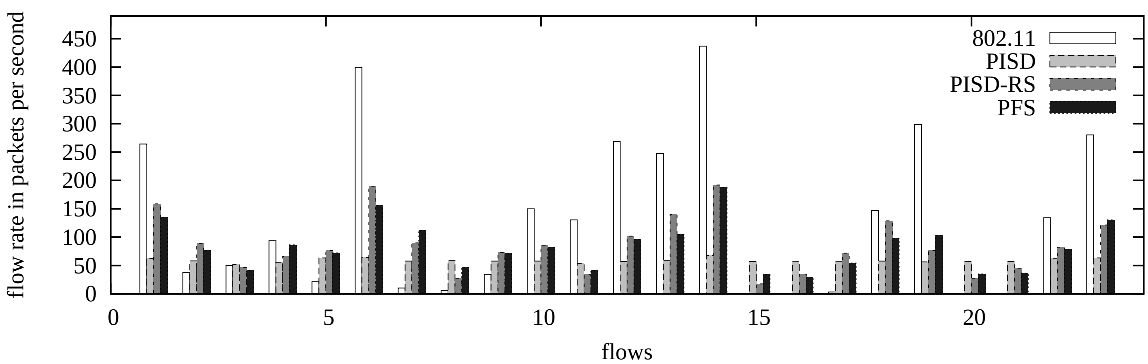


Figure 4-28. Flow rates attained under different approaches

CHAPTER 5
CONCLUSION

Two important problems in wireless networks are studied. They are end-to-end service differentiation and rate assurance for multihop flows, and fairness for single-hop MAC-layer links.

We present a new adaptive rate control function based on two novel protocols, dynamic weight adaptation (DWA) and proportional packet scheduling (PPS), which together enable prioritized rate assurance and sophisticated bandwidth differentiation among end-to-end flows in multihop wireless networks. The new adaptive function represents a paradigm change in fine-level bandwidth management without resource reservation and admission control. Three objectives are achieved: rate assurance objective, bandwidth differentiation objective, and no-starvation objective.

We demonstrate that CSMA/CA networks, including IEEE 802.11 networks, have severe fairness problem in their MAC layer. Most existing solutions require nodes to overhear transmissions made by their neighbors, and the effectiveness of these solutions may be very limited when the carrier sensing range and the interference range are much larger than the transmission range. We propose a new rate control protocol, PISD, which is able to provide fairness with only localized operations and without relying on overhearing. Moreover, as enhancements to PISD, PISD-RS and PFS are proposed to achieve proportional fairness in networks consisting of multiple contention groups.

REFERENCES

[1] A. Banchs and X. Perez, "Providing Throughput Guarantees in IEEE 802.11 Wireless LANs," *Proc. IEEE WCNC*, 2002.

[2] S. Shah, K. Chen, and K. Nahrstedt, "Dynamic Bandwidth Management in Single-Hop Ad Hoc Wireless Networks," *Proc. IEEE PERCOM*, 2003.

[3] C.-T. Chou and K. G. Shin, "Analysis of Adaptive Bandwidth Allocation in Wireless Networks with Multilevel Degradable Quality of Service," *IEEE Transactions on Mobile Compuping*, vol. 3, no. 1, 2004.

[4] P. Gupta, Y. Sankarasubramaniam, and A. L. Stolyar, "Random-Access Scheduling with Service Differentiation in Wireless Networks," *Proc. IEEE INFOCOM*, 2005.

[5] S. Sarkar and L. Tassiulas, "End-to-end Bandwidth Guarantees Through Fair Local Spectrum Share in Wireless Ad-hoc Networks," *Proc. IEEE Conference on Decision and Control*, 2003.

[6] A. Veres, A. Campbell, M. Barry, and L. Sun, "Supporting Service Differentiation in Wireless Packet Networks Using Distributed Control," *IEEE JSAC, Vol. 19, No. 10*, 2001.

[7] S. Kang and M. Mutka, "Provisioning Service Differentiation in Ad Hoc Networks by The Modification of Backoff Algorithm," *Proc. IEEE ICCCN*, 2001.

[8] G. Ahn, A. Campbell, A. Veres, and L. Sun, "SWAN: Service Differentiation in Stateless Wireless Ad Hoc Networks," *Proc. IEEE INFOCOM*, 2002.

[9] K. Karenos, V. Kalogeraki, and S. V. Krishnamurthy, "A Rate Control Framework for Supporting Multiple Classes of Traffic in Sensor Networks," *Proc. IEEE Real-Time Systems Symposium*, 2005.

[10] "IEEE Standard for Telecommunications and Information Exchange Between Systems - LAN/MAN Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Medium Access Control (MAC) Quality of Service Enhancements. IEEE Std 802.11e," Nov 2005.

[11] Y. Yang and R. Kravets, "Contention-Aware Admission Control for Ad Hoc Networks," *IEEE Transactions on Mobile Computing*, vol. 4, no. 4, 2005.

[12] M. Kazantzidis, M. Gerla, and S.-J. Lee, "Permissible Throughput Network Feedback for Adaptive Multimedia in AODV MANETs," *Proc. IEEE ICC*, 2001.

[13] G. Bianchi, "Performance Analysis of the IEEE 802.11 Distributed Coordination Function," *IEEE JSAC*, vol. 18, no. 3, 2000.

[14] F. Cali, M. Conti, and E. Gregori, "Dynamic Tuning of the IEEE 802.11 Protocol to Achieve a Theoretical Throughput Limit," *IEEE/ACM Transactions on Networking*, vol. 8, no. 6, 2000.

[15] B. Li and R. Battiti, "Performance Analysis of An Enhanced IEEE 802.11 Distributed Coordination Function Supporting Service Differentiation," *Int'l Workshop on Quality of Future Internet Service*, 2003.

[16] Y. Yang and R. Kravets, "Throughput Guarantees for Multi-priority Traffic in Ad Hoc Networks," *Elsevier Ad Hoc Networks Journal*, vol. 5, no. 2, 2007.

[17] Y. Jian, S. Chen, L. Zhang, and Y. Fang, "New Adaptive Protocols for Fine-Level End-to-End Rate Control in Wireless Networks," *Proc. IEEE ICNP*, 2008.

[18] B. Bensaou, Y. Wang, and C. Ko, "Fair Medium Access in 802.11 Based Wireless Ad Hoc Networks," *Proc. ACM MOBIHOC*, 2000.

[19] X. Huang and B. Bensaou, "On Max-Min Fairness and Scheduling in Wireless Ad Hoc Networks: Analytical Framework and Implementation," *Proc. ACM MOBIHOC*, 2001.

[20] N. H. Vaidya, P. Bahl, and S. Gupta, "Distributed Fair Scheduling in a Wireless LAN," *Proc. ACM MOBICOM*, 2000.

[21] H. Luo, J. Cheng, and S. Lu, "Self-Coordinating Localized Fair Queueing in Wireless Ad Hoc Networks," *IEEE Transactions on Mobile Compuping*, vol. 3, no. 1, 2004.

[22] S. Chen and Z. Zhang, "Localized Algorithm for Aggregate Fairness in Wireless Sensor Networks," *Proc. ACM Mobicom*, 2006.

[23] A. Rao and I. Stoica, "An Overlay Mac Layer for 802.11 Networks," *Proc. ACM MOBISYS*, 2005.

[24] S. Cai, Y. Liu, and W. Gong, "Analysis of an AIMD based Collision Avoidance Protocol in Wireless Data Networks," *Proc. IEEE CDC*, 2003.

[25] 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," *Proc. ACM SIGCOMM*, 2005.

[26] Q. Xue, W. Gong, and A. Ganz, "Proportional Service Differentiation in Wireless LANs Using Spacing-based Channel Occupancy Regulation," *Mobile Networks and Applications*, vol. 11, no. 2, 2006.

[27] Y. Grunenberger, M. Heusse, F. Rousseau, and A. Duda, "Experience with an Implementation of the Idle Sense Wireless Access Method," *Proc. CoNEXT*, 2007.

[28] Y. Jian and S. Chen, "Can CSMA/CA networks be made fair?" *Proc. ACM MOBICOM*, 2008.

[29] H. Luo, S. Lu, and V. Bhurghawn., "A New Model for Packet Scheduling in Multihop Wireless Networks," *Proc. ACM MOBICOM*, 2000.

[30] L. Tassiulas and S. Sarkar, "Maxmin Fair Scheduling in Wireless Networks," *Proc. IEEE INFOCOM*, 2002.

[31] H. Luo, P. Medvedev, J. Cheng, and S. Lu., "A Self-Coordinating Approach to Distributed Fair Queueing in Ad Hoc Wireless Networks," *Proc. IEEE INFOCOM*, 2001.

[32] Y. Wang and B. Bensaou, "Achieving Fairness in IEEE 802.11 DFWMAC with Variable Packet Lengths," *Proc. IEEE GLOBECOM*, 2001.

[33] B. Li, "End-to-End Fair Bandwidth Allocation in Multi-Hop Wireless Ad Hoc Networks," *Proc. IEEE ICDCS*, 2005.

[34] T. Nandagopal, T. Kim, X. Gao, and V. Bharghavan, "Achieving MAC Layer Fairness in Wireless Packet Networks," *Proc. ACM MOBICOM*, 2000.

[35] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, "MACAW: A Media Access Protocol for Wireless LANs," *Proc. ACM SIGCOMM*, 1994.

[36] The Network Simulator - ns-2, http://www.isi.edu/nsnam/ns/.

[37] M. Barry, A. Campbell, and A. Veres, "Distributed Control Algorithms for Service Differentiation in Wireless Packet Networks," *Proc. IEEE INFOCOM*, 2001.

[38] C. R. Lin and M. Gerla, "MACA/PR: An Asynchronous Multimedia Multi-Hop Wireless Network," *Proc. IEEE INFOCOM*, 1997.

[39] F. Kelly, A. Maulloo, and D. Tan, "Rate Control in Communication Networks: Shadow Prices, Proportional Fairness and Stability," *Journal of the Operational Research*, vol. 49, 1998.

[40] V. Jacobson, K. Nichols, and K. Paduri, "An Expedited Forwarding PHB," *IETF Network Working Group RFC 2598*, June 1999.

[41] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski, "Assured Forwarding PHB Group," *IETF Network Working Group RFC 2598*, June 1999.

[42] L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zappala, "RSVP: A New Resource ReSerVation Protocol," *IEEE Network*, Sep 1993.

[43] H. Zhang, "Service Disciplines for Guaranteed Performance Service Inpacket-Switching Networks," *Proc. IEEE*, vol. 83, no. 10, 1995.

[44] S. Chen and N. Yang, "Congestion Avoidance based on Light-Weight Buffer Management in Sensor Networks," *IEEE Transactions on Parallel and Distributed Systems, Special Issue on Localized Communication and Topology Protocols for Ad Hoc Networks*, vol. 17, no. 9, Sep 2006.

[45] S. Sarkar and L. Tassiulas, "End-to-end Bandwidth Guarantees Through Fair Local Spectrum Share in Wireless Adhoc Networks," *IEEE Transactions on Automatic Control*, vol. 50, no. 9, September 2005.

[46] D. M. Chiu and R. Jain, "Analysis of the Increase and Decrease Algorithms for Congestion Avoidance in Computer Networks," *Computer Networks and ISDN Systems*, vol. 17, pp. 1–14, 1989.

[47] Y. R. Yang and S. S. Lam, "General AIMD Congestion Control," *Proc. IEEE ICNP*, 2000.

[48] S. .Jin, L. Guo, I. .Matta, and A. Bestavros, "TCP-friendly SIMD Congestion Control and Its Convergence Behavior," *Proc. IEEE ICNP*, 2001.

[49] J. Crowcroft and P. Oechslin, "Differentiated End-to-End Internet Services Using a Weighted Proportional Fair Sharing TCP," *ACM SIGCOMM Computer Communication Review*, vol. 28, no. 3, pp. 53–69, 1998.

[50] C.-Y. Wan, S. B. Eisenman, and A. T. Campbell, "CODA: Congestion Detection and Avoidance in Sensor Networks," *Proc. ACM SenSys'03*, November 2003.

[51] T. He, J. A.Stankovic, C. Lu, and T. F. Abdelzaher, "SPEED: A Stateless Protocol for Real-Time Communication in Sensor Networks," *Proc. ICDCS*, May 2003.

[52] K. Xu, M. Gerla, and S. Bae, "How effective is the IEEE 802.11 RTS/CTS handshake in ad hoc networks?" *Proc. IEEE GLOBECOM*, 2002.

[53] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda, "Performance Anomaly of 802.11b," *Proc. IEEE INFOCOM*, 2003.

[54] S. Low, "A Duality Model of TCP and Queue Management Algorithms," *IEEE/ACM Transactions on Networking*, vol. 11, no. 4, pp. 525–536, August 2003.

BIOGRAPHICAL SKETCH

Ying Jian was born in Beijing, China, in 1979. He received his B.E. and M.E. degrees in computer science and technology from Tsinghua University, China, in 2001 and 2004, respectively. In 2004, he joined the Department of Computer and Information Science and Engineering at the University of Florida, to pursue his Ph.D. degree. His advisor is Dr. Shigang Chen. His research focused on quality of service in wireless networks and network security.