OPPORTUNISTIC LINK SCHEDULING FOR MULTIHOP AD HOC WIRELESS NETWORKS

OPPORTUNISTIC LINK SCHEDULING FOR MULTIHOP AD HOC WIRELESS NETWORKS

BY

MIN SHEN, B.ENG., M.ENG.

WUHAN UNIVERSITY, PEKING UNIVERSITY, P.R.CHINA

JANUARY 2008

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

AND THE COMMITTEE ON GRADUATE STUDIES

OF MCMASTER UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTERS OF APPLIED SCIENCE

MASTER OF APPLIED SCIENCE (2008) McMaster University

(Electrical and Computer Engineering) Hamilton, Ontario

TITLE: Opportunistic Link Scheduling for Multihop Ad Hoc Wireless Networks

AUTHOR: Min Shen, B.Eng., M.Eng.

Wuhan University, Peking University, P.R.China

SUPERVISOR: Dr. Dongmei Zhao, Assistant Professor, Ph.D

NUMBER OF PAGES: 64,

Abstract

This thesis studies throughput improvement for TCP traffic in IEEE 802.11-based multihop ad hoc wireless networks. Due to the incompatibility between TCP and the IEEE 802.11 distributed coordination function (DCF) protocols, the reaction of TCP in case of packet losses can significantly reduce TCP end-to-end throughput. In this thesis, we propose an opportunistic link scheduling (OLS), which is a simple enhancement to the IEEE 802.11 DCF protocol and intends to improve the compatibility between TCP and MAC layer protocols in multihop ad hoc networks. With OLS, a link with a good channel condition is allowed to transmit multiple packets consecutively as a burst, while the burst size depends on both physical channel fading and MAC layer collisions. The protocol also includes a mechanism to prevent starvation of nodes with poor channel conditions. An analytical model is developed for a four-hop chain to study the effect of the burst size and TCP congestion window size on the end-to-end transmission throughput in opportunistic link scheduling. Our results show that OLS can significantly improve the end-to-end transmission throughput, while keeping reasonably low transmission delay. The protocol is easy to implement, and requires slight modifications to the IEEE 802.11 protocol.

Acknowledgments

First of all, I would like to thank my supervisor, Dr. Dongmei Zhao, for her valuable assistance and help during the courses and research program. Dr. Zhao teaches me how to start and go deep with a research topic. I really appreciate her supervision and depth of knowledge.

I would also like to express my sincere thanks to Dr. Terence D. Todd and Dr. Polychronis Koutsakis. Thank them for the insightful advice on the improvement of the research project. I should also thank all my graduate colleagues, Jun Zou, Bin Wang, Rafal, Mohammad Smadi, Vahid Azhari, Amir Sayegh, Ahmed Kholaif, Dr. Sasti, Ghada and Lu Qiao, who are very supportive and provide a nice and competitive research environment.

Finally, I thank my parents, who give me courage and love to finish the research program. They support me whenever I encounter difficulties. I should share with them my research progress and all the happiness I pursuit.

Contents

List of Tables

List of Figures

List of Abbreviation

MAC Medium Access Control

UDP User Datagram Protocol

CW Contention Window

TCP Transmission Control Protocol

RTS Request to Send

CTS Clear To Send

NAV Networking Allocation Vector

ACK Acknowledgement

OAR Opportunistic Auto Rate

OLS Opportunistic Link Scheduling

DCF Distributed Coordination Function

CSMA/CA Carrier Sense Multiple Access with Collision Avoidance

DIFS DCF Interframe Space

SIFS Short Interframe Space

SINR Signal-to-Interference-plus-Noise Ratio

WUB	Window Size Upper Bound

Chapter 1

Introduction

1.1 Overview

Multihop wireless ad hoc networks based on IEEE 802.11 are becoming increasingly popular for providing extended Wi-Fi coverage. Capacity degradation resulting from co-channel and external interference in such networks becomes an important and interesting research problem. Unstable radio channel propagation conditions can also introduce packet losses. The end-to-end throughput in a multihop ad hoc network can be degraded dramatically with the increase of number of hops due to erroneous transmissions. This situation is true even for a simple chain topology [MPK05b][MK04][ZFG03]. When a packet is unsuccessfully transmitted, it is retransmitted at the link layer. In the 802.11 medium access control (MAC) protocol, a retransmission involves a longer backoff time and a lower priority in accessing the channel, compared to a new transmission. A packet is dropped by a link after a certain number of retransmissions. For TCP traffic, the source node will eventually timeout and retransmit the packet at the transport layer. For an H-hop path, an end-to-end retransmission can be triggered by a dropped packet at hop H (i.e., the last hop), even if all the $H-1$ upstream hops have transmitted successfully. An analytical model is developed in [IH06] for analyzing the number of retransmissions in an IEEE 802.11-based multihop network. Performance of TCP traffic in IEEE 802.11-based multihop networks has been studied analytically in [TIA06] and [TIA05] for batch transmission. In addition, TCP regards packet losses as a sign of network congestion and unnecessarily reduces its congestion window size, which decreases the end-to-end throughput furthermore. Therefore, reducing link layer transmission failures is important for reducing the number of retransmissions and improving the end-to-end packet transmission throughput in a multihop ad hoc network.

One of the unique problems in multihop transmissions is that a downstream link cannot transmit a packet until the packet has been successfully forwarded through all upstream links. The TCP window size limits the number of outstanding packets along an end-to-end path. For example, in an H-hop path, the throughput is at most $1/H$ of the link capacity when the TCP window size is 1, since there is at most one outstanding packet along the end-to-end path at any given time, and therefore at most one link transmitting. Once a node has received the packet, it is the time for it to forward the packet to the next hop, regardless of its channel condition. This is inefficient when the link is in poor channel condition. Meanwhile, all other links have to wait, even though they may be all in perfect channel conditions. A new packet cannot be transmitted until the previous packet has successfully reached the destination. Having a larger TCP window size allows more buffered packets, and potentially more nodes competing for channel access. If nodes with better channel conditions can have a higher priority to access the channel, the shared channel can be better utilized without increasing transmission collisions. This is the basic idea of the opportunistic link scheduling protocol proposed in this thesis.

1.2 Related Research Work

There has been some work in the literature on improving end-to-end throughput in IEEE 802.11-based multihop transmissions. Unnecessary RTS/data frames can be transmitted due to that a receiver does not respond to an RTS or data frame when it is busy in transmitting/receiving or prohibited from transmitting by networking allocation vectors (NAVs) set by surrounding nodes. This is the receiver blocking problem pointed out in [HZF06], where a new MAC protocol is proposed to solve this problem by using an out-of-band busy tone and two communication channels for control frames and data frames, respectively. Controlling the offered traffic load at the sources is another approach to reducing collisions and packet losses in a multihop network as shown in [NL05], where a quantitative analysis is provided to estimate the optimal offered load that maximizes the multihop transmission throughput. The approach used in [HZ06] is to give a higher priority to a node in accessing the channel as soon as it receives a packet. The scheme also uses a backward-pressure congestion control that gives the transmission opportunity to a congested node while keeping its upstream nodes from transmissions. The approach proposed in [MPK] is to tune the backoff mechanism of 802.11 in order to efficiently utilize the radio resources and improve the throughput. Besides collisions due to co-channel interference, channel fading is another important reason causing erroneous packet transmissions. As it is shown in [MPK05a], moving objects, such as people, in the propagation path can introduce significant variation in the received signal strength. This can happen even if both the transmitter and the receiver nodes are fixed. Approaches that can reduce the hidden terminal effects may not solve the transmission errors due to channel fading. Some research work, e.g., [SBG56], suggests to distinguish packet losses due to channel impairment from that due to collisions and treat each type differently. For TCP traffic, the transport layer ACKs are transmitted as data packets at the link layer. A dynamic approach is proposed in [dOB05] to minimize the number of transport layer ACKs in order to reduce the link layer traffic load and collisions and improve the TCP throughput.

Controlling the traffic arrival rate from the source, limiting TCP congestion window size upper bound, and offering downstream nodes a higher priority for forwarding packets can all help limit the number of concurrent transmission links so that to alleviate the interference and enhance the end-to-end throughput in a network without channel fading. The opportunistic link scheduling proposed in this thesis adopts a different idea, which allows more packets buffered in intermediate nodes along a multihop flow, so that the nodes can make use of the good channel conditions at both the physical and MAC layers opportunistically and transmit more packets successfully without retransmissions.

Opportunistic scheduling schemes have been proposed in [BSK02] and [JWF04], but for different purposes. The opportunistic packet scheduling and MAC protocol considered in [JWF04] is specifically designed for the scenario where a node has packets to different destinations. The protocol allows a packet with good channel quality to be transmitted without being blocked by other packets which arrive earlier than the considered packet but go to different destinations. The opportunistic auto rate (OAR) protocol proposed in [BSK02] allows multiple back-to-back data packets to be transmitted in a link. This is combined with the multi-rate feature in 802.11. When the channel conditions allow a link to transmit above the base rate, OAR grants channel access for multiple packet transmissions in proportion to the ratio of the achievable data rate over the base rate. The scheme is specifically designed for single-hop multi-rate ad hoc networks. The opportunistic link scheduling protocol in this thesis, although designed for multihop networks, can also be applied to single hop ad hoc networks, and is not restricted to a specific physical transmission rate.

1.3 Motivation and Overview of the Proposed Work

In this thesis, we propose an opportunistic link scheduling (OLS) protocol for improving the end-to-end transmission throughput in IEEE 802.11-based multihop networks. The basic idea is to take advantage of the channel condition changes by giving a higher priority to the links with good channel conditions and allowing the links to transmit multiple consecutive packets successfully. The channel condition includes both physical channel fading and MAC layer transmission collisions. An indication of good channel condition is a successfully received CTS or ACK frame. In this case, the transmitter can keep holding the channel and transmitting multiple packets without repeatedly contending for channel access. This mechanism provides links in good channel conditions with a better chance to access the channel. It further reduces packet loss probability at the link layer, decreases the number of retransmissions at both the link and transport layers, and reduces the amount of idle channel time that is otherwise necessary for nodes to backoff before retransmitting packets. Therefore, higher packet transmission throughput is expected using the OLS protocol. OLS does not differentiate packet losses due to channel impairment and due to network congestion. It simplifies the implementation. By transmitting multiple back-to-back packets at the link layer, OLS allows packets to be buffered at intermediate nodes. This provides the nodes with a chance to take advantage of the link condition changes and transmit packets when their channel conditions become good. Meanwhile, OLS limits the number of back-to-back packets transmitted in each link in order to avoid starving some links with poor channel conditions and prevent buffer overflow. In this way, OLS can hide some effects of poor channel conditions at both the physical and link layers from the transport layer and improve the compatibility between the MAC and TCP layer protocols.

1.4 Organization of Thesis

The remainder of this thesis is organized as follows. In Chapter 2, the IEEE 802.11 DCF and effect of the TCP congestion window size on end-to-end throughput in multihop wireless networks are introduced. In Chapter 3, the proposed OLS protocol is presented. An analytical model is then developed in Chapter 4 to study the effect of burst size and TCP window size on end-to-end performance in multihop transmissions with opportunistic link scheduling. In Chapter 5, we evaluate the end-to-end transmission performance using OLS and compare it with the IEEE 802.11 DCF by extensive simulation tests. Finally, Chapter 6 draws conclusions and summarizes possible directions for future work.

Chapter 2

Background

In this chapter, an overview of IEEE 802.11 DCF protocol is first given, followed by a discussion about the incompatibility issue between TCP and 802.11 DCF layer protocols.

2.1 IEEE 802.11 DCF

Multihop wireless networks based on the IEEE 802.11 are becoming increasingly popular for providing extended Wi-Fi coverage. These networks can distribute wide bandwidth with minimal wired Internet connections and leased back-haul links. The IEEE 802.11 DCF is a kind of CSMA/CA (carrier sense multiple access with collision avoidance) MAC protocols. It has been widely adopted in wireless multihop ad hoc networks due to its simple implementation and distributed nature. Basically, the CSMA/CA of the DCF works as a listen-before-talk scheme. Before sending a packet, the transmitter senses the channel first. If the channel is idle, the transmitter defers one DCF interframe space (DIFS). If the channel is still idle at that time, the transmitter sends one packet. If the channel is not idle, the transmitter starts a backoff process by selecting a random backoff count randomly selected between $[0, CW-1]$, where CW is the contention window. When the counter reaches zero, one packet is transmitted. After receiving one packet successfully, the receiver waits for one short interframe space (SIFS) and sends ACK back to the transmitter. When the packet is determined not to be received correctly at the receiver, i.e. the transmitter does not receive ACK in time, the contention window is exponentially increased until the maximum value is reached and the backoff counter is randomly selected from $[0, CW-1]$ and so on.

There are two kinds of carrier sensing in DCF: physical carrier sensing and virtual carrier sensing. By physical carrier sensing, the channel is regarded to be busy when the signal strength sensed is above a certain threshold. By virtual carrier sensing, the channel is considered to be busy during the time period indicated by a network allocation vector (NAV) in the request-to-send/clear-to-send (RTS/CTS) frames. The mechanism is illustrated in Figure 2.1 [pap99a].

Before transmitting a data frame, a node transmits an RTS frame, and the receiver replies with a CTS frame. If the CTS frame is not received within a predefined time interval, the RTS frame is retransmitted by performing a backoff algorithm. The RTS and CTS frames include the information of how long time it will take to transmit the subsequent data frame and the corresponding ACK response. Thus, mobile nodes, that either hear the RTS from the transmitter or receive CTS from the receiver will not start any transmissions; instead, they set NAVs to indicate the time period

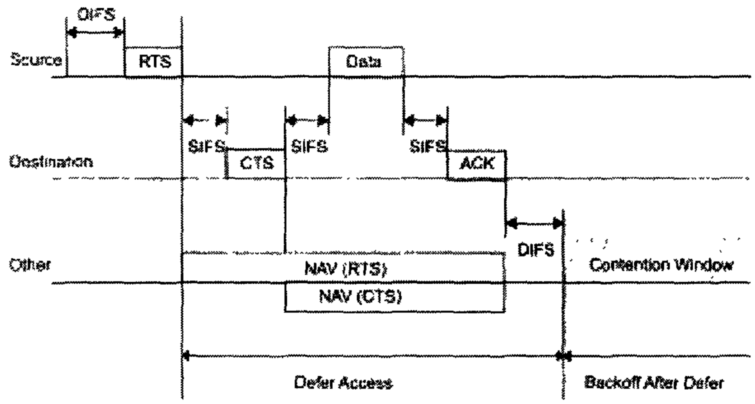

Figure 2.1: IEEE 802.11 virtual carrier sensing mechanism

reserved. Using NAVs ensures that operations between the transmitter and receiver are not interrupted. Between two consecutive frames in the sequence of RTS, CTS, data, and ACK frames, an SIFS is used. The SIFS is shorter than the DIFS value so that other mobile stations will not collide with ongoing transmissions.

2.2 Throughput of TCP Traffic

The DCF protocol works well under low traffic load condition, but suffers from significant throughput degradation in high traffic load conditions [KW01]. When a transmission fails because of either bad channel condition (e.g. fading) or interference from hidden terminals, retransmission will occur and induce more backoff delay. When the number of retransmissions reaches a certain limit, the transport layer retransmissions will be called, which will introduce more delay and degrade the end-to-end throughput furthermore. This is especially serious for TCP traffic.

TCP is a widely used transport layer protocol that provides reliable end-to-end transmission in wireline networks. When used in wireless networks, TCP throughput degrades greatly. Unstable radio propagation channel, node mobility, and compatibility with MAC layer protocol can all introduce packet losses, which TCP regards as signals of network congestion and unnecessarily reduces its congestion window size. Improving TCP throughput performance over wireless networks has been an important research topic.

Recently, research has been done on the TCP performance in a chain topology by considering spatial reuse properties of the IEEE 802.11 MAC layer protocol. Incompatibility between TCP at the transport layer and the IEEE 802.11 MAC protocols can decrease the end-to-end packet transmission throughput [XP04] furthermore. In the 802.11-based multihop wireless networks, the number of concurrent transmission links is spatially limited. If an excessive number of neighboring links are transmitting simultaneously within a limited area, packet losses will occur due to co-channel interference (mainly from hidden terminal problems). The TCP congestion window size plays an important role in the end-to-end transmission throughput. When the network is congested, packet losses due to collisions increase. In this case, TCP reduces its congestion window in order to reduce the traffic load. By limiting the TCP congestion window size, the maximum number of outstanding packets along the path is limited. This further limits the number of links competing for channel access and reduces the collision rate. On the other hand, a small TCP congestion window size may prohibit too many links from transmitting and lead to low throughput. For a typical chain topology with H hops, it is found [ZFG03] that TCP throughput is maximized if the maximum size of the TCP congestion window is upper bounded by $H/4$ when there is no fading in the physical channel and the interference range of a node is slightly larger than twice the transmission range. This result becomes more accurate when h is larger because of the reduced boundary effect. A transmission is successful if and only if the receiver is within the transmission range of the transmitter and not within the interference range of any interferer. However, to keep the TCP window size at the optimum value for a multihop flow is quite challenging. In [ZFG03] it is found that the reaction of TCP to network congestion in IEEE 802.11 multihop networks is much delayed. The TCP window size keeps increasing to some value much larger than the optimum before network congestion can be detected at the transport layer. In order to help TCP maintain its congestion window in a certain range that results in a higher throughput, [ZFG03] proposed a scheme that drops buffered packets once early packet drops are detected at the link layer. On the other hand, if a transmission loss is due to temporary channel fading but not network congestion, the TCP congestion window size should not be reduced. Although the MAC layer can hide some transmission failures, e.g., by link layer retransmissions, once TCP detects the packet loss, it will consider this as a sign of network congestion and reduce its congestion window size, causing throughput degradation. Therefore, TCP window size adjustment at the transport layer can negatively affect the end-to-end throughput in IEEE 802.11 multihop networks.

In the next chapter, we will propose an opportunistic link scheduling scheme that can improve the compatibility between the 802.11 DCF MAC and the TCP protocols.

Chapter 3

Opportunistic Link Scheduling

In this chapter, the proposed opportunistic link scheduling (OLS) protocol is presented, which includes one algorithm for the transmitter and one for the receiver. The protocol requires a slight modification to the IEEE 802.11 ACK frame header.

3.1 Overview

We consider a multihop ad hoc network where all transmissions share the same frequency channel. Each node is equipped with a single radio, and cannot transmit and receive at the same time. We assume that a routing mechanism exists and the route is selected before packet transmissions. We assume that all nodes are relatively static, so that the connectivity of the network can be considered as fixed for a relatively long time. Each node, if it is eligible to transmit, first sends an RTS. In the original IEEE 802.11 DCF, a node is eligible to transmit if its buffer is not empty and it senses the channel to be idle. Upon successfully receiving the RTS, a node sends a CTS back to the transmitter. Both the RTS and CTS specify a network allocation vector (NAV) period. Any node overhearing the RTS and/or CTS will not transmit until the time period specified by the NAV is over. The sender then can transmit a data frame and the receiver replies an ACK frame without collision. Either the RTS or the CTS can be lost due to channel impairment or collision. According to IEEE 802.11 DCF, the sender will retransmit for up to 7 times if virtual carrier sensing is enabled and it does not receive the CTS after a SIFS time.

The basic approach in the proposed OLS protocol is to allow links to transmit multiple packets consecutively. The protocol specifies a maximum burst size, $N_{\max}$, that a node is allowed to transmit. The actual burst size is dynamically changed according to channel transmission conditions and other network conditions. The protocol also includes mechanisms to avoid buffer overflow and prevent starvation of links with poor link conditions. The basic algorithms used at a transmitter and a receiver are shown in **Algorithms** 1 and 2, respectively. In the algorithms, q_i represents the buffer occupancy of node i, and $0 \leq q_i \leq q_{\max}$, and $q_{\max}$ is the maximum number of packets that can be buffered in every intermediate node that a flow traverses. The value of q_i is increased by 1 when a data packet is received, and decreased by 1 after a data packet is successfully forwarded; Q_STATUS is a binary variable to indicate whether a particular node is able to receive more packets for a flow; and N_i denotes the number of back-to-back packets that node i has transmitted in the current burst.

3.2 Algorithm for Transmitter

Algorithm 1: OLS for transmitter i

Virtual carrier sensing based on IEEE 802.11 (if virtual carrier sensing is enabled)
Initialize $N_i = 0$ and MORE_PACKETS =1
while MORE_PACKETS = 1 **do**
    Transmit one data packet from buffer
    if ACK is received in time **then**
        $q_i = q_i - 1$ %Remove the packet from buffer
        $N_i = N_i + 1$
        if $q_i = 0$ **then**
            MORE_PACKETS = 0
        end if
        if $N_i = N_{\max}$ **then**
            MORE_PACKETS = 0
        end if
        if Q_STATUS = FULL **then**
            MORE_PACKETS = 0
        end if
    else
        MORE_PACKETS = 0
        Retransmit packet based on IEEE 802.11 DCF
    end if

21: **end while**

A node starts sensing the channel whenever its buffer is not empty (Line 1 in Algorithm 1). If virtual carrier sensing is enabled, it follows the standard IEEE 802.11 for transmitting the RTS frame. When the CTS frame is returned successfully, the sender transmits the first data packet (Line 4 in Algorithm 1), i.e., it waits for a SIFS period and transmits the first data packet. The node then waits for the ACK frame from the receiver. If the ACK is not received after a SIFS period, the transmitted packet is lost and the node follows the standard IEEE 802.11 DCF for retransmitting the packet (Line 19 in Algorithm 1). Otherwise, if the ACK frame is received in time, the sender considers the channel condition good, waits for a SIFS period and transmits the next data packet. A node can keep transmitting data packets until any of the following events occurs: i) no ACK is received in time, ii) there is no packet to be transmitted in its buffer (Lines 8-10 in Algorithm 1), iii) the total number of back-to-back packets that it has transmitted reaches $N_{\max}$ (Lines 11-13 in Algorithm 1), or iv) the receiver cannot accept any more packets (Lines 14-16 in Algorithm 1), which is indicated by a Q_STATUS bit in the ACK frame returned from the receiver and will be detailed later. When the virtual carrier sensing is disabled, the only change to Algorithm 1 is to remove Line 1.

Transmitting a burst of packets for links with good channel conditions can avoid unnecessary channel competitions and improve the channel utilization. However, transmitting too many back-to-back packets can easily result in the situation that the downstream links are overloaded and neighboring links with poorer channel conditions seldom have chances to transmit. In this case throughput cannot be improved. In order to prevent some links with good channel conditions from occupying the channel all the time, OLS specifies a maximum number of packets that a node can transmit consecutively in a burst, $N_{\max}$. Once the total number of back-to-back packets that a node has transmitted reaches $N_{\max}$, the node returns to follow standard 802.11 DCF (by setting MORE_PACKETS=1 in Algorithm 1). A larger value of $N_{\max}$ provides a better chance to make use of good channel conditions of the links, but may also result in more buffered packets. The effect of $N_{\max}$ on the end-to-end packet transmission performance will be examined by both analytical model (in chapter 4) and computer simulation (in chapter 4).

3.3 Algorithm for Receiver

Algorithm 2: OLS for receiver i

```
1: Upon successfully receiving RTS
2: Send CTS back
3: Upon successfully receiving a data frame
4: q_i = q_i + 1
5: if q_i = q_max then
6:     Q_STATUS = FULL
7: else
8:     Q_STATUS = NOT_FULL
9: end if
10: Send ACK back
```

The algorithm used at the receiver in OLS is given in Algorithm 2. Upon receiving a packet, a node checks its buffer occupancy (Lines 6-10 in Algorithm 2). It sets the value of the binary variable Q_ STATUS to FULL if it is unable to receive more packets (Line 7 in Algorithm 2). The value of Q_STATUS is included in the ACK frame. This is achieved by slightly modifying the duration field of the ACK frame in the original 802.11 as shown in Figure 3.1. The duration field carries the association identity of the node. The standard specifies 16 bits for the duration field, among which 14 bits are used and the remaining 2 bits are reserved. As shown in Figure 3.1, one of the reserved bits is used in the OLS protocol to represent Q_STATUS. By using this simple backward blocking mechanism, the OLS can not only prevent buffer overflow, but also limit the burst transmissions of links with good channel conditions and allow other links to transmit. For multihop transmissions, this can prevent the upstream hops from excessively injecting packets to the downstream links.

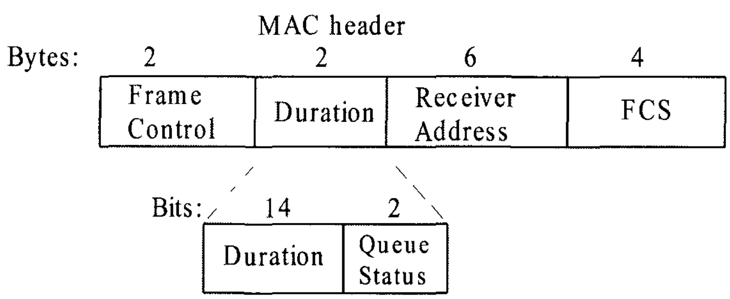

Figure 3.1: Modified ACK frame

3.4 Discussions

It is seen from the above description that the number of back-to-back packets that a node can transmit in a burst is not pre-specified, but determined by the transmission conditions. The variable burst size is achieved by dynamic NAV updates. In OLS, the initial RTS/CTS exchange, if enabled, and the first data/ACK frames in a burst transmission along a link, set NAVs for only one data packet transmission. This process is exactly the same as in DCF without burst transmissions. The NAV in OLS is then renewed for another packet transmission only if the ACK frame for the previous packet has been successfully received and the current burst size has not reached $N_{\max}$.

Burst transmissions may be allowed in IEEE 802.11 DCF, however there is a major difference between OLS and DCF in NAV setting. In standard IEEE 802.11 DCF, the NAV is pre-specified based on a pre-determined burst size, but does not take into consideration channel condition changes during the burst transmissions. In legacy IEEE 802.11, the burst size depends on fragmentation of the data from the higher layer. For example, if a TCP packet is fragmented and transmitted in multiple packets at the link layer, then these packets will be transmitted as a burst. Burst transmissions are enabled in IEEE 802.11e by allocating a transmission opportunity (TXOP) to a node. The pre-specified NAVs are not accurate and can cause inefficient channel utilization. This is because, first, the RTS/CTS exchange does not eliminate hidden terminal problems, since nodes that are beyond the transmission range of the transmitter cannot correctly decode the RTS frame. Therefore, successful virtual carrier sensing before a burst transmission does not guarantee collision-free burst transmissions.

Besides, even if virtual carrier sensing is enabled, successful RTS/CTS exchange, which is transmitted at the base rate, does not necessarily mean good transmission conditions for the data packets which are generally transmitted at a higher rate. Therefore, a data packet can be lost due to poor physical channel condition after successful virtual carrier sensing. When the sender failed to receive an ACK frame after transmitting a data packet, OLS can stop burst transmission immediately and allow other links with buffered packets to transmit. In contrast, DCF with pre-specified NAVs does not have such flexibility. In OLS $N_{\max}$ specifies the maximum burst size in number of packets, but the actual burst size is determined by the channel and other network conditions. The burst is shorter in case of stronger co-channel interference, higher collision rate, or poorer physical channel fading.

OLS considers transmission failure as an indication of poor transmission condition, and does not distinguish packet losses caused by co-channel interference and physical channel fading. This leads to simple algorithms and easy implementation. When $N_{\max} > 1$, packet transmissions along links with poor channel conditions are more likely to be delayed. Temporarily buffering a packet that experiences poor transmission conditions can avoid unnecessary transmissions and retransmissions. In slow-fading channels, if the current channel condition is bad, the channel condition in the next packet transmission time is very likely to be bad again. If a transmission failure is caused by network congestion, it is very unlikely that the congestion can be resolved before the next packet transmission time. In these cases, other links that have buffered packets and are in good channel conditions should be allowed to transmit their packets successfully.

By limiting the maximum number of back-to-back transmissions, OLS also limits the number of buffered packets in intermediate nodes, and this alleviates the network congestion. On the other hand, if a packet transmission fails, either because of a corruption in the data frame or the ACK frame, the sender does not receive the ACK frame in time, the node returns to the normal state and follows the standard IEEE 802.11 DCF for competing the channel.

Although OLS is a MAC layer protocol, the opportunistic scheduling improves the compatibility between the TCP and MAC layer protocols. That is, a large TCP window size does not increase the transmission collisions, but improves the performance of opportunistic scheduling, for nodes can better take advantage of the channel condition changes.

Chapter 4

Analytical Model of End-to-end Throughput

This chapter develops an analytical model to study the effect of TCP window size and the maximum burst size on the end-to-end transmission throughput in multi-hop transmissions with a simplified MAC protocol and opportunistic link scheduling. Some numerical results are then demonstrated.

4.1 Problem Formulation

It should be mentioned that the analysis below is not intended to match the performance of the proposed OLS protocol quantitatively, which involves both TCP and 802.11 DCF protocols and is very difficult to model accurately. The model, however, does catch some important features about the effect of burst size and TCP window size on end-to-end throughput.

We consider an end-to-end path of H hops. There are $H+1$ nodes indexed as $i = 0, 1, \ldots, H$, where node 0 is the source node which always has data to transmit, and node H is the destination node. All data traffic is from node 0 to node H. We assume that all packets have the same size. Transmission collisions of ACK frames are not considered as the collision rate is much lower than that of data packets and can be ignored. The physical channel is divided into equal size time slots with each of which for one packet transmission. Denote W the TCP window size, which is fixed in order to study its effect on the end-to-end throughput in opportunistic link scheduling. We adopt the Gilbert-Elliott two-state channel model [Gil60] [Ell77]. Each link can have Good and Bad channel conditions. The transition probabilities from the Good to Bad and from the Bad to Good states are P_{gb} and P_{bg}, respectively. When a link in the Good channel condition transmits, the successful transmission probability is 1, provided there is no co-channel interference. When a link in the Bad channel condition transmits, the successful transmission probability is 0. As will be shown later the two-state channel model is sufficient to demonstrate some important qualitative effects of both $N_{\max}$ and W on the end-to-end throughput in opportunistic link scheduling. Extending the analytical model to a more general channel model, such as the finite state Markov channel model in [WM95], can be done by modifying the expressions of state transition probabilities below accordingly.

We use a bold low case letter to represent a vector, and a bold capital letter to represent a set. Denote the buffer occupancy of node i as w_i, where $i = 1, 2, \ldots, H-1$. Then $\mathbf{w} = [w_1, w_2, \ldots, w_{H-1}]$ represents the buffer state of the chain, where $w_i \geq 0$ and $\sum_{i=1}^{H-1} w_i \leq W$. We assume that data packets received by node H are passed to the higher layer immediately, therefore, the buffer occupancy at node H is not considered. We use n_i to denote the current burst size for the link from node i to node $i+1$, where $i = 0, 1, \ldots, H-1$, and $0 \leq n_i \leq N_{\max}-1$. When the burst size reaches $N_{\max}$, it is reset immediately to 0, at the same time the node is set to have a low priority to transmit (detailed later). We further define a binary variable x_i, $i = 0, 1, \ldots, H-1$. When $x_i = 0$, node i has successfully transmitted a packet in the previous time slot and is holding the channel for transmitting more packets, and $x_i = 1$ otherwise. Therefore, when $x_i = 0$, $n_i > 0$, and we define node i as having a high priority to transmit (eligible for burst transmissions); and when $x_i = 1$, $n_i = 0$, and the node is having a low priority to transmit. Then $\mathbf{n} = [n_0, n_1, \ldots, n_{H-1}]$ and $\mathbf{x} = [x_0, x_1, \ldots, x_{H-1}]$ together can represent the burst transmission state along the end-to-end path. We use $\mathbf{s} = [\mathbf{w}, \mathbf{x}, \mathbf{n}] = [w_1, w_2, \ldots, w_{H-1}, x_0, x_1, \ldots, x_{H-1}, n_0, n_1, \ldots, n_{H-1}]$ to represent the system state, and s_i is the ith element in vector $\mathbf{s}$, e.g., $s_1 = w_1$, $s_{H+2} = x_2$, and $s_{2H} = n_0$. The system then can be modeled as a Markov chain, since the change from one state to the next only depends on current buffer occupancy and channel conditions and does not depend on what happened in the past. Let $\Pr\{\mathbf{s}\}$ represent the steady state probability of state $\mathbf{s}$, and $\Pr\{\mathbf{s}'|\mathbf{s}\}$ the one step transition probability from state $\mathbf{s}$ to state $\mathbf{s}'$. We will find the one-step transition probabilities. The steady state probabilities can be found from the transition probabilities and the condition $\sum_{\text{All } \mathbf{s}} \Pr\{\mathbf{s}\} = 1$.

The indication of a successful end-to-end transmission of a packet is a successful packet transmission from node $H-1$ to node H. Let C be the link transmission rate which is assumed to be the same for all links. Then the end-to-end throughput can be found as

$$T = C \sum_{\text{All s}} \Pr\{\mathbf{s}\} \sum_{\mathbf{s}' \in \mathbf{S}'} \Pr\{\mathbf{s}'|\mathbf{s}\}, \tag{4.1}$$

where $\mathbf{S}'$ is a set of states with $\mathbf{s}' \in \mathbf{S}'$ if $w'_{H-1} = w_{H-1} - 1$.

Although a general expression of the state transition probability for any H value is possible, the number of states and possible transitions increase dramatically with H. For a clear presentation, below we consider the case when $H = 4$ which represents the simplest scenario with packet transmission collisions. In this multihop chain, the MAC layer guarantees that there is at most one transmission within a 2-hop distance. The only possible data transmission collision is due to that a transmission from node 0 is corrupted by a concurrent transmission from node 3. Since there is at most one successful transmission at any given time, we have $\sum_{i=0}^{3} x_i \geq 3$.

We define binary variables A_i's, $i = 0, 1, 2$ and 3, as

$$A_0 = (x_0 = x_1 = x_2 = x_3 = 1 \text{ and } \sum_{j=1}^{3} w_j < W) \tag{4.2}$$

$$A_1 = (x_0 = x_1 = x_2 = x_3 = 1 \text{ and } w_1 > 0) \tag{4.3}$$

$$A_2 = (x_0 = x_1 = x_2 = x_3 = 1 \text{ and } w_2 > 0) \tag{4.4}$$

$$A_3 = (x_0 = x_1 = x_2 = x_3 = 1 \text{ and } w_3 > 0) \tag{4.5}$$

where $A_i = 1$ if the corresponding condition is true, and 0 otherwise. When $A_i = 1$, node i is competing for channel access, and there is no high priority node. We use M to denote the total number of low priority transmitting nodes, i.e., $M = \sum_{i=0}^{3} A_i$. When there is a high priority node (note that there is at most one high priority node when $H = 4$), $M = 0$, and the high priority node transmits with probability 1. Otherwise, the transmission probability of each node depends on specific situations.

When A_0 and A_3 are not both equal to 1, or $A_0 \times A_3 = 0$, node 0 and node 3 do not compete for the channel at the same time, and there is no collision. In this case we assume that all M nodes access the channel with an equal probability. When $A_0 \times A_3 = 1$, or $A_0 = A_3 = 1$, we assume that the following events occur with an equal probability: one of the $M-2$ nodes (excluding nodes 0 and 3) transmits; node 0 transmits and node 3 does not transmit; node 0 does not transmit and node 3 transmits; and both nodes 0 and 3 transmit. Therefore, the probability that there is only one among the M nodes (including nodes 0 and 3) transmitting is $\frac{1}{M+1}$, and the probability that nodes 0 and 3 transmit simultaneously is $\frac{1}{M+1}$. Let $q_i(\mathbf{s})$ be the probability that only node i transmits when the system is in state $\mathbf{s}$. Then the non-zero transmission probabilities can be found below. For $i = 1$ or 2, we have

$$q_i(\mathbf{s}) = \begin{cases} 1, & \text{if } x_i = 0 \ (M = 0) \text{ and } w_i > 0, \\ \frac{1}{M}, & \text{if } A_0 \times A_3 = 0, \ w_i > 0, \ \text{and } M > 0, \\ \frac{1}{M+1}, & \text{if } A_0 \times A_3 = 1, w_i > 0, \ \text{and } M > 0. \end{cases} \tag{4.6}$$

The transmission probability for node 0 is given by

$$q_0(\mathbf{s}) = \begin{cases} 1, & \text{if } x_0 = 0 \ (M = 0) \text{ and } \sum_{j=1}^{3} w_j < W, \\ \frac{1}{M}, & \text{if } A_0 \times A_3 = 0, \ \sum_{j=1}^{3} w_j < W, \ \text{and } M > 0, \\ \frac{1}{M+1}, & \text{if } A_0 \times A_3 = 1 \text{ and } M > 0. \end{cases} \tag{4.7}$$

For node 3,

$$q_3(\mathbf{s}) = \begin{cases} 1, & \text{if } x_3 = 0 \ (M = 0) \text{ and } w_3 > 0, \\ \frac{1}{M}, & \text{if } A_0 \times A_3 = 0, \ w_3 > 0, \ \text{and } M > 0, \\ \frac{1}{M+1}, & \text{if } A_0 \times A_3 = 1 \text{ and } M > 0. \end{cases} \tag{4.8}$$

The probability that both nodes 0 and 3 transmit is given by

$$q_{03}(\mathbf{s}) = \tfrac{1}{M+1}, \quad \text{if } A_0 \times A_3 = 1 \text{ and } M > 0. \tag{4.9}$$

The successful transmission probability depends on both the physical channel condition and possible transmission collisions. We first consider the case when nodes 0 and 3 do not transmit simultaneously.

- The state transition probability made by a successful packet transmission from node i can be written in a general form as

$$\Pr\{\mathbf{s}^{'}|\mathbf{s}\} = \begin{cases} q_i(\mathbf{s})(1-P_{gb}), & \text{if } x_i = 0, \\ q_i(\mathbf{s})P_{bg}, & \text{if } x_i = 1. \end{cases} \tag{4.10}$$

  - When $n_i < N_{\max} - 1$ and $w_i > 1$ for $i \neq 0$ (or when $n_i < N_{\max} - 1$ and $\sum_{j=1}^{3} w_j < W$ for $i = 0$), a successful transmission from node i increases n_i by 1, the value of x_i is 0 after the transmission, and the node can keep transmitting in the same burst. Based on this, the elements in $\mathbf{s}^{'}$ are given below. When $i = 0$, the transmission increases the buffer of node 1 by 1:

$$\begin{cases} w_1^{'} = w_1 + 1, x_0^{'} = 0, n_0^{'} = n_0 + 1, \\ s_j^{'} = s_j, \ \text{if } j \neq 1, 4, \text{ and } 8. \end{cases} \tag{4.11}$$

When $i = 1$ or 2, the transmission decreases the buffer of node i by 1 and increases the buffer of node $i+1$ by 1:

$$\begin{cases} w_i^{'} = w_i - 1, w_{i+1}^{'} = w_{i+1} + 1, x_i^{'} = 0, n_i^{'} = n_i + 1, \\ s_j^{'} = s_j, \ \text{if } j \neq i, i+1, \ i+4, \text{ and } i+8. \end{cases} \tag{4.12}$$

When $i = 3$, the transmission decreases the buffer of node 3 by 1:

$$\begin{cases} w_3^{'} = w_3 - 1, x_3^{'} = 0, n_3^{'} = n_3 + 1, \\ s_j^{'} = s_j, \ \text{if} j \neq 3, \ 7, \text{ and } 11. \end{cases} \tag{4.13}$$

– When either $n_i = N_{\max} - 1$ or $w_i = 1$, node i cannot keep transmitting in the same burst after the current transmission either due to its burst size has reached the maximum limit or the node does not have any packet in its buffer (for $i > 0$). In this case, the elements in $\mathbf{s}'$ are given below. For $i = 0$,

$$\begin{cases} w_1' = w_1 + 1, x_0' = 1, n_0' = 0, \\ s_j' = s_j, \text{ if } j \neq 1, 4, \text{ and } 8. \end{cases} \tag{4.14}$$

When $i = 1$ or 2,

$$\begin{cases} w_i' = w_i - 1, w_{i+1}' = w_{i+1} + 1, x_i' = 1, n_i' = 0, \\ s_j' = s_j, \text{ if } j \neq i, i+1, \ i+4, \text{ and } i+8. \end{cases} \tag{4.15}$$

When $i = 3$,

$$\begin{cases} w_3' = w_3 - 1, x_3' = 1, n_3' = 0, \\ s_j' = s_j, \text{ if } j \neq 3, \ 7, \text{ and } 11. \end{cases} \tag{4.16}$$

- A transmission failure results in the following transition

$$\Pr\{\mathbf{s}'|\mathbf{s}\} = \begin{cases} q_i(\mathbf{s})P_{gb}, & \text{if } x_i = 0, \\ q_i(\mathbf{s})(1 - P_{bg}), & \text{if } x_i = 1, \end{cases} \tag{4.17}$$

where

$$\begin{cases} x_i' = 1, n_i' = 0, \\ s_j' = s_j, \text{ if } j \neq i+4 \text{ and } i+8. \end{cases} \tag{4.18}$$

When both nodes 0 and 3 transmit simultaneously, the transmission of node 0 fails, but the transmission of node 3 depends on its channel condition.

- When the transmission of node 3 is successful, it makes the following state transition:

$$\Pr\{\mathbf{s}'|\mathbf{s}\} = \begin{cases} q_{03}(\mathbf{s})(1 - P_{gb}), & \text{if } x_i = 0, \\ q_{03}(\mathbf{s})P_{bg}, & \text{if } x_i = 1, \end{cases} \tag{4.19}$$

where $\mathbf{s}'$ depends on n_3 and w_3.

- If $n_3 < N_{\max} - 1$ and $w_3 > 1$ before the transmission, we have

$$\begin{cases} w_3' = w_3 - 1, x_0' = 1, x_3' = 0, n_0' = 0, n_3' = n_3 + 1, \\ s_j' = s_j, \text{ if } j = 1, 2, 5, 6, 9, \text{ or } 10. \end{cases} \tag{4.20}$$

- If $n_3 = N_{\max} - 1$ or $w_3 = 1$ before transmission, we have

$$\begin{cases} w_3' = w_3 - 1, x_0' = 1, x_3' = 1, n_0' = 0, n_3' = 0, \\ s_j' = s_j, \text{ if } j = 1, 2, 5, 6, 9, \text{ or } 10. \end{cases} \tag{4.21}$$

- When the transmission of node 3 is unsuccessful, the transition is given by

$$\Pr\{\mathbf{s}'|\mathbf{s}\} = \begin{cases} q_{03}(\mathbf{s})P_{gb}, & \text{if } x_i = 0, \\ q_{03}(\mathbf{s})(1 - P_{bb}), & \text{if } x_i = 1, \end{cases} \tag{4.22}$$

where

$$\begin{cases} x_0' = 1, x_3' = 1, n_0' = 0, n_3' = 0, \\ s_j' = s_j, \text{ if } j = 1, 2, 3, 5, 6, 9, \text{ or } 10. \end{cases} \tag{4.23}$$

4.2 Numerical Results

Figs. 4.1 and 4.2 show the end-to-end throughput performance based on the above model, where $P_{gb} = P_{bg} = 0.1$. The TCP window size has two contradictory effects on the end-to-end throughput. As the window size increases, i) more links may have buffered packets and compete for channel access, which increases the collision rate; ii) meanwhile, there is a better chance to opportunistically schedule links with good channel conditions. Fig. 4.1 shows when the TCP window size is smaller than $N_{\max}$, the end-to-end throughput increases with the TCP window size. In this case, the second effect of the window size dominates and collisions are still minor. As the window size keeps increasing, the collision rate increases and the throughput curve increases slower with the TCP window size. When the TCP window size is larger than $N_{\max}$, further increasing the window size only changes the end-to-end throughput slightly up and down, as none of the two effects of TCP window size on end-to-end throughput can always dominate the other.

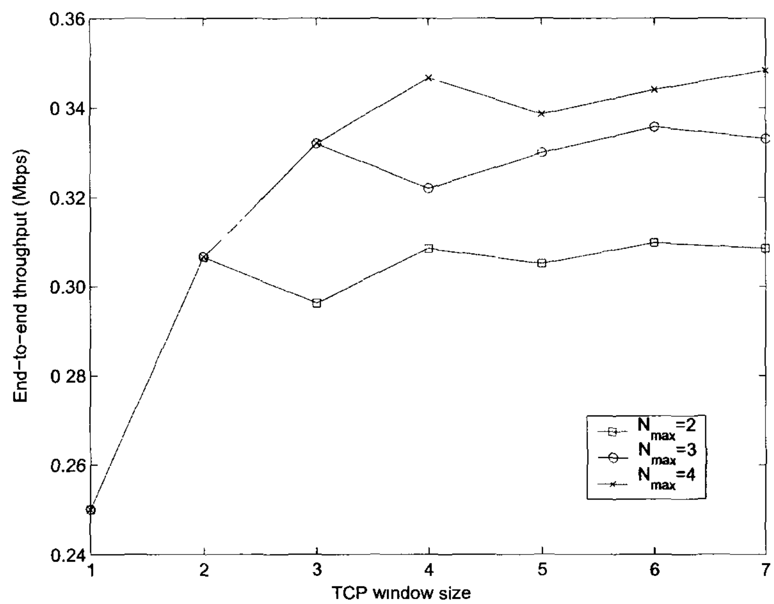

Figure 4.1: End-to-end throughput vs. TCP window size

Fig. 4.2 shows that the end-to-end throughput can be increased by increasing $N_{\max}$, provided that $N_{\max}$ is no larger than the TCP window size. As $N_{\max}$ increases, a node with good channel condition can have a better chance to take advantage of the link transmission condition to transmit more packets successfully. The effect of $N_{\max}$ on the end-to-end throughput is limited by the TCP window size, since the

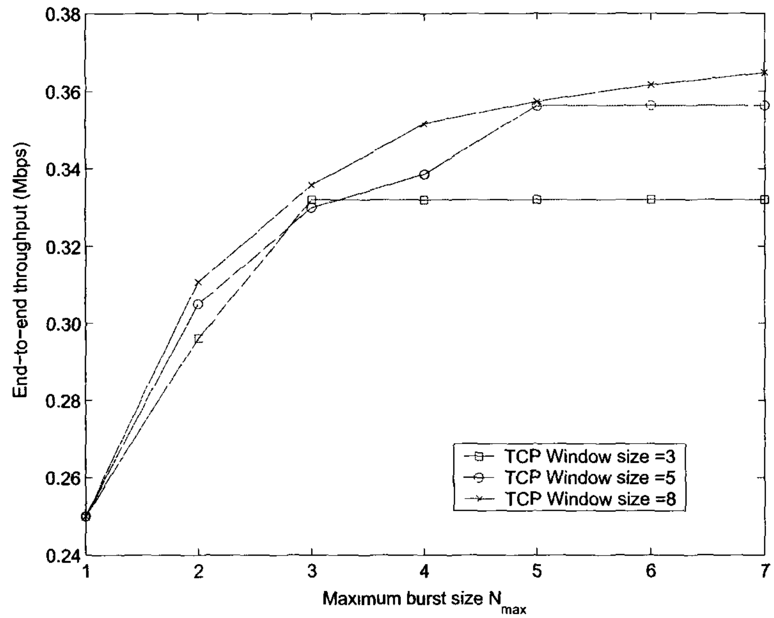

Figure 4.2: End-to-end throughput vs. maximum burst size

number of buffered packets (and therefore the actual size of burst transmissions) in an intermediate node is limited by the TCP window size.

The above results are based on fixed TCP window size. In a practical system, the TCP window size is adjusted dynamically. The slow reaction of TCP window size adjustment in IEEE 802.11-based multihop networks has been considered negative as it leads to larger window size that decreases the end-to-end throughput. By using OLS with burst transmissions, this can be an advantage as it allows longer burst transmissions and higher end-to-end throughput. The optimum value of $N_{\max}$ is equal to the TCP window size. Although it is difficult to dynamically keep $N_{\max}$ at the optimum value, it is easier to have a larger value for $N_{\max}$ which will achieve the end-to-end throughput as the optimum $N_{\max}$ does. Therefore, selecting $N_{\max}$ for maximizing end-to-end throughput is relatively easy in OLS, and this does not involve any changes to the TCP protocol. In this sense, OLS improves the compatibility between the TCP and MAC layer protocols.

Chapter 5

Simulation Results

This chapter demonstrates the performance of the proposed OLS protocol. First, we compare end-to-end throughput and delay performance of OLS and 802.11 DCF for a single flow in a chain topology. The performance improvement by using OLS is examined under different channel conditions. The effect of TCP window size upper bound and packets burst size $N_{\max}$ on the end-to-end throughput is studied. Then, the end-to-end throughput performance is studied for multiple flows and in more complicated network topologies. At last, the fairness of OLS is evaluated.

5.1 Simulation Environment

We consider a multihop wireless network as shown in Fig. 5.1, where $H+1$ nodes are equally spaced and form a chain topology. All nodes are fixed. Default system settings and parameters are as follows. The distance between two immediate neighboring nodes is 200m. TCP New Reno is adopted as the transport layer protocol, and the

TCP packet size is 1024 bytes. We do extensive simulation studies using ns2.26 to examine the performance of OLS and compare it with IEEE 802.11 DCF. A node using the 802.11 DCF transmits one packet for every channel access. Comparison of burst transmissions between OLS and DCF is also performed. We use the "Two-Ray-Ground" propagation model in ns2. That is, there is a direct path and a ground reflection path between the sender and receiver of each link. Virtual carrier sensing in 802.11 is enabled. The physical rate for the data packet transmissions is 2Mbps, and the basic rate for control packet transmissions is 1Mbps. The default value for $N_{\max}$ is 3 packets. The queue size, $q_{\max}$, at each node is set to be 50 packets. The transmitted signals along different links may experience independent Ricean fading with a Ricean factor K. A saturation case is studied. That is, the source node always has data to transmit.

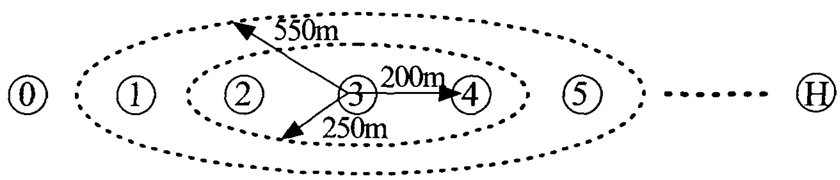

Figure 5.1: Multihop wireless mesh network topology

5.2 Simulation Results and Performance Analysis

5.2.1 Single Traffic Flow

We first consider a single data flow with all data packets transmitted from node 0 to node H along the chain. The transmission range of each node is 250m, and both the interference range and carrier sensing range are 550m. Therefore, the interference range is slightly more than twice of the transmission range for a node.

Throughput performance of OLS vs. number of hops is shown in Fig. 5.2, where the throughput of IEEE 802.11 DCF is also shown for comparison. We first look at the case without channel fading. When the total number of hops is less than 4, there is at most one transmission at any time, and the TCP throughput for an H-hop path is approximately $1/H$ of the throughput for $H = 1$. This is true for both OLS and DCF.

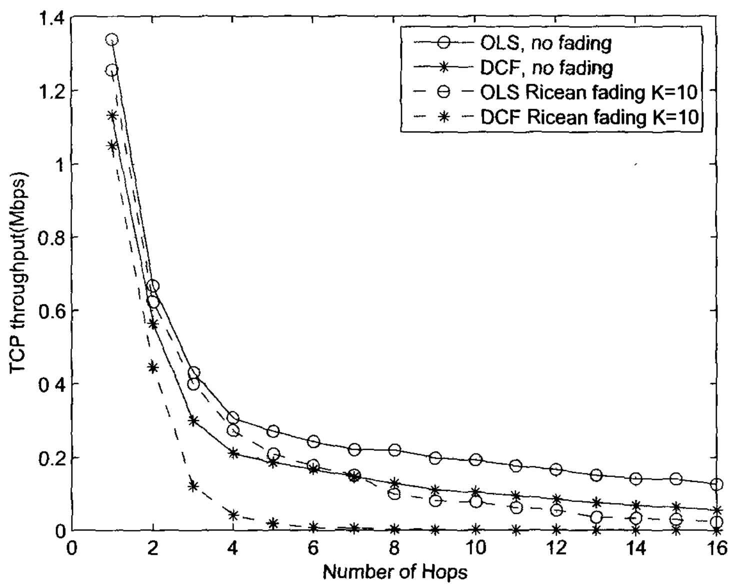

Figure 5.2: TCP throughput of a single flow

When $H \geq 4$, neither OLS nor DCF can completely eliminate transmission collisions. As H increases, more links compete for channel access, and transmission collisions caused by hidden nodes increase. Therefore, the TCP throughput decreases with number of hops for both OLS and 802.11 DCF. Fig. 5.2 also shows the OLS throughput in Ricean fading channel, where a Ricean factor of 10 represents a mild vibration of the signal strength. In the simulation, we use the Ricean fading model in [PS00]. Compared with the scenario without channel fading, packet losses in the network with Ricean fading can be caused by both co-channel interference and poor channel propagation, and therefore the throughput values are less than that without fading.

OLS achieves higher throughput than the 802.11 DCF. The percentage of end-to-end throughput improvement of OLS over DCF is shown in Fig. 5.3, which indicates that the throughput improvement increases with the number of hops. Without channel fading, the throughput improvement when $H < 4$ comes from the fact that OLS requires only one virtual carrier sensing for transmitting multiple data packets, while IEEE 802.11 DCF requires virtual carrier sensing for every transmitted packet. When $H \geq 4$, OLS further reduces the transmission collision rate. This can save a significant amount of channel time. Therefore, TCP throughput can be improved dramatically when H is large. This makes the proposed OLS protocol more attractive for multihop networks with a larger number of hops. Furthermore, the throughput improvement of OLS over the IEEE 802.11 DCF is a lot more significant than that in the case without fading. The above results indicate that the opportunistic scheduling can mitigate some effect of poor transmission conditions due to both co-channel interference and channel fading.

We then look at the impact of burst transmissions in OLS on the end-to-end transmission delay. Fig. 5.4 shows the average end-to-end delay using OLS and IEEE 802.11 DCF. It is shown that OLS does not introduce a significant extra packet transmission delay, but significantly improves the end-to-end transmission delay over DCF when there is channel fading and the number of hops is relatively large, in

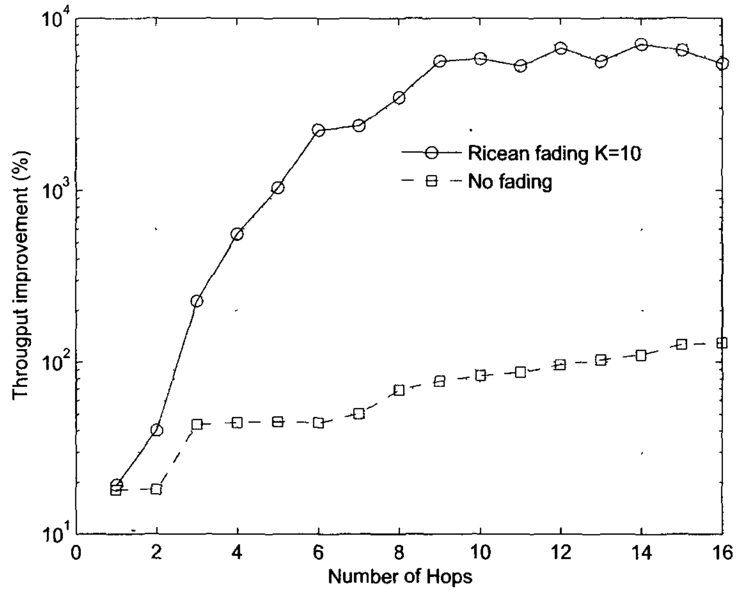

Figure 5.3: TCP throughput improvement of OLS over 802.11 DCF

which case the end-to-end packet transmission delay using IEEE 802.11 DCF increases dramatically with H while is kept around 200ms using OLS for a wide range of H values. The reason that OLS can achieve stable and relatively small delay over a long path is because it can greatly reduce the number of retransmissions at both the link and transport layers and make packet transmissions much more efficient. These results demonstrate that by temporarily buffering packets in the links with poor channel conditions and allowing those with good channel conditions to transmit a burst of packets, the OLS does not only improve the end-to-end throughput, but also achieve stable end-to-end transmission delay.

Next we compare the throughput performance of OLS and DCF for burst transmissions. Fig. 5.5 shows the throughput improvement of OLS over DCF, where the maximum burst size in OLS is set to $N_{\max} = 3$ packets and the burst size for DCF

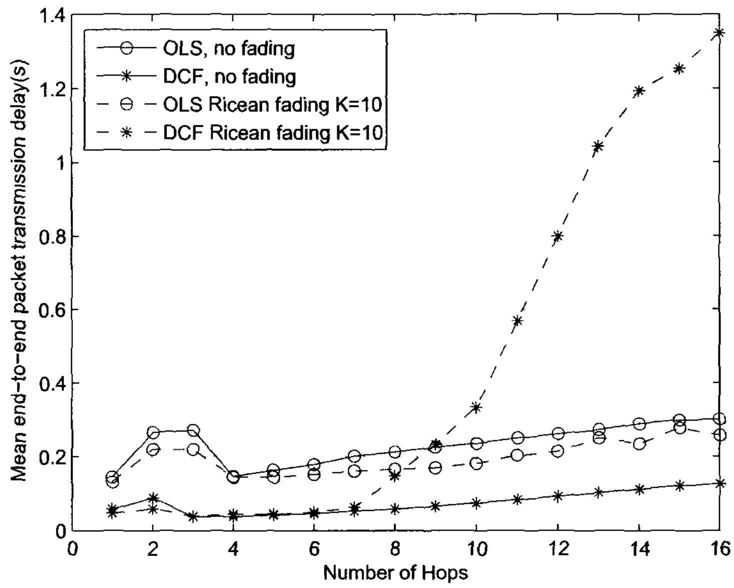

Figure 5.4: End-to-end packet delay for a single flow

is preset and fixed at 3 packets. It is shown in the figure that the throughput improvement is more significant when the number of hops increases and channel fading is severer. When a packet transmission is corrupted due to either physical channel fading or transmission collision, the sender in DCF keeps retransmitting the lost packets. During this process, nodes that were blocked from transmitting at the beginning of the burst transmission are still blocked until the NAV is expired. In OLS, once there is a transmission failure, the burst transmission is stopped, and neighboring nodes are allowed to compete for using the channel again. This provides a chance for some links with good channel conditions to transmit and results in higher channel utilization efficiency. Therefore, adaptively adjusting the burst size based on the transmission conditions in OLS is a better approach for burst transmissions.

Selecting values of $N_{\max}$ affects the packet transmission performance of OLS. Fig. 5.6 shows that TCP throughput increases with $N_{\max}$. The improvement is more

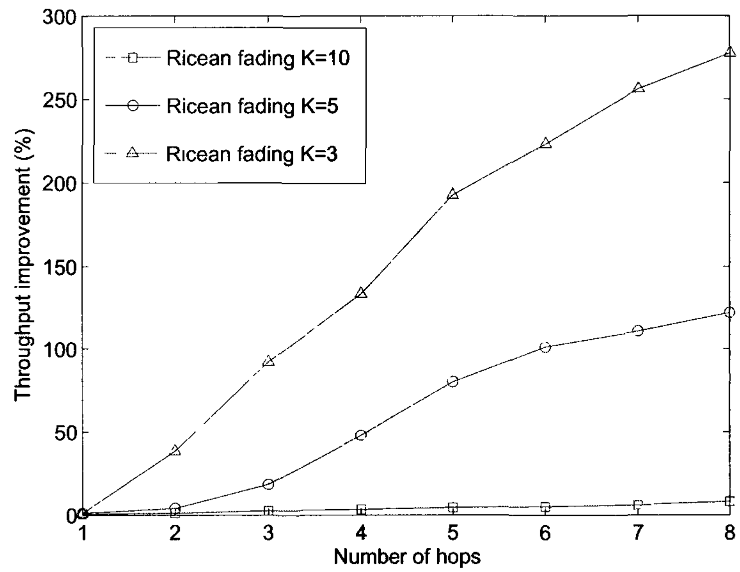

Figure 5.5: TCP throughput improvement of OLS over DCF for burst transmissions

significant when $N_{\max}$ is relatively small. Further increasing $N_{\max}$, e.g., beyond 3 in the simulated cases, only affects the end-to-end throughput slightly. This is consistent with the analytical results. As it is shown in Fig. 4.1, the effect of $N_{\max}$ on the throughput improvement is limited by the TCP window size, and the maximum value of $N_{\max}$ that can improve the end-to-end throughput is equal to the TCP window size. According to this we can find in Fig. 5.6 that the TCP window size for the $H = 4$ case is around 3, much larger than 1, which is the optimum window size if using IEEE 802.11 DCF as the MAC protocol. Fig. 5.6 also shows that the throughput improvement of OLS is more obvious when $H = 4$ than when $H = 2$, since in the former case transmission failures can be caused by both collisions and physical channel fading, while in the latter case there is no collision.

The feature that a larger value of $N_{\max}$ does not negatively affect the end-to-end throughput in OLS can ease the selection of $N_{\max}$. For example, setting unlimited

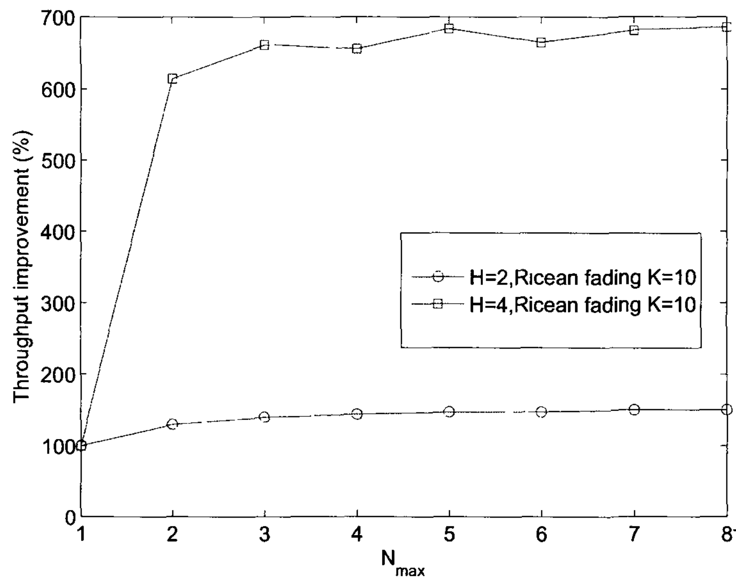

Figure 5.6: OLS: effect of $N_{\max}$ on TCP throughput

maximum burst size, i.e., $N_{\max} = \infty$, can always achieve the maximum throughput improvement. Packet transmission failures (due to physical channel fading and MAC layer collisions) and TCP window size adjustment together determine the actual burst size. Setting a smaller value of $N_{\max}$ may balance the throughput maximization and fairness in a multi-flow network, and will be studied as a separated topic.

Fig. 5.7 shows the effect of $N_{\max}$ on the end-to-end transmission delay. It can be seen that the transmission delay increases with $N_{\max}$ when $N_{\max}$ is relatively small. This is due to that a larger $N_{\max}$ increases the number of buffered packets. However, further increasing $N_{\max}$ only affects the end-to-end delay very slightly due to that the actual burst size is limited by the TCP window size and channel conditions.

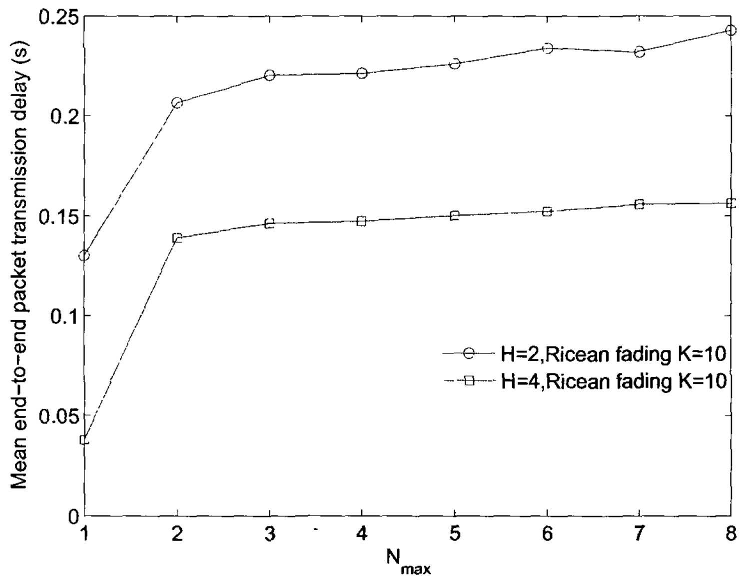

Figure 5.7: OLS: effect of $N_{\max}$ on end-to-end transmission delay

5.2.2 Effect of TCP WUB

The above results are based on an unbounded TCP window size. That is, there is no limit on the maximum value of the TCP congestion window size. Next, we study the effect of TCP window size upper bound (WUB) on the end-to-end throughput. Fig. 5.8 demonstrates that when TCP WUB is below a certain threshold, TCP throughput increases with TCP WUB. Once the TCP WUB is above a certain threshold, the TCP throughput is not very sensitive to the WUB. The reason for this result is that using OLS can greatly reduce the packet drop rate, and therefore reduce the chance to trigger congestion avoidance in TCP. In OLS, the TCP WUB should be sufficiently large in order for the links to take advantage of good link conditions. This property of OLS can ease cross-layer designs for multihop networks, since having unbounded TCP window size will not as negatively affect the end-to-end transmission performance in OLS as in DCF.

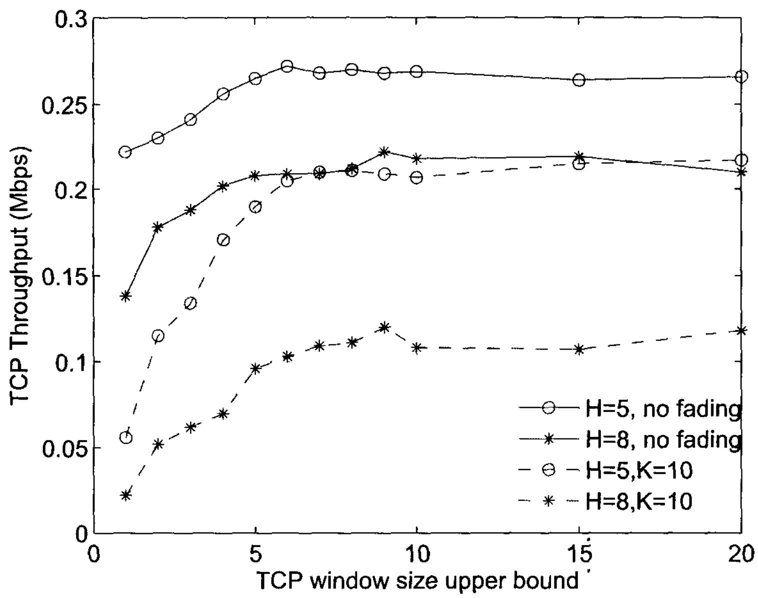

Figure 5.8: OLS: TCP throughput vs. WUB for a single flow

As the channel fading is severer, there should be more links with buffered packets in order to have at least one link with good channel condition. In this case, a larger TCP window size helps the OLS for improving transmission throughput. This is one major difference between OLS and DCF.

Fig. 5.9 shows the corresponding end-to-end packet transmission delay as the TCP WUB changes. It is seen that when the WUB is below a certain threshold, the end-to-end delay increases with TCP WUB, as more packets are allowed to be buffered in intermediate nodes. However, the end-to-end delay becomes stable as the WUB further increases, since the effect of TCP window size is also limited by the value of $N_{\max}$.

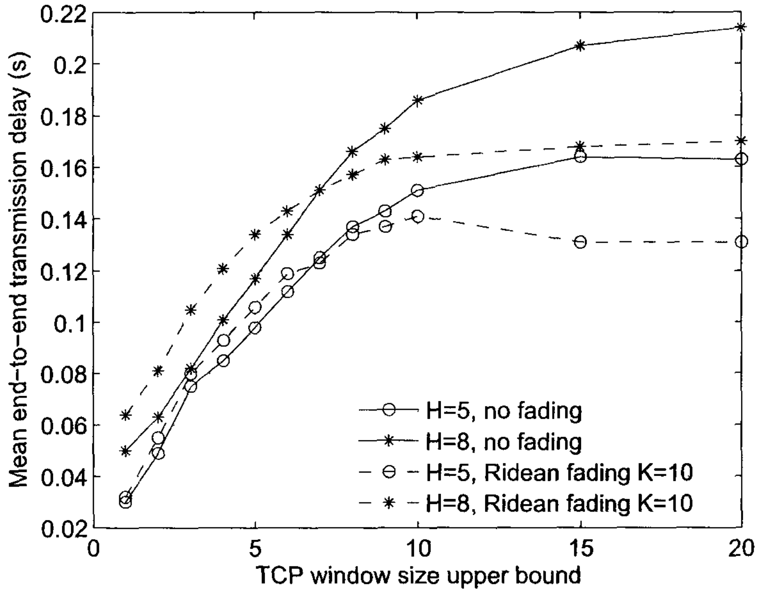

Figure 5.9: OLS: end-to-end delay vs. WUB for a single flow

5.2.3 Multiple Flows

The above results are all based on a single flow case. In Fig. 5.10 we examine the throughput performance with multiple flows in the same chain topology as shown in Fig. 5.1. Each of the nodes from node 0 to node $H-1$ initiates a TCP flow with node H as the destination. The flows start generating packets at a time uniformly distributed from 5s to 50s and keep injecting packets afterward. Since there are multiple flows coexisting in the same network, transmissions in a link experience interference not only from the same flow, but also from other flows. The aggregate TCP throughput in this scenario is shown in Fig. 5.10. The results show that OLS achieves much higher end-to-end throughput than the IEEE 802.11 DCF.

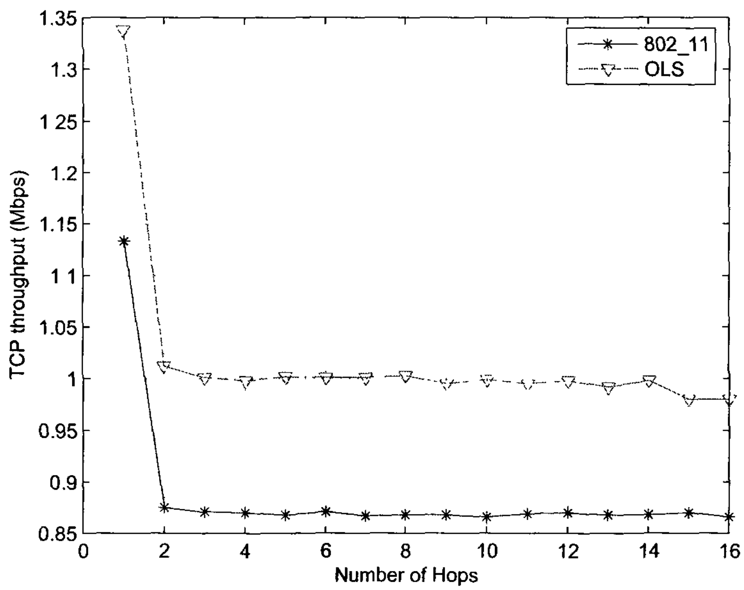

Figure 5.10: TCP throughput for multiple flows

5.2.4 Random Topology

We also simulated the end-to-end throughput in a random topology, where N nodes are randomly distributed in a 10000×10000 m^2 square area. One TCP flow is set up from node 0 to node $N-1$ by using the shortest distance routing, i.e., by finding the path with the minimum number of hops from the source to the destination. In order to keep reasonable connectivity, the distance between adjacent nodes cannot be smaller than 150m. The end-to-end throughput improvement of OLS over DCF in this scenario is shown in Fig. 5.11, and the end-to-end delay using both OLS and DCF is shown in Fig. 5.12. The results are similar to those shown in Figs. 5.3 and 5.4. The main difference between the two simulated scenarios is that the number of hops from the source to the destination in the random topology is random and can be any value from 1 to $N-1$. Therefore, the throughput improvement in the random network with N nodes (at most $N-1$ hops) is less than that in a chain network with

$N-1$ hops. Similarly, the end-to-end delay in the random network with N nodes is shorter than that in a chain network with $N-1$ hops. Nevertheless, the results still show significant throughput improvement of OLS over DCF, and OLS achieves much lower end-to-end delay in relatively large networks.

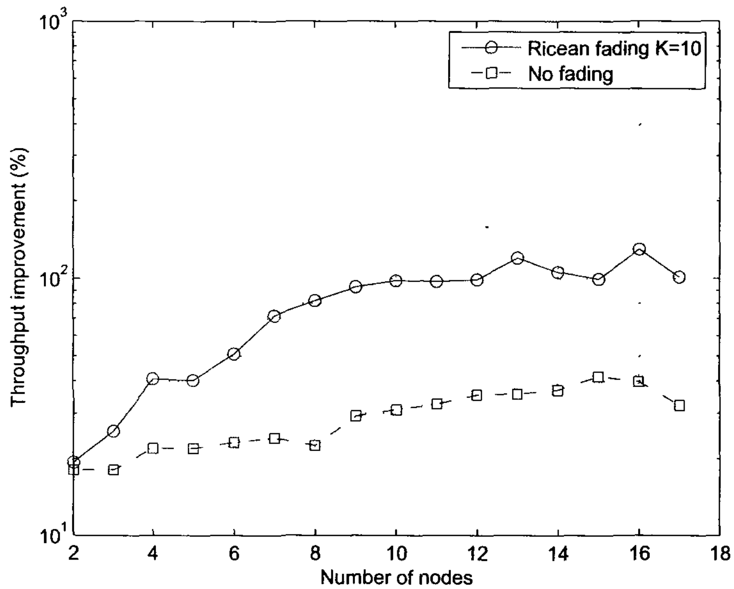

Figure 5.11: TCP throughput improvement of OLS over DCF in a random topology

5.2.5 Fairness Evaluation

Fairness is an important issue in a network with shared channel. In our simulation, three TCP flows are generated from node 0 to node H. The starting tims of the flows are 5s, 10s, and 15s respectively. The total simulation time is 500s. Fairness index [HFC05] is defined as $(\sum_f \frac{S_f}{\omega_f})^2 / \sum_f \sum_f (\frac{S_f}{\omega_f})^2$, where S_f and ω_f represent throughput and weight of flow f, respectively. Perfect fairness is achieved if fairness index is 1. The weight for every flow is set to be 1 in the simulation. The fairness index is shown in Fig. 5.13 for different channel fading conditions. It is seen that OLS can achieve

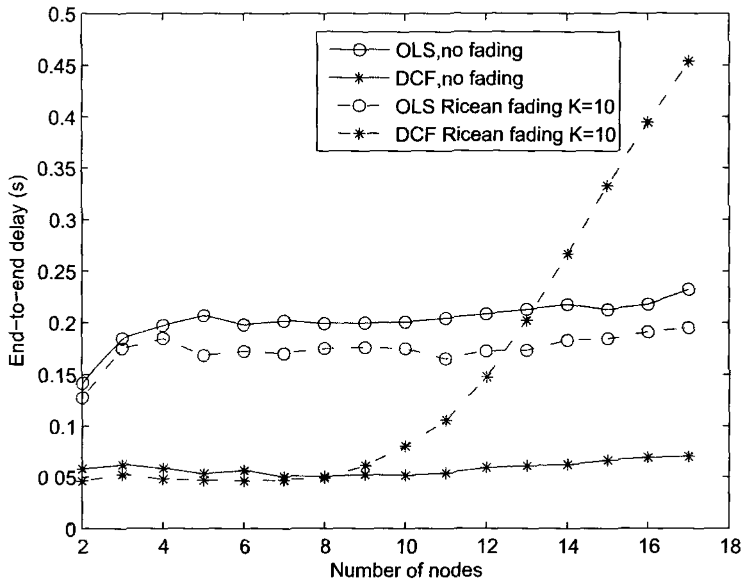

Figure 5.12: TCP end-to-end delay in a random topology

very good throughput fairness among the flows. This indicates that the opportunistic scheduling gives each flow a fair chance to access the channel. In the simulated case, all flows have the same physical channel conditions. However, we have also found that the fairness of OLS needs to be improved when multiple flows have different interference conditions. This unfairness is inherent from the IEEE 802.11 DCF.

5.2.6 More Complicated Topologies

We also simulated multi-flow scenarios in cross and grid topologies as shown Fig. 5.14. The interference in these scenarios comes from both inter-flow and intra-flow transmissions. In the cross topology, 13 nodes are distributed in two lines, each of which has 7 nodes equally spaced with 200m between two immediately adjacent nodes. Two TCP flows are set up respectively from node 0 to node 6 and from node 7 to node 12. The simulation results show that OLS improves the aggregate throughput by

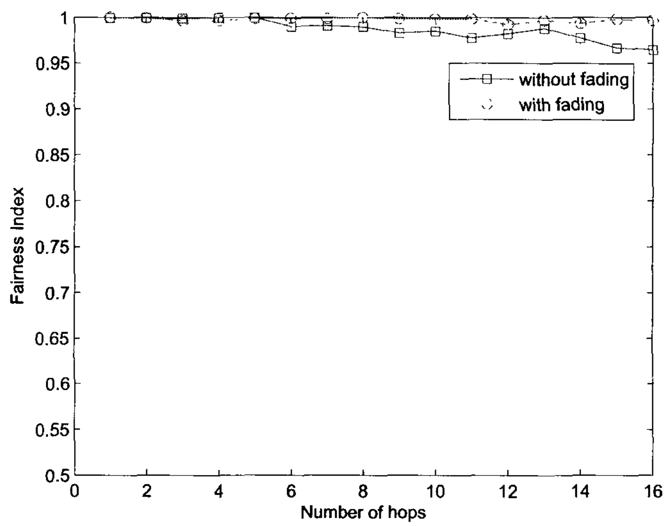

Figure 5.13: Fairness index for three flows

55.28% without physical channel fading, compared to standard IEEE 802.11 DCF. In the 9×9 grid topology, we run 9 flows with one flow in each line from the left to the right. In the 9-flow case, the aggregate TCP throughput improvement of OLS over DCF is 103.22% without fading.

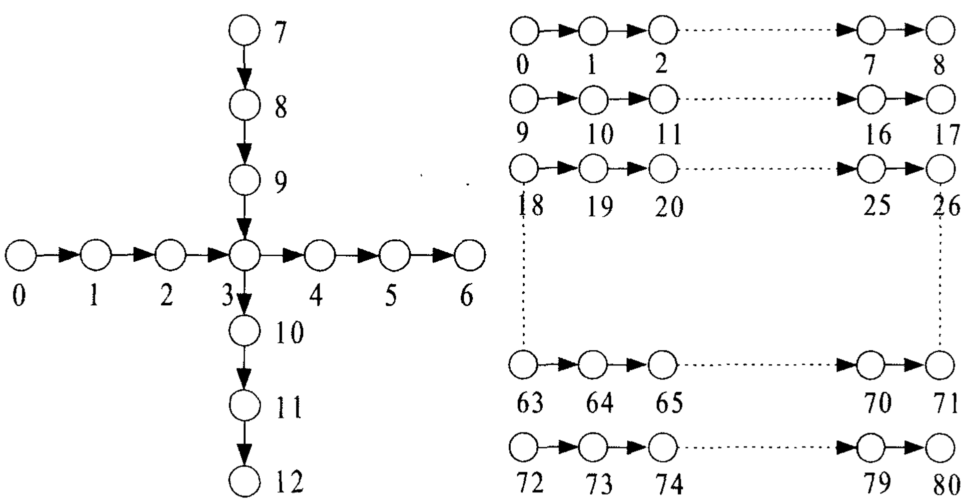

Figure 5.14: Cross and grid topologies

5.2.7 UDP Performance Evaluation

We finally simulate the chain network when supporting UDP traffic. Fig. 5.15 shows that there is also good throughput improvement of OLS over DCF when supporting UDP traffic, although not as significant as when supporting TCP traffic. Since UDP traffic does not involve complicated retransmission mechanisms as TCP, packets dropped at the link layer do not affect the source to keep generating new packets. However, OLS can still mitigate the effect of transmission failures due to both link layer transmission collisions and physical channel impairment.

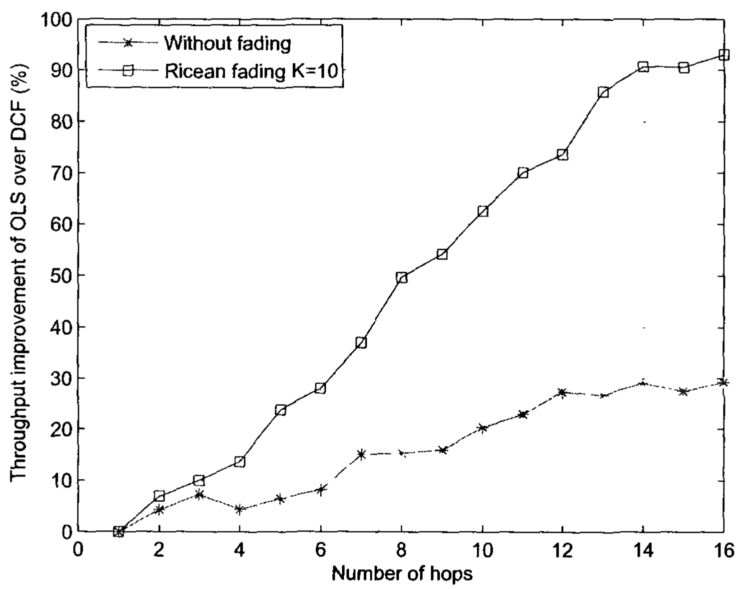

Figure 5.15: Throughput improvement for UDP traffic

Chapter 6

Conclusions and Future Work

In this thesis, we have proposed an opportunistic link scheduling protocol in IEEE 802.11-based multihop wireless networks. The protocol includes a scheme to opportunistically schedule transmissions of links with good channel conditions, and a method to prevent the network congestion and avoid starvation of nodes with poor channel conditions. The protocol is effective for combating both physical channel fading and MAC layer transmission collisions. It improves the end-to-end transmission throughput for TCP traffic, while keeping reasonably low transmission delay in multihop transmissions. OLS achieves better compatibility between the MAC layer and TCP layer protocols. It is a distributed protocol, easy to implement, and requires minor modifications to the IEEE 802.11 protocol.

While significantly improving the end-to-end throughput in multihop networks, OLS does inherit some unfairness of 802.11 in multihop transmissions and does not provide good fairness among throughput of multiple flows if their experienced interference conditions are different. In general, maximizing throughput and achieving throughput fairness are two contradictory objectives. Throughput fairness in IEEE 802.11-based multihop networks is another complicated issue that will be studied in our future work.

Bibliography

[AB02] X. Perez A. Banchs. Distributed weighted fair queuing in 802.11 wireless lan. *IEEE International Conference on Communication*, Apr. 2002.

[BSK02] A. Sabharwal B. Sadeghi, V. Kanodia and E. Knightly. OAR: an opportunistic auto-rate media access protocol for ad hoc networks. *Proceedings of ACM MobiCom'02*, Sept. 2002.

[DO05] T De Oliveira, R. Braun. A dynamic adaptive acknowledgment strategy for TCP over multihop wireless networks. *Proc. IEEE INFOCOM*, 2005.

[dOB05] R. de Oliveira and T. Braun. A dynamic adaptive acknowledgment strategy for TCP over multihop wireless networks. *Proc. IEEE INFOCOM*, pages 1863–1874, Mar. 2005.

[Ell77] E. O. Elliott. Estimates of error rates for codes on burst-noise channels. *Bell Syst. Tech. J.*, page 1977, Sept. 1977.

[Gil60] E. N. Gilbert. Capacity of a burst-noise channel. *Bell Syst. Tech. J.*, page 1253 1265, Sept. 1960.

[HFC05] C. Lee H. Ferng and G. Chiu. Designing a fairness scheduling mechanism for IEEE 802.11 wireless LANs. *IEEE Communication Letters*, pages 425–426, Apr. 2005.

[HJW06] X. Shen A. Abdrabou H. Jiang, W. Zhuang and P. Wang. Differentiated services for wireless mesh backbone. *IEEE Communications Magazine*, pages 113–119, Mar. 2006.

[HZ06] Y. Fang H. Zhai. Distributed flow control and medium access in multihop ad hoc networks. *IEEE Trans. on Mobile Computing*, pages 1503–1514, Nov. 2006.

[HZF06] J. Wang H. Zhai and Y. Fang. DUCHA: a new dual-channel MAC protocol for multihop ad hoc networks. *IEEE Transactions on Wireless Communications*, 2006.

[IH06] T. Issariyakul and E. Hossain. Performance modeling and analysis of a class of ARQ protocols in multi-hop wireless networks. *IEEE Transactions on Wireless Communications*, pages 3460–3468, Dec. 2006.

[JS03] J. Jun and M. L. Sichitiu. Fairness and QoS in multihop wireless networks. *IEEE Vehicular Technology Conference*, Oct. 2003.

[JWF04] H. Zhai J. Wang and Y. Fang. Opportunistic packet scheduling and media access control for wireless LANs and multihop ad hoc networks. *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1234–1239, Mar. 2004.

[KCK04] S.H.Shah K. Chen, Y. Xue and K.Nahrstedt. Understand bandwidth delay product in mobile ad hoc networks. *Elsevier Computer Communications Journal, Special Issue on Protocol Engineering for Wired and Wireless Networks*, pages 923–924, 2004.

[KM97] A. Kamerman and L. Monteban. WaveLAN II: a high-performance wireless LAN for the unlicensed band. *Bell Labs Technical Journal*, 1997.

[KW01] A. Kosel and A. Wolisz. Voice transmission in an IEEE 802.11 WLAN based access network. *Proceeding of the 4th ACK international workshop on wireless mobile multimedia*, pages 23–32, 2001.

[KYL06] B.-S. Lee K. Y. Lee, K.-S. Cho. Cross-layered hop-by-hop congestion control for multihop wireless networks. *Proc. IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS)*, pages 485–488, Oct. 2006.

[MK04] A. McGibney D. Pesch M. Klepal, R. Mathur. Influence of people shadowing on optimal deployment of WLAN access points. *Proc. of IEEE Vehicular Technology Conference (VTC)*, Sept. 2004.

[MPK] A. Mahasenan M. Prabakaran and A. Kumar. Analysis and enhancement of TCP performance over an IEEE 802.11 multi-hop wireless network: single session case. *Proc. IEEE International Conference on Personal Wireless Communications(ICPWC) 2005.*

[MPK05a] A. Mahasenan M. Prabakaran and A. Kumar. Analysis and enhancement of TCP performance over an IEEE 802.11 multi-hop wireless network: single session case. *Proc. IEEE ICPWC*, Aug. 2005.

[MPK05b] A. Mahasenan M. Prabakaran and A. Kumar. Analysis and enhancement of TCP performance over an IEEE 802.11 multi-hop wireless network: single session case. *Proc. IEEE ICPWC*, Jan. 2005.

[NL05] P. C. Ng and S. C. Liew. Offered load control in IEEE802.11 multi-hop ad-hoc networks. *Proc. IEEE ICMASS*, Oct. 2005.

[NVG05] P. Bahl N. Vaidya and S. Gupta. Distributed fair scheduling in a wireless LAN. *IEEE Transaction on Mobile Computing*, pages 616–629, 2005.

[pap99a] IEEE Standards Department, Part II: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications,. *IEEE Std 802.11, IEEE Press*, 1999.

[pap99b] Supplement to ieee standard for information technology telecommunications and information exchange between systems - local and metropolitan area networks - specific requirements. part 11: wireless lan medium access control (mac) and physical layer (phy) specifications: high-speed physical layer in the 5 ghz band. *IEEE 802.11 Standard*, Dec. 1999.

[pap04] Proxim, ORiNOCO 11b Client PC Card Specification. 2004.

[PS00] Nikitin P.V. Punnoose, R.J. and D.D. Stancil. Efficient simulation of ricean fading within a packet simulator. *Proc. of IEEE Vehicular Technology Conference*, pages 764–767, 2000.

[SBG56] S. Lee S. Bae, K. Xu and M. Gerla. Measured analysis of TCP behavior across multihop wireless and wired networks. *Proc. IEEE Global Telecommunications Conference*, pages 25–33, Jan. 20056.

[S.J94] S.J.Golestani. A self-clocked fair queueing scheme for broadband application. *IEEE INFOCOM*, 1994.

[SME05] M. K. Vernon S. M. Elrakabawy. Improving TCP performance for multihop wireless networks. *Proc. DSN*, 2005.

[SX04] R.Shorey S.Bansal and A.A. Xherani. Performance of TCP and UDP protocols in multi-hop multi-rate wireless networks. *Proc. IEEE WCNC*, 2004.

[SZ04] M. Shen and D. Zhao. TCP performance issues over wireless links. *IEEE Communication Magazine*, pages 52–58, 2004.

[TIA05] E. Hossain T. Issariyakul and A. S. Alfa. Markov-based analysis of end-to-end batch transmission in a multi-hop wireless network. *Proc. of IEEE International Conference on Communications (ICC)*, May 2005.

[TIA06] E. Hossain T. Issariyakul and A. S. Alfa. End-to-end batch transmission in a multi-hop and multi-rate wireless network: Latency, reliability, and throughput analysis. *IEEE Transactions on Mobile Computing*, pages 1143–1155, Sept. 2006.

[WAK02] S. Banerjee W. Atikom and P. Krishnamurthy. Starvation prevention and quality of service in wireless LANs. *The 5th International Symposium on Wireless Personal Multimedia Communications*, 2002.

[web] http://www.isi.edu/nsnam/ns/.

[web04] Proxim. ORiNOCO 11b Client PC Card Specification, 2004.

[WM95] H.S. Wang and N. Moayeri. Finite-state markov channel – a useful model for radio communication channels. *IEEE Trans. Veh. Technol.,*, pages 163–171, Feb. 1995.

[XP04] G. Xylomenos and G. C. Polyzos. TCP performance issues over wireless links. *IEEE Communication Magazine*, pages 52–58, 2004.

[YXC06] J. Lui Y. Xu, Y. Wang and D. Chiu. Balancing throughput and fairness for TCP flows in multihop ad-hoc networks. *CUHK technical report*, Oct. 2006.

[ZFG03] H. Luo S. Lu L. Zhang Z. Fu, P. Zerfos and M. Gerla. The impact of multihop wireless channel on TCP throughput and loss. *Proc. IEEE INFOCOM*, 2003.

[Zha07] D. Zhao. Throughput fairness in infrastructure-based IEEE 802.11 mesh networks. *IEEE Transactions on Vehicular Technology*, Sept. 2007.