LEARNING-BASED INTERFERENCE MITIGATION FOR WIRELESS NETWORKS

BY

CHUN-CHENG CHEN

B.S., National Taiwan University, 2000
M.S., University of California at Berkeley, 2002
M.S., University of Illinois at Urbana Champaign, 2005

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2009

Urbana, Illinois

Doctoral Committee:

Professor Nitin Vaidya, Chair
Professor Klara Nahrstedt
Professor Tarek Abdelzaher
Professor Indranil Gupta

Abstract

Wireless networks have raised great attention in the past decades because they provide tether-free connectivity. Although much of the effort in wireless network research has been spent on reducing the interference among the communication nodes, the problem remains open. In this dissertation, we propose a learning-based approach to alleviate wireless interference. The principle of the learning-based approach is based on the observation that although wireless networks are usually complex and dynamic, information can still be extracted from the data measured in the past. By learning from what was observed in the past, we can select the desired operational parameters, react intelligently, and achieve substantial performance gain. In particular, we show that interference mitigation can be achieved in three different aspects: (1) collision avoidance, (2) channel rate adaptation, and (3) spatial reuse.

To Grand Mothers, Mother, and Father.

Acknowledgments

I would like to thank my advisor, Prof. Nitin Vaidya, for his guidance and many valuable insights during the discussions of the research. I would also like to thank Dr. Haiyun Luo, who guided me during the initial years of research in wireless networking. Without them, this dissertation would not have been possible.

I would like to thank Prof. Klara Nahrstedt, Prof. Tarek Abdelzaher, and Prof. Indranil Gupta for serving on my dissertation committee and their valuable perspectives on improving the dissertation.

I would like to thank Vodafone-U.S. Foundation Fellowship and Verizon Foundation Scholarship for financially supporting me during the years of research.

Finally, I would like to thank my family, my mother Mei-chiao Chang, my father Ming-ta Chen, and my sister Hsiang-ju Chen. It is their support during these years that made this dissertation possible.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Wireless networks, e.g. 802.11, Bluetooth, and UWB, have raised great attention in the past decades because they provide tether-free connectivity to the Internet. Embracing wireless technologies, however, does not come without price. Since wireless medium is a shared resource, when one client is communicating with another, other on-going communications in the proximity might be interfered. Thus, much of the effort in wireless network research has been spent on reducing the interference among the communication nodes in wireless networks. Although people have spent decades addressing wireless interference, the problem remains open.

In this dissertation, we adopt a learning-based approach to alleviate wireless interference. In particular, we will show that interference mitigation can be achieved in three different aspects: (1) collision avoidance, (2) channel rate adaptation, and (3) spatial reuse. We introduce each of them below.

1.1 Collision Avoidance

802.11[1] wireless LANs usually rely on careful channel assignments to avoid the interference between neighboring basic service sets (BSSs). However, because there are only a very limited number of orthogonal channels (i.e., 3 for 802.11b/g, and 12 for 802.11a) and because the interference range of an 802.11 transceiver is often long compared with the communication range, the clients and access point of a BSS are often interfered by the clients and access points in neighboring BSSs operating on the same or overlapping channels. This problem is further aggravated by the widespread, autonomous installations of high-speed 802.11 home networks and hotspots. Recently published data on metropolitan area 802.11 coverage [5, 7] show that more than 40% of the access points are operating on channel 6. In Boston, a maximum number of 85 access points were detected in interference range [9, 5], which leads to at least 25 access points ($\frac{85\times 90\%}{3} = 25$) directly interfering with

[1] We use the term “802.11” to denote the IEEE 802.11 family.

each other given that more than 90% access points are 802.11b/g which has only three orthogonal channels. In April 2005, a wardriving survey of a friend's residential subdivision, where 87 single houses are located, showed an average number of 5.6 access points per house are detected active on channel 6.

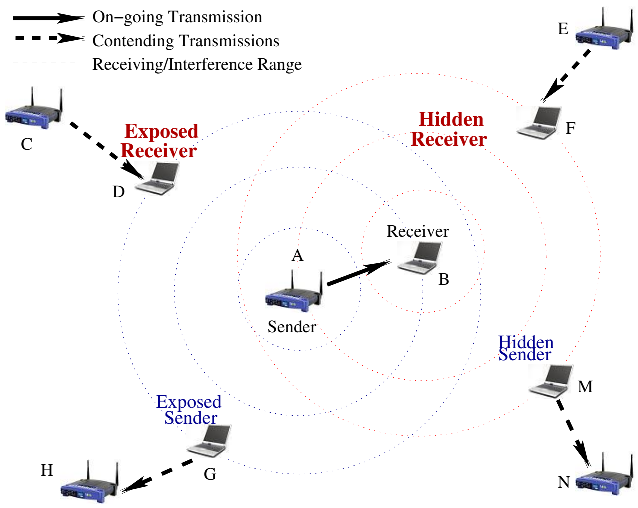


Figure 1.1: Hidden/exposed terminal problem in 802.11 networks. Sender G and receiver D are *exposed* in A's on-going transmission, while sender M and receiver F are *hidden* from A's transmission.

As a result of the inter-BSS interference, clients located around the boundaries of BSSs suffer from the well-known hidden/exposed terminal problem [14, 24, 71]. We illustrate the hidden/exposed terminal problem with Figure 1.1[2]. With 802.11 DCF, sender A and receiver B exchange Request-to-Send (RTS) and Clear-to-Send (CTS) control messages to notify potential competitors and protect the following data packet transmission. As a result, exposed sender G will defer its transmission on reception of A's RTS message and/or its physical carrier sense of A→B transmission. Hidden sender M will also defer its transmission on reception of receiver B's CTS message[3]. Therefore, 802.11's RTS/CTS handshake handles the *hidden/exposed sender problem* well.

However, neither 802.11 DCF nor any other existing research handles the *hidden/exposed receiver problem* within the 802.11's framework of single-channel operation. In Figure 1.1, receiver D is exposed while receiver F is hidden from the active sender A. They have to remain idle until receiver B receives the data packet and sender A receives the acknowledgment. However, neither sender C nor E is aware of the on-going A→B transmission. Hence C or E may initiate RTS requests to their intended receiver D or F in the middle of an A→B transmission. Sender C's RTS message

[2] We use circle to represent the receiving/interference range in Figure 1.1. However, we do not assume circular receiving/interference area in our analysis, as it could be of any dynamic shape in reality.

[3] FAMA [26] proposes an elegant improvement with *long dominating CTS* to handle hidden senders that miss the receiver's CTS and later interfere the data packet reception.

will collide with the sender A's signal at exposed receiver D[4]. Hidden receiver F cannot respond to E's RTS, because F has received a CTS from receiver B and must remain idle until the A→B transmission finishes - a mechanism named "virtual carrier sense" in 802.11. After the timer for CTS expires, sender C (or E) doubles its contention window size and engages in another round of random backoff before it tries to send an RTS again. Following the same reasoning the situation will be worse if RTS/CTS are disabled and two-way DATA/ACK handshake is employed, which is common in 802.11 WLANs.

The consequences of the hidden/exposed receiver problem are severe. First, after a number of retries, the sender drops the head-of-line data packet, resulting in contention-induced packet loss. Second, unsuccessful RTS attempts might mislead the sender to the conclusion that the intended receiver is unavailable or the channel quality at the receiver side is low. In the former case a false link breakage is triggered, resulting in routing instability and thrashing. In the latter case the sender reduces the data rate which aggravates the channel contention. Third, unsuccessful RTS attempts inflate the sender's 802.11 contention window quickly according to the binary exponential backoff algorithm and cause unfair channel access. Fourth, repeated RTS attempts prevent the sender's neighbors from transmitting, lowering the shared channel utilization. Finally, if the hidden/exposed receiver problem happens in wireless LANs, it will persist until the clients move and the contention relation changes.

Although the general hidden/exposed terminal problem has attracted a lot of attention for more than a decade, the vast majority of existing work has been devoted to the hidden/exposed sender problem only[5]. While many research efforts have been invested in mitigating the effects of hidden/exposed receiver (see Chapter 2 for a comprehensive review), the problem remains open. *The fundamental challenge lies in the lack of effective and efficient mechanisms to exchange channel availability information between the sender and the receiver, at packet level time granularity, before a channel access attempt is made.*

In Chapter 3, we propose SELECT, a self-learning collision avoidance mechanism, to address collisions due to hidden/exposed receiver problems in wireless networks. SELECT is based on the observation that *carrier sense with received signal strength (RSS) measurements at the sender and the receiver can be strongly correlated*[6]. This correlation, once established, could be used to provide the sender with information regarding the receiver's channel status and *vice versa.* Once

[4] Unless the signal from sender C is sufficiently strong to capture.

[5] Except for BAPU [15] using *dual-channel* collision avoidance.

[6] This happens especially when sender and receiver are close to each other.

a hidden/exposed receiver is detected the sender can employ appropriate mechanisms to eliminate the negative impacts.

1.2 Channel Rate Adaption

The proliferation of wireless devices on unlicensed frequency bands, e.g., 802.11, Bluetooth, and UWB, has changed the landscape of wireless networking. As the spatial and temporal intensity of such communications accelerate, wireless interference becomes one dominating factor to the success or failure of a transmission. However, the majority of existing wireless rate control algorithms, e.g, ARF [40], AARF [47], ERF [39], link adaptation [54], ONOE [4], and SampleRate [17], are based on packet losses. The implicit assumption is that packet losses signal deteriorated link conditions, and consequently the channel rate should be reduced so that the physical layer switches to a more robust modulation scheme for better interference tolerance. However, at lower channel rate it takes longer to transmit a frame. In a highly interfered wireless network where packets are lost due to interference that comes and goes, depending on the activities of interfering transceivers, a lower channel rate may cause an even higher packet loss ratio due to the prolonged packet transmission time. The increased packet loss ratio in turn further decreases the channel rate. This positive feedback in the rate control loop may quickly drive every interfered transceiver into the lowest rate possible and the entire network into the highest interference level. In the worst case, heavily interfered transceivers can be starved.

Many recent measurement studies [21, 38, 58, 68] on 802.11 wireless networks have confirmed the above phenomenon. For example, it has been shown in [58] that in an 802.11b hotspot setting most of the transmission time is spent sending at 1 Mbps, the lowest rate. The network channel rate oscillates at high frequency and only one or two frames are sent between rate switches. This problem will become worse as the autonomous installations of 802.11 home wireless networks and hotspots quickly spread. Recent reports [9, 5, 7] show that more than 40% of 802.11 home wireless routers are operating on the same channel 6 in metropolitan areas. What's more, a maximum number of 85 802.11 wireless routers were detected in the interference range in Boston [5, 9], among which at least 25 wireless routers must directly interfere with each other since 90% of them are 802.11b/g. With the current interference oblivious rate control algorithms, communications in unlicensed frequency band will soon become the victims of their own success.

Several recently proposed SINR (signal-to-interference-noise-ratio) based rate control algorithms can be applied to mitigate the above problem. For example, RBAR [31] leverages 802.11's per-frame handshake mechanism, i.e., RTS/CTS, to negotiate the best rate for the DATA message given the measured SINR of the RTS message. Although RBAR achieves fine time granularity rate control, it does not address the increase of the packet loss ratio because of the longer packet transmission time at lower channel rate. More importantly, RBAR mandates the per-frame RTS/CTS handshake, which accounts for a minimum 37% or 29% overhead[7] in throughput for 11Mbps 802.11b or 54Mbps 802.11a/g respectively. OAR [59] transmits more data frames per RTS/CTS handshake by observing that channel coherence time usually lasts longer than a single data transmission. CARA [42] differentiates packet loss due to contention from packet loss due to channel errors using RTS probing. Therefore, both of them share the same weaknesses as RBAR does.

In Chapter 4, we present rate-adaptive framing (RAF), a joint channel rate and frame size control algorithm that increases the throughput for interfered transceivers. RAF seeks to characterize the observed interference pattern at the receiver, and uses the pattern to determine the suitable channel rate and the frame size for achieving the maximum throughput. It is based on the assumption that wireless interference can be predictable in short term, since usually the interfering wireless transmissions or background wireless traffic is not purely random even at fine time scale.

1.3 Spatial Reuse

Wireless medium access control is one of the most important research topics in the past decades. Since wireless channel is a shared medium, two nearby nodes accessing the wireless medium may cause interference to each other. Take Figure 1.2 for example, when node D sends a packet to F and A sends a packet to B simultaneously, F cannot receive D's packet reliably due to A's ongoing transmission in the proximity. IEEE 802.11 DCF [33], a carrier sense multiple access (CSMA) wireless MAC protocol, adopts the carrier sensing mechanism so that a node transmits a data packet only if the sensed signal before the transmission is below a certain threshold called carrier sense threshold. In the above example, when A is transmitting, D will sense A's signal and wait until A finishes its transmission. Similarly, when flow D→F is active, A will remain silent. Thus, the transmission from D to F is safely protected from A's interference in IEEE 802.11 DCF.

Carrier sensing, however, does not always address the medium access problem properly. For example, although flows D→F and A→B in Figure 1.2 can not be reliable simultaneously, flows A→B and D→E *can* be simultaneously reliable since B and E are far away from the interference source

[7] Although RTS/CTS frames are small, there is constant per-frame PHY and MAC overhead.

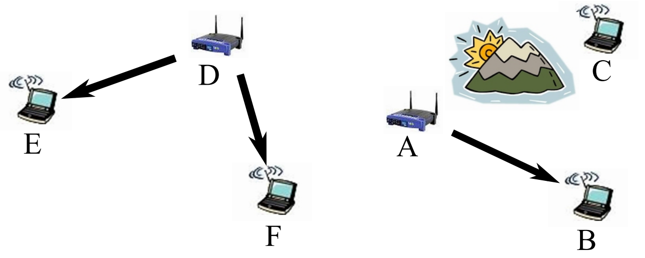


Figure 1.2: Illustration of exposed terminal and hidden terminal problems in wireless networks. A and D are two exposed senders. C is a hidden interferer from A.

D and A, respectively. If we want to protect flow D→F from the interference from A by allowing A and D to carrier sense each other, we cannot but have to sacrifice the concurrent transmission of A→B and D→E. In fact, it is also possible that interferers may not be in the proximity of wireless transmitters or that there could be obstacles separating transmitters and interferers (nodes A and C for example). In these scenarios, the interferers are hidden from the sender nodes, reducing the effectiveness of carrier sensing. Obviously, decreasing the carrier sense threshold improves the chances of detecting interferers farther away with the cost of silencing more flows in the proximity that could potentially be concurrent. On the other hand, increasing the carrier sense threshold allows more nearby flows to be concurrently active, while less number of interferers are silenced. Despite the fact that many research efforts [48, 49, 72, 77, 78] have been spent on tuning the carrier sense threshold to maximize the spatial reuse, the problem remains open.

In Chapter 5, we propose opportunistic carrier prediction (OCP), an approach to allow each wireless sender to opportunistically access the medium. OCP's rationale is based on the observation that interference from the past can be a good indicator for the outcome of future packet delivery. Therefore, each sender maintains an empirical summary of *interference relationship* (who interferes my receiver and who is interfered by me) in the proximity. When the sender overhears that an interferer is in transmission or a flow that will be interfered by the sender's transmission is active, it defers its transmission until both the interfering sender and the interfered flow finish their transmissions.

1.4 Dissertation Outline

In this dissertation, we illustrate how a learning-based approach can mitigate interference from the aspects of collision avoidance, rate adaptation, and spatial reuse. We first review the existing works on alleviating interference due to the long-haunted hidden/exposed terminal problems in wireless networks in Chapter 2. We then present in Chapter 3 the proposed self-learning collision avoidance mechanism, SELECT, to alleviate the hidden/exposed terminal interference. We present in Chapter 4 the proposed rate-adaptation mechanism, RAF, that is tailored for the interfered wireless networks. We present opportunistic carrier prediction (OCP) in Chapter 5 to tackle interference and improve space reuse. We finally discuss the future work in Chapter 6 and conclude the dissertation in Chapter 7.

Chapter 2

Literature Review

The related work can be classified into three categories. One category discusses interference mitigation and collision avoidance with wireless resource provisioning, load control, and cross-layer coordination. Another category discusses state-of-the-art rate adaptation mechanisms. The third category discusses various mechanisms for improving spatial reuse. In this chapter, we review these three categories and compare them with SELECT, RAF, and OCP, respectively.

2.1 Mechanisms for Collision Avoidance

2.1.1 Medium Access Control

Medium access control can be either contention based or schedule based. Contention based schemes are usually preferred in data networks because they achieve higher utilization due to statistical multiplexing gain, are easier to implement, and are robust to synchronization errors. Collisions have to be resolved in contention based medium access control. Different from widely adopted collision detection in wired network (e.g., IEEE 802.3 Ethernet), collision avoidance is usually adopted for wireless medium access control since it simplifies the wireless transceiver.

IEEE 802.11 [33] medium access control, predominantly Distributed Coordination Function DCF, is probably the most popular CSMA/CA MAC. 802.11 was designed for infrastructure mode, where nodes in a Basic Service Set (cell) can be *at most two hops* away from each other and *communicate only with the centralized access point.* 802.11 DCF handles hidden/exposed senders well but does not address the hidden/exposed receiver problem, since the latter usually does *not* exist in a network operating in infrastructure mode. However, as shown in [14, 24, 71] the hidden/exposed receiver problem manifests itself in multihop 802.11 networks with nodes distributed three or more hops from each other. MACAW [16] and FAMA [26] are early proposals on CSMA/CA wireless MAC. They handle hidden/exposed sender problem even better than 802.11, but leave the hidden/exposed receiver problem open.

Scheduling based MAC are proposed to achieve contention-free communication [13, 12, 56]. However, they assume that contenders have consistent knowledge of their two-hop neighborhood and traffic distribution, which may be prohibitively expensive to maintain with a single shared channel. The scheduling is based on random hashing of node identifiers, therefore does not respond well to traffic dynamics at packet-level time granularity. SELECT achieves collision avoidance at packet-level time granularity without maintaining explicit neighborhood information.

2.1.2 Multi-channel Enhancement

BAPU [15] addresses the hidden/exposed receiver problem, but it requires two channels and uses a dedicated control channel for signaling. Recently, several multi-channel variations of 802.11 medium access control are also proposed, e.g., SSCH [10] and MMAC [63], but their goal is to increase network capacity, not to handle hidden/exposed receiver. The design of single-transceiver multiple channel involves extra latency for channel synchronization and requires time-synchronization hardware [10], or introduces significant modification on 802.11 [63]. We will see in Chapter 3 that it is possible to address the hidden/exposed receiver problem using only one single shared channel without any communication overhead.

2.1.3 Receiver-based MAC and Carrier Sense Tuning

Another interesting option is to use receiver-initiated collision avoidance [28, 66]. It solves the hidden/exposed "receiver" problem (Figure 1.1), since the receiver initiates channel access. However, the hidden/exposed sender problem (Figure 1.1) emerges since a hidden/exposed sender may not be able to respond to the receiver's poll. Our proposed SELECT scheme in Chapter 3 keeps the existing collision avoidance mechanisms that are mature in dealing with hidden/exposed sender, and designs additional mechanism to handle hidden/exposed receiver.

Optimal MAC carrier sense threshold has been studied to maximize the channel reuse [72, 77]. As we will show in Section 3.1.3, simply comparing receive signal strength (RSS) with a single "adaptive" carrier sense threshold loses a lot of information about the relationship between RSS and the success ratio of channel access, leading to either conservative channel reuse or collisions due to hidden/exposed receivers.

2.1.4 Directional Antennas and Resources from Cross Layers

Directional antenna with centralized [64] and distributed [22, 55, 65] MAC enable higher channel spatial reuse (due to reduced spatial interference) and longer transmission range (due to directional antenna gain) compared with omni-directional antenna. However, new “directional” hidden terminal and “deafness” problem emerge and call for more research. Although we describe and analyze the proposed SELECT protocol in Chapter 3 in the context of 802.11 DCF omni-directional antenna, it could be tuned to work with directional antenna, as part of future research.

False blocking, as one of the negative effects resulted from hidden/exposed receiver problem, can be solved with MACAW’s DS (Data-Sending) message [16], Negative CTS [15], and RTS validation [57]. Compared with SELECT, these designs do not address other known/unknown negative effects of hidden/exposed receiver, cannot be made compatible with 802.11 standard, and therefore are difficult to deploy.

Rate-based access control [52], interference-aware routing [34], QoS admission control [74, 62], opportunistic scheduling control [59], fair scheduling [27], interference-aware queue management [70], and TCP over multihop wireless [24, 14] can all help alleviate the channel contention through improved channel utilization and/or fairness. SELECT directly attacks hidden/exposed terminal problem at MAC layer, and could work in concert with these designs.

2.2 Mechanisms for Rate-adaptation

2.2.1 Packet-loss-based Rate Adaptation

The majority of existing rate control algorithms, e.g., ARF [40], ERF [39], AARF [47], link adaptation [54], ONOE [4], and SampleRate [17], are based on packet losses. In highly interfered wireless networks, in particular those defined in the unlicensed frequency bands, reducing the channel rate on packet losses increases the contention for the shared wireless channel. It therefore aggravates interference and further increases the packet loss ratio. RAF is designed to address both interference and noise, through adaptation based on receiver side carrier sense.

2.2.2 Receiver-based Rate Adaptation

In contrast to the above rate control algorithms that are based on packet losses, RBAR [31] controls the channel rate using SINR. An RBAR receiver determines the highest data rate supported by the

SINR of the RTS message, and informs the sender of the rate with the CTS message. OAR [59] adopts similar idea. Moreover, OAR exploits the fact that the channel coherence time usually lasts longer than one data transmission time, and transmits more than one data frames per RTS/CTS handshake. CARA [42] uses RTS probing to differentiate packet loss due to contention from packet loss due to channel errors. However, all above designs mandate the RTS/CTS handshake. In realistic 802.11 WLAN or wireless mesh network deployments, the RTS/CTS option is almost never turned on because it has been shown through both analysis [69] and experiments [2] that RTS/CTS neither mitigates the hidden/exposed terminal problem nor improves the throughput, as long as carrier sense range is more than twice the communication range. RAF avoids the RTS/CTS overhead (37% and 29% overhead for 11Mbps 802.11b and 54Mbps 802.11g as shown in [19]) completely. HRC [30] uses SINR to fine-tune loss-based rate controller. RAF is different in the way SINR is used for rate control. Link adaptation based on received signal strength [23, 53] is relevant to our design in that its rate adaptation is based on the received signal strength at the receiver. Furthermore, RAF builds into its rate adaptation the frame size control, an indispensable component for throughput improvement in interfered wireless networks but currently missing from existing rate control algorithms.

Note that RBAR, OAR, CARA, HRC, and RAF's rate control algorithms are all based on SINR, while recent measurements [8, 17] on existing 802.11 mesh suggest that SINR is not a good predictive tool for the successful delivery of a packet. At the first glance, our approach seems to be contradictory to the measurements in [8]. The fact is that the SINR interface exposed by the existing 802.11 wireless card driver only outputs the *average SINR over many received packets*[1]. Indeed as shown in [8, 17], *average* SINR may not correlate well with the success or failure of *individual* packet delivery. In contrast, RAF is based on the carrier sense of the total interference and noise level at the time granularity finer than one packet transmission time. It does put higher requirement on the physical layer *carrier sense* module, of which the performance and impact have been recently evaluated [37].

There are a number of proposals on transmission power control for optimizing network capacity in the literature. For example, Akella etc. [9] propose PARF and PERF that extend ARF [40] and ERF [39] respectively for conservative power control. We do not control the transmission power in the current design of RAF. Incorporating power control into RAF's rate and frame size control framework will be part of our future work.

[1] We have confirmed this fact with the authors of [8].

2.3 Mechanisms for Improving Spatial Reuse

Many protocols have been proposed for medium access control in wireless networks. MACA [41], MACAW [16], and FAMA [26] are the proposals for handling exposed/hidden terminal problems. They mainly designed floor acquisition schemes through the exchange of specific control packets (RTS, CTS, DS, ACK, etc.) to selectively silence wireless nodes in the network for interference avoidance. IEEE 802.11 DCF [33], a CSMA/CA protocol, adopts not only physical carrier sensing but also virtual carrier sensing (RTS/CTS) so that two nodes in a cell that are two hops away know the existence of each other through RTS/CTS floor acquisition. However, since RTS/CTS incurs at least 37% and 29% overhead for 11Mbps 802.11b and 54Mbps 802.11a/g respectively [19], virtual carrier sensing is turned off by default in practice.

Since physical carrier sense is adopted in IEEE 802.11, many works have studied tuning the optimal carrier sense threshold to maximize the spatial reuse. [77] derived theoretical estimation of the optimal carrier sense threshold based on SINR interference model. They also proposed a distributed algorithm adapting carrier sense threshold in [78]. However, the above studies ignored the impact of MAC overhead in the analysis and it has been shown [72] that the aggregate throughput could suffer from a significant loss if MAC overheard is not considered properly. [48] proposed an enhanced carrier sensing mechanism by adapting the EIFS duration based on the length of packet types (RTS, CTS, DATA, ACK) observed on the medium. [49] adapts carrier sense threshold based on transmitter-receiver distance. [37] experimentally verified the efficacy of carrier sense and identified existing problems of carrier sense. All the above works proposed the solutions within the context of carrier sensing, while we go one step further to incorporate carrier sense as part of our medium access scheme in OCP.

Besides controlling the carrier sense threshold, other works have studied controlling the modulation rate [20, 31, 59], transmission power [51], or a combination of them [25, 43, 73] to allow for more concurrent active flows in one-hop or multi-hop wireless networks. [31] proposed to utilize RTS/CTS control packet and let the receiver decide the modulation scheme for the next coming DATA packet. [59] proposed to further opportunistically transmit more DATA packets when the channel condition at the receiver is good. POWMAC [51] inserted an interference margin in the CTS packet to tolerate certain amount of interference at the receiver, thereby increasing the number of concurrent active flows. Finally, [25, 73] jointly control the transmission power and carrier sense threshold, [43] proposed to tune modulation scheme, transmission power, and carrier sense threshold all together to improve spatial reuse in multi-hop wireless networks. OCP only focuses on the carrier sensing aspect for spatial reuse. We leave it as future work for incorporating modulation scheme or power control into the OCP framework.

Chapter 3

Self-learning Collision Avoidance

SELECT is a sender-side only collision avoidance mechanism. It addresses hidden/exposed receiver problem at the packet-level time granularity and involves zero communication overhead. It does not rely on special hardware support (e.g., multiple channel communication capability as proposed in BAPU [15]). Instead, SELECT only uses instantaneous RSS measurement, a standardized sensory function built-in in most wireless transceivers for carrier sense (e.g., off-the-shelf 802.11 wireless devices), and the success ratio of its channel access attempts, as signaled by ACK or ACK timeout, to infer the receiver's channel condition. Furthermore, SELECT does not rest on any analytical model of wireless signal propagation, which is affected by many factors and very difficult to analytically appraise [76, 8]. Finally, SELECT complies with 802.11's PHY and MAC specifications and is completely compatible with other non-SELECT 802.11 devices. Only small non-disruptive enhancements to collision avoidance are required to incorporate SELECT into 802.11 DCF, as we will further elaborate in Section 3.2.

We first use real-radio measurement data to demonstrate the correlation between sender and receiver carrier sense RSS in a multihop wireless network, which serves as our motivation for this work (Section 3.1). We then describe SELECT which exploits such correlation and *directly* characterizes the relationship between a sender's RSS and its channel access success ratios. Practical issues such as computation constraints, storage constraints, RSS measurement noise, and temporal dynamics of wireless signal propagation are also addressed through careful SELECT design (Section 3.2). We then evaluate the performance through extensive simulations (Section 3.3) and prototype experiments (Section 3.4) in terms of throughput, channel access success ratio, packet losses and fairness. Our results show that in typical hidden terminal scenarios SELECT increases throughput by up to 140% and channel access success ratio by up to 302%. It also reduces contention-induced packet drops and improves fairness. We finally summarize in Section 3.5.

3.1 Motivation

SELECT is grounded on the observation that the correlation between sender side and receiver side carrier sense RSS can be exploited for collision avoidance. In this section, we use measurement data from our multihop wireless testbed to verify the correlation and motivate two design options for further examination in the next section.

3.1.1 Testbed Setup

We choose Crossbow MICA2 sensor motes with ChipCon model CC1000 [1] single-chip RF transceivers to form our wireless network testbed. Each MICA2 mote is equipped with an 8-bit 4MHz microcontroller running a microthread operating system, called TinyOS [6], from its internal flash memory. The memory size available at each node is limited: 128KB of program memory and 4KB of data memory. In our testbed CC1000 radio works in the 433MHz frequency band, and achieves a maximum data rate of 19.2kbps. Depending on the power setting the communication range varies from 1∼20 meters in our lab.

The primary reason we use MICA2 motes in our experiments is the programmability of its CC1000 radio. The TinyOS radio stack for CC1000 transceiver is open-source, and allows byte-level transmission/reception control. Therefore, we can implement our SELECT collision avoidance and evaluate its effect in MICA2 platform. Off-the-shelf 802.11 devices do not expose the interface for carrier sense. They usually have collision avoidance built in the firmware, and the source codes are proprietary. Although CC1000 is far different from 802.11 PHY specifications, their core functions for the implementation of 802.11's DCF MAC protocol are similar (e.g., single-channel asynchronous communication and carrier sense with RSS). Besides, the computation and storage constraints of the MICA2 platform are comparable to the constraints (e.g., memory and computation constraints) when implementing SELECT in 802.11 devices in the future.

3.1.2 Sender/Receiver RSS Correlation

Every mote in our testbed runs a stripped version of 802.11 DCF, derived from [75], with RTS-CTS-DATA-ACK, physical and virtual carrier sense, and random backoff. Physical carrier sense is implemented as a sampling of the ADC (Analog-to-Digital Converter) port. The ADC reads the received signal from CC1000's analog pin and converts it into a 10-bit voltage reading. The voltage reading can be further mapped into RSS in dBm according to [1].

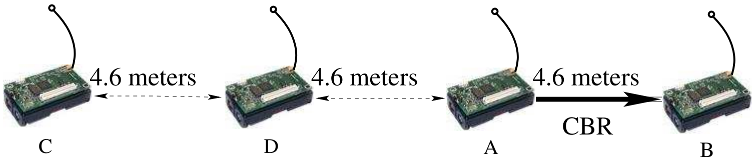
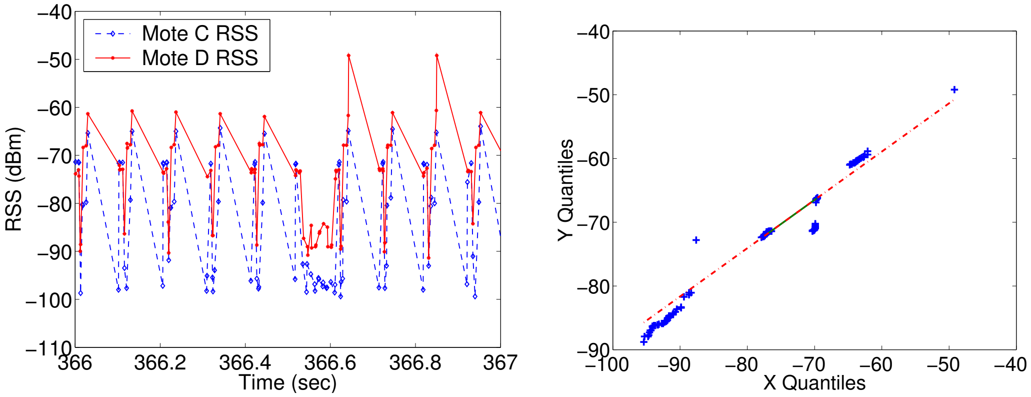


Figure 3.1: Mote D is an exposed terminal.



(a) RSS at C & D

(b) QQ-plot

Figure 3.2: RSS at motes C and D while A is transmitting to B

We first study a simple 3-hop topology with 4 motes, as shown in Figure 3.1. All 4 motes are placed 0.5 meter above the floor. Under the specific power setting mote A and B cannot communicate reliably beyond 4.6 meters. We configure mote A to transmit 65-byte packets as fast as the channel allows, and log the RSS at motes C and D as fast as possible over a 5-second interval. Note that mote D is potentially an exposed receiver if mote C initiates transmission request. Our purpose in this experiment is to construct a scenario of exposed receiver, similar to the one shown in Figure 1.1, and study the relationship between the RSS at a potential sender (C) and the RSS at an exposed receiver (D).

Figure 3.2(a) shows the RSS samples over a 1-second period. The calculated correlation coefficient between these two sequences of RSS samples is 0.878. We also show the QQ-plot[1] of these samples over a 5-second interval in Figure 3.2(b). From the QQ-plot, we can see an excellent fit of the straight line, indicating that the two sequences of RSS samples at motes C and D comply with the same distribution. Similar results are also obtained in the scenario of hidden receiver, as we reverse the flow direction (i.e., B→A).

These results are not surprising. Although the wireless signal attenuates quickly as it travels in the air, the hidden/exposed receivers and their senders usually locate within an area that is close

[1] QQ-plot plots the quantiles of the first data set against the quantiles of the second data set. If the two data sets come from the same distribution, their QQ-plot will fall along a 45° line.

enough to an on-going transmission. Note that this area could be of any shape and could change over time due to multipath fading. Although the on-going transmission (e.g., A→B or B→A) may not be strong enough to be decoded at all nodes in the area (e.g, node C), it may be strong enough to dominate their RSSs in the presence of noise and other interference caused by transmissions further away. Therefore the RSS measurements in the interference area can exhibit strong correlations, reflecting their receptions of the wireless signals from the common source.

3.1.3 RSS v.s. Success Ratio

To facilitate our discussions of the SELECT design in the next section, we perform another experiment with 8 MICA2 motes placed in the topology shown in Figure 3.3. We configure four transmissions: A→B, C→D, E→F, and G→H, where senders transmit 65-byte packets as fast as the channel allows. Note that receiver B is an exposed receiver when C→D is active, and a hidden receiver when G→H is active. Transmission E→F serves as additional interference. We log the carrier sense RSS at sender A before it initiates channel access (with RTS), and the success (reception of CTS) or failure (CTS timeout) of the attempt. We define the success ratio as the number of CTS messages received by a sender over the total number of RTS attempts. Figure 3.4 shows the RSS versus the success ratio (aggregated over 0.5dBm intervals) at mote A during a measurement time period of 75 seconds. We show the stability of such a mapping at two specific RSS readings during an experiment that lasted for 2000 seconds in Figure 3.5. The variation of the success ratio is below 20% with a sampling interval of 50 seconds. These results hold as long as the interference signal dominates the RSS measurements.

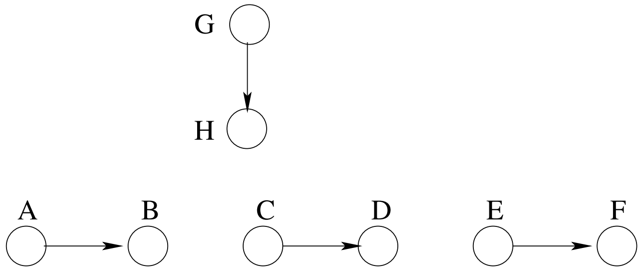


Figure 3.3: Mote B is a hidden/exposed receiver

Figures 3.4 and 3.5 clearly reveal the impacts of hidden/exposed receiver on the success ratio of sender's channel access. First, a carrier sense with low RSS at the sender (A) does not necessarily mean the channel is available at the receiver (B). In fact, all RSSs in Figure 3.4 are below the default carrier sense threshold, but the success ratio of RTS attempts can be as low as 14.3% when sender

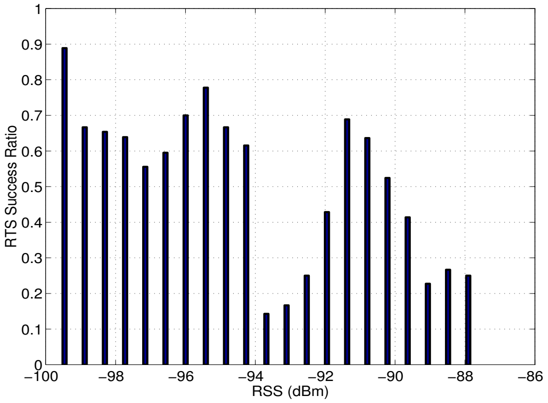


Figure 3.4: RSS v.s. RTS success ratio at mote A

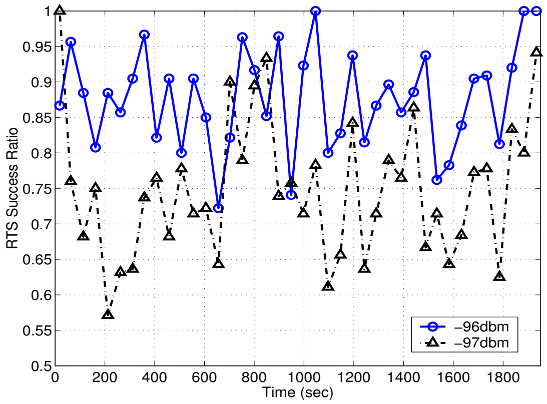


Figure 3.5: RSS v.s. RTS success ratio over time

A's RSS is around -93.68dBm, a scenario when the intended receiver B is likely exposed. Second, the relationship between the sender's RSS and the corresponding channel access success ratio is not monotonic. If a sender only contends for the channel when its RSS is lower than a carrier sense threshold, then depending on the threshold setting, the sender will either suffer from serious channel access failure even though the RSS is relatively low (-93.68dBm), or lose the opportunity to successfully grab the channel when the carrier sense RSS is relatively large (-91.37dBm) but the success ratio is actually high. Finally, the mapping between the RSS and the channel access success ratio can help a sender to significantly improve the success ratio of its channel access by deliberately choosing not to contend when the current RSS and past history suggest a low success probability. Considering the temporal variations of such mappings, the sender should continually update the mapping using historical data in the most recent time window.

3.2 SELECT: Self-learning Collision Avoidance

The analysis of the experiment data presented in Section 3.1.2 and 3.1.3 also sheds light on potential solutions. Intuitively one could leverage the strong correlation between the sender's RSS and the receiver's RSS, as shown in Section 3.1.2, and have the sender estimate the RSS and channel availability at the receiver. We did not take this approach, however, for the following two reasons. First, establishing the RSS correlation requires the receiver to feedback its RSS in a timely manner. This feedback will inevitably involve some signaling between the sender and the receiver, which complicates the MAC and/or PHY layers. Second, estimating receiver's RSS only detects an exposed receiver. The RSS at a hidden receiver could actually be low.

A receiver fails to reply with a CTS in two scenarios. At an exposed receiver the RTS collides with an on-going transmission, while at a hidden receiver the channel is already reserved by the CTS of the on-going transmission. The consequences of these two scenarios are the same from the sender's perspective. Therefore, an exact estimate of receiver's RSS is neither sufficient (when the receiver is hidden) nor necessary (since the sender only needs to know that the RSS at the receiver is low enough). Instead, the sender can skip estimating receiver's RSS, and *directly* establish the mapping between its RSS and the success ratio of its channel access attempts, as suggested in Section 3.1.3 and Figure 3.4. Since the RTS success is signified by the reception of the CTS while the RTS failure is signified by the CTS timeout, no additional signaling between the sender and the receiver is necessary.

Recent measurements [8] using existing 802.11 devices showed that signal-to-interference-noise ratio (SINR), *measured as an average during the interval of a packet transmission time*, may not be a good predictive tool for the successful delivery of the packet. It seems to be contradictory to our approach at the first glance. Note that we do not use SINR of a packet transmission to predict whether the packet will be successfully received. Instead, we seek to correlate the RSS, *measured before a node transmits a packet*, with the packet delivery success ratio. In spirit, this approach is in line with the physical carrier sense mechanism defined in the 802.11 standard and implemented in all 802.11 interfaces.

In the rest of this section we first present the details of maintaining the mapping between the RSS and the channel access success ratio (Section 3.2.1 and 3.2.2). We address the following challenges while keeping our design simple and practical:

- The mapping between the RSS and the channel access success ratio could be complex, as shown in Figure 3.4. How should one represent this mapping with reasonable storage and computation overhead?

- The mapping is affected by many factors such as traffic distribution, network topology, and wireless signal propagation. Since these factors constantly evolve, how can the mapping be adaptive to the changing environment?

We then study the appropriate integration of SELECT collision avoidance with 802.11 DCF (Section 3.2.3). Specifically we answer the following question:

- How to integrate SELECT into 802.11 DCF with minimum change to the 802.11 DCF state machine, while achieving significant performance gains?

3.2.1 RSS-SR Mapping Maintenance

Hidden/exposed receivers complicate the relationship between the sender's RSS and the channel access success ratio. This complexity and the requirement for adaptability invalidate our first thought of representing the mapping (e.g., Figure 3.4) in analytical forms (e.g., least squares fitting with a high-degree polynomial). Given the fact that most 802.11 NICs have at least 128KB SRAM embedded, we choose to maintain the histogram directly, trading a relatively small storage (in hundreds of bytes) for lower design complexity and computation overhead.

Specifically, we divide the range of the RSS, $[RSS^{min}, CS^{thred}]$, into N intervals $[RSS_i^{min}, RSS_i^{max}]$ $(i = 1, \cdots, N)$, where RSS^{min} is set to the estimated noise level or measured minimum RSS (e.g., -100dBm in Section 3.1.3) and CS^{thred} is the default carrier sense threshold. We divide intervals evenly in dBm for efficient lookup. For each RSS interval I_i three state variables are maintained: the number of successful channel access attempts S_i, the number of failed channel access attempts F_i, and the timestamp T_i^{upd} indicating the last time S_i or F_i is updated. Essentially the histogram is maintained as a simple one-dimensional array with N elements of $I_i =< S_i, F_i, T_i^{upd} >$ tuples. The size of the array N can be based on the available memory. Since close RSS measurements are aggregated and represented by a single entry I_i, it also helps suppress RSS measurement errors. We use $N = 300$ in our simulations and 52 in our implementation.

```
// Input - $rss$:      carrier sense RSS
//       - $sf$: 1 if succeed, 0 if fail, $-1$ if no new record
Upd_RSS_SR($rss$, $sf$)
1.  $i = \lfloor (rss - RSS^{min}) / I_{width} \rfloor$; // Locate element $I_i$
                // $I_{width} = (CS^{thred} - RSS^{min})/N$
2.  $\alpha = 1 - (t - T_i^{upd})/T_{win}$; // Adaptive aging factor
3.  if ($\alpha < 0$) then $\alpha = 0$;
                // if ($t - T_i^{upd} > T_{win}$) clear $S_i$ and $F_i$
4.  if ($sf == 1$)
            then $S = 1, F = 0$; // channel access succeeds
     else if ($sf == 0$)
            then $S = 0, F = 1$; // channel access fails
     else $S = 0, F = 0$;          // no new record
5.  $S_i = \alpha \cdot S_i + S$; // Update # of successful attempts
6.  $F_i = \alpha \cdot F_i + F$; // Update # of failed attempts
7.  $T_i^{upd} = t$;        // Update timestamp
```

Figure 3.6: A sender updates the mapping with new record {rss,sf}. $O(1)$ computation overhead

Figure 3.6 shows the pseudo-code for a sender to update the mapping when a new channel access attempt is made and the success/failure is determined. As we can see the histogram only covers the scenario when $RSS < CS^{thred}$, since no channel access attempt will be made if the channel is not detected idle at the sender. We also set RSS^{min} as the estimated or measured noise level, since RSS measurements below the noise level are unreliable.

For the mapping to adapt to the current operating environment, outdated records have to be removed. Therefore, both S_i and F_i, the number of successful/failed channel access attempts, have to be constrained within a recent time window T_{win}. Note that the proper setting of T_{win} depends on the dynamics of the environments, such as the traffic pattern, network topology, and signal propagation. If we denote the new record as $< rss, sf >$ where $sf = 1$ when the channel access attempt succeeds and $sf = 0$ when the channel access attempt fails, the standard approximation to the windowed sum is to apply an aging factor α on S_i and F_i periodically: $S_i = \alpha S_i + S$ and $F_i = \alpha F_i + F$, where $S = sf$ and $F = 1 - S$. The update period is set to $T_{period} \ll T_{win}$ and $\alpha = 1 - T_{period}/T_{win}$. However, updating all N S_i's and F_i's leads to $O(N)$ per-update overhead. We address this problem by adapting the aging factor α based on the time from last update T_i^{upd}:

$$\alpha = \begin{cases} 1 - \frac{t - T_i^{upd}}{T_{win}} & \text{if } t - T_i^{upd} < T_{win} \\ 0 & \text{Otherwise} \end{cases}$$

With the dynamic aging factor the mapping will only be updated *on-demand* (1) after the sender obtains a new record, and (2) before the sender queries the mapping for the channel access success ratio (see Section 3.2.2). Per-update complexity is also reduced to $O(1)$ to guarantee quick return. *The cost we pay is the maintenance of the per-entry timestamp T_i^{upd} of the last update, another tradeoff of storage for complexity.*

3.2.2 RSS-SR Mapping Lookup

With the RSS-SR mapping maintained, a sender could query the mapping to obtain the historical success ratio of channel access attempts under certain carrier sense RSS. Figure 3.7 shows the pseudocode. When a lookup request is received the mapping is firstly updated to remove the outdated records (those records that fall out of the T_{win} window). We then locate the newly updated interval I_i corresponding to the given RSS, and examine if there are enough records established. The historical success ratio is returned if the total number of successes and failures is above certain threshold. Otherwise a 100% success ratio is returned. The reason is that by default an RSS smaller than the carrier sense threshold CS^{thred} is interpreted as idle channel.

Applying a sliding time window on the histogram to remove out-dated records is critical for the system adaptability. A MAC module outputs records of successes and failures to drive the SELECT

```
// Input - rss:   carrier sense RSS
// Output:        historical channel access success ratio
RSS_SR_LookUp($rss$)
if ($rss \geq CS^{thred}$) return 0%;
            // Channel is busy at sender
Upd_RSS_SR($rss$, $-1$); // Remove outdated records
$i = \lfloor \left( rss - RSS^{min} \right) / I_{width} \rfloor$; // Locate element $I_i$
            // $I_{width} = (CS^{thred} - RSS^{min})/N$
if ($S_i + F_i > Min\_Num\_Rec$) then // Enough records
        return $S_i/(S_i + F_i)$; // Success ratio
    else                   // Not enough records
        return 100%; // Channel is idle by default
```

Figure 3.7: A sender queries for historical success ratio of channel access under certain rss. $O(1)$ computation overhead

algorithm to generate an accurate RSS-SR mapping. Without aging out the data, however, the MAC module will generally avoid attempting to access the channel under those RSSs classified by SELECT as signs of low channel access success ratio (see Section 3.2.3). Therefore, no further records will be obtained once an RSS interval is mapped to low channel access success ratio, and the system stagnates at those RSS intervals regardless of potential operating scenario changes. By limiting the valid records within a recent time window and removing the impact of old records, the system can gradually "recover" so that the MAC module will gradually start to try channel access under those RSSs that are previously believed to lead to failures.

3.2.3 Integration with 802.11 DCF

We have instantiated two interfaces in SELECT for interactions with medium access control: `Upd_RSS_SR` and `RSS_SR_LookUp`. Once a MAC module accesses the channel and the result is determined, it calls `Upd_RSS_SR` to update the RSS-SR mapping. In this section, we study in the specific context of 802.11 DCF how a sender can take advantage of better estimation of channel access success ratio provided by `RSS_SR_LookUp`. Before we discuss design options we first briefly review the current collision avoidance in 802.11 DCF.

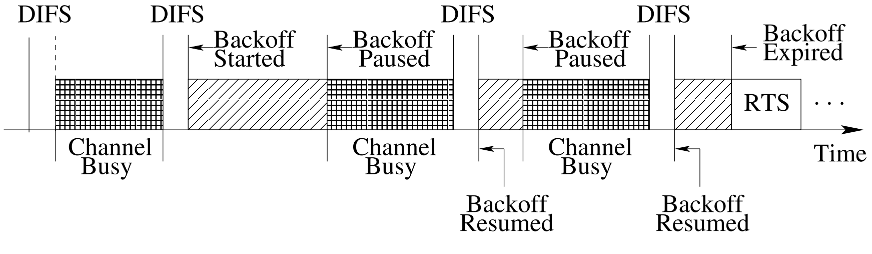


Figure 3.8: 802.11 DCF collision avoidance

As shown in Figure 3.8, an 802.11 DCF sender with a data packet to transmit first monitors the channel for a time period called DIFS: DCF interframe space. If the channel is idle for DIFS without interruption the sender chooses a random number R uniformly from the interval $(0, CW)$, and starts a backoff timer that expires after $(R \cdot aSlotTime)$. CW is the current contention window size, and $aSlotTime$ is one slot time, a constant defined by the 802.11 PHY layer [33]. The backoff timer will be paused when the channel becomes busy, and will be resumed after the channel has been idle for DIFS without interruption. The sender will contend for the wireless channel (with RTS) when the backoff timer expires. If the channel access attempt fails (CTS timeout), the sender doubles its backoff time window CW (a.k.a. exponential backoff), waits for another DIFS, and starts another round of random backoff.

One way to incorporate SELECT into the above process of collision avoidance is to measure the RSS and call `RSS_SR_LookUp` after the backoff timer expires and before the sender sends out the RTS (with four-way handshake) or the DATA (with two-way handshake). If `RSS_SR_LookUp` returns a high success ratio, say above 50%, the sender proceeds to contend for the channel. Otherwise the sender skips the RTS, and executes the 802.11 DCF protocol as if the channel access had failed (doubling CW, waiting for DIFS, and starting a new round of random backoff). This way, it saves the sender the transmission of the RTS and the timeout for the CTS. It also solves the problem of *false blocking* (as analyzed in Section 2.1.4), as the neighbors of the sender will not have to remain idle and wait for a data transmission that actually does not exist.

However, our simulation results show that above approach only improves the performance marginally (∼10%). The reason is that the above approach does not correct the flaws in applying 802.11 DCF collision avoidance to resolve the contention from other BSSs. The exponential random backoff was designed to de-synchronize the channel access among multiple senders within interference range of each other. Collisions happen *only* as a result of two or more senders contend for the channel at the same time after the random backoff, suggesting that the current contention window is not large enough. Therefore doubling the contention window size on collision is appropriate to accommodate the increased tension among competing senders. This mechanism works fine in a single BSS. However, in the case of hidden/exposed receiver due to interference from other BSSs, the collisions may result from the sender's lack of information on the channel status at the intended receiver, not the increasing number of competing senders. Doubling contention window size in this scenario is inappropriate and only causes unfair channel access or even starved flows.

SELECT's `RSS_SR_LookUp` can help the sender eliminate collisions due to the lack of information regarding receiver's channel status. Specifically, by querying RSS_SR_LookUp a sender can *suspend its backoff timer whenever its current RSS suggests low channel access success ratio, and resume the backoff timer whenever its RSS stays at those levels suggesting high channel access success ratio*. By the time the backoff timer expires the channel must be available at both the sender and receiver, with high probability[2].

By suspending the backoff timer whenever the channel access success ratio is low, the definition of "Channel Busy", as shown in Figure 3.8, is extended to representing busy channel at either the sender or the receiver. The channel is now considered "Busy" if either sender's channel is unavailable (due to physical or virtual carrier sense failure) or the channel at receiver is unlikely to be available (signified by low channel access success ratio). With this simple extension of "Channel Busy" the original state flow (Figure 3.8) still applies, implying minimum change to the 802.11 state machine since neither new state nor new state transition is introduced. The *only additional complexity is an RSS_SR_LookUp(rss) call*, after physical carrier sense and virtual carrier sense.

3.3 Simulation Evaluation

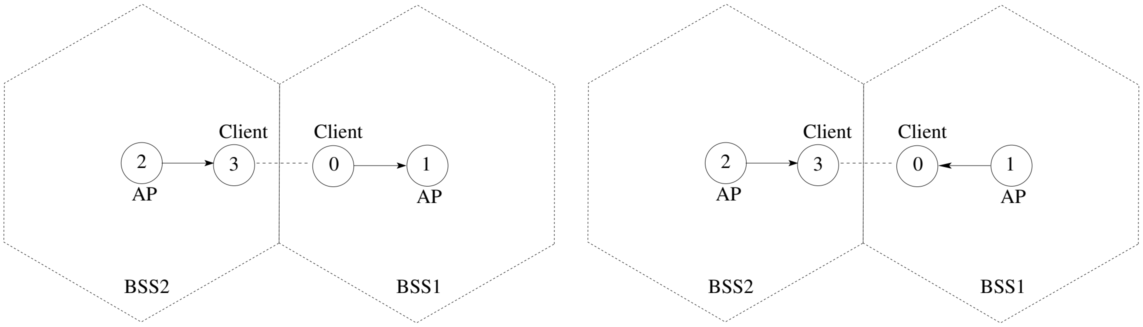
We implement SELECT in *ns-2* simulator version *2.28*. For the default ns-2.28 802.11 implementation, nodes receive packets only when the RSS from the sender is greater than a certain threshold (receive threshold), but the impact of any signal with RSS less than the carrier sense threshold is completely ignored - no matter how many those signals are. This is obviously an over-simplification of the reality. We replace this part of 802.11 functions with the ones developed in [32], so that all signals are taken into account at the receiver, and the combined signal to interference-noise (SINR) ratio is used to determine if an incoming signal can interfere or be received/captured. We use two-ray ground radio propagation model. We use 2Mbps basic rate and 11Mbps data rate based on IEEE 802.11b. Each simulation runs for 45 seconds unless otherwise stated.

In all our SELECT simulations presented in this section we require at least 10 samples in the current time window to make a success ratio prediction. We set the sliding time window (T_{win} in Section 3.2.1) to 2 seconds. Manual static routing is used in the simulation scenario with CBR/UDP traffic.

We use three metrics to evaluate the performance. **Success ratio** is the total number of data packets received reliably over the total number of RTS (DATA) transmitted if four-way handshake (two-way handshake) is used. **Data packet drops per second** denotes the total number of MAC

[2] Considering the sliding window mechanism introduced in Section 3.2.2 for closed-loop system adaptability.

layer DATA packet drops due to collisions, normalized over the simulation duration. Both success ratio and number of packet drops per second can be viewed as metrics for the effectiveness of the collision avoidance. Finally **throughput** at MAC layer (total number of data packets decoded at the receiver / simulation time) serves as the metric for protocol efficiency and fairness in channel sharing.



(a) Exposed receiver problem

(b) Hidden receiver problem

Figure 3.9: Hidden/exposed receiver problem in 802.11 WLANs.

3.3.1 Hidden/Exposed Receiver Problem

We first study the well-known exposed receiver problem as shown in Figure 3.9(a) where sender 0 and 2 are outside the carrier sensing range of each other. Client 3 is an exposed receiver since it is placed in the communication/interference range of client 0, which is associated with another access point (node 1) in a neighboring BSS. Notice that in this configuration flow 0→1 will always succeed in the channel contention because its receiver (AP 1) is not interfered by flow 2→3. We therefore vary the offered load (CBR/UDP rate) of flow 0→1, while keeping flow 2→3 always backlogged (with a 4Mbps CBR) to see how effectively SELECT can avoid collisions for flow 2→3 and keep the channel utilization high.

Figures 3.10, 3.11, 3.12, 3.13 show the number of packet drops per second due to collisions, the throughput improvement ((throughput of SELECT / throughput of 802.11)−1) for flow 2→3 compared with 802.11, channel access success ratio, and throughput profile respectively, when the exposed receiver (client 3) is out of the communication range of the interfering sender (client 0). In each graph, offered load on flow 0→1 is varied. Since sender 0 cannot receive receiver 3's CTS message, RTS/CTS handshake will not help flow 2→3 avoid collisions but increase the overhead. We therefore use two-way handshake (i.e., without RTS/CTS) in 802.11 (as well as SELECT) for the best throughput. As we can see from these figures SELECT significantly reduces the number of packet drops due to collisions by as much as 81.8% and increases the channel access success ratio by more than three-folds when flow 0→1 overloads the channel (with an offered load of 3.4Mbps). As a result, SELECT improves exposed receiver's throughput (flow 2→3) by as high as 140% (Figure 3.11), while increasing the channel utilization from 69.8% to 86.1% (Figure 3.13).

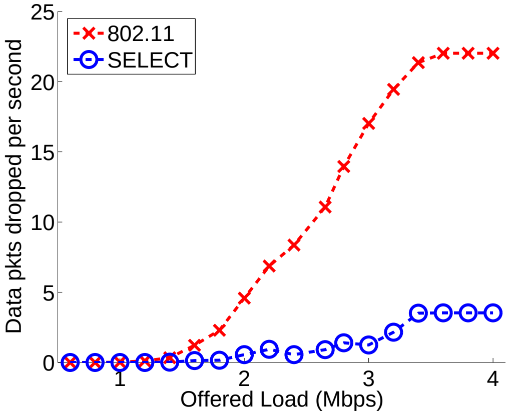


Figure 3.10: Data packet drop at node 2 (w/o RTS/CTS)

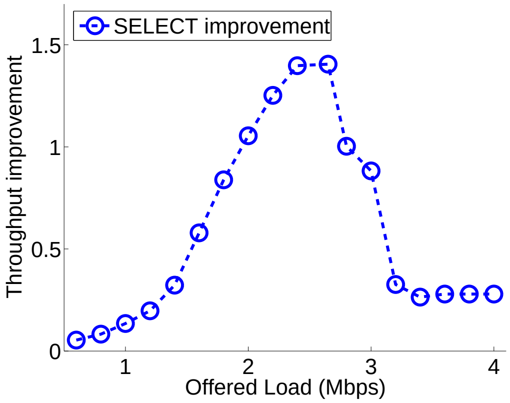


Figure 3.11: Throughput improvement at node 2 (w/o RTS/CTS)

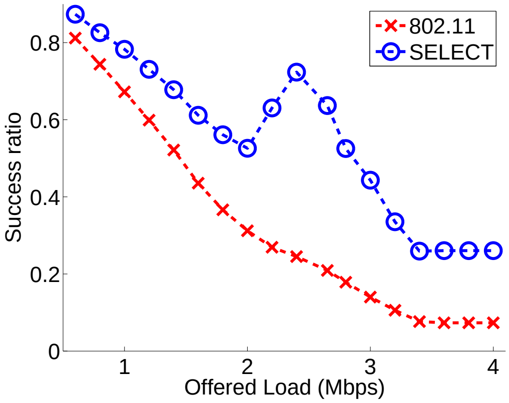


Figure 3.12: Success ratio at node 2 (w/o RTS/CTS)

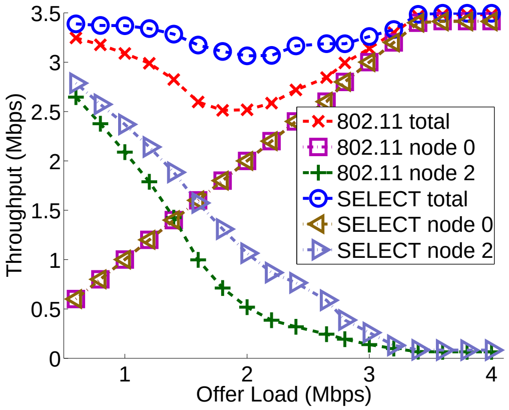


Figure 3.13: Throughput profile (w/o RTS/CTS). The figure will be more clear if it is color printed.

We then move the exposed receiver 3 closer to sender 0 so that they are within the communication range of each other. That is, the dominating sender 0 will be able to yield to the exposed receiver (node 3) if its CTS is received. We therefore enable four-way handshake (i.e., with RTS/CTS) for 802.11 (and SELECT). The results are presented in figure 3.14-3.17. Again SELECT consistently out-performs 802.11 in terms of reduced packet drop, increased channel access success ratio, and improved throughput. Moreover, with the help of RTS/CTS, it achieves almost optimal channel utilization as shown in Figure 3.17.

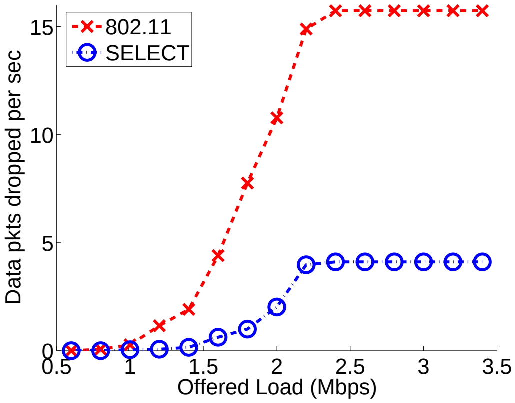


Figure 3.14: Data packet drop at node 2 (w/ RTS/CTS)

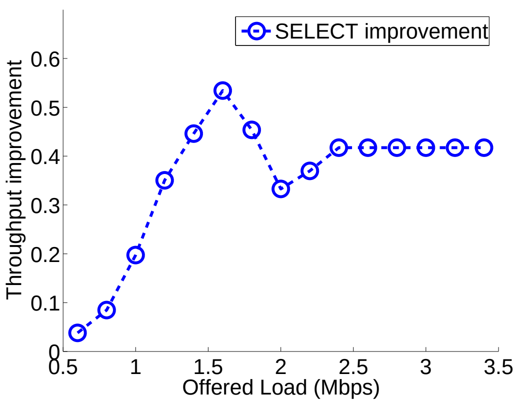


Figure 3.15: Throughput improvement at node 2 (w/ RTS/CTS)

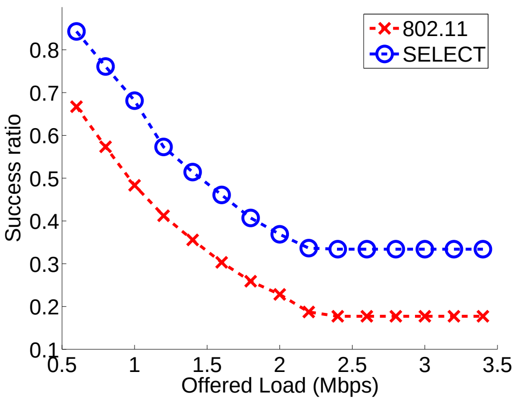


Figure 3.16: Success ratio at node 2 (w/ RTS/CTS)

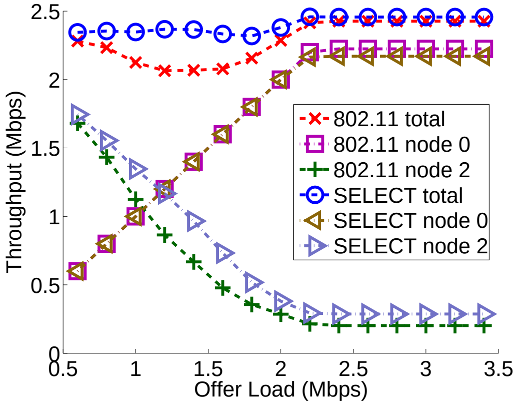


Figure 3.17: Throughput profile (w/ RTS/CTS). The figure will be more clear if it is color printed.

Finally we study the hidden receiver problem in Figure 3.9(b). Again sender 1 and 2 are outside the carrier sense range of each other, and one of receiver 0 and 3 will be the hidden receiver when the other one is actively receiving. Note that in this scenario the topology is symmetric, and 802.11 DCF is able to achieve long-term fairness among those two competing flows. However, as shown in [29], since the two flows compete the wireless medium with each other, the flow that loses the competition will double its contention window size, making it more likely to lose the competition with the other flow. This unfair competition will continue until the losing flow drops the head-of-line packet and resets its contention window size. Our simulations show that SELECT significantly improves the short-term fairness. In our simulations we keep both flows backlogged with 4Mbps CBR/UDP. The normalized throughput of both flows in consecutive 0.4 second intervals for 802.11 and SELECT are shown in Figure 3.18 and 3.19. It is clear that SELECT helps solve the short-term unfairness because both flows intelligently predict when the channel is busy and behave socially to avoid unnecessary collisions.

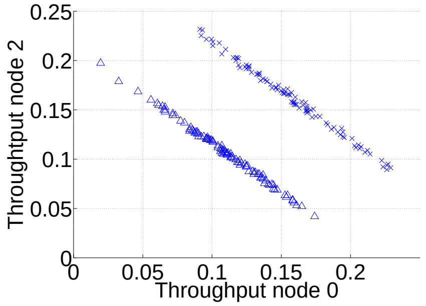


Figure 3.18: Normalized throughput for 802.11 DCF (×: w/ RTS/CTS △: w/o RTS/CTS)

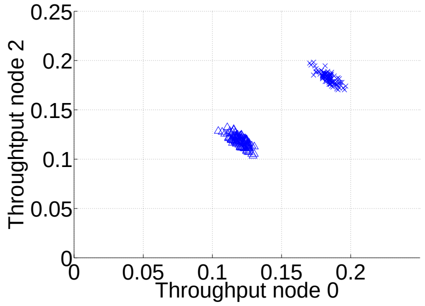


Figure 3.19: Normalized throughput for SELECT (×: w/ RTS/CTS △: w/o RTS/CTS)

3.3.2 Large Random Topologies

We now study the performance of SELECT in large random 802.11b/g WLAN topologies. We study a 5x5 hexagon BSS layout in a two-dimensional space. The size of each hexagon BSS is set so that nodes within the communication range (115m) can communicate with the access point (AP) located in the center of the hexagon. We emulate the following channel assignment strategy. We randomly go through all BSSs one by one, and assign one of the 3 non-overlapping channels that is least used in the six neighboring BSSs. We then randomly place a total number of 50 clients in the network resulting in an average of 2 clients per BSS. Each client associates with its nearest access point on the channel that is assigned to the BSS. The client also establishes a CBR/UDP connection with its access point with the flow direction randomly determined.

We found out that the above channel assignment always results in two or more neighboring BSSs on the same channel, consistent with the published measurement data [5, 7]. With an average of 2 clients per BSS, 60% of our simulated topologies[3] suffer from hidden/exposed terminal problem[4]. Figure 3.20 and 3.21 show the number of packet drops per second and throughput for those hidden/exposed terminals in 24 random topologies. In summary SELECT reduces the number of packet drops per second by 59.8% (mean) with a standard deviation of 27% (Figure 3.20), and improves the throughput of those hidden/exposed terminals by 42.6% (mean) with a standard deviation of 56.4%.

[3] The rest 40% do not suffer from hidden/exposed terminal problem since the above channel assignment algorithm addresses the problem.

[4] We define that two flows A→a and B→b suffer from hidden/exposed terminal problem if (1) sender A and B are outside carrier sense range of each other, (2) receiver a is within communication/carrier sense range of sender B, and (3) receiver b is outside carrier sense range of sender A.

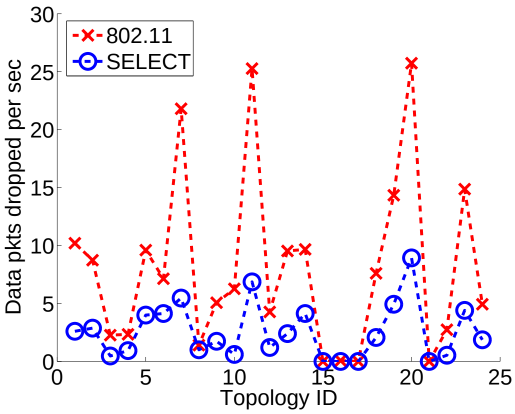


Figure 3.20: Data packet drop in random topologies

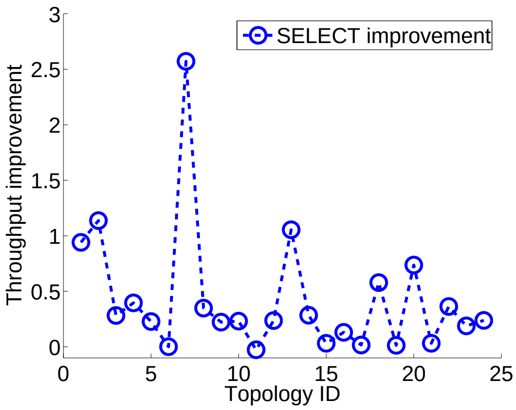


Figure 3.21: Throughput improvement in random topologies

3.3.3 The Effect of Fading

We further evaluate SELECT's performance when fading comes into play. In particular, we emulate the environment of shadowed urban area (path loss exponent equal to 4, standard deviation equal to 9dB, reference distance 1m, and reception rate 95%[5]) as listed in [3]. When fading comes into play, the receive signal strength (RSS) measured at the receiver follows a certain distribution. For example, the signal propagated from node 2 and measured at node 3 in Figure 3.9(a) is plotted in Figure 3.22. Even without interference, the RSS itself could easily range from -70 dBm to -100 dBm. One may ask, under such uncertainty, can SELECT's prediction mechanism still help avoid collisions?

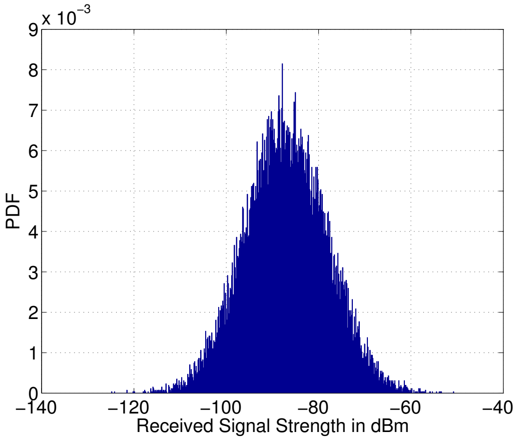


Figure 3.22: Receive Signal Strength measured at node 3 in Figure 3.9(a) with fading

Similarly to Section 3.3.1, we first evaluate the topology shown in Figure 3.9(a). When fading

[5] 95% reception rate means, given a communication distance, the packet can be received reliably 95% of the time.

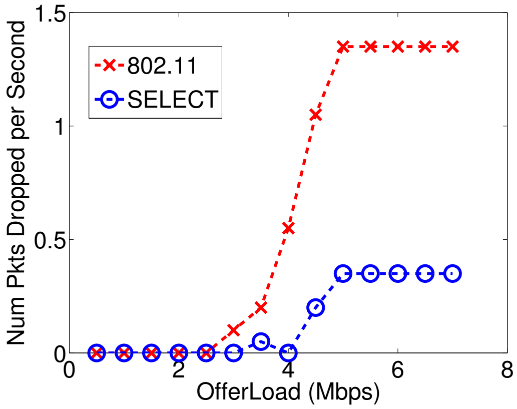


Figure 3.23: Data packet drops per second for SELECT and 802.11 for flow 2→3 in Figure 3.9(a) with fading effect

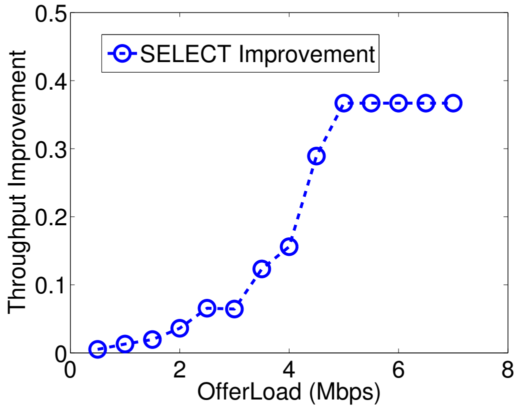


Figure 3.24: Throughput improvement of SELECT over 802.11 for flow 2→3 in Figure 3.9(a) with fading effect

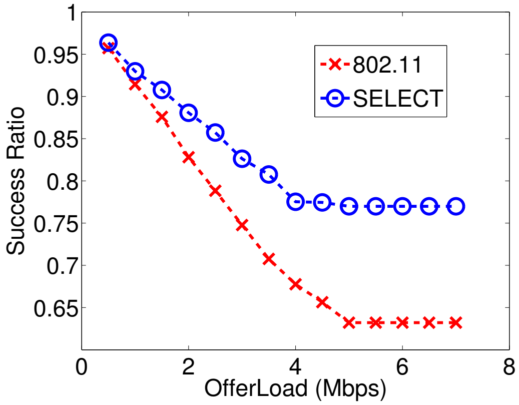


Figure 3.25: Success ratio for SELECT and 802.11 for flow 2→3 in Figure 3.9(a) with fading effect

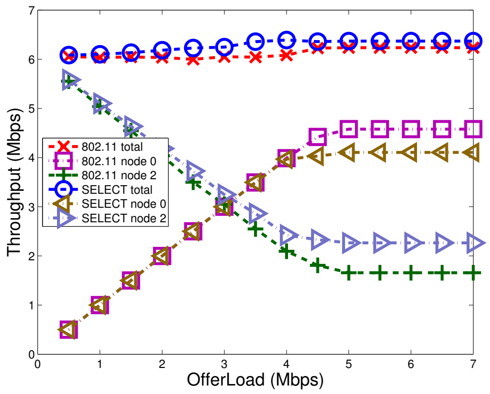


Figure 3.26: Throughput profile for SELECT and 802.11 for flow 2→3 in Figure 3.9(a) with fading effect without RTS/CTS

comes into play, the signal reception becomes probabilistic. So does the carrier sensing. Flow 0→1 may or may not interfere with flow 2→3, depending on whether node 0 can carrier sense node 2. Under our setting, nodes 0 and 2 can carrier sense each other most of the time. As a result, the number of packet drops is significantly reduced compared with Figure 3.10. SELECT, however, still reduces the number of packet drops to 33% of 802.11 (Figure 3.23) and improves the throughput by up to 40% (Figure 3.24). The success ratio is boosted up to at least 78% (Figure 3.25). We note here that 802.11's throughput in Figure 3.26 is higher than the one in Figure 3.13. This is because when there is fading, whether node 0 and node 2 can carrier sense each other becomes probabilistic. Under our setting, node 0 and node 2 can carrier sense each other most of the time, thereby increasing the throughput of 802.11 and reducing the number of packet drops per second (Figure 3.23).

More interestingly, we evaluate SELECT's performance over random topologies. We randomly place 5 (or 10) distinct flows, each running a *back-logged* CBR traffic in a 600m by 600m area. Second, we evaluate SELECT's performance in a more complicated random topology setting. We place 15 (or 30) distinct flows in a 1000m by 1000m area. In such topologies, hidden terminals may exist. Furthermore, with 15 (or 30) flows, the contention level will be more variant and thus more difficult for a SELECT-enabled sender to predict packet delivery result. For each of the above settings, we randomly generate 50 topologies and compare the performance between SELECT and the IEEE 802.11 protocol.

We first define β as the carrier sense threshold normalized by the sensitivity of receiving a packet, i.e. $\beta = CSThresh\ /\ RxThresh$. We vary the carrier sense threshold and evaluate SELECT's performance. In particular, we set the carrier sense threshold so that the corresponding carrier sense range is 768m, 512m, 384m, and 256m[6], respectively. The corresponding β value is -21, -14, -9, -2 respectively. Now, for each of the 50 random topologies, we compare the packet delivery success ratio and the throughput of SELECT with those of 802.11.

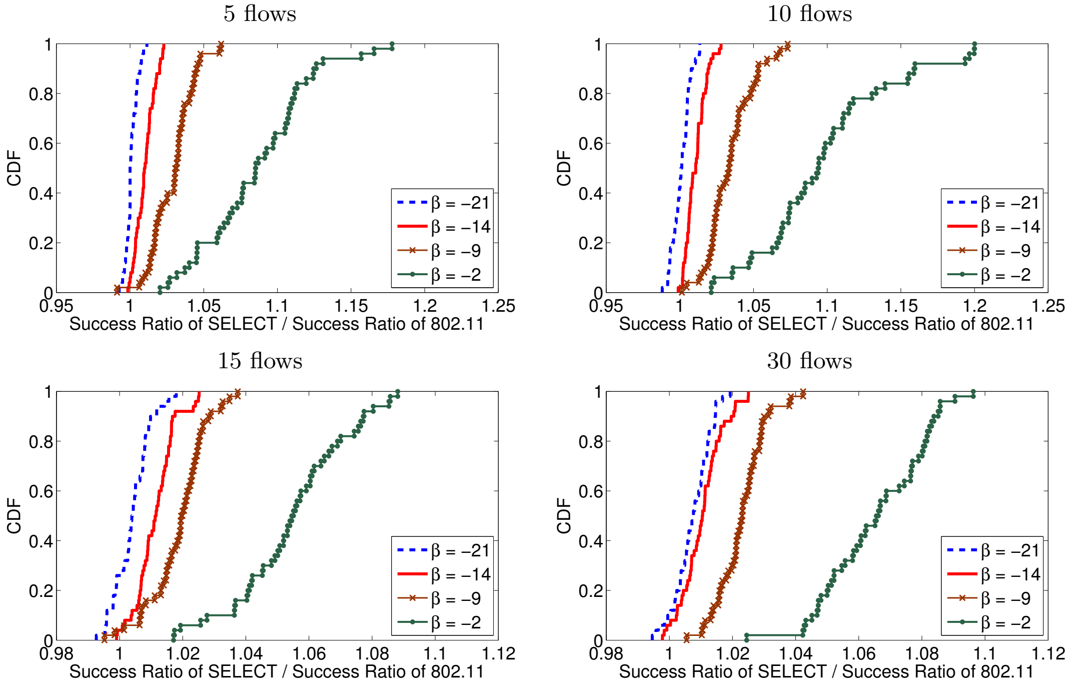


Figure 3.27: CDF of the ratio of packet delivery success ratio of SELECT to that of 802.11 over 50 random topologies at β = -21, -14, -9, -2

One naive approach to improve success ratio is to simply ensure at most one node can transmit packets at any given point of time. But this would inevitably result in unacceptably low throughput.

[6] Under this setting, nodes separated by the corresponding distances can carrier sense each other 95% of the time.

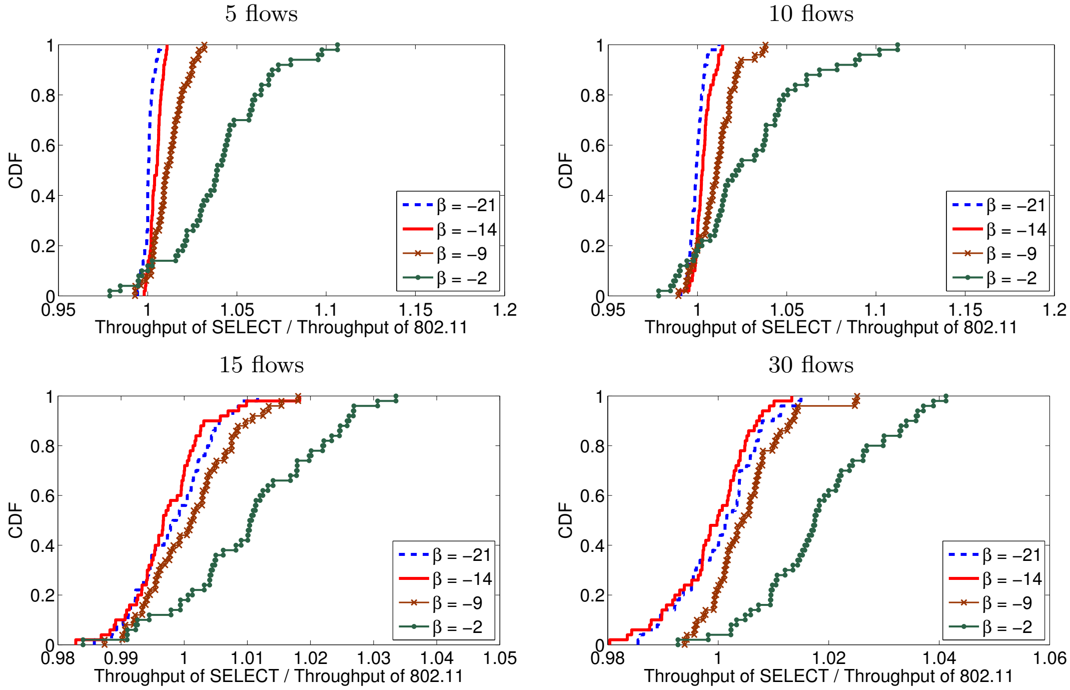


Figure 3.28: Throughput ratio of SELECT to 802.11 over 50 random topologies at β = -21, -14, -9, -2

One interesting question to ask would be: can we improve the success ratio while maintaining high throughput?

Figure 3.27 shows the CDF of the ratio of success ratio for SELECT over 802.11 for each of the 50 random topologies. When the carrier sensing range is large (e.g. β = -21 and -14), most of the interferers can be detected. The success ratio improvement is thus marginal. On the other hand, when hidden terminals exists (β = -9 and -2), the success ratio improves for the majority of the random topologies at various settings. In particular, success ratio is improved by up to 18%, 20%, 8.5%, 9.5% in random 5-flow, 10-flow, 15-flow, and 30-flow topologies, respectively. Note that when placing 15 flows (30 flows) in a larger area (1000m × 1000m), it is more likely for hidden nodes to exist. In such a case, SELECT performs much better at β = -21 and -14 when compared with 5 (10) random flows. In general, the more hidden terminals, the better is SELECT's improvement over 802.11.

Figure 3.28 shows the CDF of throughput ratio of SELECT to 802.11 (throughput of SELECT / throughput of 802.11) for each of the 50 random topologies over 5, 10, 15, and 30 random flows. When placing 5 (10) random flows in 600m×600m area, SELECT outperforms 802.11 by at least 90% (80%) of the 50 topologies at all β values. For 15 (30) flows in 1000m × 1000m area, SELECT outperforms 802.11 at β = -2 where more hidden terminals exist. When β = -21 and -14, carrier sensing is able to detect those interferers and SELECT's throughput is comparable to that of 802.11. We note that there is not much throughput improvement for SELECT over 802.11 over 15 and 30 random flows. Since SELECT design is meant for *collision avoidance*, it does not always improve the throughput. In fact, comparing Figure 3.27 and Figure 3.28, we can see that SELECT favors improving success ratio than throughput.

3.4 Experimental Results

We have implemented SELECT in TinyOS with MICA2 motes running programmable CC1000 radio (see Section 3.1 for the introduction to our testbed.). It is implemented in around 300 lines of NesC codes, in addition to the existing radio stack of around 2000 lines of NesC codes.

We modified the radio stack to record the RSS measurement before a sender sends the RTS packet and update the success/failure of the RTS channel access attempts. In MICA2 RSS is coded into a 10-bit integer with totally 1024 possible RSS values. We divide this range into 52 buckets and the width of every bucket except the last one is 20 RSS indices. For each bucket we maintain a success counter and a failure counter. The counters are updated after the channel access attempt and the result is determined. Backoff timer pausing and resuming are implemented through the manipulation of the backoff timer counter that is decremented every millisecond.

There are serious constraints in programming MICA2's radio stack. First, MICA2 has only 4KB data memory shared by both global data and function stack. The storage of SELECT's mapping between RSS and channel access success ratio takes $52 \times (2 + 2) = 208$ bytes. Since docking a sensor mote on a programming board changes its communication range irregularly, we have to store experimental data in sensor mote and collect them afterwards through the radio. Each data sample regarding one RTS attempt (RSS, timestamp, success/failure) takes $2 + 4 + 1 = 7$ bytes, and 100 samples require 700 bytes, around 17% of the total memory. Another constraint is the cost of floating point operations. MICA2 does not have hardware-supported floating point operations. Software implementations of floating point operations are too slow and break the timing constraints of the MAC protocol. Without floating point support enforcing a time window on all records with asynchronous update (see Section 3.2.1) becomes challenging and further approximation might be necessary. We do not decrement the success/failure counters in our current experiments since the motes run for a short period of time in the experiments.

We present our experiment results in a network topology shown in Figure 3.29. The distance

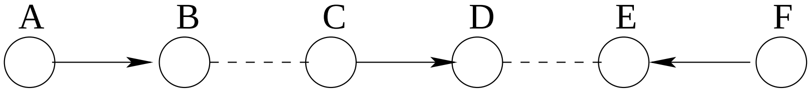


Figure 3.29: Experiment topology

Sender	RTS-CTS-DATA-ACK	w/ SELECT	Improvement
A	2.42	4.28	76.9%
C	3.98	5.08	27.6%
F	4.11	4.15	1.0%
total	10.51	13.51	28.5%
fairness	0.95	0.99	3.9%

Table 3.1: Experiment result: throughput (pkt/sec)

Sender	RTS-CTS-DATA-ACK	w/ SELECT	Improvement
A	0.36	0.63	75.0%
C	0.72	0.86	19.4%
F	0.75	0.81	8.0%
total	1.83	2.30	25.7%

Table 3.2: Experiment result: RTS success ratio

between adjacent motes is 60 cm. We configure the transmitting power so that each mote is in the transmission range of neighboring motes, but out of the carrier sense range of the motes further away. To start all the senders at the same time we broadcast a “start” packet using another mote, named commander mote, at the maximum power so that the command will be received by all senders at the same time. When a sender mote receives the “start” packet, it resets its timestamp and starts sending 100 packets to the receiver as fast as the channel allows. To measure the throughput when all the three flows are present, we used the data measured from the beginning to the time when the first mote finishes sending 100 packets.

The experiment results of throughput and RTS success ratio are shown in Table 3.1 and 3.2. Similar to our simulation results, while SELECT consistently improves the throughput and success ratio of all three flows, the flow with exposed receiver (A→B) benefits most from SELECT: 76.9% increase in throughput and 75.0% increase in RTS success ratio. SELECT also slightly improves throughput fairness by 3.9%, based on Jain’s fairness index [35].

3.5 Summary

Collision avoidance in wireless networks is complex and the correct perception of the channel availability is affected by a large number of factors. Many factors are dynamic in either temporal or spatial domains or both, and very difficult to model analytically or appraise at packet-level fine time granularity. Yet effective collision avoidance is a basic requirement for a wireless network. In this chapter we propose SELECT, an effective and efficient self-learning collision avoidance to tackle the long-haunting hidden/exposed receiver problem. Our mechanism involves zero communication overhead and easily integrates with 802.11 DCF. Our simulations show that SELECT can substantially increase the success ratio and improve the throughput, both with or without fading. We also found that SELECT favors improving success ratio over throughput when there is fading. Our research shows that non-intrusive, backward compatible enhancement to 802.11 DCF can help mitigate the hidden/exposed receiver problem.

Chapter 4

Rate-adaptive Framing

In this chapter, we present rate-adaptive framing (RAF), a joint channel rate and frame size control that improves the throughput for interfered transceivers. RAF seeks to characterize the observed interference pattern at the receiver, and uses the pattern to determine the desired channel rate and the frame size for throughput maximization. It is based on the assumption that wireless interference can be predictable in short-term, since usually the interfering wireless transmissions or background wireless traffic is not purely random even at fine time scale[1]. In more specific, for each channel rate a receiver divides a recent time window into a series of idle and busy intervals. Each idle interval is defined as a continuous time period during which the carrier sense value, as the sum of all interference and noise, is below certain threshold. See Figure 4.1 for an illustration. The threshold is derived[2] by the minimum required SINR (signal-to-interference-noise-ratio) for successful packet delivery at the channel rate and the signal strength of the last received message, assuming that the signal strength from the specific transmitter does not change significantly from the last frame. Given the set of idle intervals for a channel rate, the receiver computes the desired frame size that maximizes the achievable throughput. The receiver then compares the achievable throughput at different channel rates, and finally communicates the chosen configuration of channel rate and frame size to the transmitter in a few bits in the per-frame acknowledgement. The transmitter then applies the channel rate and frame size in the next frame transmission.

To leverage the full benefits of RAF's frame size control, aggregation of small packets from upper layers is necessary, in addition to fragmentation of large packets. Large packet fragmentation is already part of the 802.11 standard. Small packet aggregation in 802.11 was studied in [44] and relevant to the way RAF enforces its frame size control.

RAF's adaptation for channel rate and frame size resides at the receiver, while the channel access decision is enforced at the transmitter. RAF assumes a transmitter that is able to detect in real time

[1] See the literature of wireless traffic measurement, modeling and analysis, e.g., [60, 46, 50].

[2] Given a minimum required SINR value α and previously recorded signal strength β, the interference and noise can be at most $\frac{\beta}{\alpha}$.

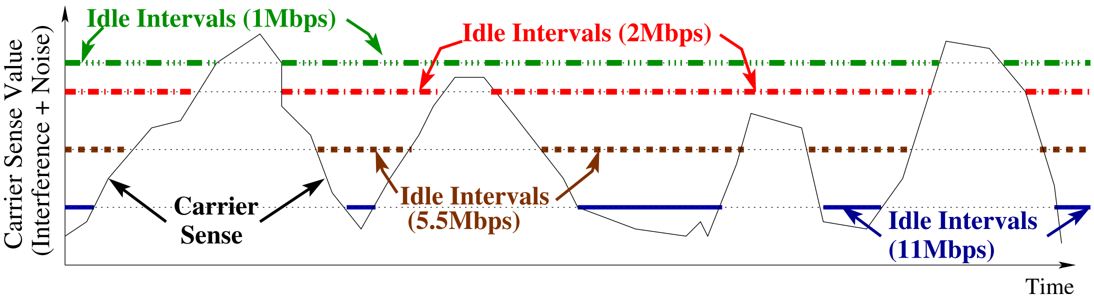


Figure 4.1: RAF receiver maintains multiple series of "idle" intervals. During each series the total interference and noise are less than certain threshold, which guarantees successful transmission and receiving at certain rate.

the beginning of each idle interval and start the transmission immediately. The proposed scheme SELECT in Chapter 3 satisfies the need. The final communications system consists of RAF receivers and SELECT transmitters. It is the RAF receiver that determines the *channel rate* and *frame size*, and the SELECT transmitter that determines *when* to transmit.

In this chapter we propose a joint rate and frame size control algorithm based on the observed patterns of the interference at the receiver. Through simulations, we present a comparison between our proposed RAF and ARF[40], RBAR[31], and OAR[59] to expose the potential gain. Our results show that RAF achieves about 20%, 65%, and 10% improvement in throughput for interfered transceivers compared with ARF, RBAR, and OAR, respectively. Our evaluation results also show that RAF adaptation converges very fast and adapts well to the system dynamics including node mobility and traffic variations.

The rest of this chapter is organized as follows. In Section 4.1 we present the details of our design and control algorithms. Section 4.2 presents the performance evaluation of RAF using extensive ns-2 simulations. We finally conclude in Section 4.3.

4.1 Rate-adaptive Framing

The design of RAF resides at the datalink layer. An RAF receiver maintains a recent history of fine-grained carrier sense from the physical layer, and computes the channel rate and frame size. The RAF receiver then communicates the desired configuration to the potential transmitters for next frame's transmission. In the following sections, we describe RAF in detail.

4.1.1 Fine-grained Carrier Sense

RAF is based on the assumption that wireless interference is predictable in short-term, since usually the interfering wireless transmissions or background wireless traffic is not purely random even at fine time scale [60, 46, 50]. RAF first maintains the history of the physical carrier sense in the forms of multiple series of idle and busy intervals. Each idle interval is defined as a continuous time period during which the carrier sense, as the sum of all interference and noise, is below certain threshold. A time period is busy if it is not idle. See Figure 4.1 for an illustration.

Note that in existing 802.11 radio a single carrier sense threshold is hardwired, and the physical layer signals whenever the carrier sense reading moves across the threshold[3]. See [37] for a brief summary of how carrier sense is implemented. RAF follows the carrier sense design logic, but requires that the physical layer be able to accept multiple configurable carrier sense thresholds (e.g., the four horizontal lines in Figure 4.1). Furthermore, the physical layer must signal in real time whenever the carrier sense reading moves across any of the thresholds.

With the input from the physical layer carrier sense RAF maintains each series of idle/busy intervals in a simple FIFO circular buffer (implemented as an array and a pointer pointing to the end of the series). Each buffer item simply records the length of an idle/busy interval. Note that RAF does not maintain the detailed carrier sense reading. RAF also implicitly controls the length of the maintained carrier sense history by simply bounding the size of the circular buffer, eliminating the need to maintain time-stamps. This simple control method automatically adapts to the dynamics of the channel status, since old records are overwritten quickly when the channel status is volatile.

Note that the thresholds in Figure 4.1 that RAF receiver submits to the physical layer may be different for different transmitters, depending on their signal strength. Even for the same transmitter, the thresholds may change over time, because its signal strength changes over time due to the dynamics in node mobility and channel fading. In RAF we use the transmitter's most recent signal strength as the reference, and calculate the carrier sense thresholds as the quotients of the transmitter's most recent signal strength and the SINR thresholds. These carrier sense thresholds are then passed to the physical layer.

[3] With necessary hysteresis to avoid thrashing.

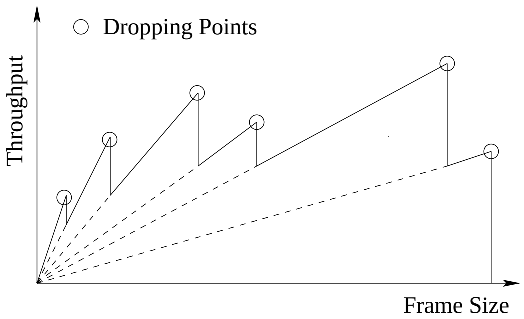


Figure 4.2: $f(s)$ versus frame size s.

4.1.2 Channel Rate and Frame Size Selection

Given the idle intervals $idle_k$, the receiver calculates the desired channel rate and frame size by maximizing the following throughput function $f(s)$:

$$\max_{i \in R, s \in S} f(s) \tag{4.1}$$

$$f(s) = s \sum_{k=1}^{n_i} \left\lfloor \frac{\max(idle_k - \text{InitBackoff}, 0)}{s/i + OH + C} \right\rfloor . \tag{4.2}$$

where R is the set of channel rates, S is the set of frame sizes, InitBackoff is the time it takes for the sender to finish its backoff and start transmitting the packet in the idle interval, n_i is the number of idle intervals for rate i maintained at the receiver[4], OH is the per-frame PHY/MAC overhead, and C is the inter frame overhead, i.e. C=(CW$_{min}$/2 · aSlotTime + SIFS + ACK + DIFS). To determine the initial backoff InitBackoff at the beginning of each idle interval, we assume the sender is able to detect when the channel becomes idle and resumes its backoff timer immediately. That is, the frame delivery fails only near the end of an idle interval, since the sender cannot predict when the idle interval will end before it starts transmitting the last frame. Since SELECT's design is for sender to predict the channel status at the receiver, we combine SELECT to work with RAF as the final proposed system. Assuming such a transmitter the expected initial backoff is InitBackoff = CW$_{min}$ · aSlotTime, since the contention window size always doubles to 2 CW$_{min}$ after the last frame is lost at the end of the previous idle interval. Our analysis below is based on this initial backoff setting.

One naive search for the maximum throughput is to enumerate all possible frame sizes, e.g., from 1 to 1500 bytes, resulting in a running time of $O(|R|\,|S|\,n_{max})$, where $n_{max} = \max_i n_i$, for $i \in R$. In the rest of this section, we describe two methods to reduce the computation overhead.

[4] For example, in Figure 4.1, $n_1 = 3$, $n_2 = 4$, $n_{5.5} = 5$, $n_{11} = 5$

Our first method is based on the observation that $f(s)$ is a saw-tooth shaped function, with each segment a linear function extending to the origin. See Figure 4.2 for an illustration. Furthermore, the slope of the linear function, i.e., the sum of the floor functions in Eqn. 4.2, monotonically decreases as the frame size increases. Since the slopes are all positive, the maximum throughput within a segment must appear at the right end, followed by a sudden drop to the next linear segment with a lower slope. We therefore can safely skip all intermediate frame sizes in our search for the one leading to the maximum throughput. In specific, we start from s_{min}, which is the minimum packet size, calculate the corresponding throughput $f(s_{min})$ and the next dropping point $s_{nextDrop}$, proceed to the next frame size which is equal to either the current frame size plus default increment or the $s_{nextDrop}$, whichever is larger.

To calculate $s_{nextDrop}$ given s, we take advantage of the fact that when $f(s)$ increases linearly with frame size s within a segment, the dropping point must occur when frame size s is just enough to fit one or multiple frame transmission time $(\frac{s}{i} + OH + C)$ into some idle interval $idle_{nextDrop}$, with no residual left. In fact, $idle_{nextDrop}$ is the idle interval with minimum normalized residual, the residual normalized by the number of frames already fit in the interval. Let the normalized residual for idle interval $idle_k$ and frame size s be $\xi(idle_k, s)$, $idle_{nextDrop}$ can be expressed as follows:

$$\begin{aligned} idle_{nextDrop} &= \text{argmin}_{idle_k \in H_i} \xi(idle_k, s) & (4.3) \\ \xi(idle_k, s) &= \frac{D - \lfloor D \rfloor}{\lfloor D \rfloor} & (4.4) \\ \text{where } D &= \frac{\max(idle_k - \text{CW}_{min} \cdot \text{aSlotTime}, 0)}{\frac{s}{i} + OH + C} \end{aligned}$$

Note that the $idle_{nextDrop}$ is readily available when computing $f(s)$. After identifying $idle_{nextDrop}$ the frame size at the next dropping point $s_{nextDrop}$ can be calculated by:

$$s_{nextDrop} = \left(\frac{adjIdle_{nextDrop}}{E} - OH - C \right) \cdot i \qquad (4.5)$$

$$adjIdle_{nextDrop} = \max(idle_{nextDrop} - \text{CW}_{min} \cdot \text{aSlotTime}, 0)$$

$$E = \begin{cases} \left\lfloor \dfrac{adjIdle_{nextDrop}}{\frac{s}{i} + OH + C} \right\rfloor, \text{ if } \left\lfloor \frac{adjIdle_{nextDrop}}{\frac{s}{i}+OH+C} \right\rfloor \neq \frac{adjIdle_{nextDrop}}{\frac{s}{i}+OH+C} \\ \\ \dfrac{adjIdle_{nextDrop}}{\frac{s}{i} + OH + C} - 1, \text{ if } \left\lfloor \frac{adjIdle_{nextDrop}}{\frac{s}{i}+OH+C} \right\rfloor = \frac{adjIdle_{nextDrop}}{\frac{s}{i}+OH+C} \end{cases}$$

The pseudo-code for the above computation are shown in line 8-12 and line 24-32 in Figure 4.3.

Our second method is based on the observation in our simulations that the number of idle intervals within certain time window, i.e., the time window (MAX_WINDOW_SIZE) that bounds the history of idle/busy intervals, could be very small, especially for lower channel rates. It turns out that when there is only one single idle interval, we can find the desired frame size in constant time without going through all the dropping points. We start from the following proposition:

Proposition 1 *With one idle interval ($n_i = 1$), $f(s)$ at the dropping points are strictly increasing.*

Proof 2 *Let m and $m-1$, $m > 0$, be the slopes of the two consecutive dropping points. Let s_1 and s_2 be the corresponding dropping-point frame sizes, $s_1 < s_2$. With some algebraic manipulation, it is easy to see that $s_1 = \frac{(idle - \mathit{InitBackoff} - m(OH+C))\ i}{m}$ and $s_2 = \frac{(idle - \mathit{InitBackoff} - (m-1)(OH+C))\ i}{m-1}$. Also, we have:*

$$\begin{array}{rcl} (m-1)\ (OH+C)\ i & < & m\ (OH+C)\ i \\ \Rightarrow\ (idle - m(OH+C))\ i & < & (idle - (m-1)(OH+C))\ i \\ \Rightarrow\ m\dfrac{(idle - m(OH+C))\ i}{m} & < & (m-1)\dfrac{(idle-(m-1)(OH+C))\ i}{m-1} \\ \Rightarrow\ m\ s_1 & < & (m-1)\ s_2 \\ \Rightarrow\ f(s_1) & < & f(s_2) \end{array}$$

Given the above proposition, we can start the search for the desired frame size from s_{max}, which is defined as the maximum packet size, and move backward for finding the last dropping point $s_{lastDrop}$[5].

$$s_{lastDrop} = \left(\frac{adjIdle}{F} - OH - C\right) \cdot i \tag{4.6}$$

$$adjIdle = \max(idle - \mathrm{CW}_{min} \cdot \mathrm{aSlotTime}, 0)$$

$$F = \begin{cases} \left\lceil \dfrac{adjIdle}{\frac{s_{max}}{i} + OH + C} \right\rceil, & \text{if } \left\lceil \frac{adjIdle}{\frac{s_{max}}{i}+OH+C} \right\rceil \neq \frac{adjIdle}{\frac{s_{max}}{i}+OH+C} \\ \\ \dfrac{adjIdle}{\frac{s_{max}}{i} + OH + C} + 1, & \text{if } \left\lceil \frac{adjIdle}{\frac{s_{max}}{i}+OH+C} \right\rceil = \frac{adjIdle}{\frac{s_{max}}{i}+OH+C} \end{cases}$$

Line 14 to 22 in Figure 4.3 show the pseudo-code for the above computation. In our simulated scenarios the two methods reduce the computation overhead by at least 75%.

Note that we do not calculate the throughput based on the expected length of the idle intervals because the variation of the idle intervals may become so large that the receiver may always choose

[5] When searching for the last dropping point, we do not include s_{max} in the search space.

```
**getRateAndPktSize**($H_1, \ldots, H_{|R|}$)
$H_i$, $i \in R$: the idle busy history
$R$: the set of available channel rates
$S$: the set of available pkt sizes
Let $windowSize$ be the smallest history length among $H_1, \ldots, H_{|R|}$
**if** $windowSize >$ MAX_WINDOW_SIZE **then**
    $windowSize \leftarrow$ MAX_WINDOW_SIZE {We do not count the data older than $windowSize$}
$maxThr \leftarrow -\infty$
**for all** $i \in R$ **do**
    cut $H_i$ to length $windowSize$
    **if** $H_i$ has more than one idle interval **then**
        **for** $s \leftarrow s_{min}, s \leq s_{max}$ **do**
            $(thr, s_{nextDrop}) =$ **getThr**$(s, H_i)$
            **if** $maxThr < thr$ **then**
                $maxThr \leftarrow thr$   $bestRate \leftarrow i$   $bestPktSz \leftarrow s$
            $s \leftarrow \max(s_{nextDrop}, s + default\_increment)$
    **else**
        compte $s_{lastDrop}$ from Eq. 4.6 with input $idle, s_{max}, i$
        Let $thr1$ be computed from Eq. 4.2 with input $idle, s_{lastDrop}, i$
        Let $thr2$ be computed from Eq. 4.2 with input $idle, s_{max}, i$
        **if** $maxThr < \max(thr1, thr2)$ **then**
            $maxThr \leftarrow \max(thr1, thr2)$   $bestRate \leftarrow i$
            **if** $thr1 > thr2$ **then**
                $bestPktSz \leftarrow s_{lastDrop}$
            **else**
                $bestPktSz \leftarrow s_{max}$
return $bestRate$, $bestPktSz$
**getThr**$(s, H_i)$
$s$: packet size
$H_i$: the idle busy history for rate $i$
$minResidual \leftarrow \infty$
$thr \leftarrow 0$
**for all** $idle_k \in H_i$ **do**
    $thr \leftarrow thr + s \left\lfloor \frac{\max(idle_k - \text{InitBackoff}, 0)}{s/i + OH + C} \right\rfloor$
    Let $normResidual$ be computed from from Eq. 4.4 with input $idle_k, s, i$
    **if** $normResidual < minResidual$ **then**
        $minResidual \leftarrow normResidual$   $dropIndex \leftarrow k$
Let $s_{nextDrop}$ be computed from Eq. 4.5 with input $idle_{dropIndex}, s, i$
return $thr, s_{nextDrop}$
```

Figure 4.3: Rate and packet size calculation

frame sizes far from the desired values. Note also that since the idle busy history is maintained in a bounded buffer, obsolete data will be removed quickly when the environment becomes more dynamic.

4.1.3 Frame Aggregation and Fragmentation

To achieve the maximum throughput, an RAF transmitter may aggregate small packets or fragment large packets from upper layers to fit into the designated frame size. Note that fragmentation is already part of the 802.11 standard and applies here. To enable aggregation, an RAF transmitter precedes every aggregated upper layer packet in the frame payload with two bytes. These two bytes specify the length of the following aggregated packet, as shown in Figure 4.4. When an RAF receiver receives a frame, it reads the first two bytes of the frame payload and extracts the first packet. If after extracting the first packet, the end of the frame is not yet reached, the RAF receiver interprets the next two bytes as the length of the next aggregated packet. It then extracts the next packet accordingly. This process continues until all aggregated packets in the frame are extracted.

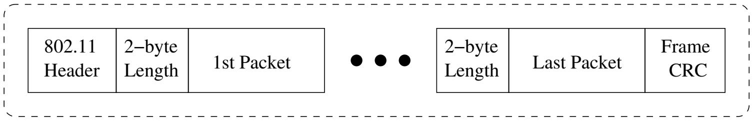


Figure 4.4: RAF frame format

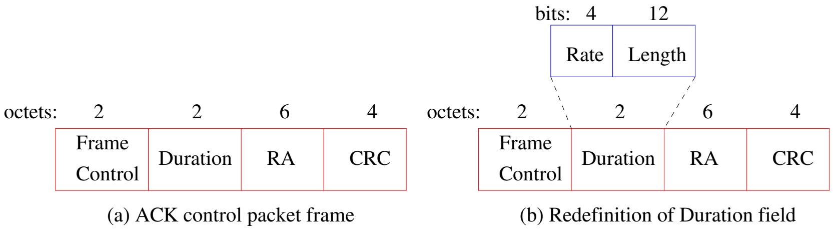
Note that the above aggregation works fine even if the last fragment of the previous packet has to be combined in a single frame with the following packets from upper layer. The 802.11 header for the frame will contain the necessary fragmentation information for the fragment, located at the beginning of the frame, to be de-fragmented at the receiver. Other packets in the same frame payload can be extracted following the same procedure as described above[6]. To avoid further complexity we do not include more than one fragment in a single frame, since there is only one 802.11 header with room for the fragmentation information of one fragment only.

4.1.4 Configuration Update

An RAF receiver piggybacks the chosen channel rate and frame size in the per-frame 802.11 ACK message, and an RAF transmitter applies the updated configuration to the transmission of the next frame. We start by introducing the 802.11 ACK frame. Figure 4.5(a) shows the format of an

[6] In fact, there could be at most two fragments in one frame, and they must appear at the beginning and the end of the frame payload respectively.

802.11 ACK message. The Duration field in original 802.11 ACK frame is set to zero unless the More Fragments bit in the Frame Control is set to 1, in which case the Duration field contains the remaining time in microseconds to finish transmitting the entire fragmented packet.



(a) ACK control packet frame

(b) Redefinition of Duration field

Figure 4.5: Original and redefined 802.11 ACK format

RAF redefines the 16-bit Duration field to carry the updated channel rate and frame size information. It divides the 16-bit Duration field into two subfields, one 4-bit Channel Rate subfield and the other 12-bit Frame Length subfield, as shown in Figure 4.5(b). All the other nodes overhearing the ACK frame will disregard the information in the duration field when More Fragments bit in the Frame Control is set to 0. When the bit is set to 1, the duration of the next fragment transmission can be easily calculated given the channel rate and frame size encoded in the frame header.

4.2 Performance Evaluation

We implement RAF in *ns-2* simulator version *2.29*. For comparison we implement ARF which works as follows. If there are two continuous packet losses, the sender decreases the channel rate to the next lower level. If there are ten continuous successful packet transmissions, the sender increases the channel rate to the next higher level. We also implement RBAR [31], and OAR [59]. The 802.11 physical layer in ns-2.29 is overly simplified. A node receives a packet only when the propagated signal from the sender is greater than the receive threshold. However, the impact of the signal with strength less than the carrier sense threshold is completely ignored—no matter how many those signals are. We replace this part of 802.11 functions with the ones developed in [32], so that all signals are taken into account at the receiver, and the combined SINR is used to determine if an incoming signal can interfere or be received/captured. We adopt the capture threshold as listed in [9] so that the SINR has to be greater than 3, 4, 8, and 12dB in order for the receiver to respectively receive the frame at 1, 2, 5.5, and 11Mbps. We use two-ray ground signal propagation model, and the transmitting power is set so that the communication range, which is defined to be the distance between two nodes within which the transmission between the two nodes is above the sensitivity level, is 115m, and the carrier sensing range is set to 200m. We use 2Mbps basic rate and 11Mbps channel rate based on IEEE 802.11b. Each simulation runs for 45 seconds unless otherwise specified.

4.2.1 Typical Hidden Terminal Topologies

We first evaluate the throughput of the simple hidden/exposed terminal topology shown in Figure 4.6. In this case, sender 0 and 2 are outside the carrier sensing range of each other. Client 3 is an exposed receiver since it is placed close to the interference source client 0. Notice that in this configuration flow 0→1 will always succeed in the channel attempt because its receiver (node 1) is not interfered by flow 2→3. We therefore vary the offered load (CBR/UDP rate) of flow 0→1 serving as varied interference level, while keeping interfered sender 2 always backlogged (with a 11Mbps CBR). We compare the throughput of RAF with ARF, RBAR, and OAR.

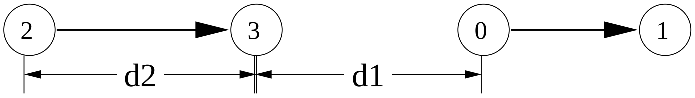
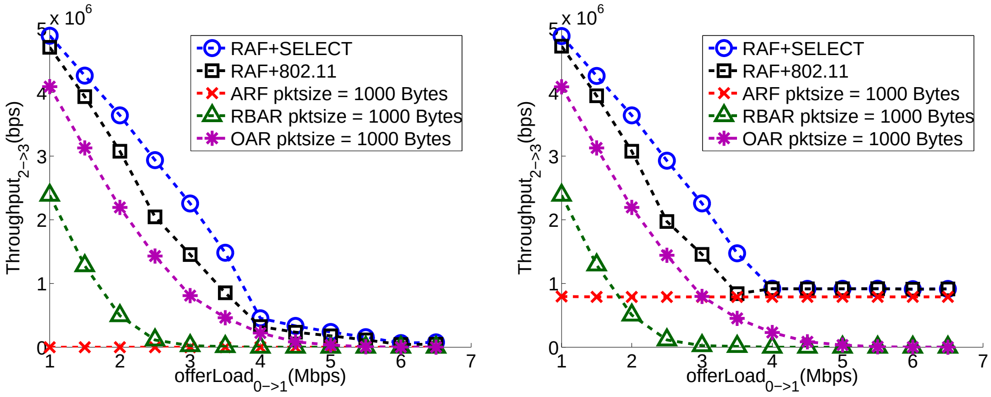


Figure 4.6: Interference from hidden transmitter

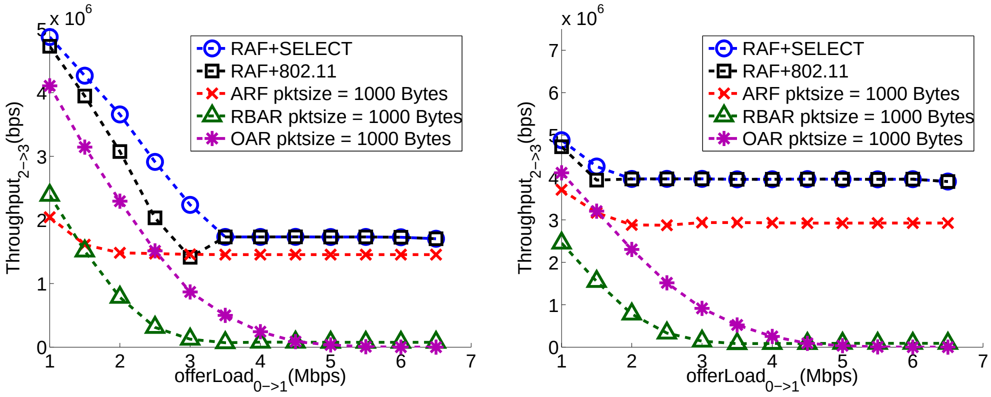
Varied Interference Levels

When the distance (d_1) between nodes 3 and 0 is equal to 120 m, the transmission of flow 2→3 can not be reliable even if the lowest 1Mbps channel rate is applied at node 2 (due to the interference from flow 0→1's transmission). As shown in Figure 4.7(a), ARF achieves almost zero throughput under this setting while RAF, RBAR, and OAR achieves significantly higher throughput. This is because under such interference levels, ARF's convergence to lower channel rate prolongs the transmission time, which in turn further increases the chances of packet collisions. In the worst case, ARF gets stuck at the lowest rate due to aggravated interference — leading to the starvation of flow 2→3. RAF, RBAR, and OAR, on the other hand, stay at higher rates with much better channel bandwidth sharing. This is because RAF, RBAR, and OAR choose the highest channel rate so that packet transmissions can be finished quickly when there is no interference. Nevertheless, RAF outperforms RBAR and OAR due to two reasons. First, RAF does not endure the RTS/CTS overhead. Since even the RTS message itself will be interfered at node 3, the efficacy of RTS/CTS handshake for rate control vanishes. Second, RBAR and OAR were not designed for rate control in interfered networks. The SINR of a received RTS message cannot figure the future interference in the rate adaptation. We also plot the throughput between an RAF receiver and a SELECT transmitter, and the throughput between an RAF receiver and a regular 802.11 transmitter. Note that both SELECT and regular 802.11 transmitters enforce their RAF receivers' control on the channel rate and frame size. As shown in Figure 4.7(a), the throughput between an RAF receiver and a regular 802.11 transmitter decreases only slightly when compared with the throughput with a SELECT transmitter, and remains consistently higher than the throughput of ARF, RBAR, and OAR.



(a) (b)

Figure 4.7: Throughput comparison between RAF and various rate control schemes for (a) (d_1 = 120m). (b) (d_1 = 135m).



(a) (b)

Figure 4.8: Throughput comparison between RAF and various rate control schemes for (a) (d_1 = 160m). (b) (d_1 = 180m).

We then increase the distance between the two flows to reduce the level of interference at node 3. Figures 4.7(b), 4.8(a), 4.8(b) again show the throughput of flow 2→3 with the varied offered load at flow 0→1 as the distance (d_1) between node 3 and 0 increases to 135m, 160m, and 180m respectively. Under these three distances the maximum channel rate at node 3 that can support reliable data

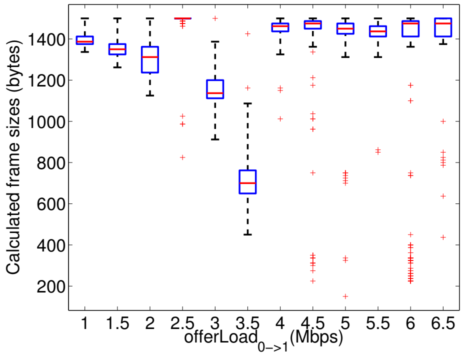
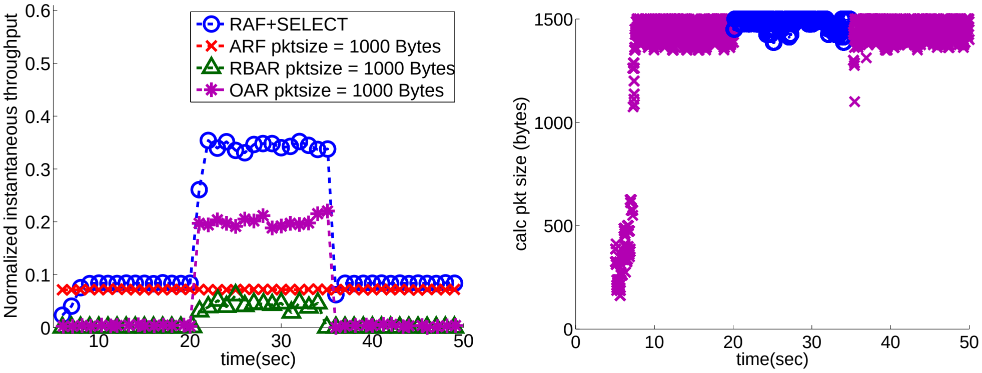


Figure 4.9: Computed frame sizes for different offerLoad$_{0\to1}$ (d_1 = 135m). Computed rate is 11Mbps for offered load < 4Mbps and 1Mbps otherwise.

transmission becomes 1Mbps, 2Mbps, and 5.5Mbps respectively. ARF is able to choose the lower channel rates and maintain the throughput at a certain level (0.8Mbps, 1.5Mbps, and 3Mbps for d_1 equals 135m, 160m, and 180m respectively) when the offered load of flow 0→1 is larger than 3Mbps (high interference). However, ARF's rate adaptation algorithm still chooses low channel rates even after we reduce the interference level by decreasing the offered load of flow 0→1. This is because after ARF switches to a higher channel rate, it is easy for two consecutive packet losses to occur, in which case ARF switches back to lower channel rates immediately. RBAR and OAR, on the other hand, are able to choose higher channel rates when the interference level is low, but failed to choose the rates that can support reliable transmissions when the interference level is high. This is because the receiver of RBAR and OAR includes the rate to be used by the sender through RTS/CTS frames. When the receiver is interfered by exposed terminals, it cannot even reply the CTS to the sender since the RSS at the receiver is larger than the carrier sense threshold. Figure 4.9 illustrates the adaptation of the calculated frame size for flow 2→3 when d_1=135m, in terms of the maximum, minimum, median, upper and lower quartiles. As we can see from the figure, when the offered load of flow 0→1 is less than 3Mbps (or the interference is low), the calculated rate for flow 2→3 is 11Mpbs and the corresponding computed frame size is large. As the offered load of flow 0→1 increases, the computed frame size decreases while the computed channel rate remains at 11Mbps. When the offered load of flow 0→1 goes beyond 3.5Mbps, or the interference level is high, the computed channel rate decreases to 1Mbps while the computed frame size increases to the maximum. Note that there are several red-crossed data points in Figure 4.9 identified as outliers[7] by the box plot function, which calculates and plots the minimum, maximum, median, 25th percentile, and 75th percentile of a data set. These data points are calculated at the start of the simulation,

[7] Data points are drawn as outliers if they are larger than $q_3 + 3 \cdot (q_3 - q_1)$ or smaller than $q_1 - 3 \cdot (q_3 - q_1)$, where q_1 and q_3 are the 25th percentile and 75th percentile, respectively.



(a) (b)

Figure 4.10: (a) Throughput$_{2\to3}$ comparison between RAF and various rate control schemes when traffic pattern of flow$_{0\to1}$changes.(b) Channel rate and frame size calculated at node 3 when traffic pattern of flow 0→1 changes (∘: 11Mbps, ×: 1Mbps). The figures will be more clear if it is color printed.

when the durations of the idle intervals are short. As a result, the calculated frame sizes are small in order to fit into the idle intervals.

Dynamic Interference Pattern

We then study how RAF adapts when the traffic pattern of the interfering flow 0→1 changes. Since our proposed system combines RAF receiver with SELECT sender, in what follows, we only show the performance result of RAF with SELECT. By fixing d_1 to 135 m, we initialize the offered load of flow 0→1 (offerLoad$_{0\to1}$) to 5Mbps. At time 20 second, offerLoad$_{0\to1}$ changes to 2Mbps, then changes back to 5Mbps at time 35 second. Node 2 remains backlogged throughout the simulations. Figure 4.10(a) shows the instantaneous throughput over 1 second period normalized to 11Mbps. From the figure we can see that RAF adjusts the channel rate and frame size within 2 seconds after the interference level goes down (where offerLoad$_{0\to1}$ changes from 5Mbps to 2Mbps), and achieves a 4-fold throughput improvement. In contrast, ARF's throughput stays at the same low level, while RBAR and OAR achieve the throughput up to 14% and 57% of that of RAF, respectively. Figure 4.10(b) shows the channel rate and frame size adaptation at flow 2→3 throughout the simulation. In specific, node 2 switches to higher channel rate (11Mbps) when the interference level becomes mild, and switches to 1Mb when the interference becomes strong. Note that the computed frame sizes quickly stabilize within 2∼3 seconds after the traffic change. From Figures 4.10(a) and 4.10(b), it is easy to see that RAF's channel rate and frame size adaptation is responsive to the traffic dynamics and achieves consistent throughput improvement compared with other alternatives.

The previous evaluation only shows RAF's adaptability for regular periodic interference with an

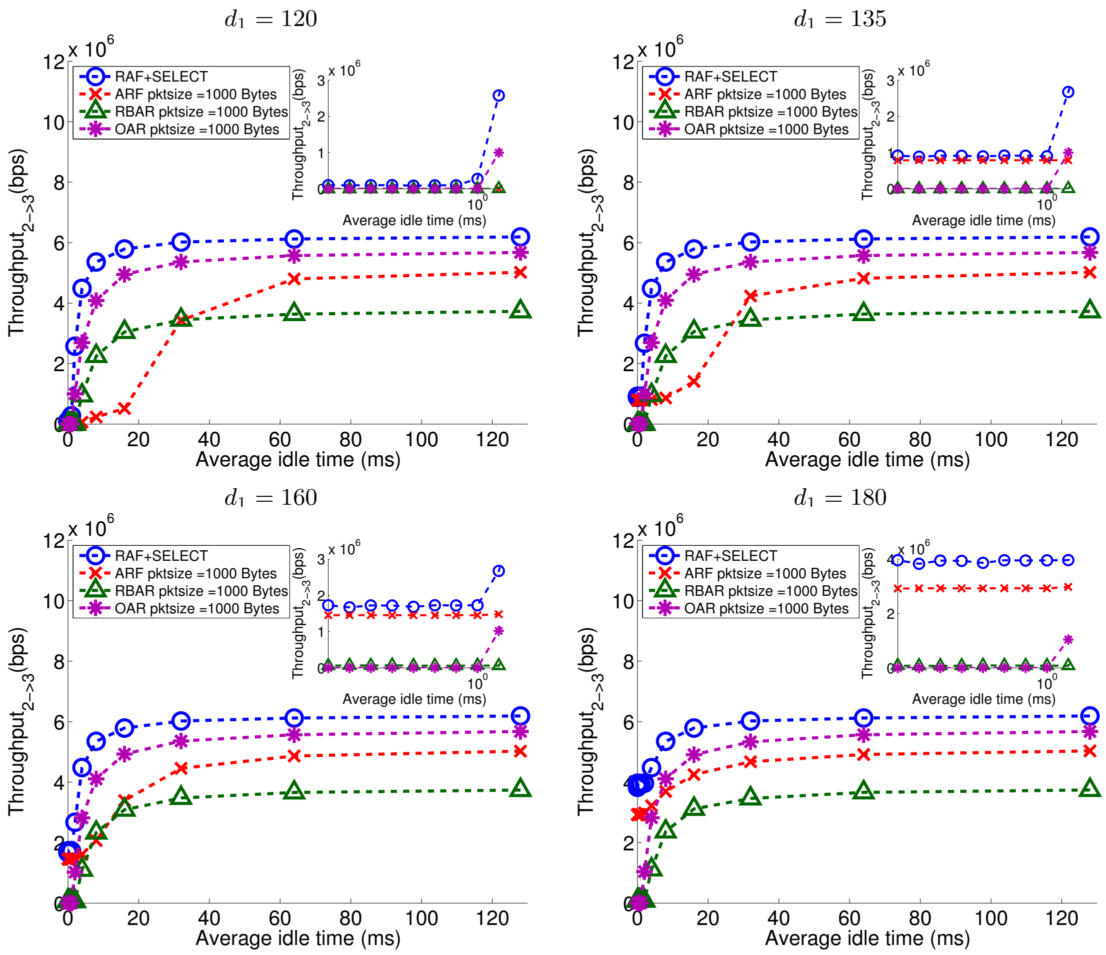


Figure 4.11: Throughput comparison between RAF and various rate control schemes for Poisson interference pattern.

instantaneous change. We next study how RAF responds to more dynamic interference patterns in general. We change the traffic from node 0→1 from CBR to VBR traffic with the packet arrival to be Poisson distribution. We vary the average idle time between each packet arrival from 8μs to 32ms and plot the corresponding throughput for flow 2→3. Figure 4.11 shows the throughput of flow 2→3 for d_1 at 120m, 135m, 160m and 180m, respectively. Despite the Poisson interference pattern, the interference level become more intense as the average idle time reduces. As a result, the throughput of flow 2→3 decreases as the average idle time reduces. We found that the relative performance between RAF, ARF, RBAR, and OAR are still comparable to the one in Figure 4.7 and 4.8. Also note that as the average idle time increases, ARF eventually exploits most of the time sending packets at high rates, thereby achieving comparable performance with RAF, RBAR, and OAR.

We then change the dynamic interference pattern at flow 0→1 to be of binary exponential on-off,

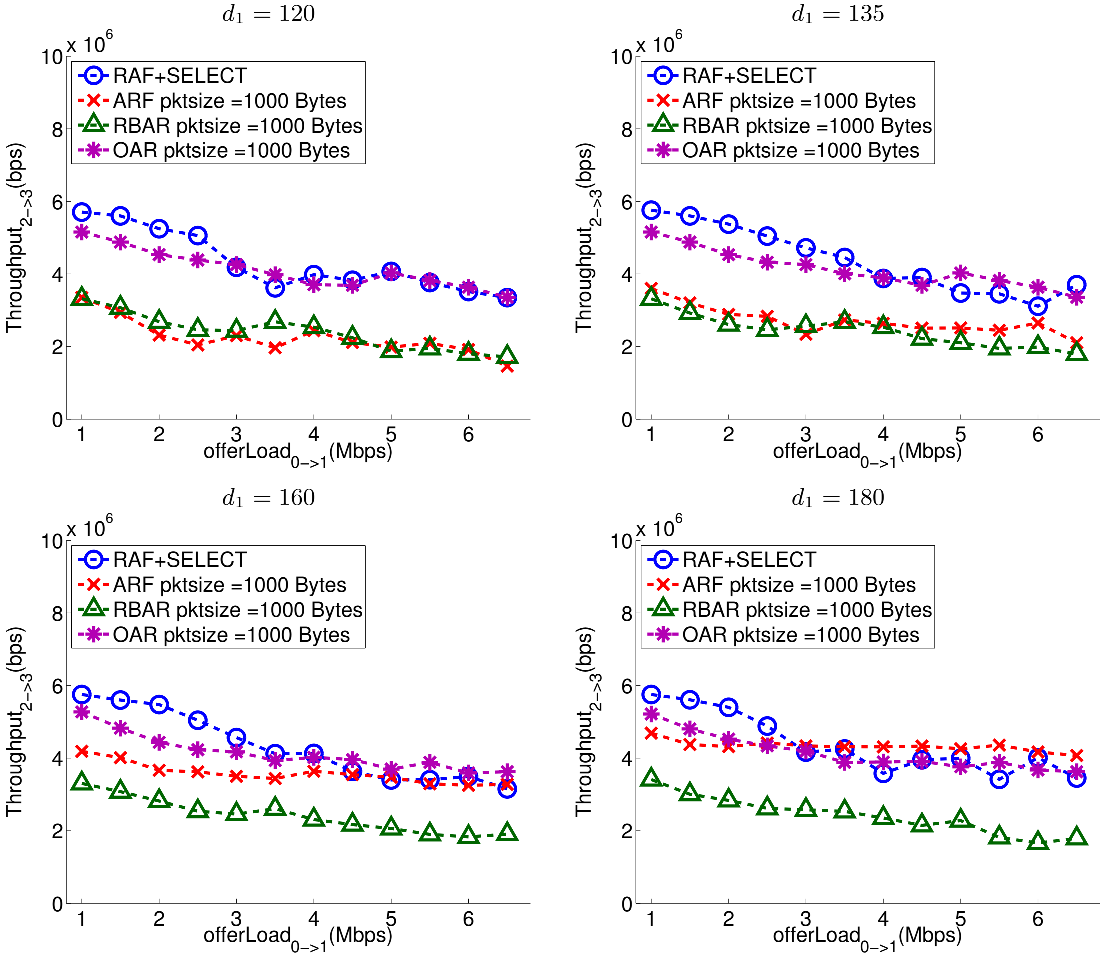


Figure 4.12: Throughput comparison between RAF and various rate control schemes for Binary Exponential On-Off interference pattern.

in which the sender stays in two states: on and off. In the “on“ state, the sender sends packets at a certain CBR rate; while in the “off“ state, no packet is transmitted. The duration in “on“ and “off“ states are both exponentially distributed. Intuitively, the longer the sender stays in “on“ state and the shorter in “off“ state, the more similar the performance is to the periodic CBR traffic. To evaluate RAF’s performance in a more dynamic scenario, we therefore set the average idle duration for the “on“ state to be 100 ms and average idle duration for the “off“ state to be 200ms and measure the throughput of flow 2→3 by varying the sending rate of flow 0→1 in the “on“ state. Figure 4.12 shows the throughput of flow 2→3 for different d_1 values. Different from the CBR and Poisson packet arrival, both RBAR and OAR maintains 2Mbps throughput even though the interference level is increased to 6Mbps for various d_1 values. This is because node 2 can send packets to node 3 successfully while flow 0→1 is in the “off“ state. Nevertheless, RAF typically outperforms ARF, RBAR, and OAR most of the time at all interference levels. Note that when the interference is not severe (d_1=180), ARF achieves the best throughput for the interference levels caused by the offered loads at flow 0→1 that are higher than 3.5Mbps. The reason is because the interference source flow 0→1 stays in the off state long enough to favor ARF for it to stabilize at the highest channel rate, despite the packet loss history.

Node Mobility

We further study RAF's adaptability when the interference level at node 3 continuously changes. The simulation runs from 5 sec to 130 sec. Node 0 is initially placed 120 m away from node 3. At time 10 sec, it starts moving back and forth from node 3 up to 200 m at 2 m/s, as shown in Figure 4.13. Figure 4.14(a) again shows the one-sec normalized instantaneous throughput for flow 2→3 when the offered load of flow 0→1 is 3Mbps. Indeed, ARF shows its ability to adapt when interference level changes. However, RAF outperforms ARF during all the time periods. Note that during interval 5∼15sec and 85∼94sec, ARF's throughput drops to zero. This is due to the strong interference resulting from the short distance between node 0 and 3, and ARF's improper selection of low channel rates which aggravates the effect of hidden terminal interferences. Also note that RBAR and OAR under this high load do not have much chance to correctly select the channel rate, resulting in the worst throughput. RAF achieves similar improvement over ARF, RBAR, and OAR for all other interference levels. We note here that RBAR and OAR do not perform better than ARF. This is because both RBAR and OAR require RTS/CTS for receiver to feedback the channel rate to the sender. In this topology, however, receiver 3 cannot send back CTS since the channel at node 3 is sensed busy when node 0 is transmitting data to node 1. Aa a result, when node 3 is interfered, it cannot reply CTS to node 2 for rate adaptation. On the other hand, when node 3 is not interfered, it replied CTS suggesting the highest channel rate, which is susceptible to future interference.

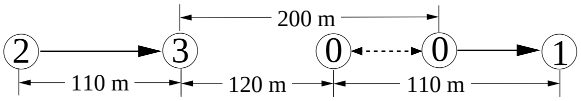
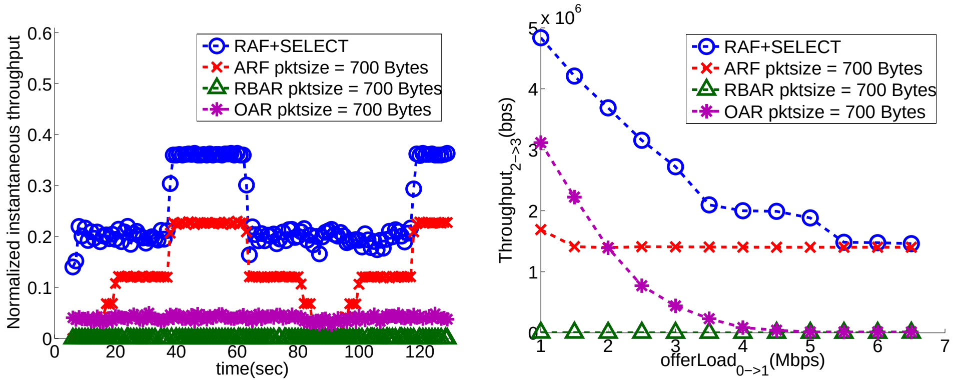


Figure 4.13: Topology where node 0 moves in an oscillatory manner.

Finally, Figure 4.14(b) shows the throughput when the offered load of flow 0→1 is varied from 1Mbps to 6.5Mbps. RAF again, performs better than ARF, RBAR, and OAR for all the offered loads and improves by up to 298% for ARF, and 70% for OAR.



(a) (b)

Figure 4.14: (a) Instantaneous throughput of flow$_{2\to3}$ over 1 sec period when node 0 is moving toward/away from node 1. (b) Throughput of flow$_{2\to3}$ when node 0 is moving toward/away from node 1, offerLoad$_{0\to1}$ ranges from 1Mb to 6.5Mb. The figures will be more clear if they are color printed.

4.2.2 Random Topologies

We also studied RAF's performance in large random topologies. Specifically, we randomly place 10 flows with distinct senders and receivers (20 nodes in total) in a 1000m by 1000m area. The process is, we randomly choose 10 senders, for each sender, we randomly choose a receiver that is not chosen yet and that is within the communication range of the sender. We generate 30 random topologies and the simulation runs for 20 seconds for each experiment. Our experiments can be categorized into 6 groups. The first 3 groups (each group runs 30 random topologies) are static random topologies with offered load of each flow set to 2Mbps, 4Mbps, and 6Mbps, respectively. For the other 3 groups, we enable node mobility and move each receiver back and forth, between 30m and 110m, from the sender at 2m/s. Note that under the same moving speed this moving pattern causes the highest signal strength variation compared with the popular random way point mobility model.

As shown in Figure 4.15-4.17, we again use box plot to show the aggregate throughput improvement[8] of RAF over ARF, RBAR, and OAR for the 30 topologies in each of the 6 groups. We can easily see the maximum, minimum, median, upper quartile, and lower quartile[9] of RAF's improvements among these 6 groups[10]. Indeed, RAF outperforms ARF, RBAR, and OAR for most of the 30 topologies among all the three different rates for both static and mobile topologies. Since the higher the rate, the more intense is the interference level, RBAR and OAR can not capture the ensuing interference that will happen. As a result, the higher the offered load (or the higher the

[8] Throughput improvement is defined as (RAF's throughput / other algorithm's throughput) -1.

[9] The maximum value is represented by the top of the upper whisker, minimum value is represented by the bottom of the lower whisker, median is shown as the line in the box, upper quartile (75th percentile) is represented by the top line of the box, lower quartile (25th percentile) is represented by the bottom of the box.

[10] S 2Mb, S 4Mb, S 6Mb represent static topologies for offered load 2Mbps, 4Mbps, and 6Mbps. M 2Mb, M 4Mb, M 6Mb represent mobile topologies for offered load 2Mbps, 4Mbps, and 6 Mbps.

interference level in the network), the higher the performance gain RAF is able to achieve. Note that here RBAR seems to perform worse than ARF since we implement ARF with RTS/CTS disabled, thus eliminating extra overhead.

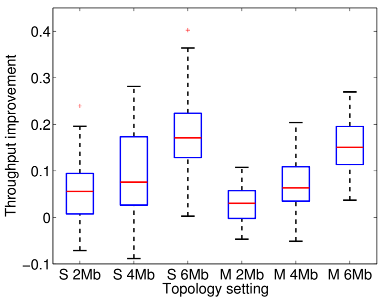


Figure 4.15: RAF's throughput improvement over ARF in 30 random static/mobile topologies.

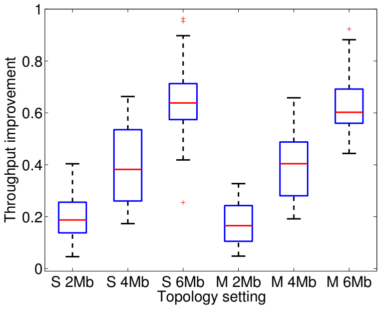


Figure 4.16: RAF's throughput improvement over RBAR in 30 random static/mobile topologies.

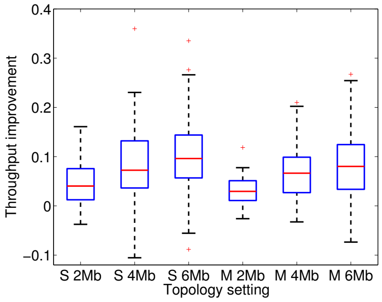


Figure 4.17: RAF's throughput improvement over OAR in 30 random static/mobile topologies.

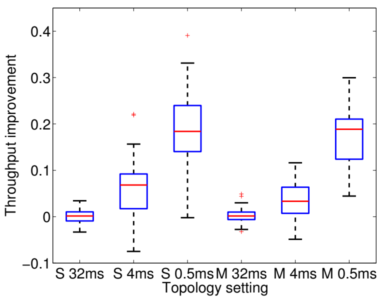


Figure 4.18: RAF's throughput improvement over ARF in 30 random static/mobile topologies with Poisson VBR traffic.

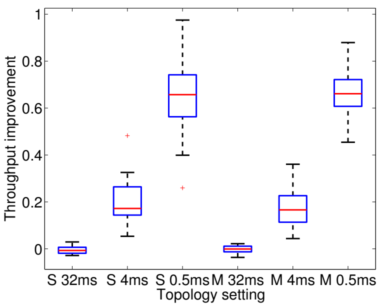


Figure 4.19: RAF's throughput improvement over RBAR in 30 random static/mobile topologies with Poisson VBR traffic.

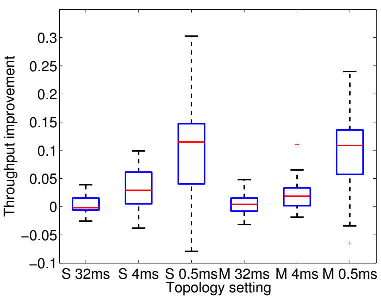


Figure 4.20: RAF's throughput improvement over OAR in 30 random static/mobile topologies with Poisson VBR traffic.

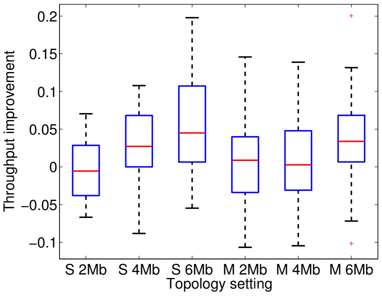


Figure 4.21: RAF's throughput improvement over ARF in 30 random static/mobile topologies with Binary Exponential On-off VBR traffic.

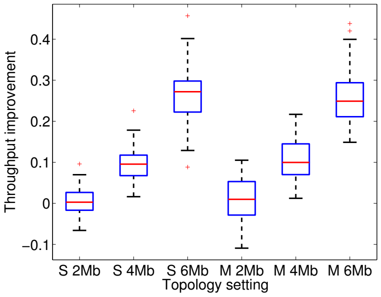


Figure 4.22: RAF's throughput improvement over RBAR in 30 random static/mobile topologies with Binary Exponential On-off VBR traffic.

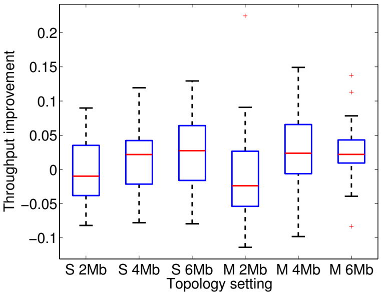


Figure 4.23: RAF's throughput improvement over OAR in 30 random static/mobile topologies with Binary Exponential On-off VBR traffic.

We further evaluate RAF's performance improvement over ARF, RBAR, and OAR for VBR traffic with packet arrival to be Poisson distribution[11] and binary exponential on-off distribution (the settings are the same as in Section 4.2.1). As expected, when traffic becomes more volatile, RAF's performance degrades more compared with a CBR traffic pattern. However, it is still able to achieve average throughput improvement up to 20% over ARF, 65% over RBAR, and 11% over OAR for Poisson packet arrivals. For binary exponential on-off packet arrivals, we observe that RAF significantly improves the throughput for RBAR (25%), while remains comparable to ARF and OAR.

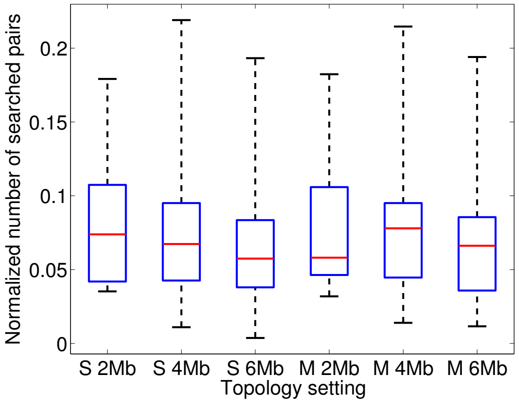


Figure 4.24: Normalized number of searched channel rate and frame size pairs for 30 random static/mobile topologies.

We finally show the effectiveness of our two proposed methods for the search of suitable rate and frame size given a recent history of channel idle intervals, as presented in Section 4.1.2. For each one of the 30 experiments, we calculate the total number of searched rate and frame size pairs when the search is enhanced by our algorithm, and normalize it by the total number of pairs with a brute-force search. Again we use box plot to show the normalized number of searched channel rate and frame size pairs for each of the 30 topologies in the 6 groups. As shown in Figure 4.24, all the medians of the computation overhead in the 6 groups are within 9% of the brute-force search (around 11 times faster), and we reduce the computation overhead by at least 75%.

4.3 Summary

As wireless devices defined in the unlicensed frequency bands proliferate, interference is becoming a dominating factor to the success or failure of a transmission. However, the majority of existing

[11] S 32ms, S 4ms, S 0.5ms represent static topologies for average packet inter-arrival time to be 32ms, 4ms, and 0.5ms. M 32ms, M 4ms, M 0.5ms represent mobile topologies for average packet inter-arrival time to be 32ms, 4ms, and 0.5ms.

rate control algorithms are interference oblivious. They often lead the system into a state where all interfered wireless transceivers operate at low data rate and the overall contention for the wireless channel stays high.

In this chapter we present rate adaptive framing (RAF), a joint channel rate and frame size control that addresses both interference and noise for achieving maximum throughput. The design of RAF leverages the patterns of interference, as a result of the spatial and temporal correlations of wireless traffic, and derives the channel rate and frame size. An RAF transmitter obtains such a configuration from the ACK message from the receiver, and applies it in the transmission of the next frame. Through simulations we have shown that RAF can outperform RBAR, OAR, and ARF under various levels of interference and traffic patterns. RAF performs better in CBR interference patterns than VBR interference patterns. This is because the variation of the length of idle intervals in VBR interference patterns degrades the predictability of channel rate and frame size selection.

Chapter 5

Opportunistic Carrier Prediction

In this chapter, we propose Opportunistic Carrier Prediction (OCP), a novel approach to allow each wireless sender to opportunistically access the medium. OCP's rationale is based on the observation that interference from the past can be a good indicator for the outcome of future packet delivery. Therefore, each sender maintains an empirical summary of *interference relationship* (who interferes my receiver and who is interfered by me) in the proximity. When the sender overhears that an interferer is in transmission or a flow that will be interfered by the sender's transmission is active, it defers its transmission until both the interfering sender and the interfered flow finish their transmissions.

To achieve the goal of OCP, we have to address the following challenges: First, how can a sender infer who is interfering its receiver and who is interfered by its transmission? Second, since each sender makes the medium access decision based on what it overhears on the channel, how can a sender efficiently extract what's going on from the wireless medium and update its channel access decision in a timely manner. Third, how do we ensure each sender correctly decodes the overheard information even in high network contention level with relatively low overhead? Finally, what is the channel access scheme if the sender does not overhear anything on the wireless medium?

In OCP, each sender infers the interference relationship by relating the overheard information on the channel to its receiver feedback of whether the previously sent packets are correctly received. Senders further exchange information to complete the interference relationship. In order for a sender to efficiently extract on-going transmissions from the wireless medium, we insert a few bits right after the physical layer preamble. These bits are used as the flow identity that consists of sender/receiver identity pair so that after decoding the overheard packet preamble the sender can immediately extract the on-going flow information. Further, these bits are modulated using the most robust scheme available so that they can be correctly decoded in the presence of high level interference. In this chapter, we assume a new physical layer decoding technique called *pre-emptive reception* that works as follows. The receiver first decodes the packet up to the receiver's identity. If the packet is destined to the receiver, it decodes the rest of the packet. Otherwise, it withdraws from the reception state. The benefit of pre-emptive reception is that once the sender finishes overhearing the on-going flow identity, it can switch to capture other flow transmissions that arise later, thereby collecting more information from the wireless medium. In case that the sender does not overhear on-going flow information from the channel, the standard carrier sensing multiple access (CSMA) is adopted. Therefore, OCP can be applied in conjunction with existing algorithms [48, 49, 72, 77, 78] that tune the carrier sense threshold to optimality.

In summary, our contribution is: First, we propose a novel wireless medium access protocol (OCP) for each sender to dynamically learn from the history and infer the interferers' identities in the proximity to address both exposed and hidden terminal problems. Second, although techniques similar to pre-emptive reception had been proposed by previous work [18, 61], we believe we are the first to utilize this technique to empirically infer the interference relationship in wireless networks. Third, prior work CMAP [67] argued that carrier sensing is too conservative and proposed a medium access scheme that purely relies on the conflict map (or interference relationship)[1], based on a complex physical layer decoding scheme called partial packet recovery (PPR) [36]. We show that to infer the interference relationship, a simple yet effective approach from receiver's feedback is sufficient. Further, we show that simply relying on the interference relationship and blindly turning off carrier sensing does not help improve the throughput in the network with high contention. In particular, we will see that such a scheme may degrade the throughput by up to 71% in random topologies. Finally, through extensive simulations, we show that OCP improves the system throughput over CMAP and CSMA in random topologies of various contention levels by up to 350% and 170% respectively, and improves the packet delivery success ratio by up to 380% and 400%, while almost removing starvation in many settings.

The rest of the chapter is organized as follows. We describe the most related work CMAP [67] in Section 5.1. We further show in Section 5.2 that the partial packet recovery (PPR) [36] technique adopted by CMAP cannot decode the packet header/receiver in a highly contented network. We present OCP in Section 5.3 and report the evaluation results in Section 5.4. We summarize in Section 5.5.

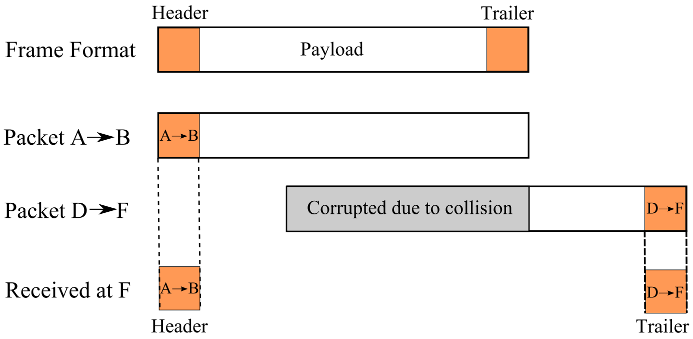


Figure 5.1: F can decode the header sent by A and trailer sent by D to infer that A→B interferes D→F.

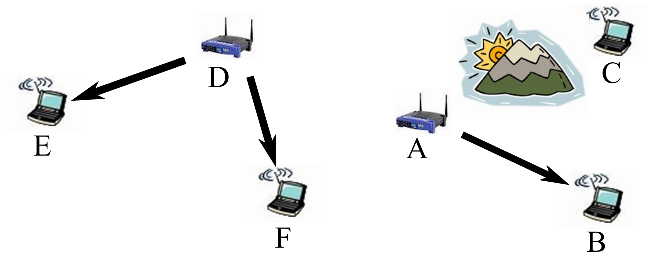


Figure 5.2: Illustration of exposed terminal and hidden terminal problems in wireless networks. A and D are two exposed senders. C is a hidden interferer from A.

5.1 CMAP Description

CMAP [67] argued that carrier sensing is too conservative and proposed to turn off carrier sensing and let each node access the medium based on the conflict map (interference relationship). They apply the partial packet recovery (PPR) technique [36] to empirically build the conflict map in each node's neighborhood. In particular, they append trailer and post-amble to each packet payload and include the flow identity into both header and trailer of each packet, as shown in Figure 5.1, based on the observation that when a collision happens, the header and trailer of the two colliding packets are usually intact and can be correctly decoded. Take Figure 5.2 for example, if A→B and D→F are active at the same time, although the two packets are colliding with each other, F can still decode the packet header sent by A and packet trailer sent by D to infer that flow A→B is interfering flow D→F. Once F infers such interference relationship, it publishes such information to sender D and all other neighboring nodes, thereby establishing the conflict map in the network. When next time A→B is active, D will defer its transmission to F until A→B is finished. Note that in such a scheme, it is the receiver who infers the interference relationship. The sender accesses the medium based on the conflict map collected from its receiver and all other neighboring nodes.

[1] We come up with the OCP idea independently with CMAP.

5.2 Why CMAP Does Not Work in General

Although CMAP's observation in Figure 5.1 is true for a simple two-packet collision, we argue that this is generally not true in a bigger network. Consider a network with 10 or 20 flows. When carrier sensing is turned off, all nodes could send out packets and the collision will likely consist of complicated overlapping of packets in the air. The interferer's header/trailer might also be interfered by a 2nd, 3rd interferer, and so on[2]. Thus, whether their claim holds in general needs more justification. In particular, we want to know how likely the header and trailer can be successfully decoded when carrier sensing is turned off under high network contention level.

Before answering the above question, we categorize a packet collision into two groups: (1) Collision In the Beginning (CIB): the interference level is too high for the receiver to even start receiving the packet. For example, the collision at F in Figure 5.1 belongs to CIB. (2) Collision In the Middle (CIM): during the middle of receiving the packet, the interference from other nodes causes the receiver to drop the currently receiving packet. For example, if we reverse the transmission order of D→F and A→B in Figure 5.1, packet collision still happens at F, but it belongs to CIM now. Note that if the packet collision can be categorized into both CIB and CIM, we give preference to CIB. The reason will be clear later in the discussion.

We implemented a more realistic physical layer capture model [45] in ns-2 simulator by considering the interference propagated from all other nodes. Whether a packet can be captured/received depends on the modulation scheme (transmission data rate) and the corresponding SINR value. We randomly generate 10 topologies, each with 20 distinct backlogged flows in a 1000m x 1000m area. More detailed setting can be found in Section 5.4.

Although CMAP assumes that it is the header/trailer *closest* to a packet collision that contains the interferer information and can be decoded, in the simulation we only require that the header/trailer of *any* packet overlapping a collided packet be decoded in order to identify the interferer. If the receiver can decode both the interferer's identity and the sender's identity in a packet collision, we say the collision is *decodable*. The ratio of the number of decodable collisions to the number of total collisions is called the *decode ratio*. We ask the question that under such network contention as described above, can we still decode the interferer information and infer the conflict relationship?

In Figure 5.3, we plot the cumulative distribution function (CDF) of the decode ratio for each flow in the 10 random 20-flow topologies. For CIB, the decode ratio is mostly under 46%. What's

[2] In their scheme, the header and trailer are modulated using the same rate as the payload.

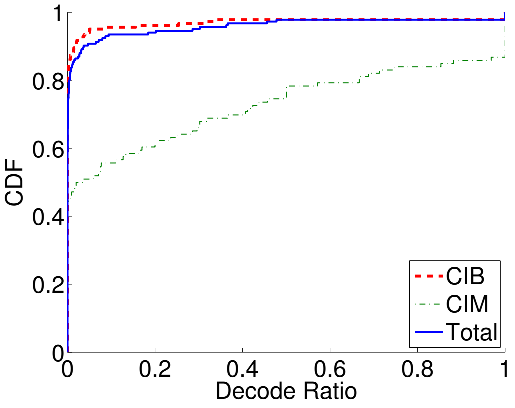


Figure 5.3: CDF of the per-flow decode ratio for 10 random 20-flow topologies

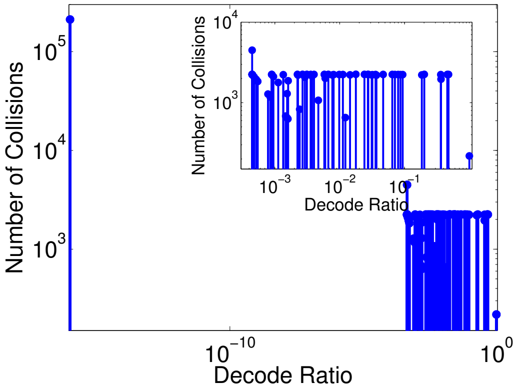


Figure 5.4: Number of collisions for each decode ratio for 10 random 20-flow topologies

worse, more than 80% of the flows have decode ratio 0%. For the case of CIM, 45% of the flows have decode ratio 0%. Although 15% of the flows whose collisions are classified as CIM have 100% decode ratio (Figure 5.3), the majority of the collisions are categorized into CIB. As a result, when combining CIB and CIM, the overall CDF of the decode ratio is not much different from that of CIB. We note here that for CIM, the header of the sender is likely to be correctly decoded and the receiver only needs to recover the trailer to decode the interferer's identity. This explains why the decode ratio for CIM is higher than that for CIB.

Since there are still tiny portion (2%) of flows with high decode ratios as shown in Figure 5.3, one question is whether this may actually help the flows suffering from hidden/exposed terminal problems. In Figure 5.4, we plot the number of collisions for each of the decode ratios over all of the 10 random topologies. As we can see (note the log-scale of x-y axes), the majority of the collisions have decode ratio 0%, therefore can not rely on partial packet recovery (PPR) to decode the header/trailer to identify the interferer's identify. What's worse, those flows with high decode ratio (therefore can apply PPR to recover the header/trailer) do not suffer from severe packet collisions.

The following points summarize our findings: First, CMAP only searches for the closest decodable header/trailer to identify who is the interferer. In reality, when there are multiple interferers, the interferer's header/trailer may also be interfered by other nodes, the true interferer which contributes the most to a collision may not be the one that is decoded by CMAP. The above experiments ignore such miscalculation, and it is clear to see that header/trailer still does not usually survive in a packet collision. Second, CMAP modulates the header and trailer using the same rate as the payload. As we have seen from the above study, such scheme does not help decode the header/trailer when multiple interferers come into play. One simple fix is to modulate the header/trailer using the lower, more robust date rate. A back-of-the-envelope calculation, however, shows that the incurred overhead (24-byte trailer plus 24-byte postamble as proposed in CMAP), when transmitted at 1 Mbps, consists of more than 35% of a 1500-byte data transmission time at 11 Mbps. Third, since the majority of the collisions can be categorized into CIB, if we can address the CIB well, most of the collisions could be avoided. Further, since for CIB the interferers' identities have already been transmitted over the air before the interfered packet being sent, it is not necessary to place the responsibility at the receiver to decode both the header and trailer of a packet collision[3] in order to infer the interference relationship.

5.3 Opportunistic Carrier Prediction

The analysis in Section 5.2 sheds light on the design of OCP. Since there is a high chance that the interferers' identities have been transmitted over the air before the interfered packet is sent, the sender can make more prudent decisions for channel access by carefully observing what's going on in the air. The main idea of OCP is to build a mapping between the overheard flow information at the *sender* side and the corresponding packet delivery success ratio (SR). In order to build the mapping, each sender in the network tries to overhear packets in the air and extracts the information of what flows (transmitter-receiver pairs) are active. All the *currently active* flows overheard by a sender are used to represent the current *channel status*. Each sender builds the mapping by relating the channel status to the SR based on its receiver feedback of whether the previously sent packets are correctly received or not. We summarize OCP as follows:

- Each node is set to promiscuous mode and overhears on-going transmissions continually (§5.3.1). Before a node transmits a data packet, all the currently active flows that are overheard are recorded as the channel status identification (CSID).
- After the node transmits the data packet, it updates the success ratio (SR) of the corresponding CSID based on whether the data is received by the receiver or not (§5.3.2).
- Each sender also periodically broadcasts the identities of its interferers so that when the interferers receive the packet and realize they are interfering some flow, they will yield to the flow when it is active (§5.3.3).

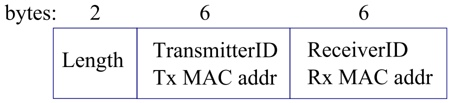
[3] Correctly decoding the packet already requires much computation effort and we believe the receiver should not be made unnecessarily complicated.

- Each sender node continually overhears currently active flows and updates its CSID accordingly. The sender uses the CSID and its current receiver to consult the (CSID,Receiver)-SR mapping before transmitting the data. The sender node accesses the channel only when no flows will be interfered and the CSID corresponds to a high success ratio, say, larger than 50% (§5.3.4).

5.3.1 Pre-emptive Reception and CSID Collection

In traditional wireless reception process, a receiving node always finishes receiving the entire packet even though the packet may be destined to other nodes. Such *non-preemptive* reception is currently implemented in most of the wireless receivers. However, in OCP, we propose that physical layer supports a *pre-emptive* reception capability. The idea is that when the node receives the packet up to the receiver's identity in the packet header, depending on whether the packet is destined to the receiver, it decides to receive the rest of the packet or not. If the packet is destined to the receiver node, it receives the rest. Otherwise, it withdraws from the reception state. Such pre-emptive reception technique is mainly used for each sender node to more efficiently overhear the on-going transmissions in the air. Once the sender decodes the sender/receiver identities in the overheard packet, it can switch to capture other ensuing flow transmissions, thereby collecting more information in the air.

One approach for preemptive reception is for physical layer to simply decode the bits all the way up to the receiver's MAC address in the header, but this inevitably forces receiver to also decode other fields in the PLCP and MAC header. A better approach for preemptive reception could be inserting a few bits right after the PLCP preamble serving as the sender/receiver identity. In this case, nodes will need to negotiate to ensure each one has distinct identity in its two-hop neighborhood. Another approach could be simply moving the MAC address to right after the PLCP preamble. Since MAC addresses are distinct, there is no need for negotiation. In this chapter, we adopt the latter approach. The detailed frame format is shown in Figure 5.5. By overhearing the *TransmitterID* and *ReceiverID*, the node knows what are the currently active flows in the air. Note that *Length* is used to indicate how long the overheard flow will last. By receiving these three pieces of information, each overhearing node knows exactly which flows will be active until when. All the three fields are modulated using the most robust scheme available so that they can be more easily captured along with existing interference.



bytes:	2	6	6
	Length	TransmitterID Tx MAC addr	ReceiverID Rx MAC addr

Figure 5.5: Frame format inserted after the physical layer preamble

5.3.2 (CSID,Receiver)-SR Mapping

Before a sender transmits DATA to the receiver, it records all the currently overheard flows. The sender concatenates the recorded flow IDs, each consisting of (*TransmitterID*, *ReceiverID*) pair, to represent the channel status (CSID) before the transmission[4]. Once the DATA is sent out, the sender waits for the ACK to update the packet delivery success ratio (SR) of the corresponding CSID. Note that the ACK may be lost even though the DATA is correctly received at the receiver. To avoid such false negative events, we redefine the ACK so that each ACK selectively acknowledges the previously received packets, which can be easily implemented using a simple bitmap.

To implement the mapping, we maintain a list of SR records (*numSuc, numFail, updTime*) containing the number of successful transmissions, number of failed transmissions, and the last time the record was updated. Note that a node may send packets to multiple receivers, and the packet delivery success ratio may be different for different receivers even with the same CSID. Take Figure 5.2 for example, when A→B is active, the corresponding CSID for both flow D→E and D→F at sender D is A→B. But, the same CSID (A→B) would give totally different prediction result at sender D for the two flows D→E and D→F. Thus, each SR record must be indexed by (*CSID, Rcvr*) where *Rcvr* is the receiver of the corresponding flow. When the transmitter receives the ACK (does not receive the ACK), it increments the *numSuc* (*numFail*) field. The packet delivery success ratio can be easily derived from *numSuc* and *numFail.* Every time the (CSID,Rcvr)-SR mapping is consulted, we require that $numSuc + numFail > 1$; otherwise, the channel is considered idle due to insufficient number of data points.

A node may temporarily move away from its sender or the channel quality may occasionally be bad, causing the SR to drop to a low value and preventing the sender from accessing the channel ever again. To address such transient events, we must age out the stale data so that senders can intermittently poll the medium and access the channel when channel quality becomes good. Each time the (CSID,Rcvr)-SR mapping is accessed, we age out the corresponding SR record by

[4] We do not distinguish the order of flow IDs if multiple flows are overheard.

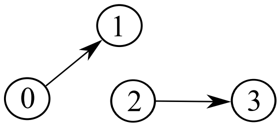


Figure 5.6: Flow 0→1 suffers from the interference from node 2; while node 2 always sends packet to 3 successfully. Node 0 will inform node 2 that it is being interfered.

multiplying the *numSuc* and *numFail* by the aging factor α,

$$\alpha = \begin{cases} 1 - \frac{t - updTime}{T_{window}} & \text{if } t - updTime < T_{window} \\ 0 & \text{Otherwise} \end{cases} \tag{5.1}$$

Note that when $t - updTime = T_{window}$, α will be zero. In our implementation, we set T_{window} to be 5 seconds, meaning that the SR records remain effective within the 5-second window.

5.3.3 Handling Dominating Interferers

From the (CSID,Rcvr)-SR mapping, each node can easily infer what are the interferers. For example, flow 0→1 in the two-flow topology shown in Figure 5.6 is suffering from the interference from node 2. Flow 2→3, on the other hand, always succeeds in packet transmission. When node 0 examines the (CSID,Rcvr)-SR mapping, it will find that the success ratio of (2→3, 1) is low (less than 50%) and identify 2 as the interferer. After each transmitter node infers who are the interferers from the (CSID,Rcvr)-SR mapping, it periodically sends out packets using the most robust modulation to tell its neighbors who are the interferers. The broadcast packet contains a list of (*interfererID, TxID, RxID*) where *interfererID* is the identity of interferer and *TxID* (*RxID*) is the transmitter (receiver) ID of the interfered flow. In the above example, when node 2 receives the broadcast packet containing $(2, 0, 1)$, it knows that flow 0→1 suffers from its interference. When next time node 2 overhears that 0→1 is transmitting, it will yield to flow 0→1 until the transmission is finished[5].

When the sender examines the (CSID,Rcvr)-SR mapping, it is possible that the record with low success ratio corresponds to the CSID consisting of multiple flows[6]. In this case, the sender does not know the interference is due to which node(s). In our implementation, we only report 1st-order interferers, i.e. the TransmitterID in CSID that consists of only one flow whose SR is less than 50%. A more advanced approach could be to attribute the interference to the flow with the strongest receive signal strength (RSS) value. More information regarding the interferers' identities could also

[5] This is possible since node 2 knows the transmission duration of 0→1 from the *Length* field.

[6] In this case, the sender decodes the flow IDs of multiple overlapping flows.

```
predict(CSID, Receiver)
if there is any flow in CSID that is interfered by me then
    return BUSY
if CSID contains no flow information then
    if interference $>$ CSThresh then
        return BUSY
    else
        return IDLE
if (CSID,Receiver)-SR mapping does not contain the record for the (CSID,Receiver) pair then
    return IDLE
if success ratio of (CSID,Receiver) $>$ 0.5 then
    return IDLE
else
    return BUSY
```

Figure 5.7: Pseudo-code for predicting the channel status at the sender node

be extracted through mining the entire (CSID,Rcvr)-SR mapping. We leave this as our future work.

5.3.4 Opportunistic Channel Access

Each sender continually overhears the on-going flows and updates its CSID accordingly. Each time the CSID is changed or the head-of-line packet is destined to a receiver different from the receiver of the previously sent packet, the sender updates its prediction of the channel status. The prediction consists of three parts. First, if a flow that will be interfered by the sender's transmission is active (contained in the CSID), then the sender predicts the channel busy. Second, if the (CSID,Rcvr) corresponds to a success ratio less than 50%, the sender predicts channel busy; otherwise, it predicts channel idle. Finally, if the sender does not overhear any CSID, then it falls back to the standard carrier sensing. To incorporate existing backoff mechanism into OCP, the backoff timer needs to be suspended (released) whenever the channel is predicted busy (idle). Therefore, we are guaranteed that after the backoff is finished, the DATA is always transmitted when the channel is predicted idle. The detail of the prediction process is shown in Figure 5.7.

Note that rather than totally relying on the conflict mapping as what is done in CMAP[67], the above prediction process ensures that the sender accesses the medium only in an opportunistic manner, i.e. when it is confident that accessing the medium will likely be successful and cause no collision to other flows. In Section 5.4, we will see that turning off carrier sensing, as in CMAP, suffers from significant throughput loss by up to 71% in random topologies when compared with CSMA-based schemes.

5.4 Performance Evaluation

We implement OCP in ns-2 simulator. Existing ns-2 does not allow for packet capture even when the SINR of one packet is much larger than the other. We implement a more realistic capture model [45] by considering the propagated interferences from all other nodes in the network. A packet can be received only if the signal to interference and noise ratio (SINR) is larger than the predefined threshold and the signal is above the sensitivity level (receive threshold). We set the SINR threshold according to the measurement study in [9] and receive threshold according to [11] so that the corresponding receive range is 232m for 11 Mbps data rate and 550m for 1 Mbps data rate (for modulating CSID).

We also implement CMAP in ns-2 simulator. Our CMAP implementation incorporates header/trailer decoding for extracting conflict relationship at the receiver, channel contention decision based on the conflict map at the sender, and ACK retransmissions. We try to follow the specification as described in [67] as much as we can.

The methodology to evaluate the efficacy of OCP is as follows. First, we study a set of typical hidden/exposed terminal topologies consisting of two or three flows. We then turn to random topologies setting. We randomly place 5 (or 10) distinct flows by randomly choosing senders and receivers that are within the communication range of each other. Each flow runs a *back-logged* CBR traffic in a 600m by 600m area. Since CSID is modulated using the lowest data rate, all nodes are likely to receive each other's CSID in this setting. Second, we evaluate OCP's performance in a more complicated random topology setting. We place 15 (or 20) distinct flows in a 1000m by 1000m area. In such topologies, hidden terminals may exist. Furthermore, with 15 (or 20) flows, the contention level will be more variant and thus more difficult for a node to infer the interference relationship. For each of the above settings, we randomly generate 50 topologies and compare the performance between OCP, CMAP, and the IEEE 802.11 protocol. In what follows, we use CSMA to denote the IEEE 802.11 protocol. Besides the two-ray ground signal propagation model, we also adopt the shadowing signal propagation model which simulates fading effects. We set the path loss exponent to 4.0, standard deviation 9.0, reference distance 1m, and reception rate 95%[7] to simulate shadowed urban area [3]. Unless otherwise stated, binary exponential backoff is turned off so that each node maintains the same contention level during the evaluation.

We try to answer the following questions in the next few sections: (§5.4.1) & (§5.4.2) What is the

[7] This means that when sender and receiver are separated by the communication range (232m), the packet can be received 95% of the time. The probability increases (decreases) with the decrease (increase) of the distance between the sender and receiver.

performance improvement of OCP for typical exposed/hidden terminal topologies? (§5.4.3) Should we turn off carrier sensing and purely rely on the inferred interference relationship as done in CMAP [67]? Further, is partial packet recovery (PPR) necessary to infer the interference relationship? (§5.4.4) Can OCP improve the throughput over existing CSMA mechanism? (§5.4.5) Can OCP improve the packet delivery success ratio over existing CSMA? (§5.4.6) Can OCP alleviate starvation in the network?

5.4.1 Typical Exposed/Hidden Terminal Topologies

As shown in Figure 5.8, we place four nodes in the network. In this topology, node 0 and node 2 are senders and can sense each other's transmission. Since receiver 1 and 3 are far away from their corresponding interferers 2 and 0 respectively, flow 0→1 and 2→3 can be active concurrently. However, existing CSMA protocol does not allow such concurrency since node 0 and 2 can sense each other. As a result, at most one flow can be active at any given point of time. Indeed, in Figure 5.9, we see that when the offered load of both flows increases, flows 0→1 and 2→3 are able to achieve fair throughput. But the total throughput cannot go beyond 6.3 Mbps for CSMA, the maximum throughput of a single flow. By introducing opportunistic medium access, node 0 and node 2 can access the channel concurrently. Therefore, the total system throughput is improved to 9.4 Mbps, a 50% improvement.

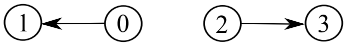


Figure 5.8: Symmetric two-flow topology, node 0 and 2 are two exposed senders.

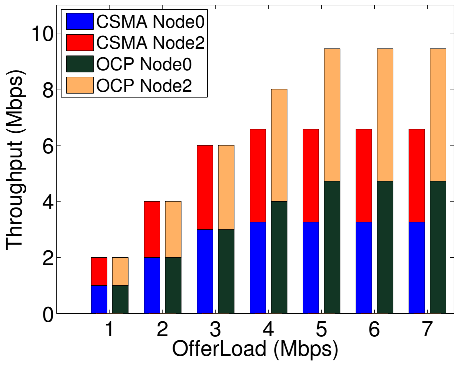


Figure 5.9: Throughput of two exposed senders for varied offered load

We then evaluate the performance of OCP for the topology shown in Figure 5.10. Since node 2 is in the proximity of node 1, it can potentially interfere node 1's reception. Further, node 2 is hidden

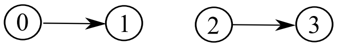


Figure 5.10: Asymmetric two-flow topology, sender 2 is hidden from sender 0.

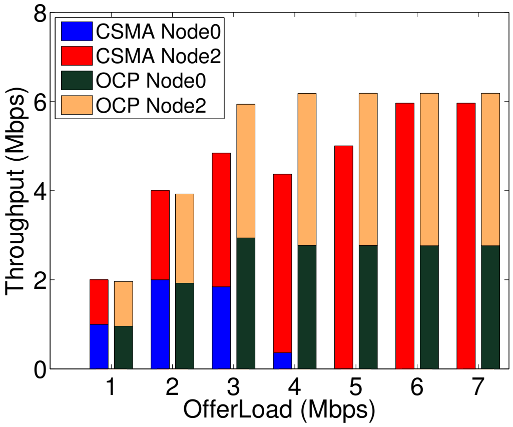


Figure 5.11: Throughput of asymmetric hidden-exposed flows for varied offered load

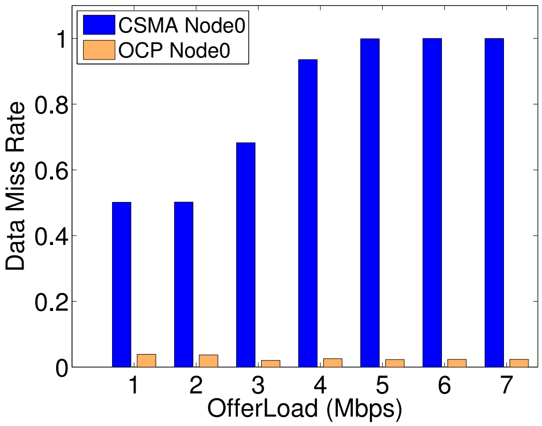


Figure 5.12: Data miss rate of asymmetric hidden-exposed flows for varied offered load

form node 0 and existing carrier sensing does not help node 0 to detect the existence of node 2. One possible solution is to simply reduce the carrier sense threshold at node 0. But this also forces node 0 to be silent when an exposed flow that can be concurrently active occurs. OCP on the other hand, allows node 0 to capture the CSID of flow 2→3 and yield to such transmission without giving up the opportunity of concurrency with exposed flows. As shown in Figure 5.11, we can see that as the offered load of *both* two flows increases, node 0's throughput reaches its maximum at 2 Mbps, then quickly drops to zero. On the other hand, OCP allows node 0 to intelligently compete with its interfering flow node 2→3 and achieve almost fair throughput among the two flows. Note also that OCP's total throughput of the two flows outperforms that of CSMA for most of the offered loads. Figure 5.12 shows the data miss rate (number of MAC layer data packet losses due to collisions / total MAC layer data transmitted) of the two flows. Since receiver 3 will not be interfered by all potential interferers, we only plot the data miss rate for flow 0→1. Clearly, due to the hidden terminal node 2, node 0 does not benefit from CSMA and thus blindly accesses the channel. On the other hand, OCP reduces node 0's data miss rate to be less than 5%.

We further evaluate the topology shown in Figure 5.2. In this topology, we first set the carrier sense threshold to be small so that nodes D and A can carrier sense each other. In this case, D→F and A→B flows alternate their occurrences, allowing F to receive the packet from D reliably. However, that D and A can carrier sense each other restricts the potential concurrency between D→E and A→B flows. As a result, D→F and D→E flow achieve only up to 1.5Mbps for CSMA as

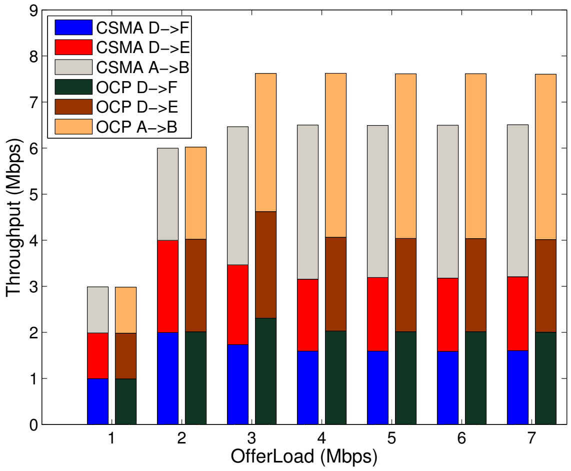


Figure 5.13: Throughput of the topology in Figure 5.2 with varied offered load when carrier sense threshold is small.

shown in Figure 5.13. On the other hand, OCP allows the concurrency between D→E and A→B flows, thereby increasing the throughput of both D→E and D→F flows by up to 2Mbps.

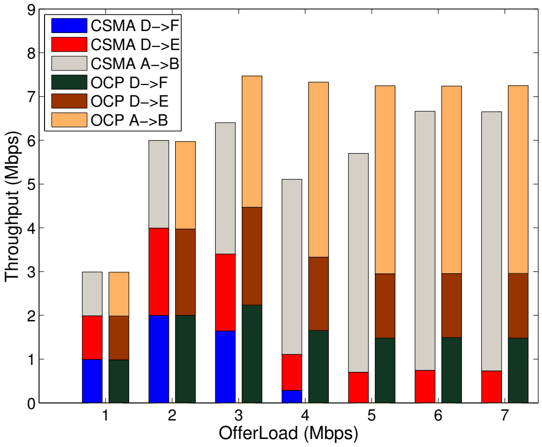


Figure 5.14: Throughput of the topology in Figure 5.2 with varied offered load, when carrier sense threshold is large.

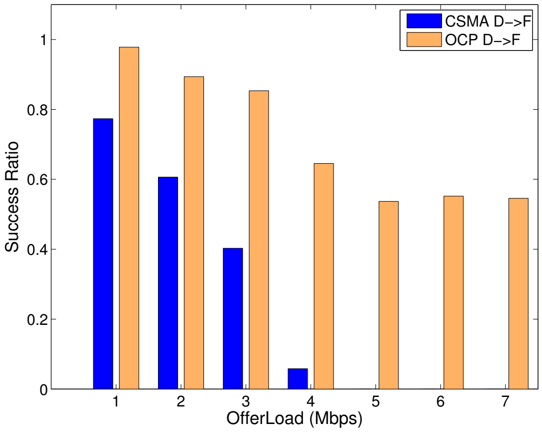


Figure 5.15: Success ratio of the topology in Figure 5.2 with varied offered load, when carrier sense threshold is large.

We then increase the carrier sense threshold so that node D and A cannot sense each other. In this case, the transmission of D→F flow cannot be reliable when A→B flow is active. Figure 5.14 shows the throughput of the three flows when the offered load of each flow is gradually increased from 1Mbps to 7Mbps. For CSMA, as the offered load increases, the throughput of flow D→F degrades to zero due to the interference from flow A→B. The inability to deliver the packet for flow D→F also restricts the throughput for flow D→E, although D→E can be concurrent with A→B flow. This is because most of the time sender D attempts to transmit/retransmit the packet for flow D→F. OCP, on the other hand, is able to allow node D to distinguish the effect of flow A→B on its two receivers E and F. As a result, the throughput of D→E and D→F remains at 1.5Mbps when the offered load increases to 7Mbps. Figure 5.15 shows the packet delivery success ratio of flow D→F[8]. As the offered load increases, flow D→F for CSMA decreases to zero, while OCP maintains the success ratio to be 55%.

5.4.2 Topology with Dominating Interferers

We next place 6 nodes as shown in Figure 5.16. In this scenario, node 4 is hidden from both nodes 0 and 2. Therefore, nodes 0 and 2 do not know the existence of such an interferer that can affect the packet reception at their respective receivers nodes 1 and 3. So, as the offered load increases, we expect that node 4 will gradually grab the channel and dominate all the transmissions in the air. As shown in Figure 5.17, when the offered load increases, the throughput of flow 0→1 and 2→3 for CSMA peaks at 2Mbps offered load. After that, their respective throughput decreases to zero at 5-Mbps offered load, while flow 4→5 dominates the medium usage. For all varied offered loads, the total throughput does not exceed 6.5 Mbps, the maximum throughput of a single flow. OCP, on the other hand, allows each sender to decode the CSID of other ongoing flows and infer the interference relationship, so that each flow has around one third of the total throughput. Further, in this scenario, flow 0→1 and 2→3 can be active simultaneously as long as flow 4→5 is not active. OCP allows such concurrency since all the senders can decode the CSIDs in the air and infer the interference relationship. The total throughput of OCP increases by 50% than that of CSMA. We also draw the data miss rate for flow 0→1 and 2→3 in Figure 5.18. As we can see, OCP reduces the data miss rate of the two flows from 70% to 15% at high offered loads, and from 40% to 3% at low offered loads.

5.4.3 Carrier Prediction Should Be Opportunistic and Carrier Sensing Should NOT Be Turned Off

As discussed in Section 5.1, CMAP [67] argued that carrier sense (CS) is too conservative and proposed to rely on the inferred conflict mapping rather than on carrier sensing. In this section, we will answer two questions (1) Shall we turn off carrier sensing all the time as proposed in CMAP [67]? (2) Is the PPR scheme (used by CMAP) necessary to infer the interference relationship?

We first define β as the carrier sense threshold normalized by the sensitivity of receiving a packet,

[8] We do not show the success ratios of flows D→E and A→B since they are both 100%.

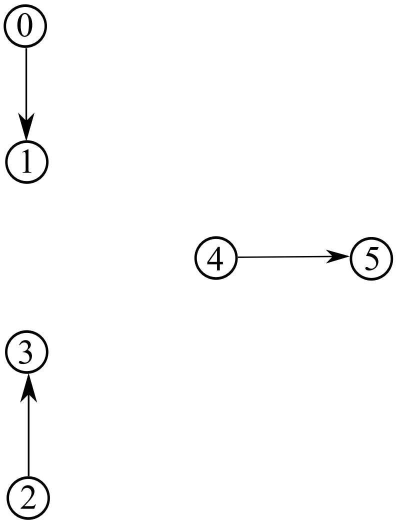


Figure 5.16: Asymmetric hidden-exposed 3-flow topology. In this scenario, node 4 is hidden from both node 0 and 2.

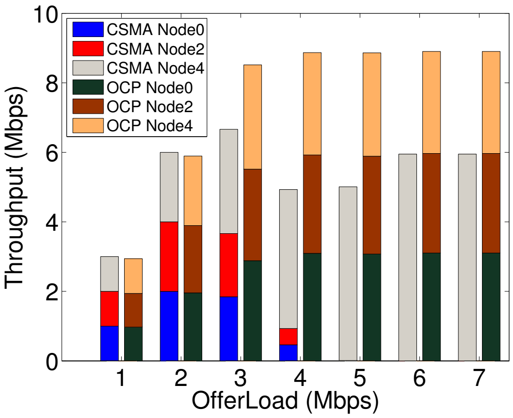


Figure 5.17: Throughput of asymmetric hidden-exposed 3-flow for varied offered load

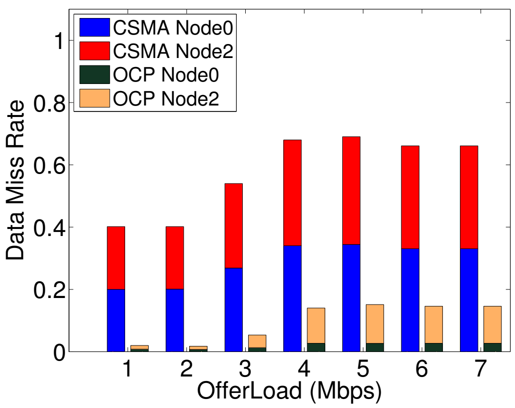


Figure 5.18: Data miss rate of asymmetric hidden-exposed 3-flow for varied offered load

i.e. $\beta = \mathit{CSThresh} \,/\, \mathit{RxThresh}$. Since an OCP-enabled sender falls back to carrier sense when it does not overhear any CSID, we vary the carrier sense threshold which OCP-enabled senders fall back to. In particular, we set the carrier sense threshold so that the corresponding carrier sense range is 768m, 512m, 384m, and 256m[9]. The corresponding β value is -21, -14, -9, -2. Now, for each of the 50 random topologies, we compare the throughput of OCP for which carrier sense threshold set to these β values with the throughput of CMAP.

Figure 5.19 shows the cumulative distribution function (CDF) of the throughput ratio (throughput of OCP / throughput of CMAP) for each of the 50 random topologies. Since CMAP appends to

[9] This is for both with fading and without fading. When there is fading, it means that two nodes separating by the corresponding distance can carrier sense each other 95% of the time. The probability increases (decreases) with the decrease (increase) of the distance between the two nodes.

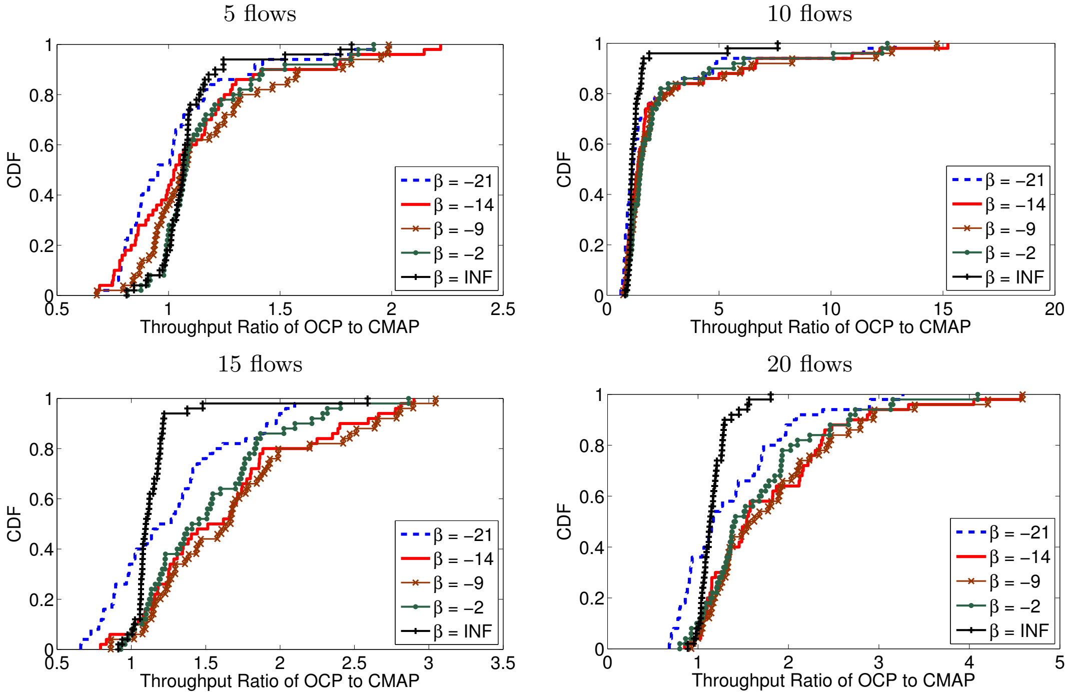


Figure 5.19: Throughput ratio of OCP at β = -21, -14, -9, -2, INF to CMAP over 50 random topologies for two-ray ground propagation. The figure will be more intelligible if it is color-printed.

each packet the trailer and postamble that consist of 35% air time of transmitting a 1500-byte packet at 11Mbps, even when OCP turns off carrier sensing (β = INF), it still outperforms CMAP for more than 93% of the time for the 5-flow random topologies. For β = -14, -9, and -2, OCP performs better than CMAP for respectively 63%, 68%, and 77% of the 50 topologies. When the number of flows increases to 10, OCP without carrier sensing outperforms CMAP for all the topologies at all β values. This is not surprising, as the contention level increases, it is more and more difficult to correctly decode the header/trailer to infer the interference relationship. Aggressively turning off carrier sensing and blindly contending for the channel with insufficient information will likely result in collisions.

When placing 15 and 20 flows in a larger area (1000m by 1000m), the contention level is more variant and hidden terminals may exist. Nevertheless, we still obtain similar results and OCP with β = -14, -9, -2, and INF outperforms CMAP for more than 95% of the 50 topologies. Note that when β = -21 in the 15-flow and 20-flow random topologies, OCP can be worse than CMAP for 40% of the 50 topologies. This is because OCP-enabled senders fall back to carrier sensing when no CSID is overheard, resulting in a more fair throughput distribution among all the nodes but less total aggregate throughput.

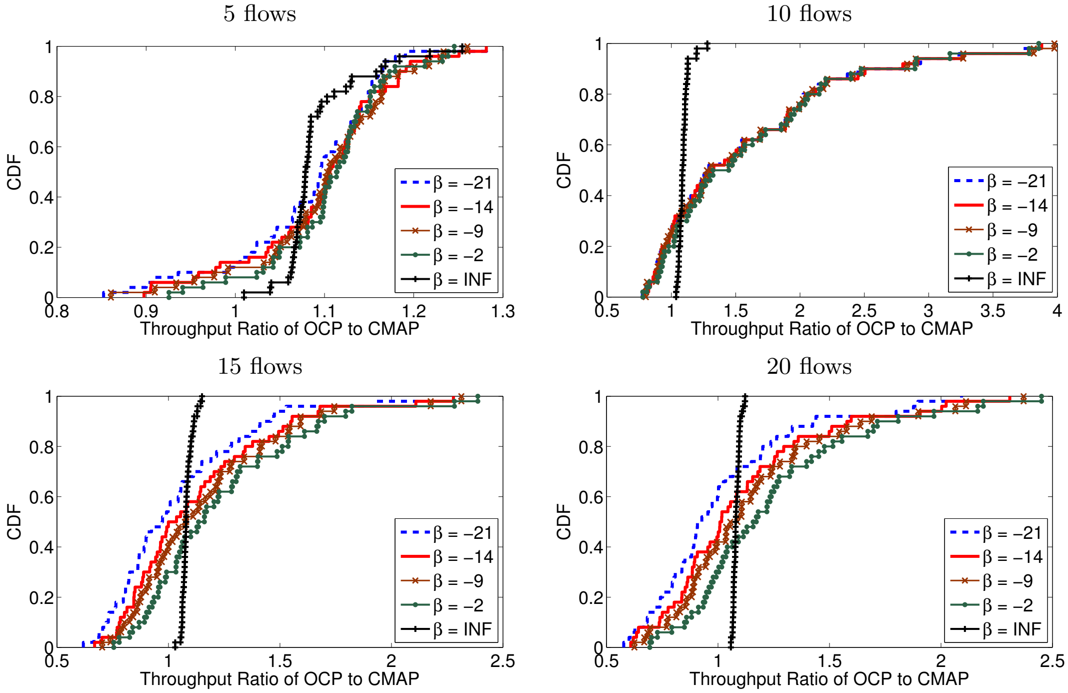


Figure 5.20: Throughput ratio of OCP at β = -21, -14, -9, -2, INF to CMAP over 50 random topologies with fading. The figure will be more intelligible if it is color-printed.

Figure 5.20 shows the throughput ratio of OCP to CMAP when fading comes into play. OCP outperforms CMAP for more than 60% of the topologies at all β values in 5-flow and 10-flow random topologies. Further, the improvement is up to nearly 3-folds. For 15 and 20-flow random topologies, OCP's can still improve up to 150%. In Figure 5.20, we found that sometimes OCP performs worse than CMAP. This is more apparent as the number of flows increases. The reason is because with fading, nodes that are out of the carrier sensing range can still sense each other with some probability. Since OCP-enabled nodes fall back to carrier sensing when no CSID is overheard, they become more conservative with fading. As a result, system fairness is improved, and the system throughput is reduced.

Figure 5.21 shows the CDF of the ratio of packet delivery success ratio of OCP to CMAP for two-ray ground signal propagation. Clearly, adopting the sender-side based interference inference is sufficient to improve packet delivery success ratio than the sophisticated and complex PPR as proposed by CMAP. Moreover, falling back to carrier sense further improves the success ratio by up to 380%, 5500%, 830%, and 1800% for 5, 10, 15, and 20 random flows, respectively. Figure 5.22 again shows the CDF of the ratio of success ratio with fading effects. Again, we see that OCP's falling back to carrier sensing significantly improves the success ratio. When turning off carrier

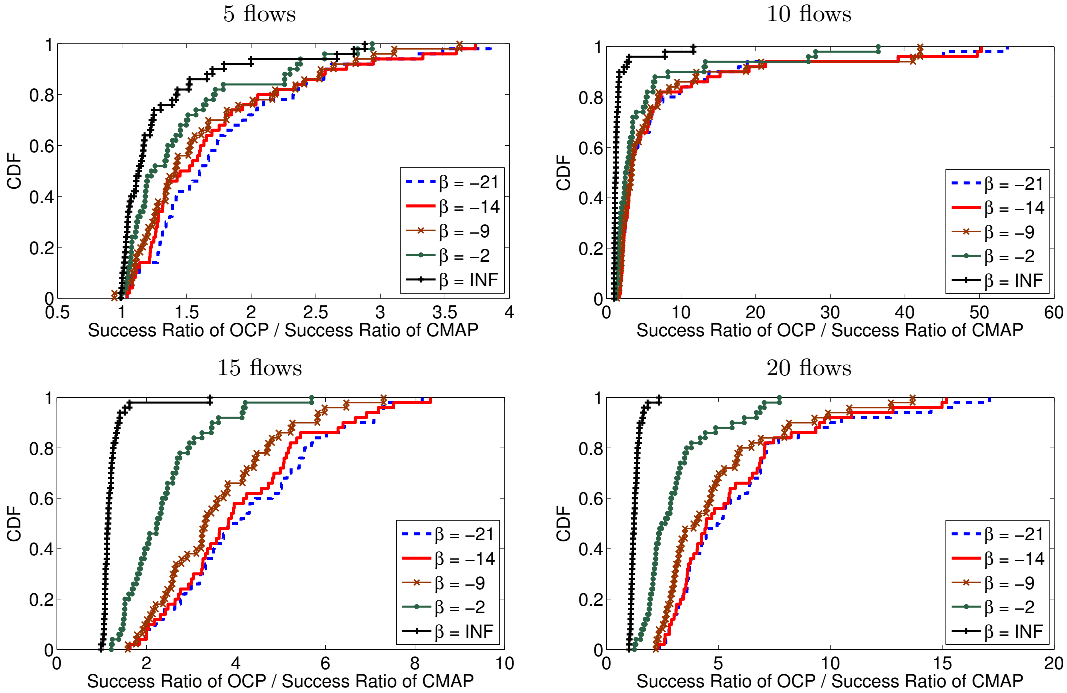


Figure 5.21: Ratio of success ratio of OCP at β = -21, -14, -9, -2, INF to CMAP over 50 random topologies for two-ray ground propagation. The figure will be more intelligible if it is color-printed.

sensing (β = INF), OCP achieves comparable success ratio performance with CMAP when there are 5 random flows. As the number of flows increases, OCP's success ratio performance (β = INF) is even better than CMAP. In particular, the improvement is up to 26%, 11%, and 8% for 10, 15, and 20 random flows, respectively. The reason is because as the interference level becomes more intense, it is more and more difficult for a CMAP-enabled receiver to decode both the header and trailer to derive the conflict map. The simple yet effective approach adopted in OCP provides a better alternative for interference inference than the one proposed in CMAP.

Note that in Figure 5.19 and 5.20 OCP without carrier sensing (β = INF) outperforms CMAP for 93% and 100% of the 50 topologies at all contention levels (5, 10, 15, 20 random flows), respectively. Further, as we have shown in Figure 5.21 and 5.22, OCP significantly improves the success ratio over CMAP for various contention levels. We believe it is more appropriate to have the sender overhear the on-going transmissions and infer the interference relationship rather than placing the burden at the already heavily loaded receiver. From Figure 5.19, 5.20, 5.21, 5.22, there is always a carrier sense setting such that CS-enabled OCP performs much better than CMAP for 5-flow, 10-flow, 15-flow, and 20-flow random topologies. Thus, carrier prediction should be opportunistic and carrier sensing should NOT be turned off.

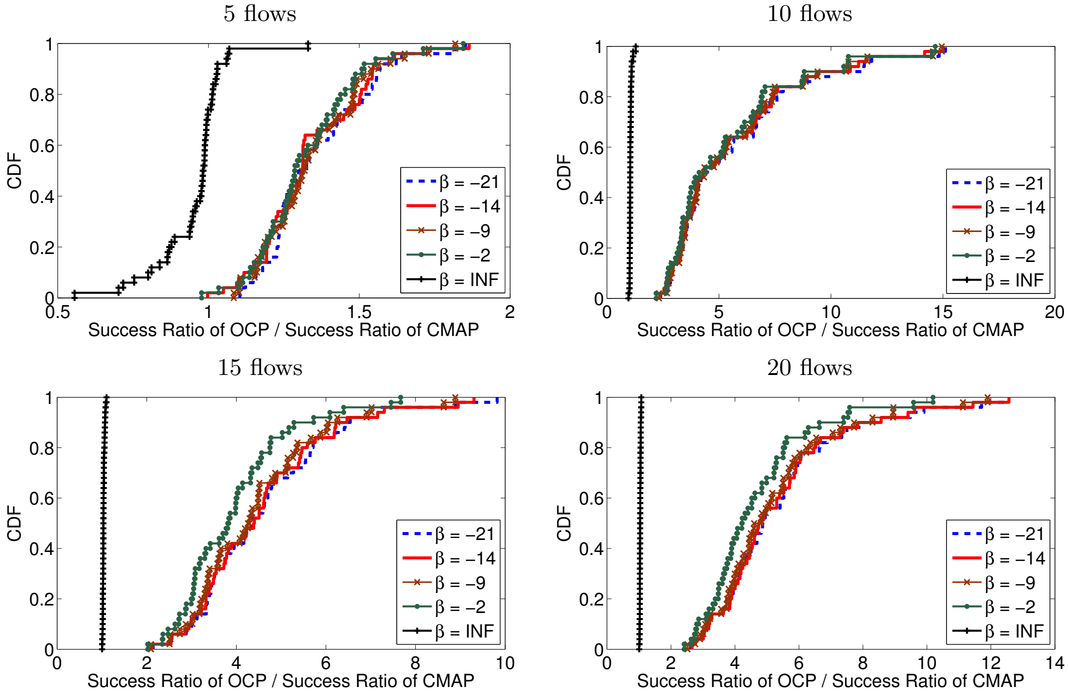


Figure 5.22: Ratio of success ratio of OCP at β = -21, -14, -9, -2, INF to CMAP over 50 random topologies with fading. The figure will be more intelligible if it is color-printed.

5.4.4 OCP Improves Throughput

In this section, we try to answer how much throughput improvement OCP has over CSMA. Since an OCP-enabled sender falls back to carrier sense when it does not overhear any CSID, we compare the performance of OCP and CSMA at varied carrier sense thresholds to evaluate how much performance gain OCP achieves.

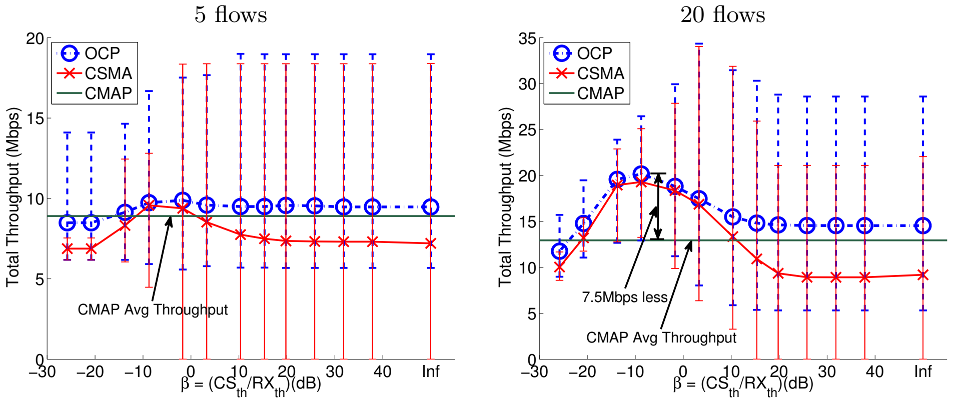


Figure 5.23: Mean, Max, and Min total throughput of 50 random topologies for OCP and CSMA at different CS$_{th}$

Figure 5.23 shows the maximum, minimum, and average throughput of the 50 random topologies

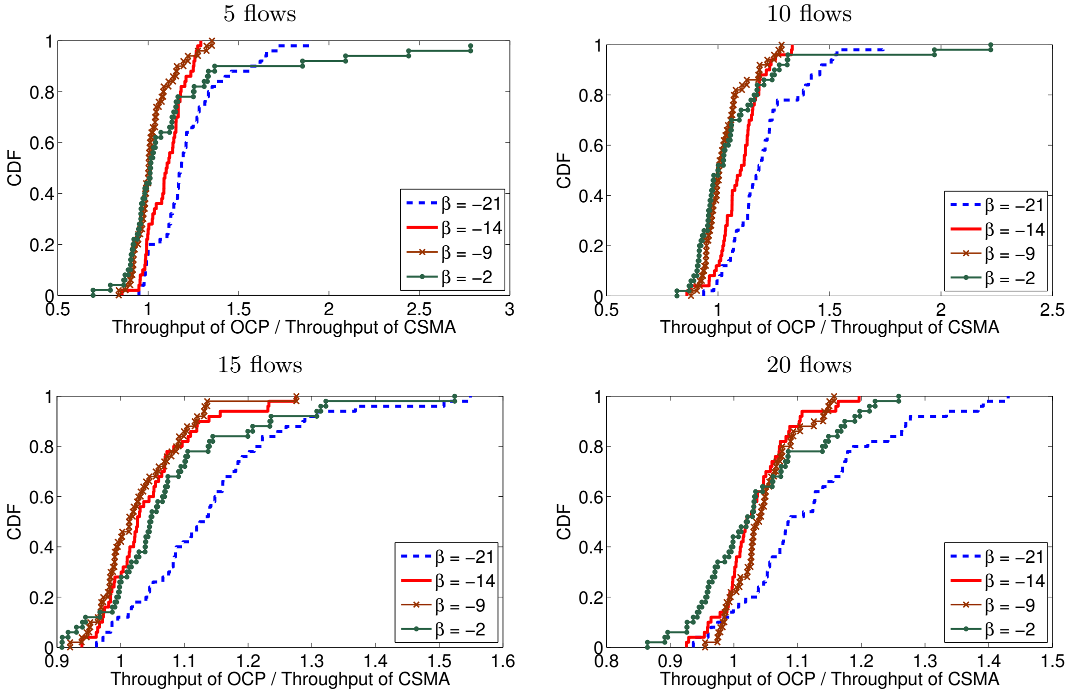


Figure 5.24: Throughput ratio of OCP to CSMA over 50 random topologies at β = -21, -14, -9, -2 with two-ray ground signal propagation. The figure will be more intelligible if it is color-printed.

for varied β values. We see that when carrier sense threshold is small, all the nodes can hear each other and the network becomes essentially a single hop network. In this case the total system throughput for CSMA does not vary much. But when carrier sense threshold is set to larger values (β >10), the throughput for CSMA could vary from 0 Mbps to as large as 20 Mbps. An OCP sender, however, accesses the medium whenever there is an opportunity. Therefore, when carrier sense threshold is low ($\beta = -26$), i.e. the network is essentially single-hop, OCP sender nodes can still grab the opportunity to boost the throughput to 14.5 Mbps for 5 random flows and 15.5 Mbps for 20 random flows. Even at the optimal carrier sense threshold ($\beta = -9$), the average throughput of OCP outperforms that of CSMA. Furthermore, at larger carrier sense thresholds, OCP improves not only the average total throughput by up to 67% but also the minimum total throughput from 0 Mbps to at least 5.5 Mbps (we will see that OCP alleviates starvation in Section 5.4.6).

Comparing the two plots in Figure 5.23, we see that OCP's average total throughput varies more significantly at different β values for 20 random flows. For example, the average total throughput of OCP is 12 Mbps at $\beta = -26$, 20 Mbps at $\beta = -9$, and 15 Mbps at $\beta =$ INF (no carrier sense). If we simply turn off carrier sense all the time, we may perform OK for 5 random flows, but we could lose up to 5 Mbps for 20 random flows compared with OCP at β = -9. Although finding the optimal

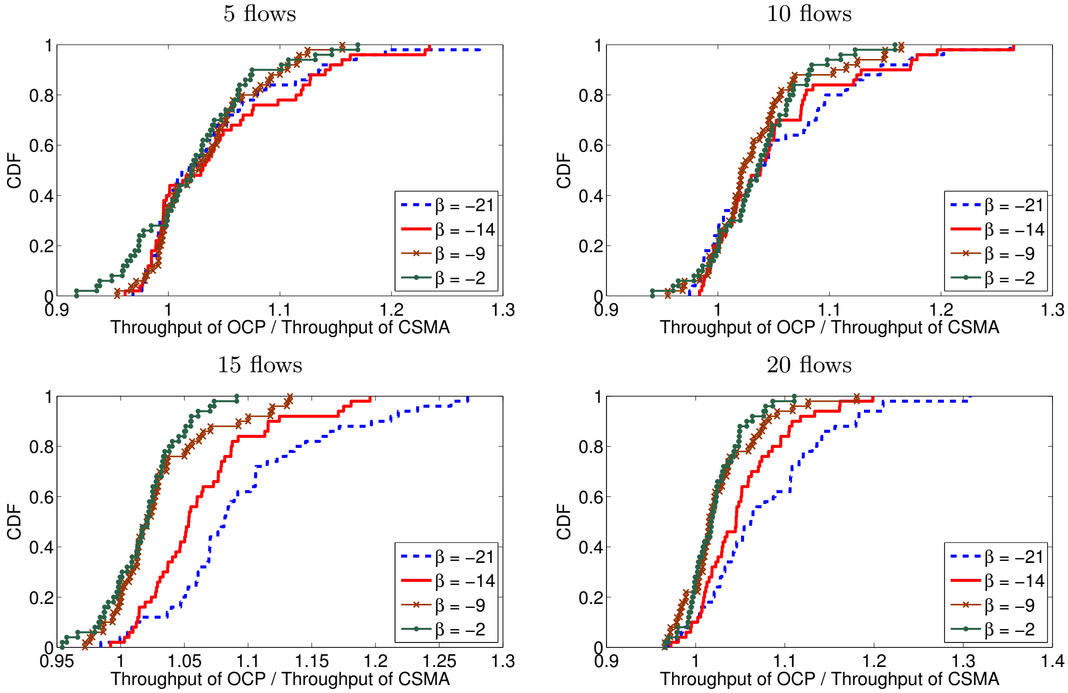


Figure 5.25: Throughput ratio of OCP to CSMA over 50 random topologies at β = -21, -14, -9, -2 with fading. The figure will be more intelligible if it is color-printed.

carrier sense threshold is out of the scope of this chapter, we point out that the effect of OCP may need to be taken into consideration when tuning carrier sense threshold to its optimality.

Note that in Figure 5.23, OCP's average throughput outperforms CMAP at most of the β values. Furthermore, CMAP performs even worse than CSMA when -10 $< \beta <$ -2 for 5 random flows and -20 $< \beta <$ 10 for 20 random flows. As the network interference level gets more intense, it is easier for CSMA to outperform CMAP. This justifies the indispensability of carrier sense and we should fall back to carrier sense and only conduct carrier prediction opportunistically.

We further compare the throughput improvement of OCP over CSMA for each of the 50 random topologies. Figure 5.24 shows the CDF of the throughput ratio of OCP to CSMA for each topology for varied β values. Let's first see the 5-flow case. For β = -9 and -2, OCP outperforms CSMA for only around 50% of the topologies. This is because carrier sense thresholds at these two values are already optimal (see Figure 5.23) and there is not much space left for OCP to opportunistically access the medium. When there is no such opportunity, OCP consumes more overhead and results in comparable performance to CSMA. When β = -21 and -14, CSMA becomes more conservative and OCP is able to exploit the opportunity of flow concurrency and improves the throughput for more than 80% of the topologies. When placing 10 random flows, OCP and CSMA are again comparable

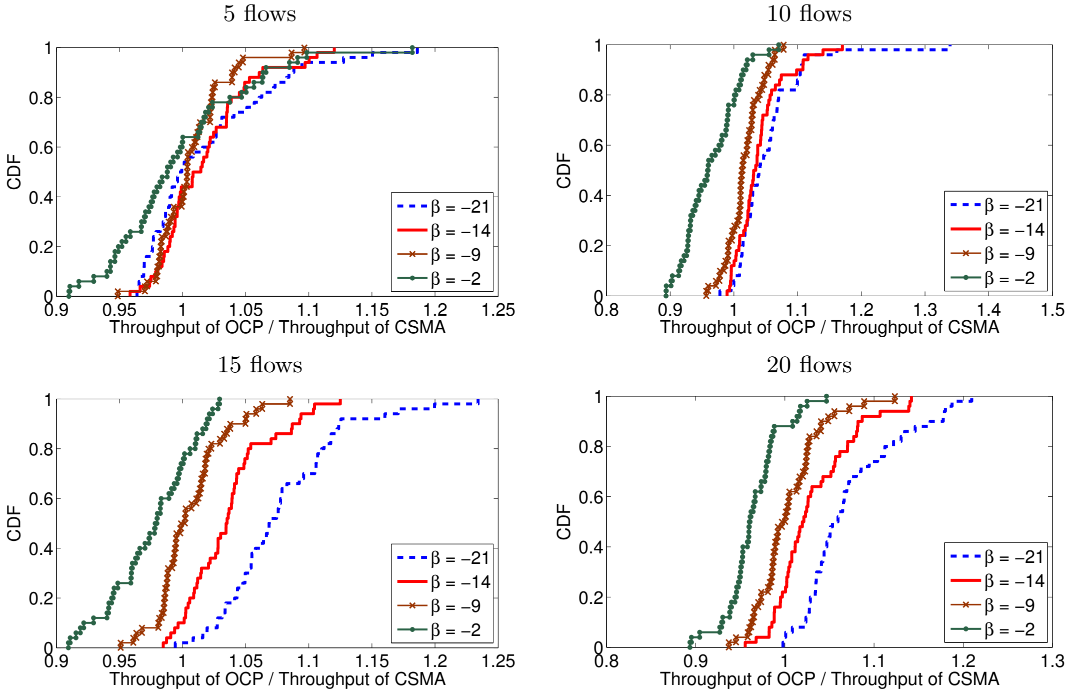


Figure 5.26: TCP throughput ratio of OCP to CSMA over 50 random topologies at β = -21, -14, -9, -2 with fading. The figure will be more intelligible if it is color-printed.

to each other at β = -9 and -2, and OCP performs much better than CSMA at β = -21 and -14.

When we place more flows (the 15-flow and 20-flow plot in Figure 5.24) in a larger area, the contention level varies more in the network and the opportunity for concurrent transmissions is more likely to occur. Indeed, for 15 random flows, OCP outperforms CSMA for more than 90%, 70%, 60%, and 71% of the topologies at β = -21, -14, -9, and -2. For 20 random flows, OCP outperforms CSMA for more than 85%, 73%, 78%, 62% of the topologies for β = -21, -14, -9, and -2.

Figure 5.25 shows the throughput ratio of OCP to CSMA with fading effects. In this case, OCP's performance is even better. For example, OCP outperforms CSMA for 73% of the topologies by up to 30% for 5 random flows, 82% of the topologies by up to 28% for 10 random flows. With 15 and 20 random flows, OCP outperforms CSMA at all β values (throughput of the majority of the topologies are improved by up to 30%).

We also evaluate OCP's performance for TCP flows. Figure 5.26 shows the throughput ratio of OCP over CSMA for various β values with fading effects. Note that for 5 TCP flows, there will be 10 contending senders in the network, leading to much higher contention level than 5 UDP flows. As a result, for β = -2, a collision is likely to occur regardless of the prediction (recall we fix the contention window size to 31). As the contention level increases, such phenomenon becomes more apparent. However, OCP still significantly outperforms CSMA (up to 30%) at β = -14 and -21.

5.4.5 Does OCP Improve Packet Delivery Success Ratio?

In this section, we evaluate OCP's performance in link layer packet delivery success ratio. We plot the maximum, minimum, and average success ratio for the 50 random topologies for OCP and CSMA with varied β value in Figure 5.27. For both 5-flow and 20-flow random topologies, OCP and CSMA's success ratios decrease when carrier sense threshold increases (β increases). This is because by setting to large carrier sense threshold each node contends for the medium more aggressively and ignores near-by transmissions. As a result, for random 5 flows CSMA's average success ratio decreases from 90% to 30% while OCP improves it to 92% and 73% respectively. For random 20 flows, CSMA's average success ratio decreases from 87% to 9%, while OCP from 83% to 23%. We also plot the average success ratio for CMAP in Figure 5.27 and note that OCP outperforms CMAP at all β values. This shows again the necessity of falling back to carrier sensing and that without using the complex decoding technique as in CMAP, OCP is still able to infer the interference relationship and improve the success ratio.

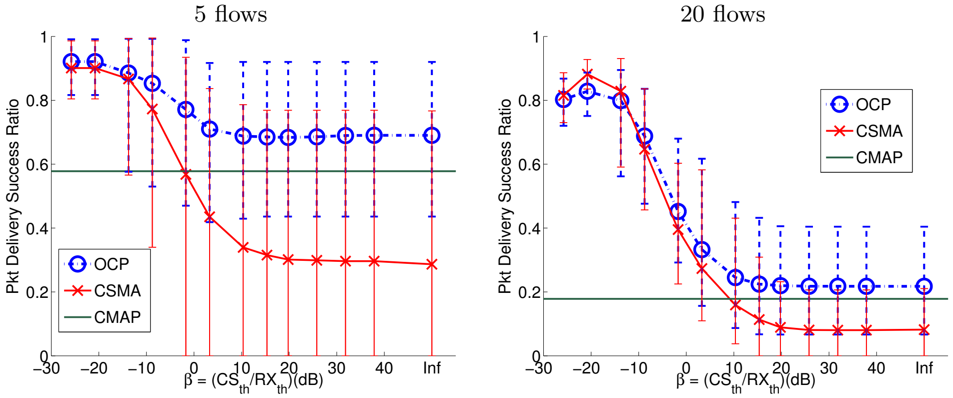


Figure 5.27: Mean, Max, and Min packet delivery success ratio of 50 random topologies for OCP and CSMA at different CS$_{th}$

Comparing the 5-flow plot with 20-flow plot in Figure 5.27, we see that OCP does not improve the success ratio in the 20-flow plot as much as in the 5-flow plot. There are three reasons. First, the contention level for 20 flows could be much more intense than 5 flows, causing more packet collisions. Second, with 20 flows, it is more likely for a sender node to decode a non-interferer while missing the packet of the true interferer when both are active. Third, we fixed the contention window size to be 31, leading to high collision probability with 20 flows. As a result, both CSMA and OCP have lower success ratios for 20 random flows than for 5 random flows.

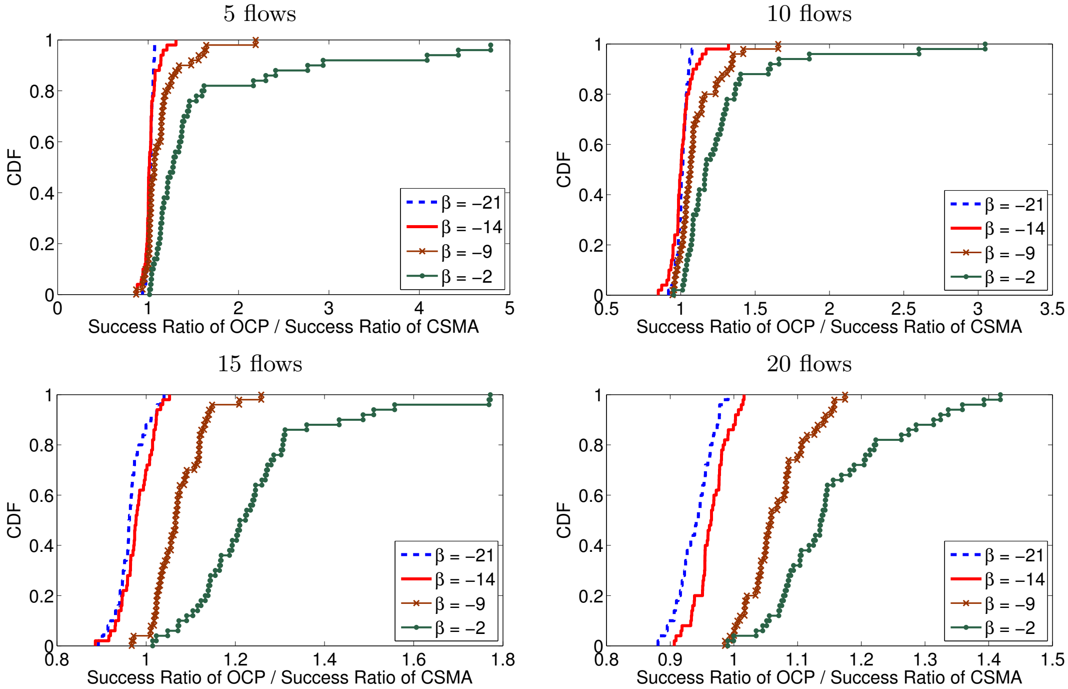


Figure 5.28: CDF of the ratio of packet delivery success ratio of OCP to that of CSMA over 50 random topologies at $\beta =$ -21, -14, -9, -2 with two-ray ground signal propagation

Note that from Figure 5.27, CMAP indeed improves the success ratio when carrier sense threshold is large ($\beta >$ -2 for 5 random flows and $\beta >$ 10 for 20 random flows). But as carrier sense threshold decreases (nodes can sense transmissions farther away), it is beaten by CSMA and OCP.

More interestingly, OCP does not improve the success ratio over CSMA for 20 random flows when β is small (Figure 5.27). This is because when β is small, the network is essentially one-hop (nodes can sense the transmission of each other) and the success ratio for CSMA is significantly increased to more than 80%. Since we encourage nodes to contend for the channel whenever the packet delivery success ratio is larger than 50%, OCP senders become more aggressive in channel access, thereby not improving the success ratio. Despite that OCP decreases the success ratio when $\beta < -10$ for 20 random flows (Figure 5.27), it improves the average total throughput by 2 Mbps (20-flow plot in Figure 5.23). To illustrate further, we compare the success ratio of OCP and CSMA for each of the 50 random topologies and draw the CDF for the ratio of success ratio in Figure 5.28. For both 5-flow and 20-flow random topologies, OCP has significant improvement over CSMA for $\beta =$ -9 and -2. On the other hand, for $\beta =$ -21 and -14, it does not perform significantly better for 5-flow random topologies and performs even worse for 20-flow random topologies. If we compare Figure 5.24 with Figure 5.28, we immediately see that while OCP performs worse for $\beta =$ -21 and -14

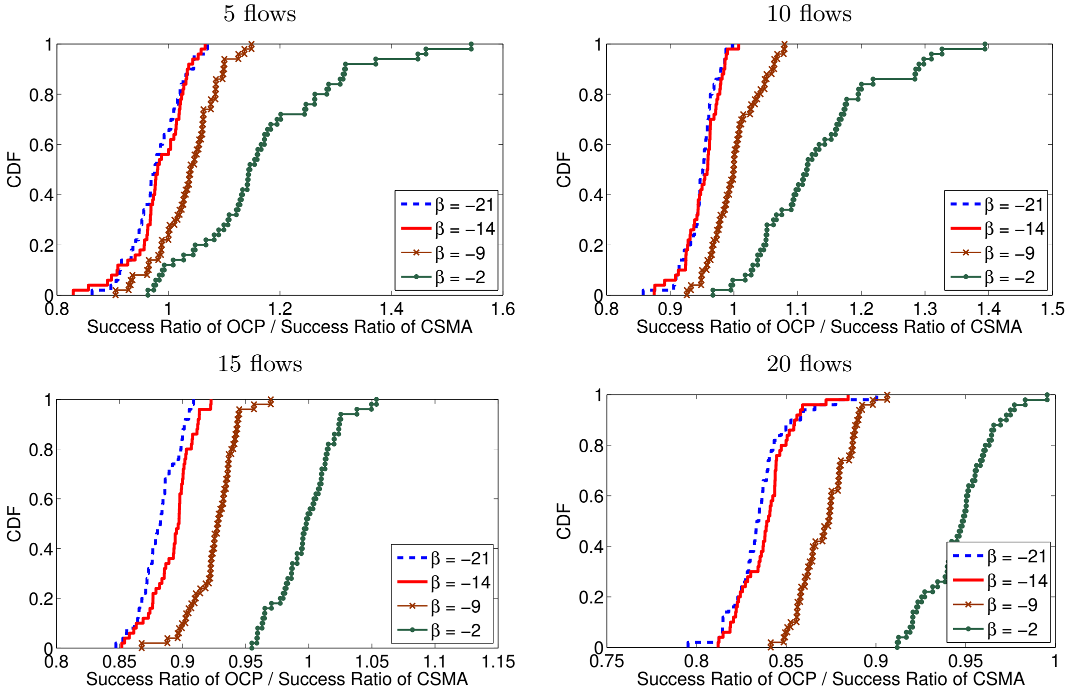


Figure 5.29: CDF of the ratio of packet delivery success ratio of OCP to that of CSMA over 50 random topologies at β = -21, -14, -9, -2 with fading

in success ratio, it is at these two β values that OCP performs much better in terms of throughput. The reason is because an OCP sender contends for the medium when it sees an opportunity under the condition that the packet delivery success ratio is at leat 50%. Such opportunistic approach may sometimes reduces packet delivery success ratio trading for more throughput. On the other hand, when there is not much opportunity to improve the throughput (β = -9 and -2 in Figure 5.24), OCP senders become more conservative rather than blindly access the channel, thereby improving the packet delivery success ratio (Figure 5.28).

The aforementioned complimentary phenomenon between throughput and success ratio also exists when fading comes into play. Again, we plot in Figure 5.29 the CDF of the ratio of success ratio of OCP to CSMA with fading. With fading, the success ratio improvement of OCP is less than that in Figure 5.28. However, this is what OCP is trading for much better throughput improvement, as shown in Figure 5.25.

Figure 5.30 shows the CDF of the ratio for TCP flows. We see again OCP's improvement over CSMA for 5 and 10 random flows. Note that when β = -2 for 20 TCP flows (40 link layer flows), OCP performs worse than CSMA both in throughput and success ratio. Recall we fix the contention window size to 31 for each packet transmission. Consequently, a collision will always occur despite

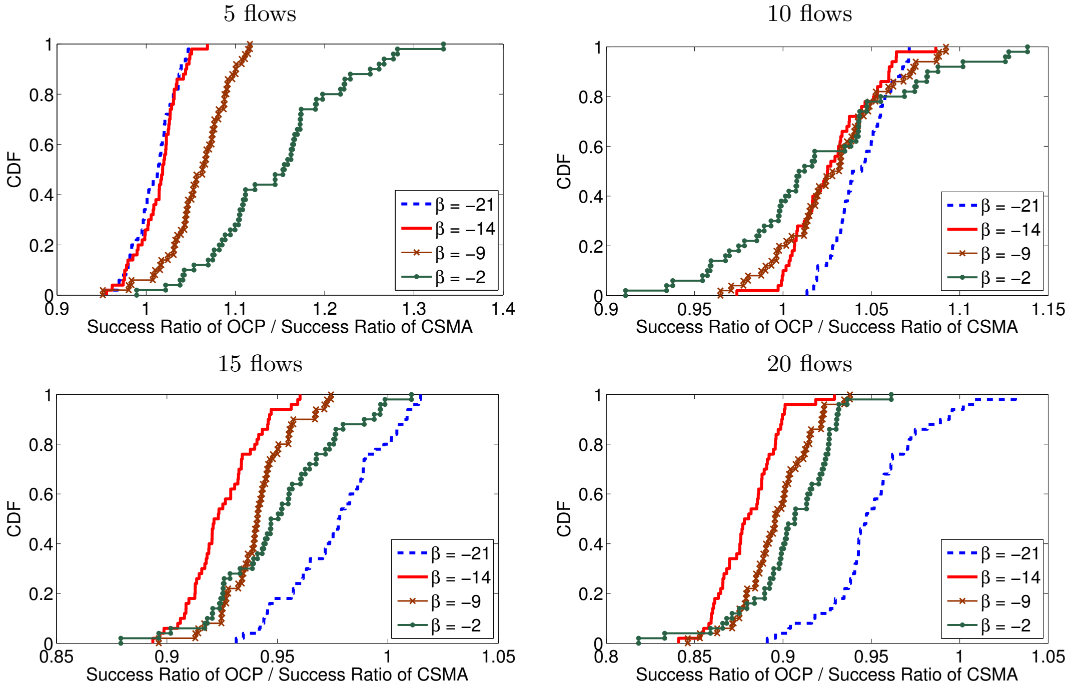


Figure 5.30: CDF of the ratio of packet delivery success ratio of OCP to that of CSMA for TCP flows over 50 random topologies at β = -21, -14, -9, -2 with fading

OCP's prediction mechanism. OCP in this case performs worse due to its modulation overhead for CSID. Similar to Section 5.4.4, this can be significantly improved with the decrease of the carrier sense threshold.

5.4.6 OCP Alleviates Starvation

Recall that the design purpose of OCP is for improving the concurrency of flow transmissions in various contention levels. We try to understand whether OCP can also mitigate the starvation of the flows. We define that a flow is starved when it gets low throughput during the simulation. For ease of exposition, let's see one random 5-flow topology in Figure 5.31. In such a topology, node 13 suffers from the interference from node 8 due to the close distance between them. Node 12, on the other hand, suffers from the interference from node 5 when both node 8 and 5 are active at the same time[10]. In such a topology, if β is set to -9, both flows 8→12 and 10→13 will be severely starved for CSMA, as shown in Figure 5.32. On the other hand, OCP totally removes the starvation of the two flows and improves the fairness among the flows. Although CMAP also alleviates the starvation over CSMA, OCP provides a much more fair throughput distribution among all the flows in the

[10] Whether the packet can be received depends on the SINR. In this case, node 2 can receive the packet from node 5 even when node 5 and node 8 are both active simultaneously.

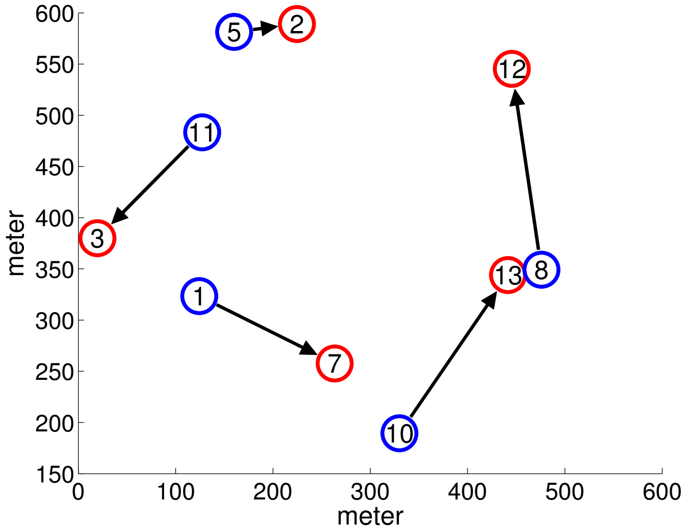


Figure 5.31: One of the 50 random 5-flow topologies

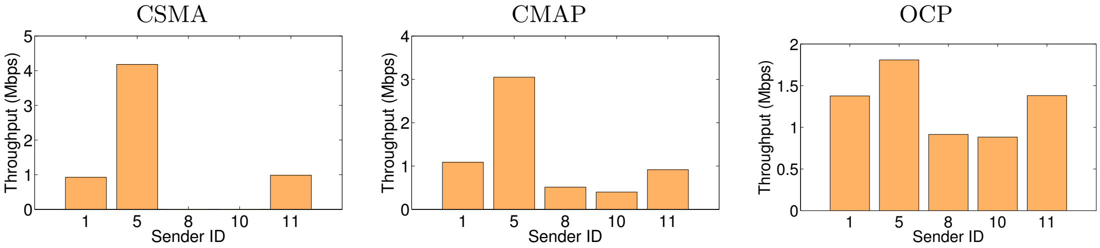


Figure 5.32: Throughput profile of the 5-flow random topology in Figure 5.31

network.

To further show that the above illustration is not a niche example, we compare OCP, CMAP, and CSMA by plotting in Figure 5.33 the portion (number of starved flows / total number of flows) of starved flows for each of the 50 random topologies for β = -9 and -2. We do not observe much starvation on other smaller β values and thus ignore those plots. Clearly, both OCP and CMAP almost completely remove the starvation for 5-flow random topologies. But when it comes to 20 random flows, CMAP actually performs the worst. This is again due to CMAP's aggressively turning off carrier sensing and contending for the channel all the time. OCP, on the other hand, significantly reduces the starvation for 20-flow random topologies. Interestingly, for the 27-th 5-flow random topology at β = -2, all the 5 flows are completely starved in CSMA (because all the senders cannot carrier sense the interfering nodes when $\beta = -2$) while OCP totally removes the starvation in this case.

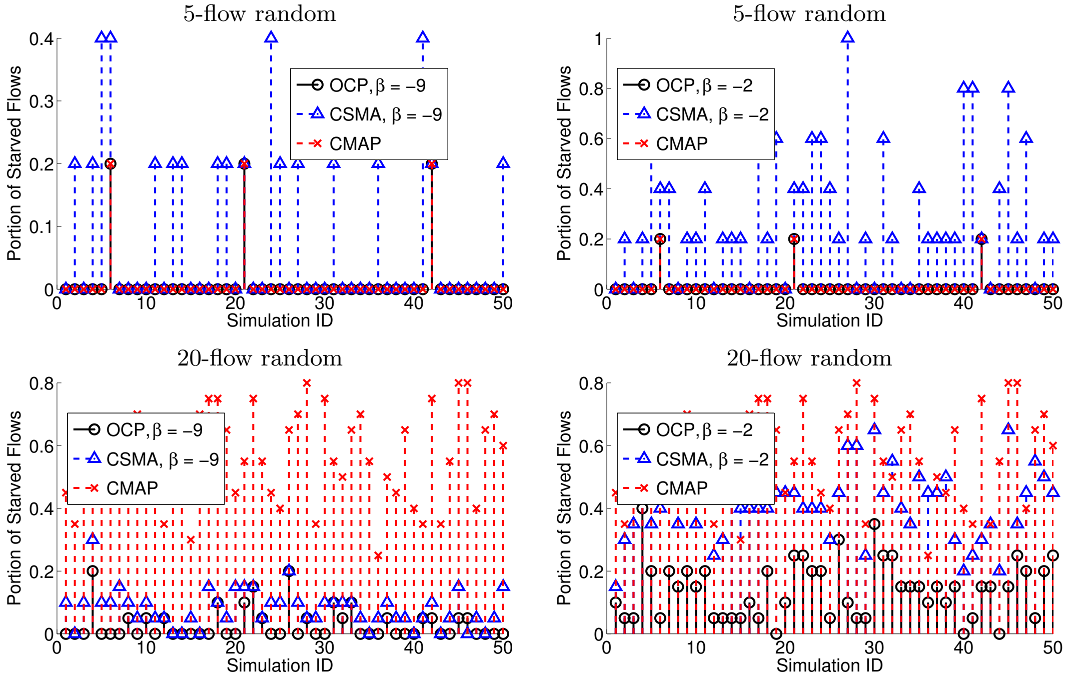


Figure 5.33: Portion of flows (number of starved flows / total number of flows) that are starved for 5-flow and 20-flow random topologies at β = -9 and -2

5.5 Summary

Since wireless medium is a shared resource, how to control the medium access scheme to reduce interference and increase spatial reuse is a crucial topic in wireless networks. In this chapter, we have presented OCP for each sender to opportunistically access the wireless medium. OCP is based on the rationale that the past interference information could be used as an indicator for future packet delivery outcome. An OCP-enabled sender accesses the medium only when it is confident that the channel access will likely to succeed and cause no collision to other flows. We propose a novel (CSID,Receiver)-SR mapping to allow for interference inference at the sender side even under high network contention. Further, we have shown that such medium access scheme needs to be done *opportunistically.* An OCP node must fall back to carrier sensing when there is no information overheard in the air, or there will be significant throughput degradation. In OCP, each receiver only needs to focus on correctly decoding the packet since OCP is a purely sender-side interference inference medium access scheme. Compared with CMAP and CSMA, we have shown that OCP significantly improves the throughput, packet delivery success ratio, and alleviates starvation in various random topologies with different contention levels.

Chapter 6

Discussion and Future Work

In this chapter, we discuss issues such as how to integrate SELECT, RAF, and OCP as a unified system, how mobility may affect the predictability, energy consumptions of each of the proposed schemes, and other future research directions.

6.1 Integration of SELECT, RAF, and OCP

As we have proposed SELECT, RAF, and OCP to tackle collision avoidance, rate adaptation, and spatial reuse, a question one might ask is how to integrate all of these schemes into a unified system? In Chapter 4, we have already proposed the system that couples RAF receiver with SELECT sender. The question is, how can we incorporate OCP into the existing framework.

SELECT essentially builds an RSS-SR mapping for the RSS values that are less than the carrier sense threshold. OCP goes one step further by indexing the channel status using CSID and building the mapping between CSID and the corresponding channel access success ratio. As we have seen that OCP sender needs to fall back to carrier sense, or there will be significant throughput degradation due to excess collisions. However, when falling back to carrier sensing, it is possible that RSS falls below the carrier sense threshold. One way to combine OCP and SELECT is to build a secondary RSS-SR mapping at the sender side when an OCP-enabled sender falls back to carrier sensing. When an OCP sender does not overhear any on-going flow information, it essentially becomes a SELECT sender. It tests if the RSS value is below the carrier sense threshold. If so, it accesses the channel only if the RSS-SR mapping indicates a high success ratio. If the RSS value is above the carrier sense threshold, the sender simply assumes the channel is busy. After integrating SELECT and OCP, incorporating RAF becomes straightforward, since SELECT and OCP are both sender-based schemes and RAF is a receiver-based scheme. The sender enabled with SELECT and OCP just needs to follow the channel rate and frame size suggested by the receiver.

6.2 Adaptation with the Presence of Mobility

The predictability of SELECT, RAF, and OCP is based on the assumption that environment does not change fast. For example, nodes may move, but only at the walking speed (2m/sec). As the mobility of the nodes in the network increases, the predictability decreases. Whether we can utilize the learning-based approach for nodes with fast moving speed is an interesting problem and we leave it as the future work.

6.3 Energy Consumption

In this section, we qualitatively analyze the energy consumption of SELECT, RAF, and OCP. SELECT's design principle is for collision avoidance, and indeed, we have shown its efficacy in improving packet delivery success ratio. A SELECT-enabled sender spends less time in retransmissions due to collisions. However, more energy is spent on maintaining the RSS-SR mapping and computation. For RAF, more energy will be spent on computing the desired channel rate and frame size. However, as shown in Chapter 4, we are able to efficiently compute the channel rate and frame size, which reduces the energy consumption compared with a brute-force search. For OCP, we note that it favors improving throughput rather than success ratio. So, OCP-enabled senders may not always save the energy on transmission. How to reduce the energy consumption on OCP, RAF, and SELECT can be a future research direction.

6.4 Other Potential Future Research Directions

- Receiver-based MAC protocols differ from sender-based MAC protocols by letting the receiver pull the packets from the sender, rather than for the sender to push the packets towards the receiver. The benefit of receiver-based MAC is that the receiver has more channel condition information to decide whether to initiate the transmission or not. When receiver-based MAC comes into play, however, hidden terminal problems still exist. For example, the receiver B of flow A→B in Figure 3.1 does not know whether there is another receiver D near its sender A and the transmission from A may interfere the reception at D. Whether and how learning-based approach can be applied to such a protocol is an exciting and challenging problem.

- One drawback of RAF is that the receiver does not know exactly when the sender will contend for the channel in current 802.11 MAC. As a result, the receiver has to estimate the expected backoff value at the sender. One way to address such a problem is for the sender to predetermine the up-coming backoff value, calculate the statistics such as mean-time-to-channel-access, embed the information in the data message, and send it to the receiver. When the receiver receives the data, it can utilize the information to decide the data rate and frame size more accurately.

- The idea of RAF can also be applied to WiMax or other TDMA-based MAC protocols. We expect that the actual performance of RAF applied to TDMA-based MAC protocols will be even better than CSMA-based protocols due to the removal of random channel access mechanism. How to apply this to existing TDMA-based protocols is an interesting future work.

Chapter 7

Conclusion

As wireless devices operated in the unlicensed frequency bands proliferate, how to effectively and efficiently control the medium access scheme to reduce interference is a crucial topic in wireless networks. However, the majority of existing medium access control schemes are interference oblivious. They often lead the system to a state where all interfered wireless transceivers operate at low data rate, suffer from severe packet collisions, achieve low throughput, and waste energy in unsuccessful packet transmissions, while the overall contention for the wireless channel stays high.

Interference mitigation in wireless networks is complex and the correct perception of the channel availability and operating parameters is affected by a large number of factors. Many factors are dynamic in either temporal or spatial domains or both, and very difficult to model analytically or appraise at packet-level fine time granularity. Yet effective and efficient interference mitigation is a basic requirement for a wireless network.

In this dissertation, we have discussed various learning-based interference-aware schemes to avoid packet collisions, improve channel rate adaptation, and increase spatial reuse in wireless networks. More specifically, in Chapter 3 we propose SELECT, an effective and efficient self-learning collision avoidance to tackle the long-haunting hidden/exposed receiver problem. In the SELECT design, a sender conforms to carrier sensing as in conventional CSMA based schemes. When the receive signal strength (RSS) falls below the carrier sense threshold, however, SELECT does not regard the channel as idle. Instead, it carefully selects the transmission timing that corresponds to high packet delivery success ratio. To further incorporate SELECT into existing CSMA framework, backoff timer is suspended when the channel is predicted busy even if the RSS value is below the carrier sense threshold.

In Chapter 4, we present rate adaptive framing (RAF), a joint channel rate and frame size control protocol that addresses both interference and noise. The design of RAF leverages the patterns of interference, as a result of the spatial and temporal correlations of wireless traffic. From the learned interference patterns, a RAF-enabled receiver derives the desired channel rate and frame size. An

RAF transmitter obtains such a configuration from the ACK message, and applies it to the next transmission. Through simulations we have shown that RAF outperforms ARF, RBAR, and OAR in throughput under various levels of interference and traffic patterns.

In Chapter 5, we presented OCP for each sender to opportunistically access the wireless medium. An OCP-enabled sender accesses the medium only when it is confident that the channel access will likely succeed and cause no collision to other flows. Different from SELECT which relies on RSS-SR mapping only for RSS values less than the carrier sense threshold, each OCP-enabled sender maintains a (CSID,Receiver)-SR mapping to allow for interference inference even under high network contention. Further, senders exchange interference information with each other periodically. We also showed that such medium access scheme needs to be done *opportunistically* and must fall back to carrier sensing when there is no information overheard in the air. In OCP, each receiver only needs to focus on correctly decoding the packet. Compared with CMAP and CSMA, we have shown that OCP significantly improves the throughput and packet delivery success ratio, and alleviates starvation in various random topologies with different contention levels.

The design principle of the learning-based schemes in this dissertation is based on the observation that although wireless networks are usually complex and dynamic, information can still be extracted from the data measured in the past. By learning from what was observed in the past, we can extract useful information, select the desired operational parameters, and react intelligently, while achieving substantial performance gain. It is our belief that this methodology could be applied to other open challenging problems in networking and systems research.

References

[1] CC1000 single chip very low power RF transceiver. http://www.chipcon.com.

[2] MIT roofnet. http://www.pdos.lcs.mit.edu/roofnet/.

[3] Network simulator. `http://www.isi.edu/nsnam/ns/`.

[4] ONOE. `http://madwifi.org/browser/trunk/ath_rate/onoe`.

[5] Place Lab. http://www.placelab.org/.

[6] TinyOS. http://www.tinyos.net/.

[7] WiFi Maps. http://www.wifimaps.com/.

[8] Daniel Aguayo, John Bicket, Sanjit Biswas, Glenn Judd, and Robert Morris. Link-level measurements from an 802.11b mesh network. In *Proceedings of ACM SIGCOMM*, 2004.

[9] Aditya Akella, Glenn Judd, Peter Steenkiste, and Srinivasan Seshan. Self management in chaotic wireless deployments. In *Proceedings of ACM MobiCom*, 2005.

[10] Paramvir Bahl, Ranveer Chandra, and John Dunagan. SSCH: Slotted seeded channel hopping for capacity improvement in IEEE 802.11 ad-hoc wireless networks. In *Proceedings of ACM MobiCom*, 2004.

[11] Sorav Bansal, Rajeev Shoreyy, and Arzad Kherani. Performance of tcp and udp protocols in multi-hop multi-rate wireless networks. In *Prof. of IEEE WCNC*, 2004.

[12] Lichun Bao and J.J. Garcia-Luna-Aceves. A new approach to channel access scheduling for ad hoc networks. In *Proceedings of ACM MobiCom*, 2001.

[13] Lichun Bao and J.J. Garcia-Luna-Aceves. Hybrid channel access scheduling in ad hoc networks. In *Proceedings of IEEE ICNP*, 2002.

[14] Dan Berger, Zhenqiang Ye, Prasun Sinha, Srikanth Krishnamurthy, Michalis Faloutsos, and Satish K. Tripathi. TCP-friendly medium access control for ad-hoc wireless networks: Alleviating self-contention. In *Proceedings of IEEE MASS*, 2004.

[15] Vaduvur Bharghavan. Performance evaluation of algorithms for wireless medium access. In *Proceedings of IEEE Performance and Dependability Symposium*, 1998.

[16] Vaduvur Bharghavan, Alan Demers, Scott Shenker, and Lixia Zhang. MACAW: A medium access protocol for wireless LANs. In *Proceedings of ACM SIGCOMM*, 1994.

[17] John C. Bicket. Bit-rate selection in wireless networks. Master's thesis, Department of EECS, MIT, February 2005.

[18] An Chan and Soung Chang Liew. Merit of phy-mac cross-layer carrier sensing: A mac-address-based physical carrier sensing scheme for solving hidden-node and exposed-node problems in large-scale wi-fi networks. *LCN*, 2006.

[19] Chuncheng Chen and Haiyun Luo. The case for heterogeneous wireless MACs. In *Proceedings of HotNets*, 2005.

[20] Chuncheng Chen, Haiyun Luo, Eunsoo Seo, Nitin Vaidya, and Xudong Wang. Rate-adaptive framing for interfered wireless networks. In *Proceedings of IEEE INFOCOM*, 2007.

[21] Sunwoogn Choi, Kihong Park, and Chong kwon Kim. On the performance characteristics of wlans: Revisited. In *Proceedings of ACM SIGMETRICS*, 2005.

[22] Romit Roy Choudhury, Xue Yang, Ram Ramanathan, and Nitin Vaidya. Medium access control in ad hoc networks using directional antennas. In *Proceedings of ACM MobiCom*, 2002.

[23] Javier del Prado Pavon and Sunghyun Choi. Link adaptation strategy for IEEE 802.11 WLAN via received signal strength measurement. In *Proceedings of ICC*, 2003.

[24] Zhenghua Fu, Petros Zerfos, Haiyun Luo, Songwu Lu, Lixia Zhang, and Mario Gerla. The impact of multihop wireless channel on TCP throughput and loss. In *Proceedings of IEEE INFOCOM*, 2003.

[25] Jason A. Fuemmeler, Nitin H. Vaidya, and Venugopal V. Veeravalli. Selecting transmit powers and carrier sense thresholds in csma protocols for wireless ad hoc networks. In *Proc. of WICON06*, 2006.

[26] Chane L. Fullmer and J.J. Garcia-Luna-Aceves. Solutions to hidden terminal problems in wireless networks. In *Proceedings of ACM SIGCOMM*, 1997.

[27] Violeta Gambiroza, Bahareh Sadeghi, and Edward W. Knightly. End-to-end performance and fairness in multihop wireless backhaul networks. In *Proceedings of ACM MobiCom*, 2004.

[28] J.J. Garcia-Luna-Aceves and Asimakis Tzamaloukas. Receiver-initiated collision-avoidance in wireless networks. *ACM Wireless Networks, Special Issue on Selected Papers from Mobicom 99*, 8(2/3):249–263, 2002.

[29] M. Garetto, J. Shi, and E. Knightly. Modeling media access in embedded two-flow topologies of multi-hop wireless networks. In *Proceedings of ACM MobiCom*, 2005.

[30] Ivaylo Haratcherev, Koen Langendoen, Reginald Lagendijk, and Henk Sips. Hybrid rate control for IEEE 802.11. In *Proceedings of ACM MobiWac*, 2004.

[31] Gavin Holland, Nitin Vaidya, and Paramvir Bahl. A rate-adaptive MAC protocol for multi-hop wireless networks. In *Proceedings of ACM MobiCom*, 2001.

[32] Chunyu Hu and Jennifer C. Hou. A reactive channel model for expediting wireless network simulation. In *ACM SIGMETRICS Poster*, 2005.

[33] IEEE Computer Society. Wireless LAN medium access control (MAC) and physical layer (PHY) specifications. IEEE standard 802.11, 1999.

[34] Kamal Jain, Jitendra Padhye, Venkat Padmanabhan, and Lili Qiu. Impact of interference on multi-hop wireless network performance. In *Proceedings of ACM MobiCom*, 2003.

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

[36] Kyle Jamieson and Hari Balakrishnan. PPR: Partial Packet Recovery for Wireless Networks. In *ACM SIGCOMM*, 2007.

[37] Kyle Jamieson, Bret Hull, Allen K. Miu, and Hari Balakrishnan. Understanding the real-world performance of carrier sense. In *Proceedings of ACM SIGCOMM E-WIND Workshop*, 2005.

[38] Amit P. Jardosh, Krishna N. Ramachandran, Kevin C. Almeroth, and Elizabeth M. Belding-Royer. Understanding link-layer behavior in highly congested ieee 802.11b wireless networks. In *Proceedings of ACM SIGCOMM E-WIND Workshop*, 2005.

[39] G. Judd and P. Steenkiste. Using emulation to understand and improve wireless networks and applications. In *Proceedings of NSDI*, 2005.

[40] Ad Kamerman and Leo Monteban. WaveLAN-II: a high-performance wireless LAN for the unlicensed band. *Bell Labs Technical Journal*, 2(3):118–133, August 1997.

[41] Phil Karn. MACA: A new channel access method for packet radio. In *Proceedings of IEEE Computer Network Conference*, 1990.

[42] Jongseok Kim, Seongkwan Kim, Sunghyun Choi, and Daji Qiao. CARA: Collision-aware rate adaptation for IEEE 802.11 WLANs. In *Proceedings of IEEE INFOCOM*, 2006.

[43] Tae-Suk Kim, Hyuk Lim, and Jennifer C. Hou. Improving spatial reuse through tuning transmit power, carrier sense threshold, and data rate in multihop wireless networks. In *Proc. of MobiCom06*, 2006.

[44] Youngsoo Kim, Sunghyun Choi, Kyunghun Jang, and Hyosun Hwang. Throughput enhancement of IEEE 802.11 WLAN via frame aggregation. In *Proceedings of IEEE VTC*, 2004.

[45] Andrzej Kochut, Arunchandar Vasan, A. Udaya Shankar, and Ashok Agrawala. Sniffing out the correct physical layer capture model in 802.11b. In *ICNP '04*.

[46] David Kotz and Kobby Essien. Analysis of a campus-wide wireless network. In *Proceedings of ACM MobiCom*, 2002.

[47] Mathieu Lacage, Mohammad Hossein Manshaei, and Thierry Turletti. IEEE 802.11 rate adaptation: A practical approach. In *Proceedings of ACM MSWiM*, 2004.

[48] Zhifei Li, Sukumar Nandi, and Anil Gupta. Improving mac performance in wireless ad-hoc networks using enhanced carrier sensing (ecs). In *In Proc. of IFIP NETWORKING*, 2004.

[49] Nyein Aye Maung Maung, Taku Noguchi, and Makoto Kawai. Maximizing aggregate throughput of wireless ad hoc networks using enhanced physical carrier sensing. 2008.

[50] Xiaoqiao (George) Meng, Starsky H.Y. Wong, Yuan Yuanz, and Songwu Lu. Characterizing flows in large wireless data networks. In *Proceedings of ACM MobiCom*, 2004.

[51] Alaa Muqattash and Marwan Krunz. A single-channel solution for transmission power control in wireless ad hoc networks. In *Proc. of MobiHoc04*, 2004.

[52] Thyagarajan Nandagopal, Tae-Eun Kim, Xia Gao, and Vaduvur Bharghavan. Achieving MAC layer fairness in wireless packet networks. In *Proceedings of ACM MobiCom*, 2000.

[53] Daji Qiao and Sunghyun Choi. Goodput enhancement of IEEE 802.11a wireless LAN via link adaptation. In *Proceedings of IEEE ICC*, 2001.

[54] Daji Qiao and Sunghyun Choi. Fast-responsive link adaptation for IEEE 802.11 WLANs. In *Proceedings of IEEE ICC*, 2005.

[55] Bhaskaran Raman and Kameswari Chebrolu. Revisiting MAC design for an 802.11based mesh network. In *Proceedings of HotNets*, 2004.

[56] Ananth Rao and Ion Stoica. An overlay MAC layer for 802.11 networks. In *Proceedings of ACM MobiSys*, 2005.

[57] Saikat Ray, Jeffrey B. Carruthers, and David Starobinski. RTS/CTS-induced congestion in ad hoc wireless lans. In *Proceedings of IEEE WCNC*, 2003.

[58] Maya Rodrig, Charles Reis, Ratul Mahajan, David Wetherall, and John Zahorjan. Measurement-based characterization of 802.11 in a hotspot setting. In *Proceedings of ACM SIGCOMM E-WIND Workshop*, 2005.

[59] B. Sadeghi, V. Kanodia, A. Sabharwal, and E. Knightly. Opportunistic media access for multirate ad hoc networks. In *Proceedings of ACM MobiCom*, 2002.

[60] Aimin Sang and Sanqi Li. A predictability analysis of network traffic. In *Proceedings of IEEE INFOCOM*, 2000.

[61] Naveen Santhapuri, Justin Manweiler, Souvik Sen, Romit Roy Choudhury, Srihari Nelakuduti, and Kamesh Munagala. Message in message (mim): A case for reordering transmissions in wireless networks. In *Proceedings of HotNets*, 2008.

[62] Kimaya Sanzgiri, Ian D. Chakeres, and Elizabeth M. Belding-Royer. Determining intra-flow contention along multihop paths in wireless networks. In *Proceedings of Broadnets Wireless Networking Symposium*, 2004.

[63] Jungmin So and Nitin H. Vaidya. Multi-channel MAC for ad hoc networks: Handling multichannel hidden terminals using a single transceiver. In *Proceedings of ACM MobiHoc*, 2004.

[64] Karthikeyan Sundaresan and Raghupathy Sivakumar. A unified MAC layer framework for ad-hoc networks with smart antennas. In *Proceedings of ACM MobiHoc*, 2004.

[65] Mineo Takai, Jay Martin, Aifeng Ren, and Rajive Bagrodia. Directional virtual carrier sensing for directional antennas in mobile ad hoc networks. In *Proceedings of ACM MobiHoc*, 2002.

[66] Fabrizio Talucci, Mario Gerla, and Luigi Fratta. MACA-BI (MACA by invitation) a receiver oriented access protocol for wireless multihop networks. In *Proceedings of IEEE PIMRC*, 1997.

[67] Mythili Vutukuru, Kyle Jamieson, and Hari Balakrishnan. Harnessing Exposed Terminals in Wireless Networks. In *Proceedings of NSDI*, 2008.

[68] Zhibin Wu, Sachin Ganu, Ivan Seskar, and D. Raychaudhuri. Experimental investigation of PHY layer rate control and frequency selection in 802.11-based ad-hoc networks. In *Proceedings of ACM SIGCOMM E-WIND Workshop*, 2005.

[69] Kaixin Xu, Mario Gerla, and Sang Bae. Effectiveness of rts/cts handshake in ieee 802.11 based adhoc networks. *Ad Hoc Network Journal*, 1:107–123, 2003.

[70] Kaixin Xu, Mario Gerla, Lantao Qi, and Yantai Shu. TCP unfairness in ad hoc wireless networks and a neighborhood RED solution. In *Proceedings of ACM MobiCom*, 2003.

[71] Shugong Xu and Tarek Saadawi. Does the IEEE 802.11 MAC protocol work well in multihop wireless ad hoc networks? *IEEE Communication Magazine*, 39(6):130–137, June 2001.

[72] Xue Yang and Nitin H. Vaidya. On the physical carrier sense in wireless ad-hoc networks. In *Proceedings of IEEE INFOCOM*, 2005.

[73] Xue Yang and Nitin H. Vaidya. Spatial backoff contention resolution for wireless networks. In *Proceedings of IEEE WiMesh*, 2006.

[74] Yaling Yang and Robin Kravets. Contention-aware admission control for ad hoc networks. Technical Report UIUC-DCS-R-2003-2337, UIUC, May 2004.

[75] Wei Ye, John Heidemann, and Deborah Estrin. An energy-efficient MAC protocol for wireless sensor networks. In *Proceedings of IEEE INFOCOM*, 2002.

[76] Gang Zhou, Tian He, Sudha Krishnamurthy, and John A. Stankovic. Impact of radio irregularity on wireless sensor networks. In *Proceedings of ACM MobiSys*, 2004.

[77] Jing Zhu, Xingang Guo, L. Lily Yang, and W. Steven Conner. Leveraging spatial reuse in 802.11 mesh networks with enhanced physical carrier sensing. In *Proceedings of IEEE ICC*, 2004.

[78] Jing Zhu, Xingang Guo, L. Lily Yang, W. Steven Conner, Sumit Roy, and Mousumi M. Hazra. Adapting physical carrier sensing to maximize spatial reuse in 802.11 mesh networks: Research articles. *Wirel. Commun. Mob. Comput.*, 4(8):933–946, 2004.

Author's Biography

Chun-cheng Chen was born in Hualien, Taiwan. He received the B.S. degree in Civil and Environmental Engineering from National Taiwan University in 2000, M.S. degree in Civil and Environmental Engineering from the University of California at Berkeley in 2002, and M.S. degree in Computer Science from the University of Illinois at Urbana-Champaign in 2005. He is the recipient of the Vodafone-U.S. Foundation Fellowship for the years 2006$\sim$2007, 2007$\sim$2008 and awarded the Verizon Foundation Scholarship in 2007$\sim$2008.