

國立交通大學 資訊工程學系

博士論文

無線區域網路通道之存取與 QoS 排程機制

Wireless Channel Access and QoS Scheduling Schemes for WLANs

研究生：葉向榮

指導教授：曾煜棋 教授

中華民國九十四年十月

無線區域網路通道之存取與 QoS 排程機制

Wireless Channel Access and QoS Scheduling Schemes for WLANs

研究生：葉向榮

Student : Shiang-Rung Ye

指導教授：曾煜棋

Advisor : Yu-Chee Tseng

國立交通大學
資訊工程學系
博士論文



A Dissertation

Submitted to Department of Computer Science and Information Engineering

College of Electrical Engineering and Computer Science

National Chiao-Tung University

in Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

in

Computer Science and Information Engineering

Hsinchu, Taiwan, Republic of China

October 2005

中華民國九十四年十月

無線區域網路通道之存取與 QoS 排程機制

學生：葉向榮

指導教授：曾煜棋教授

國立交通大學資訊工程研究所

摘 要

於無線網路裏，媒體存取控制為仲裁移動站之間使用無線通道的一項機制。此機制最主要的目的是讓網路能夠提供最大的效能和高品質服務。網路的效能會受到許多因素所影響。當移動站傳資料之前，會先執行一個隨機延遲程序。一個計設良好的延遲程序可以有效地降低資料碰撞的機率，進而提升網路的效能。我們提出了一個多鍊結的延遲方法。它能使網路達到更高的效能，並且讓無線通道被公平的使用。此方法利用多個延遲鍊結，每個延遲鍊結分別適用在不同的網路擁擠狀態。移動站可以藉由使用不同的延遲鍊結，以降低碰撞的機率。在多重跳躍網路裏，移動站隱藏問題也會對網路的效能造成影響。我們在無線通道裏以傳送雜訊以佔據無線通道的方式來避免移動站隱藏問題的發生。此方法把來源移動站和目的地移動站的資料，分別放在不同的無線通道裏傳送，並在這兩個無線通道裏，傳送雜訊以防止移動站隱藏問題的發生。在多媒體網路裏，frame dropping rate 可以用來衡量一個網路是否提供高品質服務。我們對 IEEE 802.11e 的 transmission opportunity (TXOP) 機制，做了一些改善。增加了一個 deadline constraint，限制移動站在 TXOP 裏，只能傳緊急的資料，以避免別的移動站的緊急資料，無法在時限內傳出去。根據這個機制，我們提出一個 deadline constraint scheduling algorithm。它能有效地降低 frame dropping rate 並且增加網路的效能。我們對以上所提的方法，都有評量其效能，並且和文獻上的方法做比較。並做了一些數學上的分析，和討論如何選取參數，使網路達到最大效能。

關鍵字 — 802.11, 802.11e, 移動站隱藏問題, 媒體存取控制, 高品質服務, 隨機延遲, 排程, 無線區域網路.

Wireless Channel Access and QoS Scheduling Schemes for WLANs

Student : Shiang-Rung Ye

Advisor : Dr. Yu-Chee Tseng

Department of Computer Science and Information Engineering

National Chiao Tung University

ABSTRACT

In wireless networks, medium access control (MAC) plays a significant role in coordinating mobile stations' access to the wireless channel. This coordination is desirable in order to achieve the goals of maximizing channel throughput and supporting quality of service (QoS). The throughput of a MAC protocol is influenced by several factors. In most WLANs, a backoff procedure is executed before frame transmission begins. This procedure helps to prevent frame collision and increase the throughput. We propose a multi-chain backoff algorithm that employs multiple backoff chains to provide a higher throughput while maintaining fair channel access. In a multi-hop ad hoc network, the hidden terminal problem also has a significant impact on the channel throughput. In this respect, we propose a jamming-based MAC protocol which separates source stations' traffic from destination stations' traffic into different channels to prevent the hidden terminal problem. For multimedia applications, frame dropping rate is another useful metric to measure how well a scheduling algorithm can support real-time applications. We propose a deadline-constraint scheduling algorithm for IEEE 802.11 WLANs. The scheduling algorithm imposes a deadline constraint on each TXOP to make urgent frames to be transmitted before their deadlines. The performance of the above access scheme and scheduling algorithm are evaluated and compared to prior arts. For some of them, we also analyze their performance and show how to choose proper parameters to obtain optimal throughput.

Key words — 802.11, 802.11e, hidden terminal problem, medium access control, quality of service, random backoff, scheduling, WLANs.

Acknowledgements

I would like to express my sincere gratitude to my advisor Prof. Yu-Chee Tseng for his patient guidance over the past years. Without his supervision, assistant, and perspicacious advices, I can not complete this dissertation. Also thank my committee members: Prof. Rong-Hong Jan, Prof. Li-Chun Wang, Prof. Jang-Ping Sheu, Prof. Wanjiun Liao, Prof. I-Shyan Hwang, and Prof. Yuh-Shyan Chen. Their invaluable comments on this dissertation make the dissertation a perfect work.

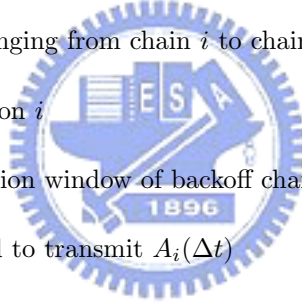
I also would like to give my thanks to colleague in Lab HSCC for their helps and the beautiful memory we share during these years.

Finally I am very grateful to my parents and fiancée Ganhua for their firm supports encouragement over the past years.

Notations

- α : the ratio of the bandwidth of R channel to that of S channel
- $\hat{\alpha}$: the optimal α that gives the maximum throughput
- χ_i : the probability that a station will detect a collision event on the wireless channel during its backoff period in stage 0 of chain i
- ρ_i : the mean data rate of traffic stream i
- ρ_i^h : the peak data rate of traffic stream i
- ρ_i^l : the minimum data rate of traffic stream i
- τ : the probability that a station will transmit in a randomly chosen backoff slot
- τ_i : the HOLD of the i -th traffic stream in the sorted list.
- (i,j,k) : a backoff state at stage j of chain i with backoff value k
- $A_i(\Delta t)$: the maximum amount of data generated by traffic stream i during a time interval Δt ,
- B_i : the maximum burst size of traffic stream i
- $b_{i,j,k}$: the stationary probability that a station is in state (i, j, k)
- C : the set of all traffic streams in the network
- C_i : the set of traffic streams of station i
- c : the number of backoff chains
- D_i : the delay bound of traffic stream i
- $HOLD_i$: the deadline of the head-of-line frame of traffic stream i
- L_i : the nominal frame size of traffic stream i
- M_i : the maximum frame size of traffic stream i

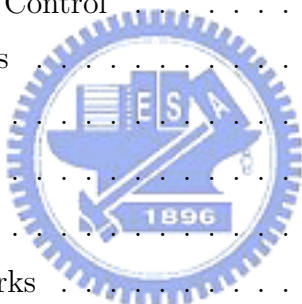
- m_i : the maximum backoff stage of backoff chain i
- $P_{i,j,k|i',j',k'}$: the probability that a station transits from state (i', j', k') to (i, j, k)
- p : the collision probability of each transmission
- Q_i : the current bucket size of station i
- R_i : the minimum PHY transmission rate of traffic stream i
- T_B : the beacon interval
- T_{cp} : the total time reserved for contention periods in a beacon interval.
- $TXOP_i$: the transmission opportunity for traffic stream i or station i
- t_{nb} : the next beacon transmission time
- u_i : the probability of changing from chain i to chain $i + 1$, given $f_{col} = 1$
- v_i : the probability of changing from chain i to chain $i - 1$, given $f_{col} = 0$
- r_i : the token rate of station i
- w_i : the minimum contention window of backoff chain i
- $Z_i(\Delta t)$: the time required to transmit $A_i(\Delta t)$



Contents

1	Introduction	1
1.1	Performance Issue	1
1.2	Quality-of-Service Issue	3
2	Related Work	5
2.1	Random Backoff	5
2.2	The Hidden Terminal Problem	6
2.3	QoS Scheduling for IEEE 802.11e WLANs	12
3	A Multi-Chain Backoff Algorithm	16
3.1	The Proposed Backoff Algorithm	16
3.2	Performance Analysis	18
3.2.1	Saturation Throughput	19
3.2.2	Optimal Values of u and v	23
3.3	Performance Evaluation	23
3.3.1	Saturation Throughput	25
3.3.2	The Number of Backoff Chains	25
3.3.3	Comparison to Existing Algorithms	27
3.4	Concluding Remarks	33
4	A Jamming-Based MAC Protocol to Improve the Performance of Wireless Multihop Ad Hoc Networks	36

4.1	The Proposed Protocol	37
4.2	Tuning the Factor α	40
4.3	Performance Evaluation	42
4.4	Concluding Remarks	46
5	A Deadline-Constraint Scheduling Algorithm for IEEE 802.11e WLANS	50
5.1	The Proposed Deadline-Constraint Scheduling Algorithm	50
5.1.1	Length Constraint	51
5.1.2	Deadline Constraint	52
5.1.3	Updating HOLDS	53
5.1.4	Allocation of TXOPs	53
5.1.5	Admission Control	54
5.2	Simulation Results	54
5.2.1	Scenario 1	55
5.2.2	Scenario 2	56
5.2.3	Scenario 3	58
5.3	Concluding Remarks	61
6	Conclusions and Future Work	62
6.1	Conclusions	62
6.2	Future Work	63



List of Tables

3.1	Simulation parameters.	24
4.1	MAC Layer parameters.	45
4.2	Physical layer parameters.	45
4.3	Parameters of the simulated wireless network.	45
5.1	MAC parameters used in our simulation	55
5.2	The traffic parameters for VoIP and MPEG TSs in scenario 1.	56
5.3	The TSPECs of MPEG TSs in scenario two.	59
5.4	Throughput, dropping rate, and access delay in scenario two.	59
5.5	TSPECs of TS1 and TS2 in scenario three.	60
5.6	Throughput, dropping rate, and access delay for different traffic sources in scenario three when the delay bound of TS2 is 30 ms.	60
5.7	Throughput, dropping rate, and access delay for different traffic sources in scenario three when the delay bound of TS2 is 60 ms.	61

List of Figures

1.1	The hidden terminal problem.	3
2.1	Erroneous reservations caused by: (a) busy destination, (b) frame collision, (c) transmission error, and (d) the problem itself.	10
2.2	(a) TXOP without deadline constraint mechanism, and (b) TXOP with deadline constraints mechanism. The number in each block is the deadline of the frame.	15
3.1	The transition diagram of MCB. The (i, j) denotes j -th backoff stage of chain i . Symbols s and f denote success and failure transmission events, respectively.	17
3.2	The Markov chain model of MCB.	19
3.3	The ratio of optimal u to v under different n and c	24
3.4	Saturation throughput under different u and v with the frame size of 1024 bytes.	26
3.5	Saturation throughput under different u and v with the frame size of 128 bytes.	26
3.6	Saturation throughput versus frame sizes.	27
3.7	Throughput of MCB with u and v which are chosen for $n = 6$ and $n = 46$	28
3.8	Fairness Index.	28

3.9	Saturation throughput of four-chain MCB : analysis versus simulation.	29
3.10	Saturation throughput of MCB and GDCF for frame size 128 bytes.	30
3.11	Saturation throughput of MCB and GDCF for frame size 1024 bytes.	31
3.12	Fairness index of MCB and GDCF for frame size 1024 bytes. . .	31
3.13	Saturation throughput of MCB, IEEE 802.11, and MILD.	32
3.14	Saturation throughput of MCB and EIED(x,y) with fixed $y = 2$. . .	33
3.15	Throughput of MCB and EIED(x,y) with fixed $x = 2$	34
3.16	Fairness index of MCB and EIED(x,y) with fixed $x = 2$	34
4.1	The procedures of transmission and receipt of control/data frames in JMAC.	38
4.2	The timing diagram of transmission of control/data frames.	38
4.3	Frame exchange and transmission of jamming signal in JMAC. . .	41
4.4	Concurrent transmissions in JMAC.	41
4.5	Data frame size vs. the optimum alpha $\hat{\alpha}$	43
4.6	Frame exchange time in IEEE 802.11 and in JMAC for different data frame sizes and different α	43
4.7	Aggregate throughput versus frame arrival rate.	45
4.8	Mean throughput versus frame arrival rate.	47
4.9	Aggregate throughput versus network density.	47
4.10	Mean throughput versus network density.	47
4.11	Aggregate throughput versus frame size.	48
4.12	Mean throughput versus frame size.	48
4.13	Mean access delay versus frame arrival rate.	48
5.1	Throughputs comparison under different numbers of stations in scenario one.	57

5.2 Frame dropping rate under different numbers of stations in scenario one. 57

5.3 Comparison of access delay under different numbers of stations in scenario one. 58



Chapter 1

Introduction

With the maturity of wireless communications technologies, more and more wireless local area networks (WLANs) have been deployed in schools, airports, railway stations, megastores, cafes, etc.. The research on WLANs can be classified into two directions. One explores the modulation and encoding schemes of wireless communications systems with the goals of providing higher data rates and lower bit error rates. The other research direction concerns MAC sublayer. The MAC sublayer further divides into two research areas: the network management and channel access. The network management includes authentication, association, security, etc.. Channel access studies channel allocations such as random access and centralized scheduling. In this dissertation, we focus on two issues of the channel access: the performance issue and QoS issue.

1.1 Performance Issue

The performance of a MAC protocol is influenced by several factors. In most WLANs, before transmission, a random backoff procedure is executed in order to prevent frame collision. In 1970s Norman Abramson and his colleagues first proposed an elegant MAC protocol, called ALOHA [1]. In ALOHA, stations

are allowed to transmit immediately upon receiving data from upper layers. A variant of ALOHA divides time into contiguous time slots and allows transmission at beginning of a time slot. This reduces the vulnerable time of the pure ALOHA. It has been shown that with Poisson arrival process, the maximum throughputs for pure ALOHA and slotted ALOHA are only 0.184 and 0.368, respectively [2].

The inefficacy of ALOHA protocols results from its high collision probability in heavy traffic load. To decrease the collision probability, carrier sense multiple access (CSMA) scheme [2] requires stations to sense carriers on the wireless channel before transmitting data. In this scheme, if the medium is busy, stations have to defer their transmission until the medium becomes idle. This prevents stations's frames from colliding with ongoing transmission frames of other stations. When a station detects the medium is busy, it can persistently wait for the medium to become idle, and then transmit with a probability of one or p , $0 < p < 1$. The former is called 1-persistent CSMA and the latter is p -persistent CSMA. Alternatively, a station can stop monitoring the wireless medium. After a random time period, it listens to the medium again to check whether the medium has become idle. This is called nonpersistent CSMA.

The p -persistent CSMA requires a station to wait for a random period of time before transmitting their data after the medium becomes idle. This is called a backoff before transmission. In p -persistent CSMA, a geometry backoff is used to prevent frame collision. Much effort has been devoted to designing adaptive backoff algorithms. The goal is to offer a higher throughput and maintain fair channel access at the same time. On this issue, we propose a multi-chain backoff algorithm that uses multiple backoff chains for stations to adapt to different congestion levels.

When ALOHA or CSMA protocols are applied in multihop ad hoc networks, they will suffer from the well-known hidden-terminal problem. An *ad*

hoc network is a spontaneous network that consists solely of mobile stations without base stations. It can be applied in many contexts such as military communication, disaster rescue, and outdoor/indoor activities due to its fast deployment and flexibility in reconfiguration. In a multihop ad hoc network, a source station may be n -hop away from its destination station. It relies on intermediate stations to forward its messages. The hidden-terminal problem occurs when the neighbors of the destination station transmit while the destination station is receiving data. Fig. 1.1 shows an example. Station B is in the transmission range of station A and C, but station C is out of the transmission range of station A. During the transmission from station A to station B, if station C transmits to D, the frames from station A to station B will be corrupted at station B. In the literature, the hidden terminal problem can

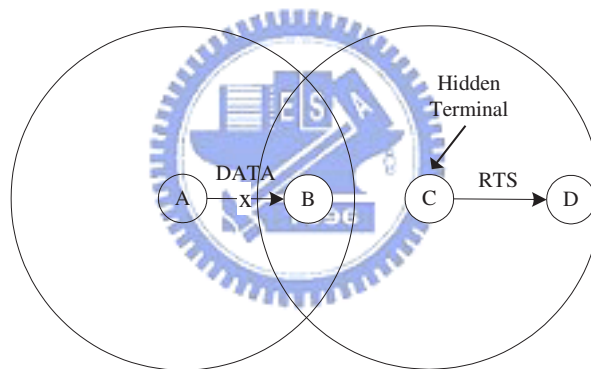


Figure 1.1: The hidden terminal problem.

be solved by the well-known busy tone solution. We propose a *jamming-based MAC (JMAC)* protocol which doesn't require extra bandwidth for busy tones but still can solve the hidden terminal problem.

1.2 Quality-of-Service Issue

With the increasing popularity of multimedia applications, the demand for quality-of-service (QoS) is expected to grow. The IEEE 802.11e [3] is de-

signed to support QoS in WLANs. It defines two channel access schemes: *enhanced distributed channel access (EDCA)* and *HCF controlled channel access (HCCA)* [3–5]. The EDCA is a contention-based channel access, providing prioritized QoS, and the HCCA is a contention-free channel access that supports parameterized QoS. In HCCA, a station can specify its traffic parameters, called *traffic specification (TSPEC)*, and submit it to the hybrid coordinator (HC). The HC may admit or reject the TSPEC, depending on the current available bandwidth. For those admitted TSPECs, HC computes a schedule and allocates TXOPs to the TSs by polling the corresponding stations.

Scheduling of TSs' transmissions is one of core components of HCCA. We propose a deadline-constraint scheduling algorithm which can decrease the frame dropping rate and increase the network throughput. The rest of this dissertation is organized as follows. In Chapter 2 we review prior work on random backoff, the hidden terminal problem, and QoS scheduling. For these issue, we propose our solutions in Chapter 3, Chapter 4, and Chapter 5, respectively. Chapter 6 concludes this dissertation and gives some future research directions.

Chapter 2

Related Work

2.1 Random Backoff

In the literature, there has been many studies of backoff algorithms [6–16]. In [6], to prevent the contention window of binary exponential backoff (BEB) from oscillation, the *multiplicative increase, linear decrease (MILD)* algorithm increases the contention window by 1.5 times when collision occurs, and decreases the contention window by one when transmission succeeds. To ensure fair access to the wireless medium, a station is required to attach its contention window in each transmitted frame. Whenever other station overhears this frame, it has to adopt the window size. However, since a station with a smaller contention window has a better chance to win the channel, this may force other stations with a large window size to adopt this small window size. When the network is in high traffic load, this may increase collision probability. In [7, 9], it is suggested to choose a contention window according to the estimated number of competing stations. While this may significantly improve performance, it relies on the accurate estimation of the number of competing stations.

The *exponential increase, exponential decrease (EIED)* algorithm [14, 17, 18]

increases the contention window by a multiple when collision occurs and exponentially decreases the window size when transmission succeeds. With a relatively small decrement of the window size, compared to the increment, the EIED can outperform 802.11 DCF. However, our simulation results show that with such a relatively small decrement, the EIED may suffer from unfair channel access when the number of contending stations is small. The *linear increase, linear decrease (LILD)* algorithm always adjusts the contention window by a constant [18, 19]. This is not suitable for the network with a large population. The GDCF backoff algorithm [20] doubles the contention window after each unsuccessful transmission and halves the window size after c consecutive successful transmissions. Although GDCF greatly improves the performance of 802.11 DCF, it may cause unfair medium access for some values of c . The works [8, 10] show that the performance of DCF is highly related to the number of contending stations and the minimum contention window (CW_{min}). The CW_{min} has to change with the number of contending stations. Motivated by this observation, we propose a multi-chain backoff algorithm that employs multiple backoff chains, each of which is used in a different congestion level; thereby, stations can adapt to different congestion levels by switching among the chains.

2.2 The Hidden Terminal Problem

The BTMA [21] first proposes a busy tone solution to avoid the hidden-terminal problem in an infrastructure network. In the BTMA, the whole bandwidth is divided into a data channel and a busy-tone channel. When a base station detects carrier on the data channel, it transmits busy-tone signal (a sine wave) to indicate the busy state of the data channel. Other mobile station that detects the busy-tone signal will defer its transmission until the end of the busy tone.

In the receiver-initiated busy-tone multiple protocol [22], a receiver will transmit a busy tone after it receives a request frame from a sender. This busy tone not only serves as a response to the request but also prevents other stations from accessing the channel; other attempting source that detects the busy tone will postpone its transmission. However, in a multihop ad hoc network, a sender may overhear other stations' busy tone and cannot tell whether its receiver or other station turns on the busy tone.

Based on the RTS-CTS-DATA dialogue, DBTMA protocol [23] [24] uses two busy tones to convey channel states. It divides the communication bandwidth into a control channel, a data channel, and two busy tones (BT_t and BT_r). RTS and CTS frames are exchanged on the control channel. If the RTS-CTS exchange succeeds, the source station will transmit its data on the data channel and turn on the BT_t . Similarly, while the destination is receiving data, it turns on the BT_r . The neighbors of the source station and destination station can tell whether they are hidden terminal or expose terminal by sensing the two busy tones. However, DBTMA require extra bandwidth for the two busy tones and an extra transmitter hardware. Moreover, the busy tone detection time is not negligible and must be taken into consideration [21].

Another solution to the hidden terminal problem is the RTS-CTS reservation scheme. The MACA protocol [25] first introduces the RTS-CTS exchange to prevent the hidden-terminal problem. In this scheme, a source station transmits a RTS frame to the destination station before transmitting its data frame. If the intended destination correctly receives the RTS frame, it sends a CTS frame to the source station. Other stations that overhear the RTS or CTS frame will reschedule their transmissions to a later time. In contrast to MACA, the MACA-BI [26] which is a receiver-oriented protocol tries to improve the channel utilization by removing the RTS part of RTS/CTS handshake. A destination station sends a RTR (ready-to-receive) frame to invite

a source station to transmit data. The RTR frames are transmitted at a rate that matches the source station's incoming traffic rate. However, this method is not suitable for networks with bursty traffic.

The MACAW [25] protocol suggests a new frame exchange sequence, RTS-CTS-DS-DATA-ACK. A DS (Data-Sending) frame is transmitted by a source station to confirm the use of the medium after it receives a CTS frame. However, since a DS frame may collide with frames of other stations, neighboring stations may not correctly receive the DS frame. The ACK frame is transmitted after a receiver correctly receives a data frame. This acknowledgement improves the reliability of a wireless link, and avoids the long recovery cost at the upper layer (e.g., TCP).

The IEEE 802.11 standard adopts the RTS-CTS-DATA-ACK. As in MACAW, RTS and CTS frames may be lost due to collision or channel error. So the hidden-terminal problem remains unavoidable (but will be reduced). Besides, the use of RTS/CTS in 802.11 also causes an erroneous reservation problem, which may occur when RTS/CTS exchange fails but the channel is reserved by the RTS/CTS frames. This inhibits neighboring stations from accessing the medium even though the medium is idle. Such "holding-while-waiting" scenario may waste much bandwidth when the traffic load is high. The erroneous reservation may occur under the following conditions: busy destination, frame collision, transmission error, and the problem itself. For example, in Fig. 2.1(a), assume that stations A and B have successfully completed RTS/CTS exchange and started their transmission. In the meanwhile, if station D transmits a RTS to C , the circle centered at D will be incorrectly reserved. Similarly, if E sends a RTS to F , the circle centered at E will be incorrectly reserved too.

Frame collision may also cause erroneous reservations. Fig. 2.1(b) shows that if the CTS_Timeout interval is smaller than the length of a CTS frame,

the erroneous reservation may occur. In the example, A and C transmit RTS frames at the same time. The RTS frame from A is collided with that from C , but the RTS frame from C is successfully transmitted to D . While D responds with a CTS frame to C , if A retries to send a RTS frame, the CTS frame is collided at C and the circle centered at D is erroneously reserved.

Fig. 2.1(c) demonstrates that transmission errors can also cause erroneous reservations. In the figure, B successfully receives the RTS from A , but A fails to receive the CTS from B due to transmission error. Then the circles centered at both A and B will be incorrectly reserved. It is also possible that after a station incorrectly reserves the channel, this incorrect reservation causes another reservation. In Fig. 2.1(d), after A incorrectly reserves the channel, D also incorrectly reserves the channel.

Besides, the use of RTS/CTS in 802.11 also causes an erroneous reservation problem, which may occur when RTS/CTS exchange fails but the channel is reserved by the RTS/CTS frames. This inhibits neighboring stations from accessing the medium even though the medium is idle. Such “holding-while-waiting” scenario may waste much bandwidth when the traffic load is high. The erroneous reservation may occur under the following conditions: busy destination, frame collision, transmission error, and the problem itself. For example, in Fig. 2.1(a), assume that stations A and B have successfully completed RTS/CTS exchange and started their transmission. In the meanwhile, if station D transmits a RTS to C , the circle centered at D will be incorrectly reserved. Similarly, if E sends a RTS to F , the circle centered at E will be incorrectly reserved too.

Frame collision may also cause erroneous reservations. Fig. 2.1(b) shows that if the CTS_Timeout interval is smaller than the length of a CTS frame, the erroneous reservation may occur. In the example, A and C transmit RTS frames at the same time. The RTS frame from A is collided with that from C ,

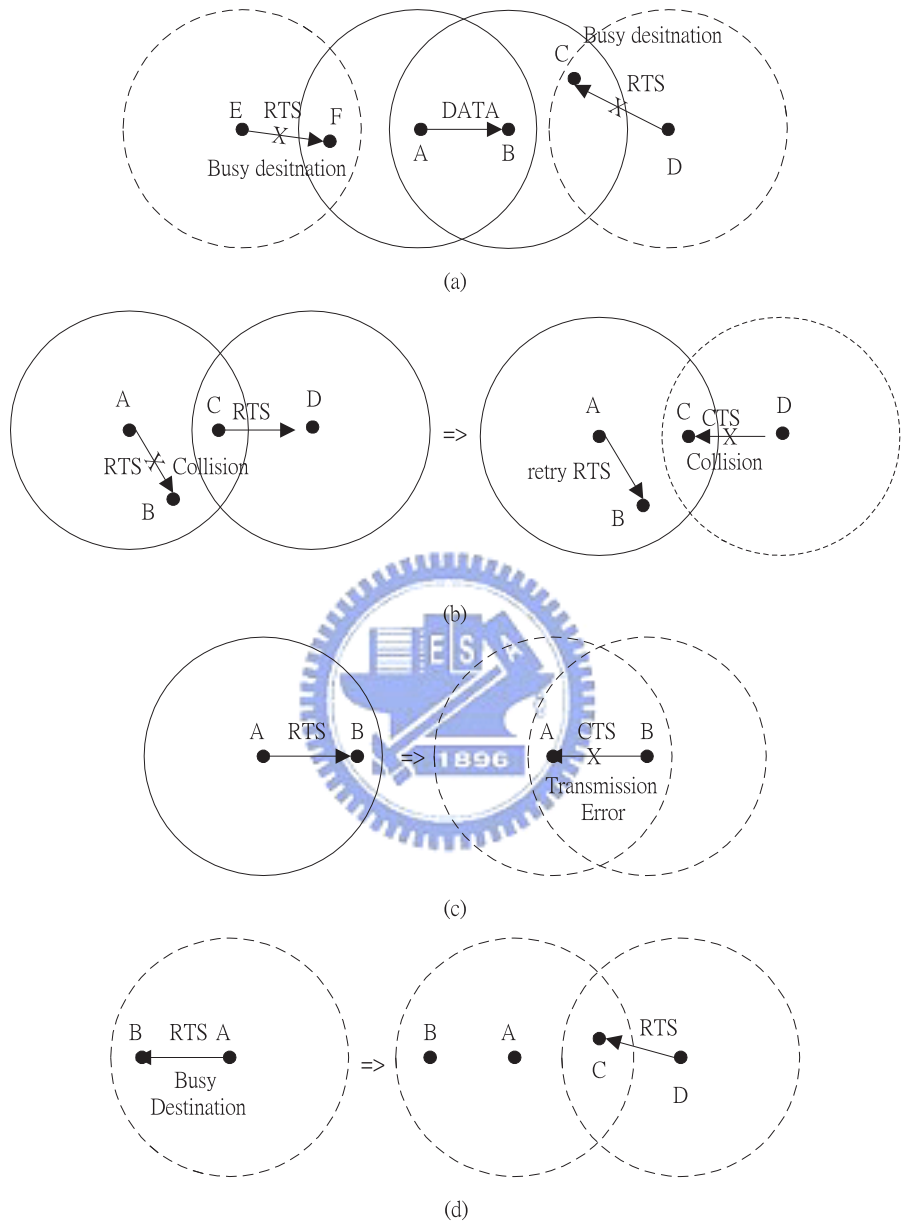


Figure 2.1: Erroneous reservations caused by: (a) busy destination, (b) frame collision, (c) transmission error, and (d) the problem itself.

but the RTS frame from C is successfully transmitted to D . While D responds with a CTS frame to C , if A retries to send a RTS frame, the CTS frame is collided at C and the circle centered at D is erroneously reserved.

Fig. 2.1(c) demonstrates that transmission errors can also cause erroneous reservations. In the figure, B successfully receives the RTS from A , but A fails to receive the CTS from B due to transmission error. Then the circles centered at both A and B will be incorrectly reserved. It is also possible that after a station incorrectly reserves the channel, this incorrect reservation causes another reservation. In Fig. 2.1(d), after A incorrectly reserves the channel, D also incorrectly reserves the channel.

Frame collision may also cause erroneous reservations. Fig. 2.1(b) shows that if the CTS_Timeout interval is smaller than the length of a CTS frame, the erroneous reservation may occur. In the example, A and C transmit RTS frames at the same time. The RTS frame from A is collided with that from C , but the RTS frame from C is successfully transmitted to D . While D responds with a CTS frame to C , if A retries to send a RTS frame, the CTS frame is collided at C and the circle centered at D is erroneously reserved.

Fig. 2.1(c) demonstrates that transmission errors can also cause erroneous reservations. In the figure, B successfully receives the RTS from A , but A fails to receive the CTS from B due to transmission error. Then the circles centered at both A and B will be incorrectly reserved. It is also possible that after a station incorrectly reserves the channel, this incorrect reservation causes another reservation. In Fig. 2.1(d), after A incorrectly reserves the channel, D also incorrectly reserves the channel.

In this dissertation, we propose a *jamming-based MAC (JMAC)* protocol that can satisfactorily solve the hidden terminal problem. The basic idea behind the JMAC is to separate source stations' traffic from destination stations' traffic into different channels, and explicitly signal the channel status by jam-

ming the channels. Although the division of the shared medium into two channels incurs some cost, as shown in simulation results, the advantages of being free from the and the hidden terminal problem and erroneous reservation problem, and the benefits of more concurrent transmissions will compensate the cost and provide higher channel utilization when data frame size is median or large.

2.3 QoS Scheduling for IEEE 802.11e WLANs

In IEEE 802.11e, channel access is based on the concept of TXOP. A TXOP is a time period during which a station has exclusive access to the wireless medium. During a TXOP, a station can transmit more than one frame before its TXOP expires. There are two types of TXOPs: *EDCA TXOP* and *HCCA TXOP*. An EDCA TXOP is obtained via contention according to EDCA rules, while a HCCA TXOP is assigned by HC. The lengths of a EDCA TXOP and consecutive HCCA TXOPs are limited by `dot11EDCATableTXOPLimit` and `dot11CAPLimit`, respectively.

In HCCA, HC can allocate a HCCA TXOP in a contention period (CP) or in a contention-free period (CFP) by transmitting a QoS(+)Poll frame to a station. The QoS(+)Poll represents a set of the following polling frames: QoS Poll, QoS Data+CF-Poll, QoS CF-Ack+CF-Poll, and QoS Data+CF-Ack+CF-Poll.

A station can request a HCCA TXOP in two different ways. It may send a data frame with a QoS control field to specify its desired TXOP length or its queue size. In the other way, it can request a traffic stream (TS) by sending a request frame with a TSPEC. A TSPEC is used to specify traffic parameters of a TS. The parameters include mean data rate, peak data rate, minimum data rate, nominal MSDU size, maximum MSDU size, minimum service interval, and maximum service interval. The last two items specify the minimum and

the maximum time intervals, respectively, between two successive TXOPs of a TS. For burst traffic, it can specify its maximum burst size, which is defined to be the maximum amount of data arriving at the *MAC Service Access Point (MAC_SAP)* at peak data rate. For real-time traffic, it can specify a delay bound, which is the time between the arrival of a MSDU at the MAC sublayer and the end of the successful reception of an ACK frame. A minimum PHY rate also has to be specified. For each TS, it has an access policy. The access policy can be HCCA, EDCA, or the hybrid of HCCA and EDCA. With the hybrid access policy, a TS can be polled as well as being transmitted in contention periods.

IEEE 802.11e/D13 [3] provides a simple scheduling algorithm for allocating HCCA TXOPs. This scheduler defines a common *service interval (SI)* for all TSs. The length of the SI is set to the minimum one of maximum service intervals of all TSs in the network. During a SI, each TS i is allocated a HCCA TXOP with a length

$$TXOP_i = \max \left(\left\lceil \frac{SI \cdot \rho_i}{L_i} \right\rceil \cdot \left(\frac{L_i}{R_i} + O \right), \frac{M_i}{R_i} + O \right), \quad (2.1)$$

where ρ_i is the mean data rate, L_i is the nominal MDSU size, R_i is the minimum PHY rate, M_i is the maximum MDSU size of TS i , and O is a fixed communication overhead. The admission control can be done as follows. When a station requests a TS k , we first compute its $TXOP_k$ according to Eq. (2.1). Then, the request is admitted if

$$\frac{TXOP_k}{SI} + \sum_{j \in C} \frac{TXOP_j}{SI} \leq \frac{T_B - T_{cp}}{T_B}, \quad (2.2)$$

where C is the set of TSs which have already been admitted to the network, T_B is the beacon interval, and T_{cp} is the time reserved for contention periods in a beacon interval.

The SETT-EDD (Scheduling Based on Estimated Transmission Times - Earliest Due Date) algorithm [27] computes an aggregate schedule to poll sta-

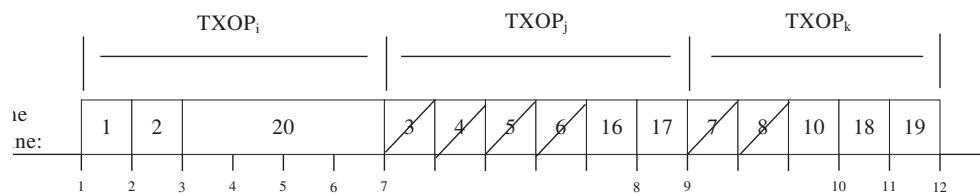
tions. For each station v , it computes a minimum service interval, mSI , and a maximum service interval, MSI , as follows:

$$mSI_v = \min_{i \in C_v} (L_i / \rho_i)$$

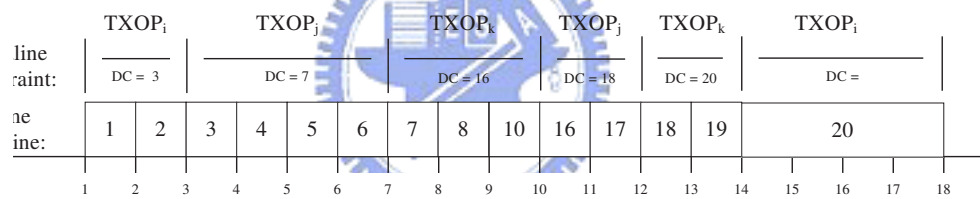
$$MSI_v = \min_{i \in C_v} (\beta \cdot (D_i - MTD))$$

where C_v is the set of TSs of station v , D_i is the delay bound of TS i , MTD is the maximum length of a HCCA TXOP, and $0 < \beta \leq 1$. Let t_v be the last time when station v was polled. Then station v must be polled again during $(t_v + mSI_v, t_v + MSI_v)$. Since there may be more than one station satisfying the condition, HC will choose the station i with the smallest $t_i + MSI_i$ to poll. The length of the TXOP that can be allocated to station i is derived from a token bucket. The token bucket has a depth of MTD and a token rate of $\sum_{j \in C_i} TXOP_j / mSI_i$, where $TXOP_j$ is calculated by Eq. (2.1) with mSI_i as the service interval. This token bucket is maintained by HC.

The drawback of SETT-EDD algorithm is that once a TXOP is allocated to a station, the station can transmit urgent as well as not urgent frames as long as the TXOP does not expire. This may cause that other stations drop their urgent frames. Fig. 2.2(a) shows an example. In this example, the transmission of the third frame of station i causes four frames of station j being dropped, which further causes two frames of station k being dropped. To relieve this problem, we propose imposing a deadline constraint, in addition to the length constraint, on each TXOP to restrict frames to be transmitted in a TXOP. Fig. 2.2(b) demonstrates the same scenario, but transmissions in a TXOP are restricted by a given deadline. Due to the deadline constraint, there is no frame drop in the example.



(a)



(b)

Figure 2.2: (a) TXOP without deadline constraint mechanism, and (b) TXOP with deadline constraints mechanism. The number in each block is the deadline of the frame.

Chapter 3

A Multi-Chain Backoff

Algorithm

3.1 The Proposed Backoff Algorithm

In MCB, each station maintains a transition diagram, as demonstrated in Fig. 3.1, to determine its contention window. The diagram consists of c backoff chains, numbered from 0 to $c-1$, each of which represents a sequence of backoff stages and is defined by the following parameters:

- w_i : the minimum contention window of chain i .
- m_i : the maximum backoff stage of chain i .
- u_i : the transition probability from chain i to chain $i + 1$. In case of $i = c - 1$, $u_{c-1} = 0$.
- v_i : the transition probability from chain i to chain $i - 1$. In case of $i = 0$, $v_0 = 0$.

For each backoff chain i , we define $w_0 = CWmin$, and $w_{c-1} = CWmax$. For $i = 1 \cdots c - 2$, w_i could be

$$w_i = CWmin + i \cdot \left\lfloor \frac{CWmax - CWmin}{c - 1} \right\rfloor.$$

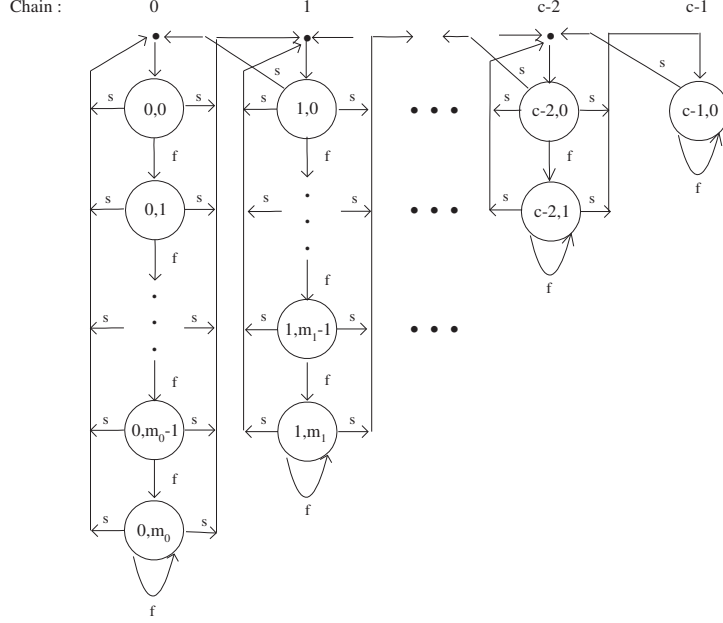


Figure 3.1: The transition diagram of MCB. The (i, j) denotes j -th backoff stage of chain i . Symbols s and f denote success and failure transmission events, respectively.

Alternatively, we may increase w_i in an exponential manner as follows,

$$w_i = (CWmin + 1) \cdot \left[\frac{CWmax + 1}{CWmin + 1} \right]^{\frac{i}{c-1}} - 1 .$$

Within a backoff chain, the contention window is doubled for the next backoff stage but is limited to $CWmax$.

Parameters u_i and v_i are probabilities for a station to switch from its current chain to the next chain and the previous chain, respectively. The simplest assignment is to let all u_i be the same, and all v_i be the same. In this case, the optimal values for u_i and v_i are related to the number of competing stations, the length of a backoff slot, and the average size of data frames. This will be addressed in Section 3.2.2.

With the above defined parameters, the MCB algorithm works as follows. Initially, each station is in stage 0 of chain 0. Before transmitting data, a station randomly chooses a backoff value from the current contention window.

If the medium is idle for a DIFS period, the station starts to perform backoff. For each idle slot being detected, its backoff counter is decreased by one. If the medium is busy during a backoff slot, the backoff counter is frozen and the station has to wait until the medium becomes idle. Once its backoff counter reaches zero, the station can start to transmit a frame. During the backoff period, the station shall also detect any collision event caused by other stations. A collision flag f_{col} will be set to 1 if a station itself experiences a collision or it detects that the medium has been busy for a duration longer than the transmission time of the smallest frame, but this doesn't result in a receipt of a correct frame. Assume that a station is transmitting in stage j of chain i . In case that the transmission fails, it will move to stage $j + 1$ of chain i if $j < m_i$, or stay in the same stage if $j = m_i$. In case that the transmission succeeds, if f_{col} is set, it will move to stage 0 of chain $i + 1$ with probability u_i , but move to stage 0 of chain i with probability $1 - u_i$. In case that the transmission succeeds and the f_{col} is not set, it will move to stage 0 of chain $i - 1$ with probability v_i , but stay in stage 0 of chain i with probability $1 - v_i$. After each successful transmission, the f_{col} is cleared to 0. Intuitively, when a station encounters or detects a collision event, it moves to a chain with a larger minimum contention window with probability u_i , and if no collision is encountered and detected, it moves to a chain with a smaller minimum contention window with probability v_i .

3.2 Performance Analysis

In this section, we analyze the saturation throughput of MCB, which is defined to be the maximum achievable throughput, obtained by continuously increasing the traffic load to a limit. To operate the network in a saturation condition, all transmit queues of stations are assumed to be nonempty all the time. It is also assumed that, under such a condition, the collision probability

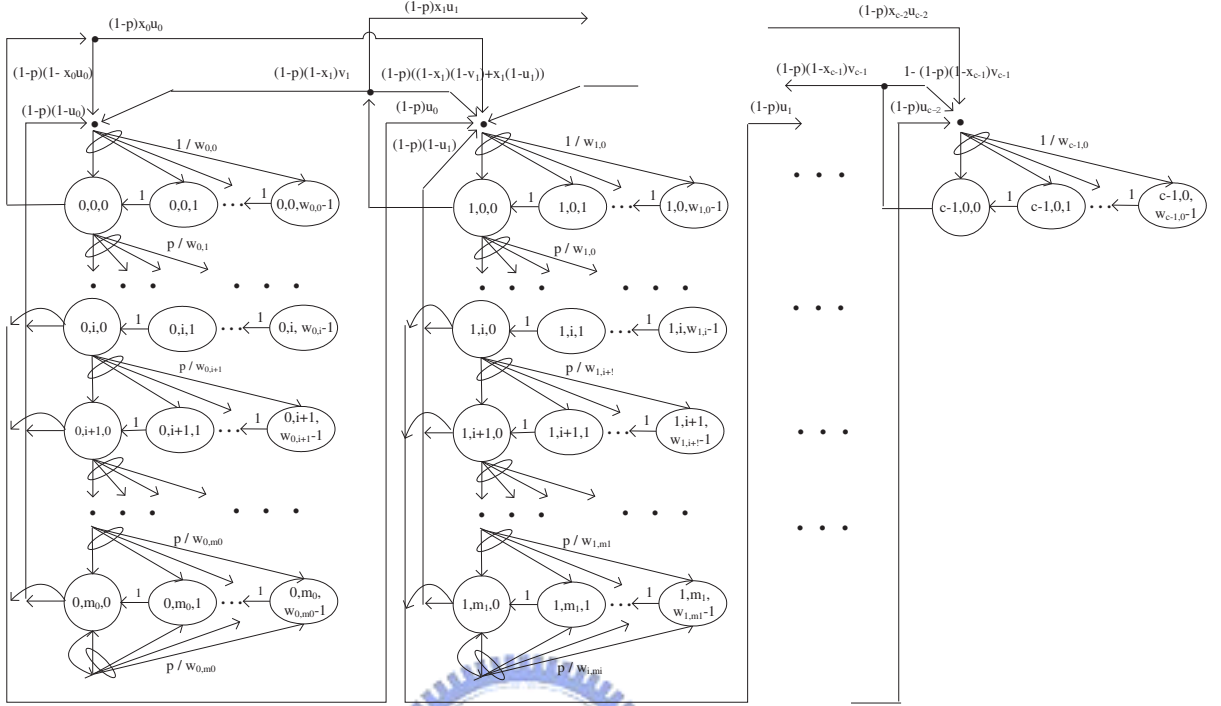


Figure 3.2: The Markov chain model of MCB.

for each transmission attempt is a constant and independent value, p .

3.2.1 Saturation Throughput

Fig. 3.2 shows the Markov chain of MCB. Each state is represented by a triple (i, j, k) , which means that the station is in stage j of chain i and has a backoff value k . Let τ be the probability that a station will transmit in a randomly chosen backoff slot, and let χ_i be the probability that a station will detect a collision event on the wireless channel during its backoff period in stage 0 of chain i . For a given backoff slot, the probability that a station will not detect any collision event is $((1 - \tau)^{n-1} + \tau(n - 1)(1 - \tau)^{n-2})$, where n is the total number of stations. Since a station will choose a backoff value with an equal

probability of $(1/(w_i + 1))$, we have

$$\begin{aligned}\chi_i &= \sum_{k=0}^{w_i} \frac{1 - ((1 - \tau)^{n-1} + (n - 1)\tau(1 - \tau)^{n-2})^k}{w_i + 1} \\ &= 1 - \frac{1 - ((1 - \tau)^{n-1} + (n - 1)\tau(1 - \tau)^{n-2})^{w_i+1}}{(w_i + 1)(1 - (1 - \tau)^{n-1} - (n - 1)\tau(1 - \tau)^{n-2})}.\end{aligned}\quad (3.1)$$

For a station, the transmission probability from stage j of chain i to stage 0 of chain $i + 1$ with probability u_i is $(1 - p) \cdot \chi_i \cdot u_i$. Similarly, transition probability from stage j of chain i transits to stage 0 of chain $i - 1$ is $(1 - p) \cdot (1 - \chi_i) \cdot v_i$. Let $P_{i,j,k|i',j',k'}$ denote the probability that a station transits from state (i', j', k') to (i, j, k) . The nonnull one-step transition probabilities are summarized as follows:

$$\left\{ \begin{array}{l} P_{i,j,k-1|i,j,k} = 1 \\ P_{i,0,k|i,j,0} = \frac{(1-p)(1-u_i)}{W_{i,0}}, 0 < j \leq m_i, i < c - 1 \\ P_{c-1,0,k|c-1,0,0} = \frac{1 - (1-p)(1-\chi_{c-1})v_{c-1}}{W_{c-1,0}}, \\ P_{i,0,k|i,0,0} = \frac{(1-p)(\chi_i(1-u_i) + (1-\chi_i)(1-v_i))}{W_{i,0}}, 0 < i < c - 1 \\ P_{0,0,k|0,0,0} = \frac{(1-p)(1-\chi_0 \cdot u_0)}{W_{0,0}} \\ P_{i,j+1,k|i,j,0} = \frac{p}{W_{i,j+1}}, j < m_i \\ P_{i,m_i,k|i,m_i,0} = \frac{p}{W_{i,m_i}}, i < c - 1 \\ P_{i+1,0,k|i,j,0} = \frac{(1-p)u_i}{W_{i+1,0}}, j > 0, i < c - 1 \\ P_{i+1,0,k|i,0,0} = \frac{(1-p)\chi_i u_i}{W_{i+1,0}}, i < c - 1 \\ P_{i-1,0,k|i,0,0} = \frac{(1-p)(1-\chi_i)v_i}{W_{i-1,0}}, i > 0 \end{array} \right. \quad (3.2)$$

where $W_{i,j} = (w_i + 1) \cdot 2^j$. Let $b_{i,j,k}$ be the stationary probability that a station stays in state (i, j, k) . Since $b_{i,j,0} = b_{i,j-1,0} \cdot p$,

$$\begin{aligned}b_{i,j,0} &= b_{i,0,0} \cdot p^j, 0 < j < m_i \\ b_{i,m_i,0} &= b_{i,0,0} \cdot \frac{p^{m_i}}{(1-p)}\end{aligned}\quad (3.3)$$

With Eq. (3.3), we can express each $b_{i,j,k}$ in terms of $b_{i-1,0,0}$, $b_{i,0,0}$, and $b_{i+1,0,0}$.

For $0 < j < m_i$, and $0 \leq k < W_{i,j}$,

$$\begin{aligned} b_{i,j,k} &= \frac{W_{i,j} - k}{W_{i,j}} \cdot b_{i,j-1,0} \cdot p \\ &= \frac{W_{i,j} - k}{W_{i,j}} \cdot b_{i,0,0} \cdot p^j, \end{aligned} \quad (3.4)$$

and for $j = m_i$, and $0 \leq k < W_{i,m_i}$,

$$\begin{aligned} b_{i,m_i,k} &= \frac{W_{i,m_i} - k}{W_{i,m_i}} (b_{i,m_i,0} \cdot p + b_{i,m_i-1,0} \cdot p) \\ &= \frac{W_{i,m_i} - k}{W_{i,m_i}} \cdot b_{i,0,0} \cdot \frac{p^{m_i}}{1 - p}. \end{aligned} \quad (3.5)$$

In the case that $j = 0$, for $0 < i < c - 1$, and $0 \leq k < W_{i,0}$,

$$\begin{aligned} b_{i,0,k} &= \frac{W_{i,0} - k}{W_{i,0}} (b_{i,0,0} ((1 - u_i)(p - p^{m_i+1}) \\ &\quad + (1 - p)(1 - \chi_i \cdot u_i - v_i + \chi_i \cdot v_i)) \\ &\quad + b_{i-1,0,0} \cdot u_{i-1} (\chi_{i-1} - p\chi_{i-1} + p - p^{m_{i-1}+1}) \\ &\quad + b_{i+1,0,0} \cdot (1 - p) \cdot (1 - \chi_{i+1}) \cdot v_{i+1}). \end{aligned} \quad (3.6)$$

For $j = 0$, $i = 0$, $b_{0,0,k}$, and $0 \leq k < W_{0,0}$,

$$\begin{aligned} b_{0,0,k} &= \frac{W_{0,0} - k}{W_{0,0}} (b_{0,0,0} ((1 - u_0)(p - p^{m_0+1}) \\ &\quad + (1 - p)(1 - \chi_0 \cdot u_0)) \\ &\quad + b_{1,0,0} \cdot (1 - p) \cdot (1 - \chi_1) \cdot v_1), \end{aligned} \quad (3.7)$$

and for $j = 0$, $i = c - 1$, and $0 \leq k < w_{c-1,0}$,

$$\begin{aligned} b_{c-1,0,k} &= \frac{W_{c-1,0} - k}{W_{c-1,0}} (b_{c-2,0,0} \cdot u_{c-2} (p - p^{m_{c-2}+1}) \\ &\quad + (1 - p)\chi_{c-2}) + b_{c-1,0,0} \\ &\quad \cdot (1 - (1 - p)(1 - \chi_{c-1})v_{c-1}). \end{aligned} \quad (3.8)$$

From Eq. (3.7), $b_{1,0,0}$ can be written as

$$b_{1,0,0} = \frac{b_{0,0,0}}{(1-p)(1-\chi_1)v_1} \cdot (1 - (1-u_0)(p - p^{m_0+1}) - (1-p)(1-\chi_0 \cdot u_0)). \quad (3.9)$$

From Eq. (3.6), when $1 < i < c-1$, $b_{i,0,0}$ can be in a recurrence form

$$b_{i,0,0} = \frac{b_{i-1,0,0}}{(1-p)(1-\chi_i)v_i} (1 - ((1-u_{i-1})(p - p^{m_{i-1}+1}) + (1-p)(1 - v_{i-1} - \chi_{i-1} \cdot u_{i-1} + \chi_{i-1} \cdot v_{i-1}))) - \frac{u_{i-2}(\chi_{i-2} - p\chi_{i-2} + p - p^{m_{i-2}+1})}{(1-p)(1-\chi_i)v_i} \cdot b_{i-2,0,0}, \quad (3.10)$$

and for $i = c-1$,

$$b_{c-1,0,0} = \frac{u_{c-2}(p - p^{m_{c-2}+1} + (1-p)\chi_{c-2})}{(1-p)(1-\chi_{c-1})v_{c-1}} b_{c-2,0,0}. \quad (3.11)$$

By Eq. (3.3) to Eq. (3.11), all stationary probabilities are expressed in terms of $b_{0,0,0}$, p , and τ . Since the sum of all probabilities must be 1,

$$\sum_{i=0}^{c-1} \sum_{j=0}^{m_i} \sum_{k=0}^{W_{i,j}-1} b_{i,j,k} = 1. \quad (3.12)$$

Moreover, since a station only transmits when its backoff counter is 0, it follows that

$$\tau = \sum_{i=0}^{c-1} \sum_{j=0}^{m_i} b_{i,j,0}. \quad (3.13)$$

From Eq. (3.12) and Eq. (3.13), we have an equation with two unknown variables, p and τ . The collision probability can be expressed in terms of τ ,

$$p = 1 - (1 - \tau)^{n-1}. \quad (3.14)$$

By solving Eq. (3.13) and Eq. (3.14), we can obtain p and τ . The saturation throughput, S , is given by

$$S = \frac{E[\text{amount of data transmitted in a time slot}]}{E[\text{length of a time slot}]} = \frac{P_s P_{tr} T_{data}}{(1 - P_{tr})\rho + P_s P_{tr} T_s + (1 - P_s) P_{tr} T_c}, \quad (3.15)$$

where $P_{tr} = 1 - (1 - \tau)^n$ is the probability that a transmission occurs in a randomly chosen slot, $P_s = (n\tau(1 - \tau))^{n-1}/P_{tr}$ is the probability that a transmission succeeds in a time slot, ρ is the length of a backoff slot, T_s is the time required to complete a frame exchange sequence, and T_c is the length of a colliding duration. T_s and T_c are $(DIFS + DATA + ACK + SIFS)$ and $(DIFS + DATA)$, respectively, if direct transmission is used, and are $(DIFS + RTS + CTS + DATA + ACK + 3SIFS)$ and $(DIFS + RTS)$, respectively, when RTS-CTS exchange is used.

3.2.2 Optimal Values of u and v

In case that all u'_i 's are the same and all v'_i 's are the same, optimal u and v can be obtained from the optimal transmission probability τ which maximizes the saturation throughput. Eq. (3.15) can be rewritten as

$$S = \frac{T_{data}}{T_s - T_c + 1/f(\tau)},$$

where

$$f(\tau) = \frac{n\tau(1 - \tau)^{n-1}}{T_c/\rho - (1 - \tau)^n(T_c/\rho - 1)}.$$

The saturation throughput S is maximized when $f(\tau)$ is maximized. Taking the derivative of $f(\tau)$ and setting it to zero,

$$f'(\tau) = (1 - \tau)^n - T_c/\rho(n\tau - (1 - (1 - \tau)^n)) = 0,$$

under the condition $\tau \ll 1$, we have $\tau \approx 1/(n \cdot \sqrt{T_c/(2\rho)})$. With optimal τ we have an equation that relates the optimal u and v from Eq. (3.13). Fig. 3.3 shows the ratios of u to v that give the optimal throughput under different n and c .

3.3 Performance Evaluation

This section presents our simulation results of the performance of MCB as opposed to MILD, DCF, GDCF, EIED, and LILD algorithms. The custom

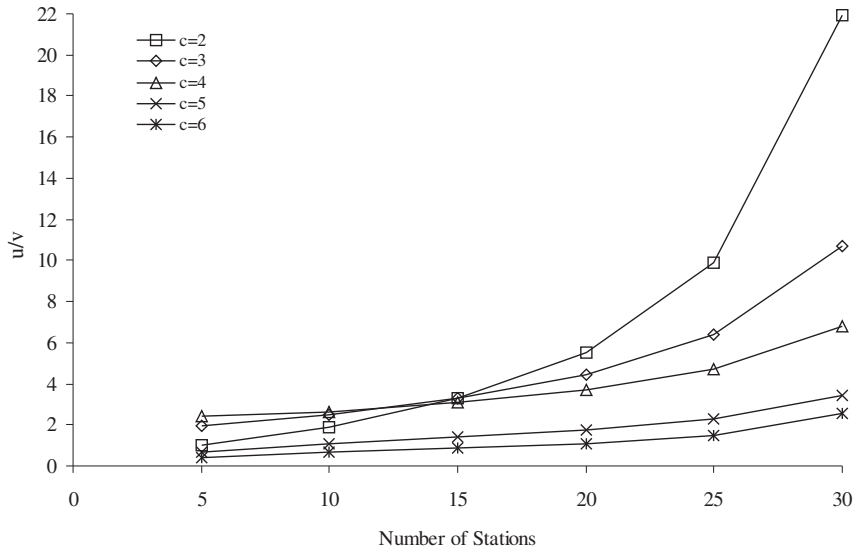


Figure 3.3: The ratio of optimal u to v under different n and c .

Table 3.1: Simulation parameters.

Channel Rate	1 Mbps
aSlotTime	20 us
aSIFS	10 us
aDIFS	50 us
CWmin	31
CWmax	1023
DATA	1024 bytes
ACK	120 bits

simulation programs are written in C++ that simulate networks with an ideal wireless channel (i.e., no hidden terminals). A fairness index (FI) [28] is used to examine the fairness property of a backoff algorithm,

$$FI = \frac{(\sum_i S_i)^2}{n \cdot \sum_i (S_i)^2}, \quad (3.16)$$

where S_i is the saturation throughput received by station i . FI is bounded in the interval $[1, 0]$. An algorithm is fair as its FI is close to 1. Table 3.1 lists MAC and PHY parameters in our simulations. For ease of discussion, we assume the same u and the same v for all chains through out our simulations.

3.3.1 Saturation Throughput

Fig. 3.4 presents the saturation throughput under different u and v . The frame size is 1024 bytes. First, we vary u from 0 to 1 with fixed $v = 0.5$. The figure shows that saturation throughput decreases as the number of competing stations n increases. Given a fixed n , the throughput increases as u increases. Next, we fix $u = 1$ and change the v from 0 to 1. When v is small, the saturation throughput drops first and then increases as n increases. This drop of throughput is due to bad ratios of u to v , which cause large backoff overheads. However, as n increases, the overheads will decrease and the throughput will increase.

Fig. 3.5 shows the saturation throughput, under the frame size of 128 bytes. In Fig. 3.5, we first fix $v = 0.5$ but vary u . The throughput increases when u increases from 0 to 0.1. Further increase of u will degrade the throughput. Then, we vary v from 1 to 0 with fixed $u = 0.1$. The result indicates that a v_i around 0.3 could be a good choice. Fig. 3.6 shows the relations between throughput and frame sizes. When the frame size increases, since less backoff overhead is incurred, the throughput also increases.

3.3.2 The Number of Backoff Chains

In Fig. 3.7, we show the throughput of MCB under different number of chains. The u and v are chosen to maximized the throughput for $n = 6$ and $n = 46$, respectively. In the case that the u and v are chosen for $n = 6$, the throughput of two-chain MCB drops more when the number of stations n increases. In the case that u and v are chosen for $n = 46$, the throughput of two-chain MCB decreases more when n is small. In Fig. 3.3, it has been shown that the curve of the ratio of u to v that gives the optimal throughput is flatter when more backoff chains are used. This may imply that when more backoff chains are used, the optimal ratio of u to v for a given n will be closer to optimal ratios

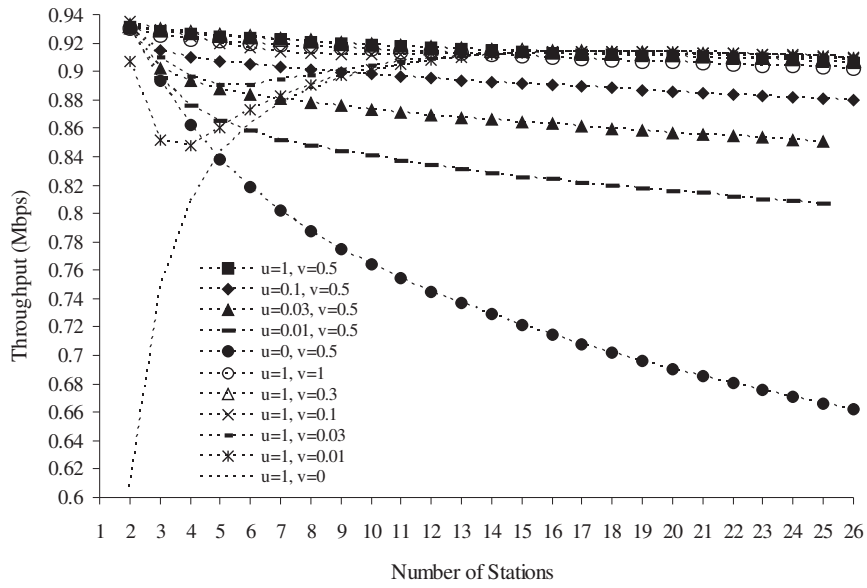


Figure 3.4: Saturation throughput under different u and v with the frame size of 1024 bytes.

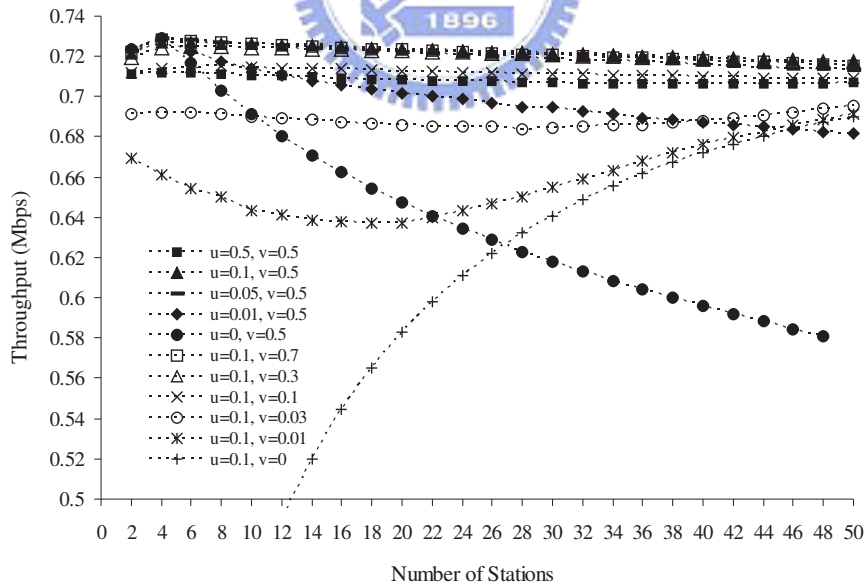


Figure 3.5: Saturation throughput under different u and v with the frame size of 128 bytes.

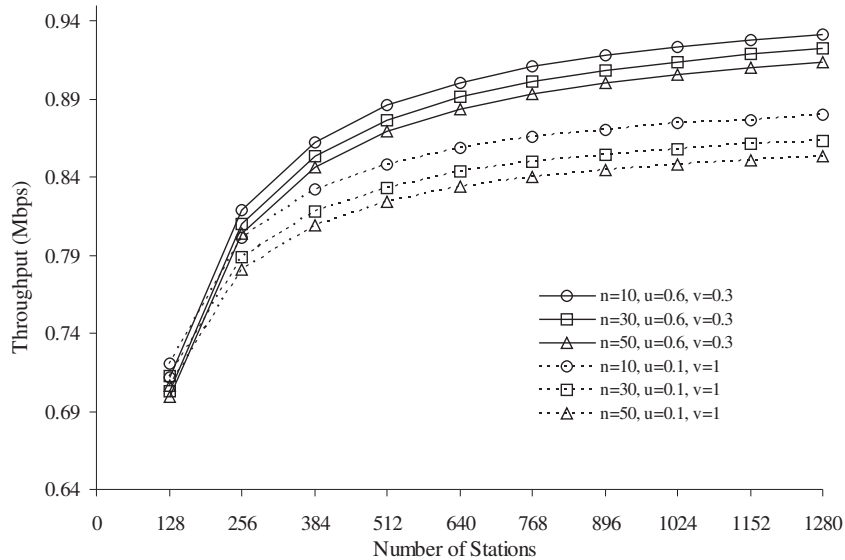


Figure 3.6: Saturation throughput versus frame sizes.

for $n + 1$ or $n - 1$ than when less backoff chains are used. Fig. 3.8 shows the corresponding fairness indexes. The fairness index tends to oscillate when more backoff chains are used. Fig. 3.9 compares the analysis results to the simulation results of saturation throughput of four-chain MCB. The figures show that the analyzed throughput has the same trend as the simulation throughput and for some values of u and v , the analysis results match the simulation results.

3.3.3 Comparison to Existing Algorithms

In the following, we compare the performance of four-chain MCB to existing algorithms. The minimum contention windows of the four chains are 31, 127, 511 and 1023, respectively. The u and v are chosen from simulation results in Fig. 3.5 and Fig. 3.4. They are 0.1 and 0.3, respectively, for the frame size of 128 bytes, and are 1 and 0.3, respectively, for the frame size of 1024 bytes.

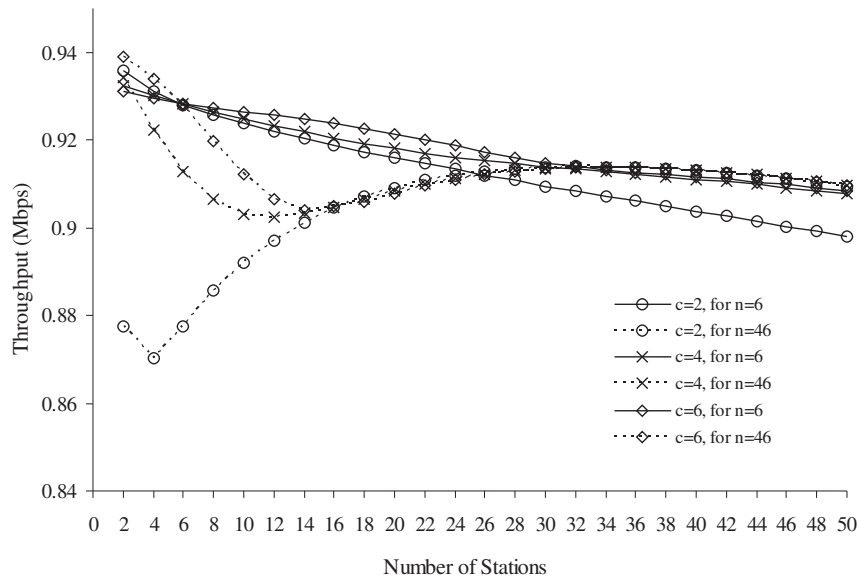


Figure 3.7: Throughput of MCB with u and v which are chosen for $n = 6$ and $n = 46$.

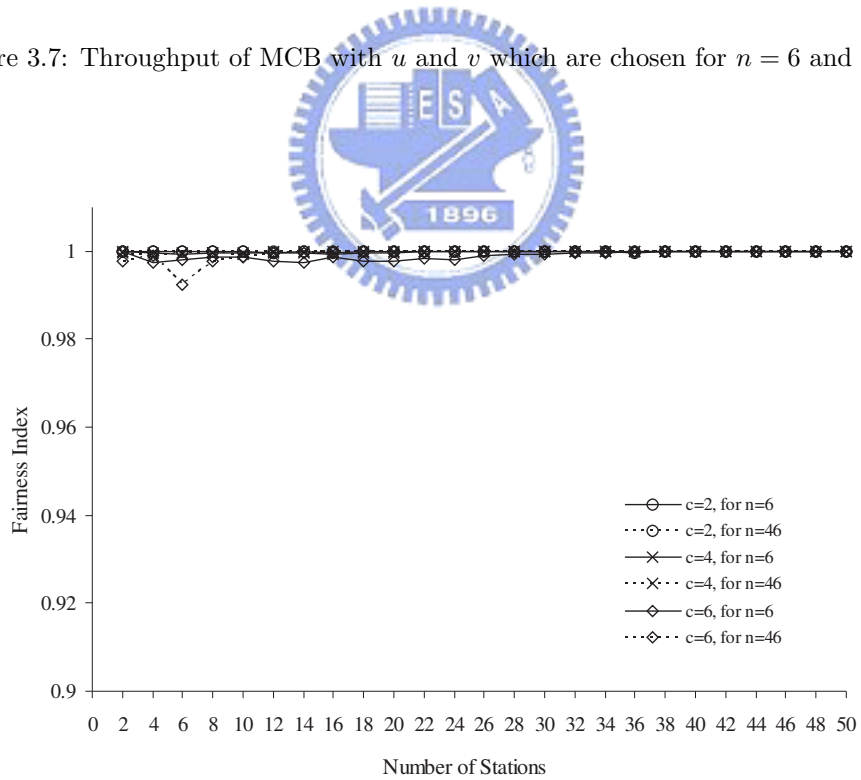
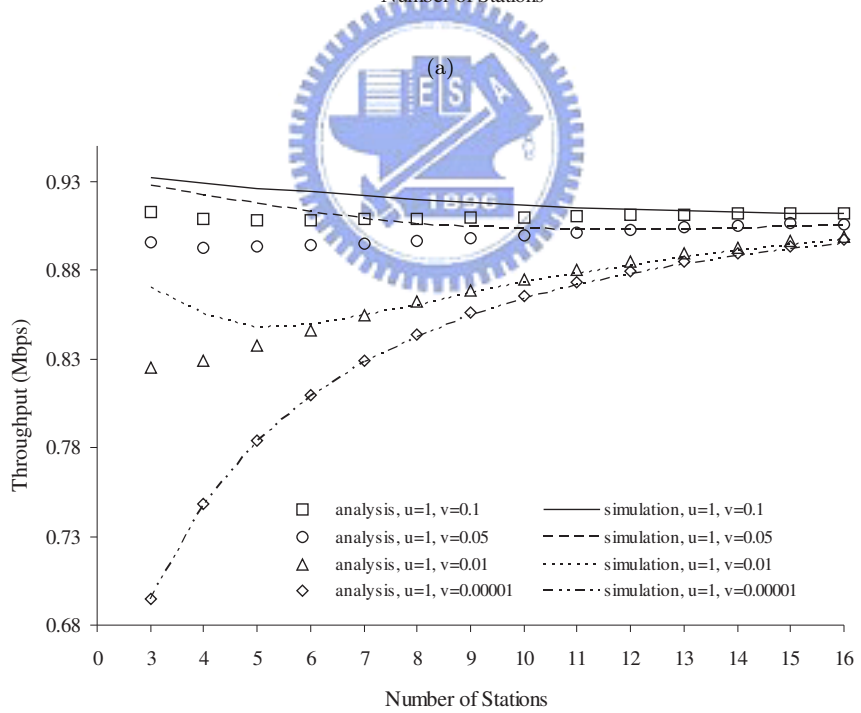
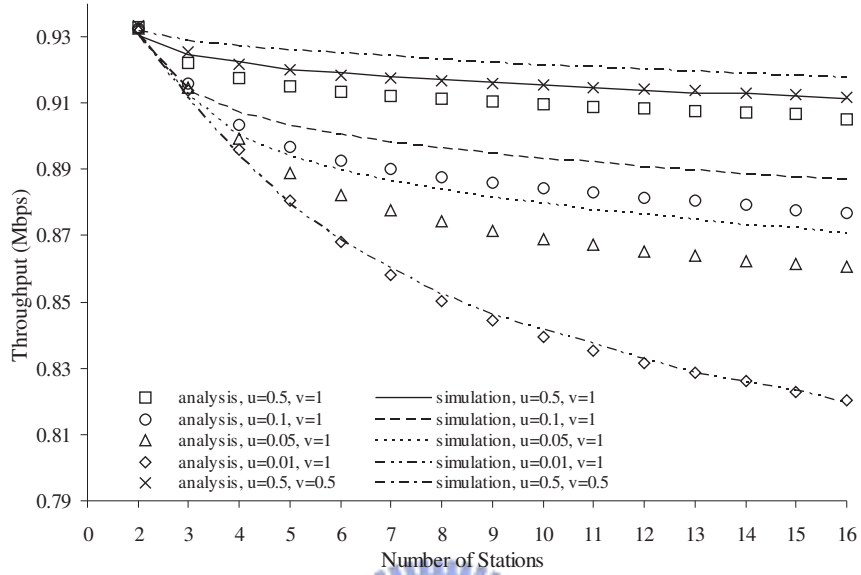


Figure 3.8: Fairness Index.



(a)

Figure 3.9: Saturation throughput of four-chain MCB : analysis versus simulation.

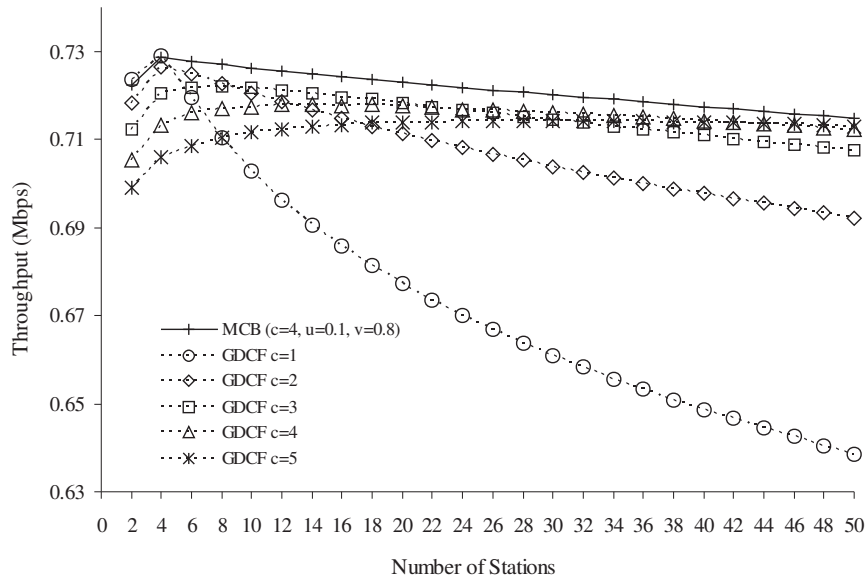


Figure 3.10: Saturation throughput of MCB and GDCF for frame size 128 bytes.

Comparing with GDCF

In Fig. 3.10, we compare the throughput of MCB to that of GDCF, assuming the frame size is 128 bytes. For GDCF, we increase its parameter c from 1 to 5. With a smaller c , although GDCF achieves higher throughput for small n , the throughput drops dramatically as n increases. However, with a larger c , it is clear that MCB outperforms GDCF.

Fig. 3.11 compares MCB and GDCF when the frame size is 1024 bytes. With a larger c , GDCF performs as well as MCB. However, as shown in Fig. 3.12, GDCF suffers from unfair channel access when n is small. This is because that for $n = 2$, when window sizes of the two stations are different, a collision event will double the difference of their window sizes. Since a station with a small contention window has shorter backoff time, it may reach c successful transmissions and then halve its contention window faster than the other station.

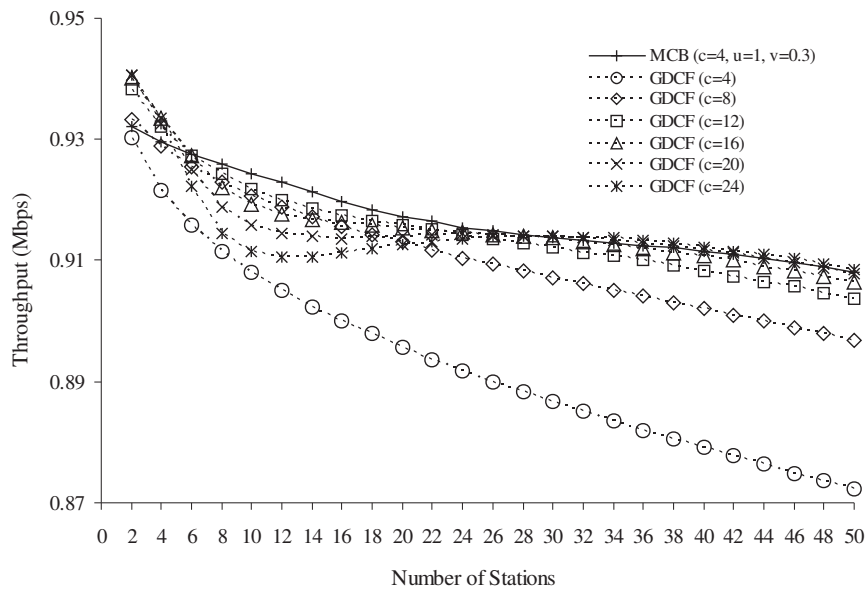


Figure 3.11: Saturation throughput of MCB and GDCF for frame size 1024 bytes.

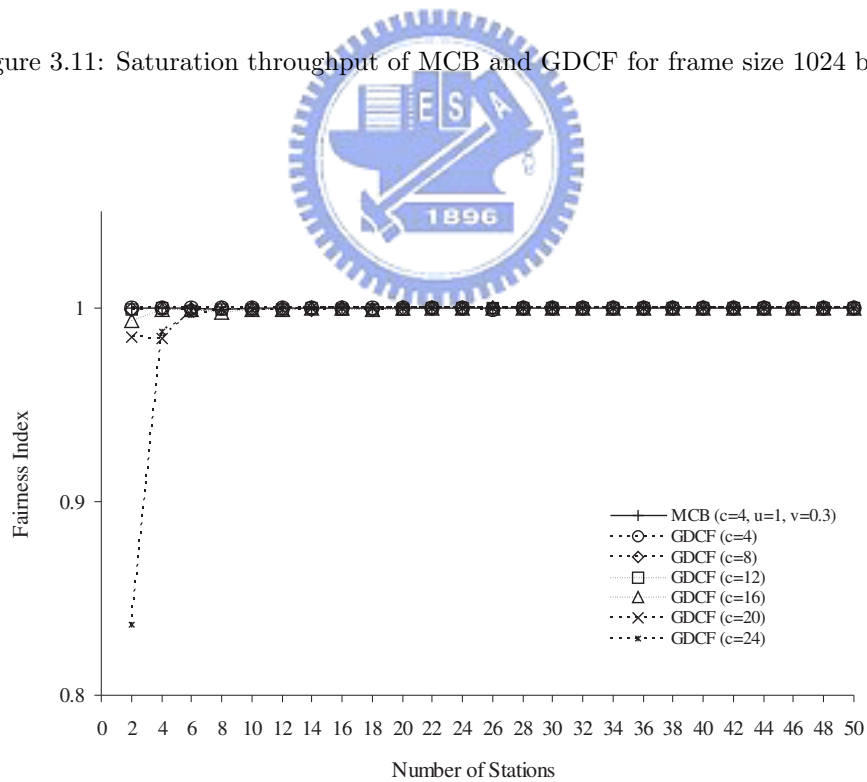


Figure 3.12: Fairness index of MCB and GDCF for frame size 1024 bytes.

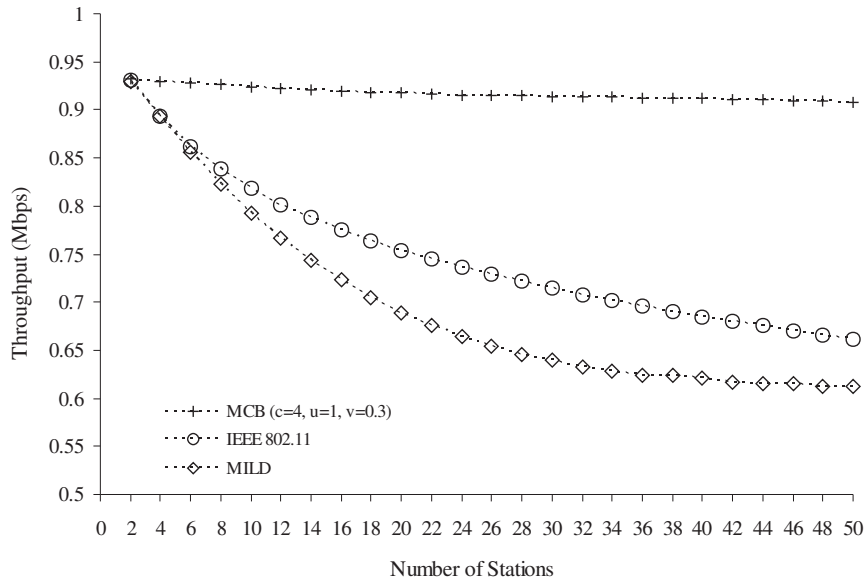


Figure 3.13: Saturation throughput of MCB, IEEE 802.11, and MILD.

Comparing with IEEE 802.11 and MILD

Fig. 3.13 shows the throughputs of MCB, IEEE 802.11, and MILD. The throughput of MILD is lower than those of MCB and IEEE 802.11. In MILD, a station can advertise its contention window only if no other station successfully transmits during its backoff period. However, there is a high probability that a station with a smaller contention window successfully transmits during this period. This will force other stations to adopt this small contention window. In high traffic load, this will increase collision probability. The IEEE 802.11 is outperformed by MCB since MCB offers more than one chain, allowing stations to adapt to different congestion levels.

Comparing with EIED

Fig. 3.14 compares the throughput of MCB to that of EIED. We use $EIED(x,y)$ to denote that if a collision occurs, $CW_{new} = \min(x \cdot (CW_{old} + 1) - 1, CW_{max})$,

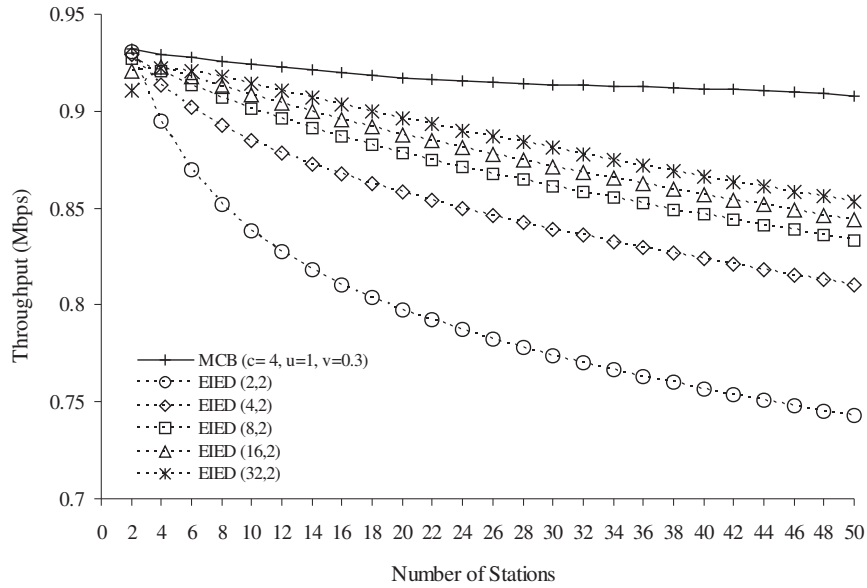


Figure 3.14: Saturation throughput of MCB and EIED(x,y) with fixed $y = 2$.

and if a transmission succeeds, $CW_{new} = \max(\lfloor (CW_{old} + 1)/y \rfloor - 1, CW_{min})$. In Fig. 3.14, we fix $y = 2$ and increase x from 2 to 32. In Fig. 3.15, we fix $x = 2$ and vary y from 1.01 to 2. For EIED(2, 1.01), Fig. 3.16 shows that the wireless medium is unfairly utilized when $n < 8$. For EIED(2, 1.01) at $n = 2$, when a collision occurs, the difference between the two window sizes is doubled. Since the decrement of contention windows is slow (the decrement is 1 when $CW < 100$), the window sizes are hardly reduced to CW_{min} after a number of successful transmissions. Once a collision occurs, the difference is doubled.

3.4 Concluding Remarks

In this chapter, we have proposed a new MCB algorithm. MCB explores the possibility of using multiple backoff chains with different minimum contention windows and considering collision events on the wireless channel as hints to

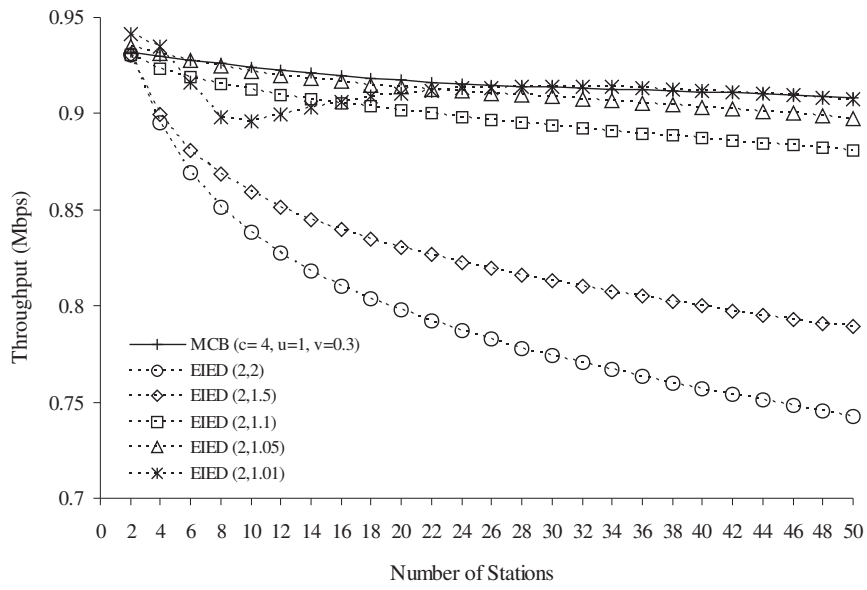


Figure 3.15: Throughput of MCB and EIED(x,y) with fixed $x = 2$.

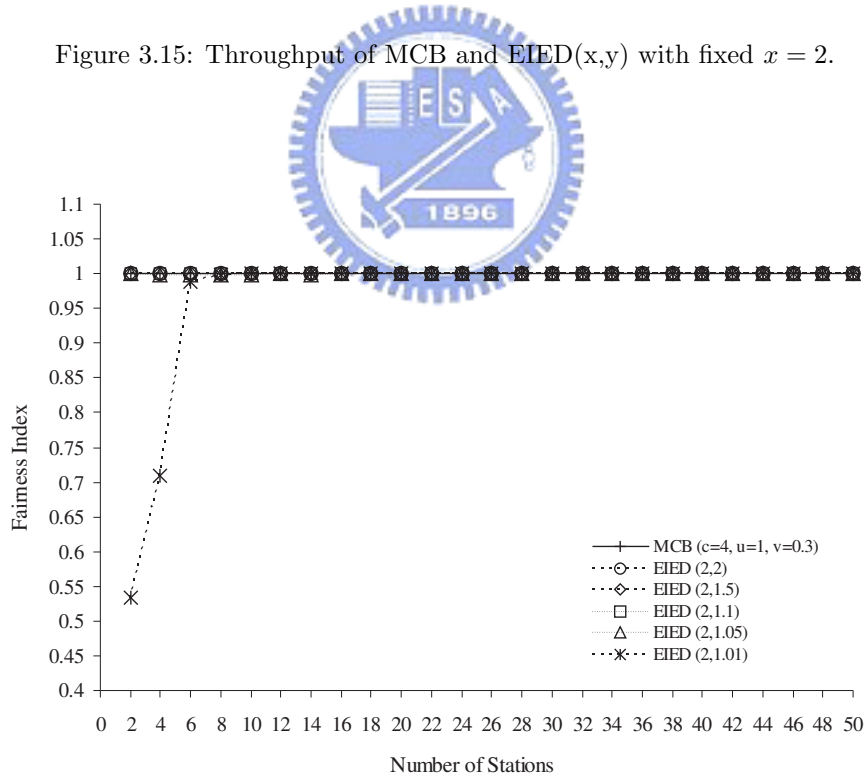



Figure 3.16: Fairness index of MCB and EIED(x,y) with fixed $x = 2$.

choose a proper chain. With the capability of switching to different backoff chains, MCB offers higher throughput than the existing protocols, such as GDCF, IEEE 802.11, MILD, EIED, and LILD, yet still provides fair access to the wireless channel.



Chapter 4

A Jamming-Based MAC Protocol to Improve the Performance of Wireless Multihop Ad Hoc Networks



In this chapter, we describe the proposed MAC protocol, called *JMAC*, that is designed based on the concept of traffic separation and jamming mechanism. In *JMAC*, the medium is divided into two channels: *S* channel and *R* channel. RTS and DATA frames (source stations' traffic) are transmitted on the *S* channel, and CTS and ACK frames (destination stations's traffic) are transmitted on the *R* channel. It is assumed that each station is equipped with one transmitter and one receiver device, one tuned to the *S* channel and the other tuned to the *R* channel. The ratio of bandwidth allocated to the *R* and *S* channels is assumed to be $\alpha : (1 - \alpha)$, where $0 < \alpha < 1$. How to choose an appropriate α will be further discussed in Section 4.2.

4.1 The Proposed Protocol

In JMAC, a source station always transmits RTS/DATA frames on the S channel but receives CTS/ACK frames on the R channel. It also transmits jamming signal on the S channel while waiting or receiving a CTS/ACK frame on the R channel. For a destination station, while it is waiting or receiving a DATA frame on S channel, it jams the R channel to prevent the hidden terminal problem. Jamming signal is the one with sufficient energy causing the medium to become busy. No data is carried in jamming signal and there is no need to decode its content. The overlapping of a jamming signal and a data signal is considered as a jamming signal too, and thus can not be correctly recognized.

The procedures and the timing diagram of transmission and receipt of RTS, CTS, DATA, and ACK frames are shown in Figs. 4.1 and 4.2, respectively. Before transmitting a RTS frame on the S channel, a source station first senses the R channel. If it is idle for a DIFS period, the source station sends a RTS on the S channel, and then listens to the R channel for a CTS frame. If a station detects R channel to be busy before sending a RTS frame, this implies that some neighbors may be receiving data so the source station is not allowed to access the medium. While waiting for the CTS frame from an intended destination, the source station is required to jam the S channel. The purpose of jamming the S channel is similar to the reservation function of RTS in 802.11, but the difference is that the medium is jammed as long as needed, depending on the result of RTS-CTS exchange. If the RTS-CTS exchange fails (indicated by a CTS_Timeout), the sender will stop jamming the S channel and will start the backoff procedure as in 802.11.

After the destination station receives the RTS from the S channel, it responds with a CTS frame on the R channel, and then listens to the S channel for a DATA frame. While it is waiting for the DATA frame, it also jams

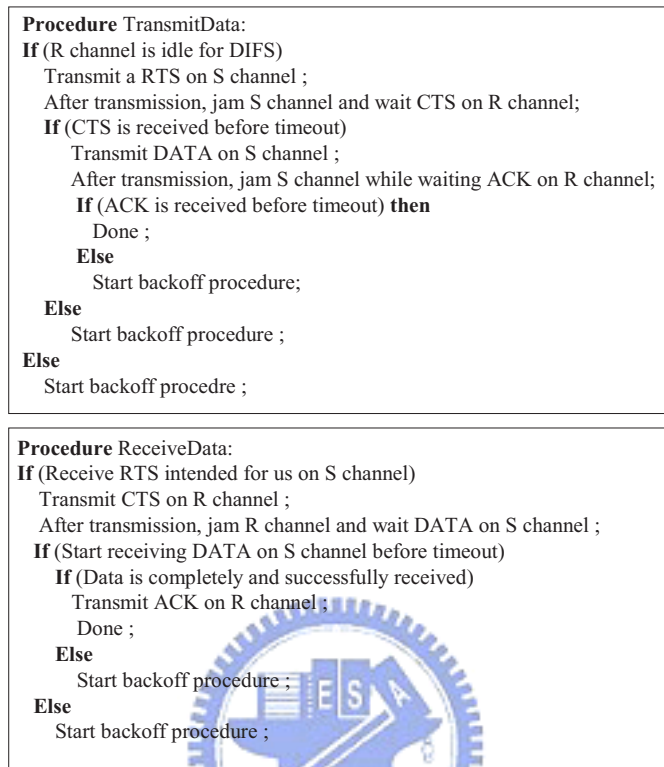


Figure 4.1: The procedures of transmission and receipt of control/data frames in JMAC.

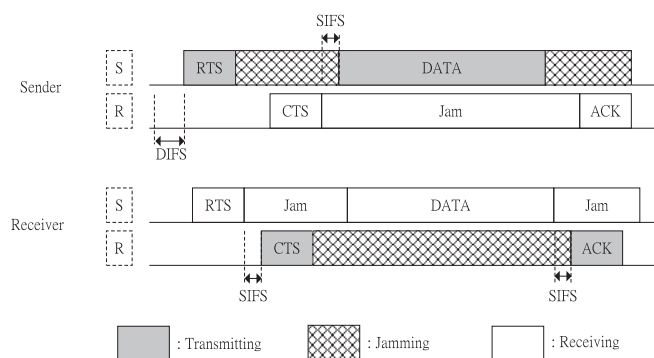


Figure 4.2: The timing diagram of transmission of control/data frames.

the R channel to prevent the hidden terminal problem. If the DATA frame unfortunately fails to appear after timeout, it will stop jamming the R channel.

The rest of access procedure is similar. After receiving the CTS, the source transmits its DATA frame, and then jams the S channel while waiting for an ACK on the R channel. For the destination, after receiving the data frame, it transmits an ACK frame on the R channel. Note that although we adopt the RTS-CTS-DATA-ACK frame exchange sequence, stations may also apply the DATA-ACK sequence directly to transmit data frames.

In JMAC, the backoff procedure starts after the R channel becomes idle for a DIFS period, and it is independent of the status of the S channel. This is because the transmission of a RTS on the S channel doesn't interfere with neighboring stations' reception of CTS/ACK frames on the R channel. Fig. 4.3 illustrates the exchange of RTS, CTS, DATA, and ACK frames in JMAC. The double circles, in the example, represent the stations that are currently transmitting RTS, CTS, DATA, or ACK frames. Fig. 4.3 (a) and Fig. 4.3 (b) show that A is transmitting a RTS frame to B and then jamming the S channel, respectively. Since the RTS frame is transmitted by broadcasting, F also hears the RTS frame. After receiving the RTS frame, B responds with a CTS frame and then jams the R channel in Fig. 4.3(b) and Fig. 4.3(c) respectively. This will prevent the hidden terminal problem and erroneous reservation problem. Assume that if C misses the CTS frame from B due to collision or transmission error, C will not be a hidden terminal since it will detect the busy R channel and know that some of its neighbors are receiving data. Also, since B will stop jamming the R channel if it doesn't receive a DATA frame after timeout, the erroneous reservation problem will not occur in JMAC. After A receives the CTS frame from B , it transmits a DATA frame and jams the S channel in Fig. 4.3(d). It can be easily observed that the jamming signal transmitted by A also protects its receipt of the ACK frame from B . We comment that if F

doesn't successfully receive the RTS frame from A in Fig. 4.3(a), since A will transmit RTS/DATA frames and jamming signal on the S channel, any RTS frame to F will be collided at F . Therefore, F will not transmit any CTS or ACK frame when A is receiving CTS and ACK frames from B .

JMAC allows more concurrent transmission/receipt activities for stations within each other's transmission range. Taking Fig. 4.3(b) as an example, while A is transmitting DATA to B , D and F are allowed to concurrently request transmission to C and E , respectively. After C and E receive RTS frames from D and F , they can safely reply with CTS frames without collision. This is shown in Fig. 4.4(a) and Fig. 4.4(b).

4.2 Tuning the Factor α

In this section, we discuss how to choose the ratio α that determines the bandwidth of the S channel and R channel. Let the system transmission rate be r . After dividing the total bandwidth into two sub-channels, the transmission rates for the S channel and R channel are assumed to be $\alpha \times r$ and $(1 - \alpha) \times r$, respectively. The basic idea is to find a value of α such that the time of a RTS-CTS-DATA-ACK exchange, denoted by $f(\alpha)$, is minimized. The $f(\alpha)$ can be expressed as follows:

$$f(\alpha) = \frac{RTS + DATA}{\alpha \times r} + \frac{CTS + ACK}{(1 - \alpha) \times r} \quad (4.1)$$

Differentiating $f(\alpha)$ with respect to α , we have

$$\frac{df(\alpha)}{d\alpha} = \frac{\alpha^2 \times (CTS + ACK) - (1 - \alpha)^2 \times (RTS + DATA)}{\alpha^2 \times (1 - \alpha)^2 \times r} \quad (4.2)$$

Set Eq. (4.2) to zero, we have the optimal value $\hat{\alpha}$,

$$\hat{\alpha} = \frac{RTS + DATA \pm \sqrt{(RTS + DATA) \times (CTS + ACK)}}{RTS + DATA - CTS - ACK} \quad (4.3)$$

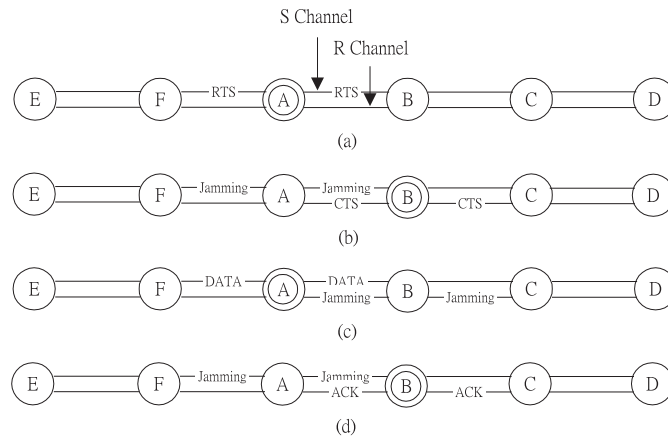


Figure 4.3: Frame exchange and transmission of jamming signal in JMAC.

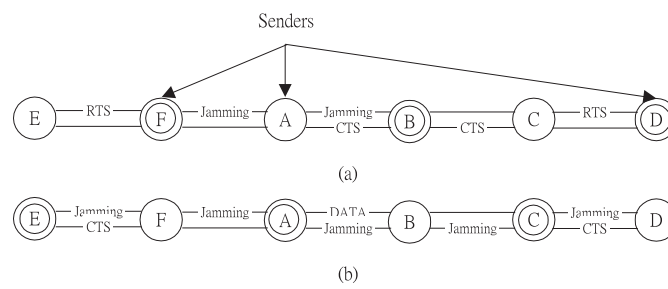


Figure 4.4: Concurrent transmissions in JMAC.

Since $0 < \hat{\alpha} < 1$,

$$\hat{\alpha} = \frac{RTS + DATA - \sqrt{(RTS + DATA) \times (CTS + ACK)}}{RTS + DATA - CTS - ACK} \quad (4.4)$$

Although the sizes of RTS, CTS, and ACK frames are all fixed, the size of data frames may vary. Fig. 4.5 shows the relation of the optimum values of $\hat{\alpha}$ and the sizes of data frames, s . The optimal $\hat{\alpha}$ tends to increase as the frame size increases. In other words, a larger data frame would require more bandwidth to be assigned to the S channel. The figure also shows that there is no globally optimal $\hat{\alpha}$ for all frame sizes. So we turn to search an approximation to the global optimal for a range of frame sizes specified in 802.11. In Fig. 4.6, we plot the frame exchange time of IEEE 802.11 and JMAC for different frame sizes and different α values. Note that for JMAC, the curve of $f(\text{optimal})$ represents the ideal case where the data frame size is always known and we can always choose the best $\hat{\alpha}$ to minimize the frame exchange time. As can be seen in the figure, $\alpha = 0.7 \sim 0.8$ is a good approximation to the curve of $f(\text{optimal})$ when the frame size falls in the range of 128 \sim 2048 bytes.

The RTS-CTS-DATA-ACK exchange time in 802.11 is also shown in Fig. 4.6. Since 802.11 fully utilizes channel bandwidth, its frame exchange time is shorter than that of JMAC. This reflects the cost incurred by JMAC due to channel division. However, the above analysis is under the ideal assumption that IEEE 802.11 always successfully completes its frame exchange. As discussed earlier, IEEE 802.11 may suffer from the hidden terminal and erroneous reservation problems. Thus, such an advantage may be offset by these factors.

4.3 Performance Evaluation

In this section, we present our simulation results and compare the performance of JMAC to that of IEEE 802.11. Parameters in Table 4.1, Table 4.2, and Table 4.3 are used in our simulation. We uniformly put N stations in a

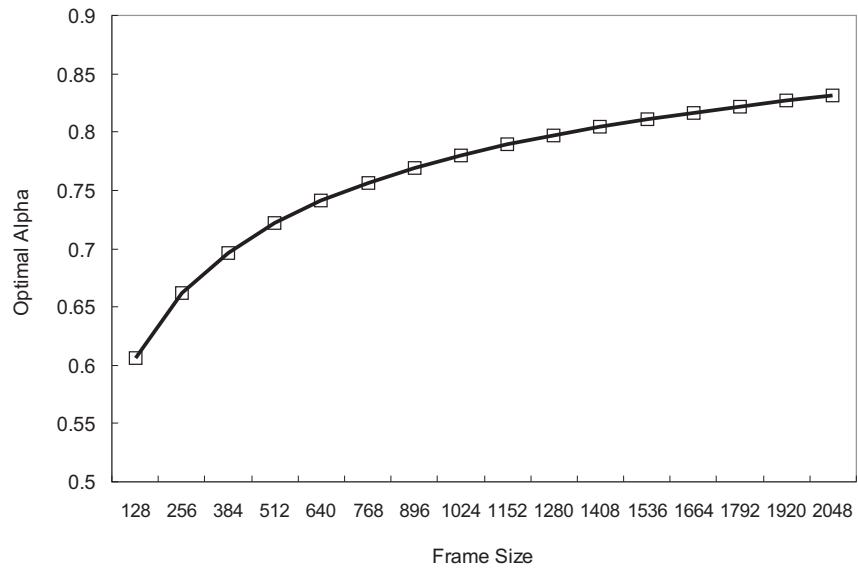


Figure 4.5: Data frame size vs. the optimum $\hat{\alpha}$.

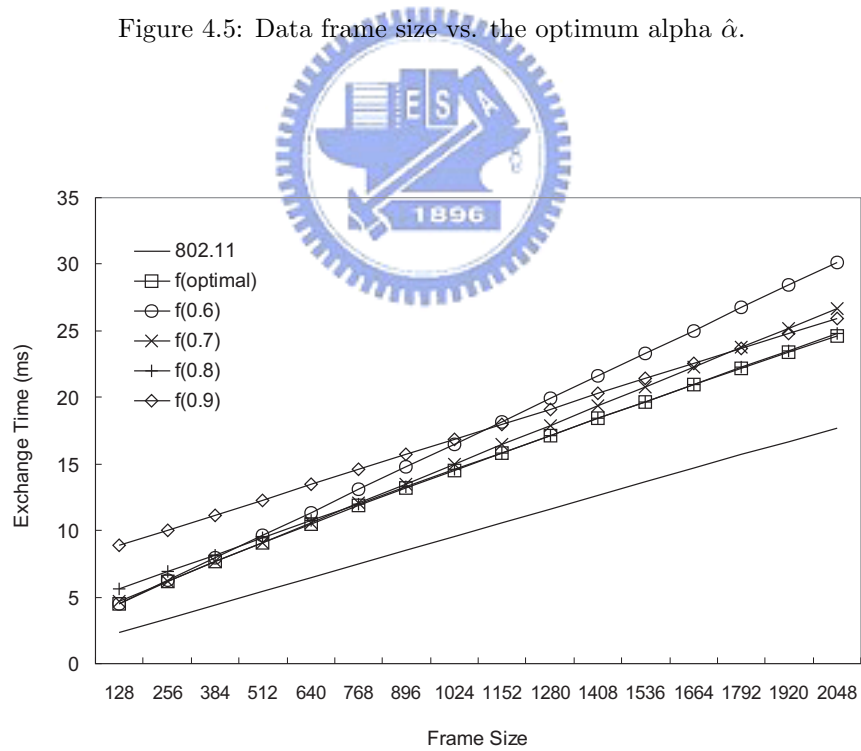


Figure 4.6: Frame exchange time in IEEE 802.11 and in JMAC for different data frame sizes and different α .

120m×120m area. The transmission range of each station is $R = 30$ m. The transmission rate is 1 Mbps. In each individual simulation run, data frame size is assumed to be fixed. The movement of a station follows a two-state model in which each station transits from the *moving* state to the *still* state with probability P_S and from the still state to the moving state with probability P_M . When transiting from the still to the moving state, a station chooses one of eight directions and moves in that direction with a constant speed of 1 m/sec. The total simulation time is 50 minutes in each simulation run, and frames are assumed to arrive at each station according to the Poisson process. As to the bandwidth of the S channel and R channel, we choose $\alpha = 0.78$, which is the optimal α for the frame size of 1024 bytes.

Figs. 4.7 and 4.8 show the aggregate throughput and mean throughput, respectively, under different traffic loads. At light loads (from 1 to 5 frames/sec), the mean throughputs of JMAC and IEEE 802.11 are very close. However, as the traffic load increases, unless for small N (such as $N = 20$), JMAC outperforms IEEE 802.11. This implies that at this stage, the hidden terminal and erroneous reservation problems start to degrade the performances of IEEE 802.11. On the contrary, JMAC is quite resistant to such effects, and thus can still perform very well. As classic multiple access protocols such as ALOHA and CSMA, IEEE 802.11 also exhibits instability on channel throughput. After reaching the overload condition, the performance of 802.11 starts to degrade. For the case of $N = 20$, since the traffic load is still light, the network is not saturated yet and the throughputs of JMAC and 802.11 are quite the same. But as traffic load increases, shown in Fig. 4.9, the similar behavior can also be observed.

Figs. 4.9 and 4.10 further demonstrate saturation throughput under different network sizes N . Although, saturation throughput decreases as N increases for both JMAC and 802.11, JMAC still shows much better performance than

Table 4.1: MAC Layer parameters.

MAC Parameters	
Transmission Rate(r)	1 (Mb/sec)
RTS	20 octets
CTS	14 octets
ACK	14 octets
DATA	1024 octets
Retry_Count	7
CWmin	31
CWmax	1023
CTS_Timeout	SIFS + $2 \times \tau$
Data_Timeout	SIFS + $2 \times \tau$
ACK_Timeout	SIFS + $2 \times \tau$
Duration/ID (for RTS)	CTS + DATA + ACK + $3 \times \text{aSIFSTime} + 3 \times \tau$
Duration/ID (for CTS)	DATA + ACK + $2 \times \text{aSIFSTime} + \tau$

Table 4.2: Physical layer parameters.

DSSS PHY Specification for 2.4G Band	
AslotTime	20 μs
ASIFSTime	10 μs
ADIFSTime	50 μs
PLCP Preamble	24 octets
PLCP Header	6 octets

Table 4.3: Parameters of the simulated wireless network.

Simulated Environment	
Bandwidth Ratio (α)	0.784
Propagation Delay (τ)	1 (μs)
Transmission Range (R)	30 (m)
Simulated Area (A)	$4R \times 4R$ (m^2)
Moving Speed (v)	1 (m/s)
Still Probability (P_s)	0.1
Moving Probability (P_M)	0.9

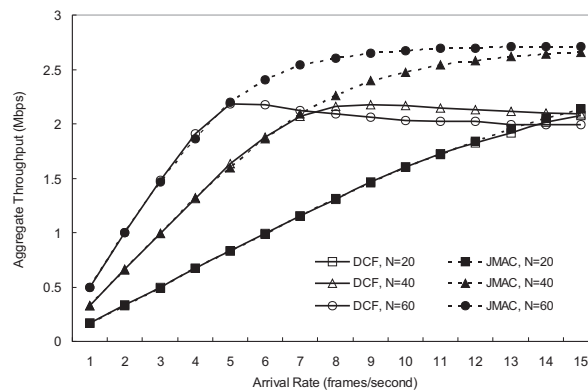


Figure 4.7: Aggregate throughput versus frame arrival rate.

802.11. The saturation throughput may be affected by many factors: the number of contending stations in the network, the maximum backoff window, retry count, and the hidden terminal problem [7, 8]. In this simulation, the same parameters are used for both JMAC and 802.11, which may imply that the hidden terminal problem and the erroneous reservation problem are two causes of degradation of the performance of 802.11.

In the above simulations, the frame size is fixed at 1,024 bytes. In this part, we vary the data frame size to observe its effect. As Figs. 4.11 and 4.12 show, both JMAC and 802.11 benefit from larger data frames due to less control overheads incurred by RTS, CTS, and ACK frames. The results also show that smaller frames will favor 802.11, but larger frames will favor JMAC. For 802.11, the impact of the hidden-terminal problem will become more serious for larger data frames (due to higher penalties caused by collisions). On the contrary, JMAC doesn't suffer from such a problem and can thus benefit from this factor. Hence, JMAC is more efficient than IEEE 802.11 for transmitting medium or large data frames (> 512 bytes/frame).

Fig. 4.13 presents mean access delay of data frames versus frame arrival rate. The access delay increases with frame arrival rate. Under low traffic load, the access delay of JMAC is longer than that of 802.11. This is because the time to complete the frame exchange sequence is longer than that in 802.11. But as traffic load increases, 802.11 will suffer from more collision, and thus its access delay dramatically increases.

4.4 Concluding Remarks

Besides the well known hidden terminal problem, the IEEE 802.11 also suffers from the erroneous reservation problem. In this paper, we propose a jamming-based MAC (JMAC) protocol which is free from both the hidden terminal and erroneous reservation problems. JMAC separates source stations' traffic

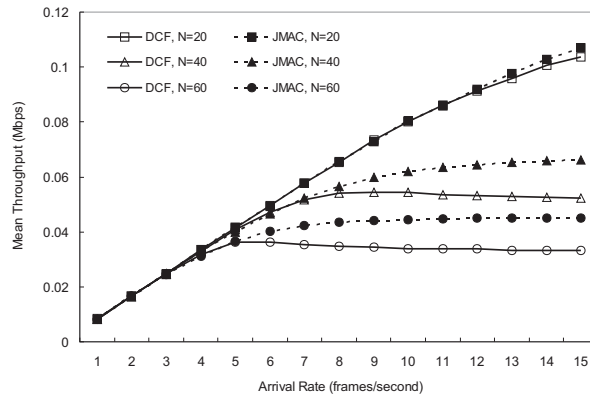


Figure 4.8: Mean throughput versus frame arrival rate.

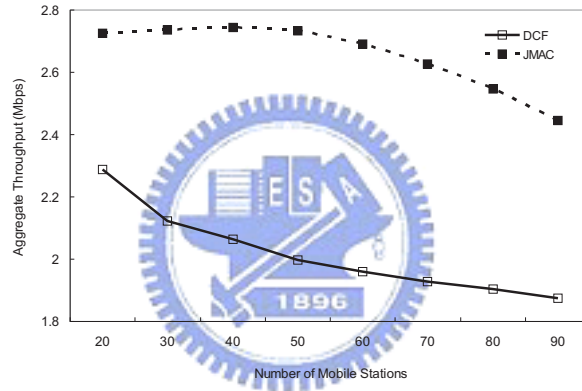


Figure 4.9: Aggregate throughput versus network density.

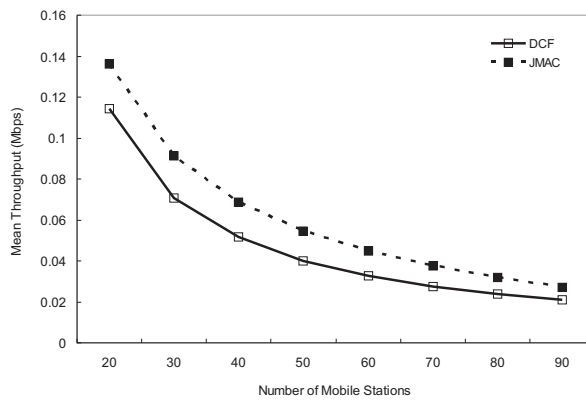


Figure 4.10: Mean throughput versus network density.

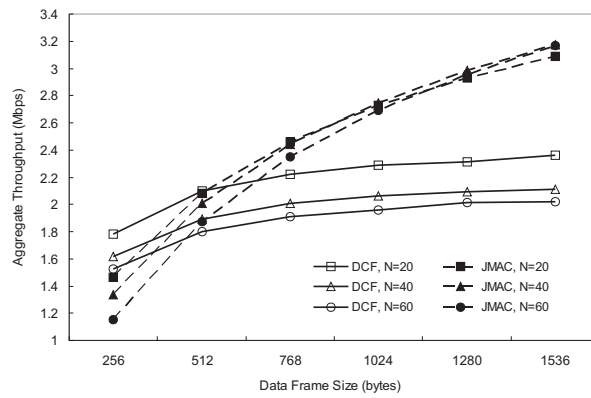


Figure 4.11: Aggregate throughput versus frame size.

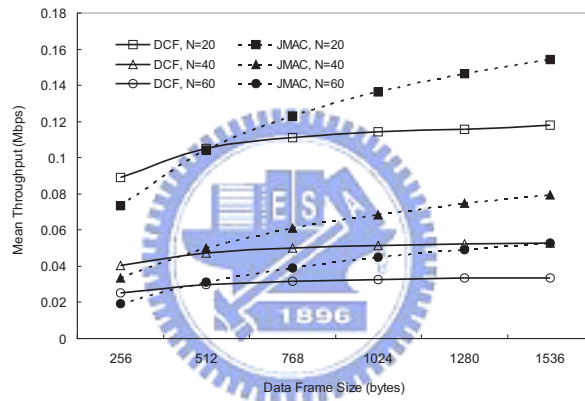


Figure 4.12: Mean throughput versus frame size.

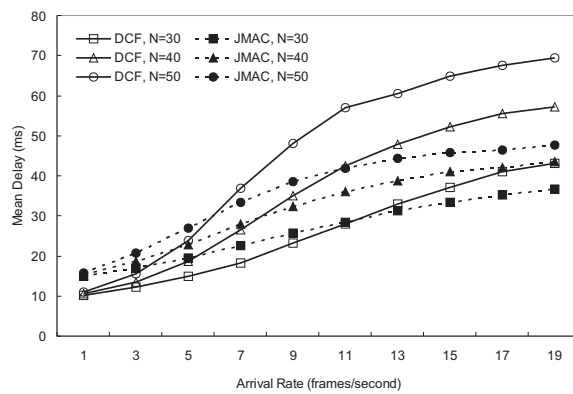


Figure 4.13: Mean access delay versus frame arrival rate.

from destination stations' traffic into different channels and signals the channel status by jamming the channels. We also discuss how to choose a proper ratio for determining the bandwidths of the channels. It is shown that, the optimal ratio changes with the data frame size so there is no global optimal value for all sizes. But the values in the range of 0.7 and 0.8 would be a good approximation for frame size ranging from 128 to 2048 bytes. It is also shown that channel division incurs some cost in terms of the transmission time of a RTS-CTS-DAT-ACK exchange sequence; however, from simulation results, the advantages of being free from the erroneous reservation and the hidden terminal problems and the benefit of more concurrent transmissions of JMAC can compensate the cost of channel division when data frame size is median or large.



Chapter 5

A Deadline-Constraint Scheduling Algorithm for IEEE 802.11e WLANs

5.1 The Proposed Deadline-Constraint Scheduling Algorithm



We consider an IEEE 802.11e WLAN with a QAP (QoS AP) and multiple QSTAs (QoS stations). Since the PCF is not mandatory in 802.11, we will only assume EDCA and HCCA running on the top of DCF. For each arriving MSDU, a deadline is associated with it. For each TS i , we assume that the nominal and maximum MSDU sizes are L_i and M_i , respectively, the mean and peak data rates are ρ_i and ρ_i^h , respectively, the delay bound is D_i , the maximum burst size is B_i , and the minimum PHY transmission rate is R . For a TS i , whenever a frame of size S arrives, it is assigned a deadline equal to its arrival time plus $(D_i - (S/R_i + O))$, where O is a fixed overhead. For a fragmented MSDU, the deadlines of all fragments should be the same. There is a hybrid coordinator (HC) resident in the QAP, which maintains an aggregate

token bucket for each QSTA v . Specifically, let C_v be the set of TSs of QSTA v , the token rate r_v for QSTA v is

$$r_v = \sum_{i \in C_v} \frac{\rho_i}{L_i} \cdot \left(\frac{L_i}{R_i} + O \right). \quad (5.1)$$

The HC also maintains a *Head-of-Line Deadline (HOLD)* for each TS, which is the deadline of the head-of-line frame of the TS. Each QSTA is required to piggyback the HOLDS of its TSs in its transmitted frames.

5.1.1 Length Constraint

In each beacon interval T_B , a fraction of T_B , denoted by T_{cp} , is reserved for contention periods. Let S_t be the sum of the lengths of CAPs which are allocated during previous beacon transmission time and time t . Let S'_t be the length of the current CAP up to time t if t is in a CAP. HC is allowed to allocate a TXOP at time t if $S_t \leq T_B - T_{cp}$ and $S'_t \leq \text{dot11CAPLimit}$.

Given HOLDS of all TSs in the network, HC sorts TSs according to their HOLDS in an ascending order. For each TS i in the sorted list, HC checks if the following condition holds

$$\min\{Q_v, t_{nb} - t, T_B - T_{cp} - S_t, \text{dot11CAPLimit} - S'_t\} - (M_i/R_i + O) \geq 0, \quad (5.2)$$

where t is the current time, t_{nb} is the next beacon transmission time, and Q_v is the current bucket size of QSTA v . Eq. (5.2) guarantees that the transmission of a data frame will not overlap with the next beacon transmission time and that the remaining CAP is long enough to transmit at least one frame. There may be more than one TS which satisfies Eq. (5.2). HC will poll the QSTA with the TS whose HOLD is smallest. The length constraint of the TXOP for the winning QSTA v is

$$\min\{Q_v, t_{nb} - t, T_B - T_{cp} - S_t, \text{dot11CAPLimit} - S'_t\}, \quad (5.3)$$

This length constraint will be specified in the QoS control field of the Polling frame to QSTA v .

5.1.2 Deadline Constraint

In Section 5.1.1, HC sorts all TSs according to their HOLDS. Let δ_i , $i = 1, 2, 3, \dots$, be the i -th TS in the sorted list and τ_i be its HOLD. Assume TS δ_1 is selected in Section 5.1.1, and it is a TS of QSTA v . HC will poll QSTA v and allocate a TXOP to it. A deadline constraint will be imposed on the TXOP. The chosen deadline constraint of the TXOP has to meet two requirements. It shall not cause frames of other TSs being dropped. And, it shall allow the selected QSTA v to transmit as many frames as possible to reduce polling overheads.

Recall that HC will sort all TSs according to their HOLDS. Let τ_i , $i \geq 1$, be the HOLD of i -th TS in the sorted list, and let δ_i be the i -th TS. We will pick one of these HOLDS as the deadline. To calculate the deadline constraint, we need to derive the maximum amount of data $A_i(\Delta t)$ that may be generated by TS i during a time interval Δt ,

$$A_i(\Delta t) = \min(M_i + \rho_i^h \cdot \Delta t, B_i + \rho_i \cdot \Delta t). \quad (5.4)$$

The time required to transmit $A_i(\Delta t)$ bytes can be approximated by

$$Z_i(\Delta t) = \left\lceil \frac{A_i(\Delta t)}{L_i} \right\rceil \cdot \left(\frac{L_i}{R_i} + O \right). \quad (5.5)$$

Suppose that we set τ_k as the deadline constraint of the TXOP. Then we need to guarantee that after QSTA v transmits, other QSTAs' frames that have deadlines before τ_k can be transmitted by τ_k . For QSTA v , the amount of data that is to be transmitted in its TXOP with the deadline constraint τ_k is

$$\sum_{i \in C_v} Z_{\delta_i}(\tau_k - \tau_i).$$

The expected amount of data in all other QSTAs that has a deadline before τ_k is

$$\sum_{i \in C - C_v} Z_{\delta_i}(\tau_k - \tau_i),$$

where C is the set of all TSs in the network. This implies that if frame dropping should be avoided and if τ_k is chosen as the deadline constraint, then the following condition must be true

$$\tau_k - t - \sum_{i \in C_v} Z_{\delta_i}(\tau_k - \tau_i) - \sum_{i \in C - C_v} Z_{\delta_i}(\tau_k - \tau_i) \geq 0, \quad (5.6)$$

where t is the current time. HC will check Eq. (5.6) from $k = 3$ to $k = |C|$. If $\tau_{k'}$ fails Eq. (5.6), $\tau_{k'-1}$ is chosen to be the deadline constraint.

5.1.3 Updating HOLDs

HOLDs are required to be updated through piggyback information. However a HOLD may expire before a QSTA piggybacks its information. Such TSs are called *asynchronous* TSs. Otherwise, a TS is *synchronous*.

For an asynchronous TS, at the end time of a TXOP, we need to predict its next HOLD. The next HOLD of an asynchronous TS i can be approximated by assuming that the asynchronous TS generates frames with the same frame size of L_i and these frames are generated with the mean rate or peak rate. Therefore, a predicted HOLD is

$$HOLD_i^{next} = HOLD_i^{pre} + \left\lceil \frac{x \cdot (t - HOLD_i^{prev})}{L_i} \right\rceil \cdot \frac{L_i}{\rho_i}, \quad (5.7)$$

where t is the current time, $HOLD_i^{prev}$ is the previous HOLD of TS i , and x is set to ρ_i or ρ_i^h . We require QSTA to piggyback a bit for each of its TSs. This bit is set if the head-of-line frame of the corresponding TS is generated with peak rate. HC will use peak rate to predict the next HOLD if the bit for previous HOLD is 1, and use mean rate if it is 0.

5.1.4 Allocation of TXOPs

To give an opportunity for frames to be transmitted in contention periods, HC may decide to stop polling if the following conditions hold: (i) the total

lengths of contention periods don't exceed T_{cp} , (ii) the contention period shall not cause frames of TSs being dropped, and (iii) the new contention period shall be long enough to transmit at least one frame.

Condition (i) is easy to check. To check conditions (ii) and (iii), suppose that the current time is t . We need to guarantee the following condition to be true for each $k = 1, \dots, |C|$,

$$\tau_k - t \geq \left(\frac{M}{R} + O \right) + T_g + \sum_{i=1}^k Z_{\delta_i}(\tau_k - \tau_i), \quad (5.8)$$

where T_g is a guard time for HC to regain the control of the wireless medium, and M and R are the maximum MSDU size and minimum PHY transmission rate in IEEE 802.11e, respectively. In Eq. (5.8), $(M/R+O)$ is the time required to transmit a maximum frame and $Z_{\delta_i}(\tau_k - \tau_i)$ is the time required to transmit frames of TS i with a deadline before τ_k . If Eq. (5.8) is satisfied, then it implies that after the transmission of one frame in the contention period, there is sufficient time to transmit frames of TSs and no frame is expected to be dropped. After each transmission in the contention period, the HC will verify Eq. (5.8) again to determine whether it should take back the control of the medium again.

5.1.5 Admission Control

The admission control needs to guarantee that during a beacon interval, the maximum amount of data generated by all TSs can be transmitted. When a new TS i requests to enter the network, it will be admitted if $Z_i(T_B) + \sum_{j \in C} Z_j(T_B) \leq T_B - T_{cp}$.

5.2 Simulation Results

In this section, we present simulation results and compare our performance to the performance of the simple scheduler and SETT-EDD. In the first two

Table 5.1: MAC parameters used in our simulation

Beacon interval	100 <i>ms</i>
aSlotTime	20 μ s
aFragmentationTreshold	1024 octets
SIFS	20 μ s
PIFS	40 μ s
DIFS	60 μ s
Retry_limit	7

scenarios, we simulate a network with VoIP, MPEG-4, and Poisson traffic as in [27]. For each VoIP traffic source, it generates a 60-octet packet every 20 ms. The resulting data rate is 24 Kb/s. For MPEG-4 video sources, real MPEG-4 trace files [29] are used to generate I, P, and B frames. For each Poisson source, frames arrive following a Poisson arrival process and each frame may have a size of 64, 128, 256, 512, 1024, and 1518 octets with a probability of 0.6, 0.06, 0.04, 0.02, 0.25, and 0.03, respectively. The resulting data rate is 200 Kb/s.

For the second scenario, each TS is assumed to be a two-state traffic source. A TS i may switch between a high-rate state and a low-rate state. The peak data rate ρ_i^h and the minimum data rate ρ_i^l of TS i are used in the high- and low-rate states, respectively. State transition is controlled by two probabilities p and q , where the transition probability from the high- to the low-rate state is $1 - p$, and the transition probability from the low- to the high-rate state is $1 - q$. Table 5.1 lists the MAC parameters used in our simulations.

5.2.1 Scenario 1

In this scenario, we assume $T_{cp} = T_B/3$. Each QSTA has two TSs. One is a VoIP source and the other is a MPEG-4 source. Table 5.2 lists the TSPECs of the VoIP TS and MPEG-4 TS. For the MPEG-4 source, trace file is generated from the movie of Aladdin [29]. These TSs are transmitted with a hybrid

Table 5.2: The traffic parameters for VoIP and MPEG TSs in scenario 1.

	VoIP	MPEG-4 (Aladdin)
Mean data rate	24 kb/s	256 kb/s
Delay bound	40 <i>ms</i>	40 <i>ms</i>
Nominal MSDU size	60 octets	1067.2 octets
Maximum MSDU size	60 octets	1024 octets
Maximum burst size	120 octets	13966 octets
Peak data rate	24 kb/s	3.35Mb/s
User priority	6	5

access policy. In addition, each QSTA also has a Poisson traffic source, whose access policy is EDCA. The minimum and maximum contention windows for the VoIP, MPEG, and Poisson arrival traffic are (7, 31), (15, 63) and, (31, 255), respectively.

Applying these parameters to Eq. (2.1) and Eq. (2.2), it can be derived that the simple scheduler can support at most seven QSTAs. For the sake of fair comparison, if there are more than seven QSTAs, the TS of extra QSTAs will be transmitted using EDCA.

Fig. 5.1, Fig. 5.2, and Fig. 5.3 show the throughput, frame dropping rate, and access delay, respectively. Our TXOP/DC scheme does provide a higher throughput because of a lower dropping rate for MPEG-4 TSs than the other schedulers. As can be seen in Fig. 5.3, this lower dropping rate also causes a slightly higher access delay because more frames are successfully delivered without being dropped. For VoIP TSs, since the burst size is relatively small, we see that all schemes perform very close to each other.

5.2.2 Scenario 2

In this scenario, we set $T_{cp} = 0$ and place ten QSTAs in the simulated network. Among four of the ten QSTAs, each QSTA has one VoIP TS, one MPEG-4 TS,

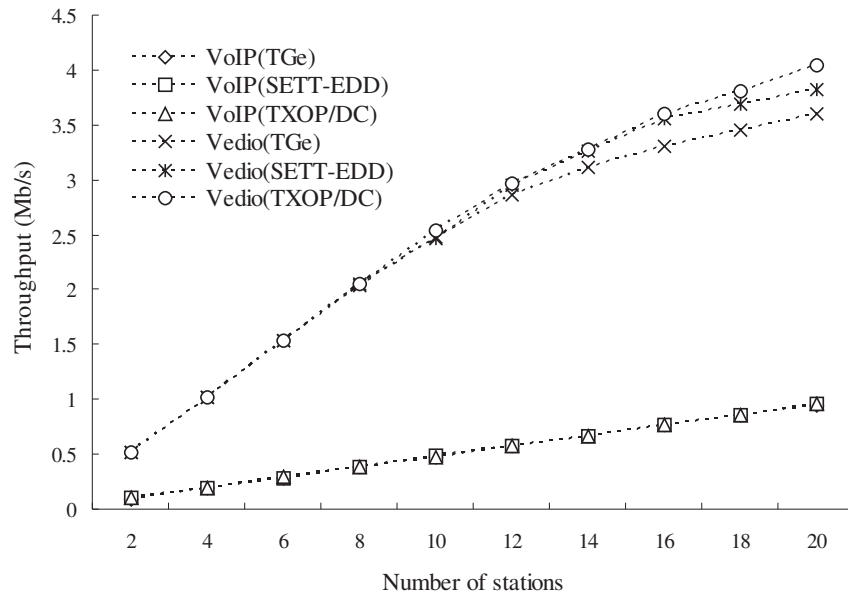


Figure 5.1: Throughputs comparison under different numbers of stations in scenario one.

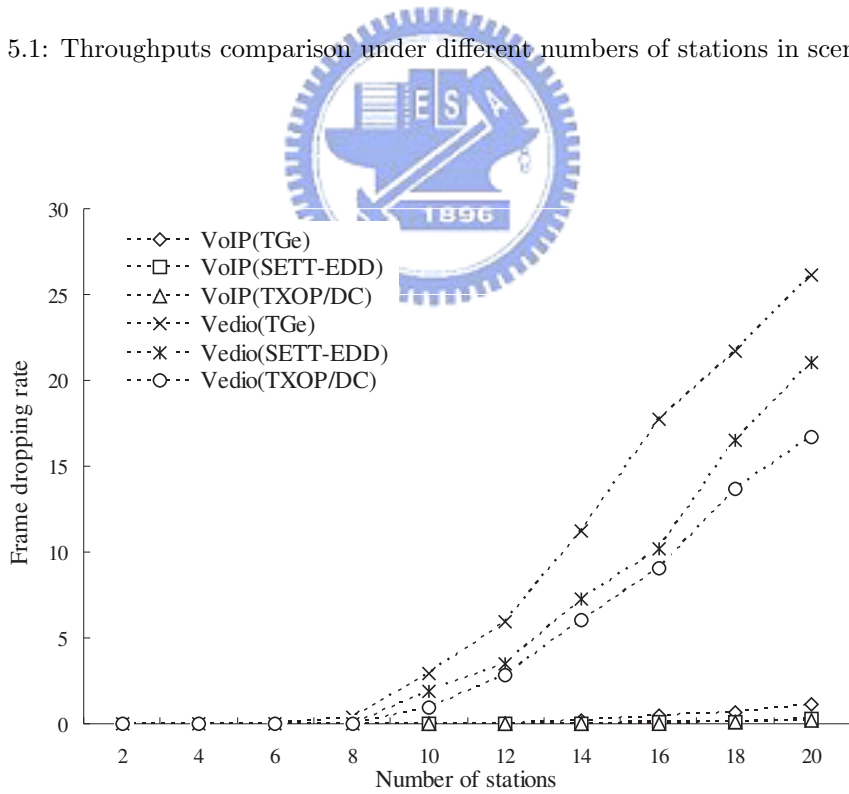


Figure 5.2: Frame dropping rate under different numbers of stations in scenario one.

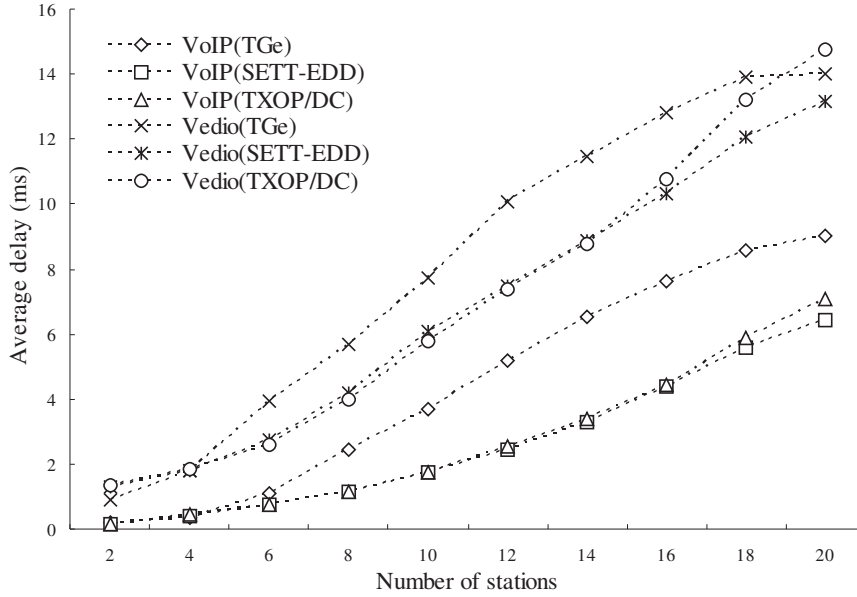


Figure 5.3: Comparison of access delay under different numbers of stations in scenario one.

and one Poisson traffic source. These four MPEG-4 traffic sources are driven by traffics generated from movies Aladdin, Start War V, Jurassic Park I, and Jurassic Park II, respectively [29]. The TSPECs of the last three movies are shown in Table 5.3. Each of the rest of the six QSTAs has only one Poisson traffic source which will be transmitted using EDCA. The PHY transmission rate is set to be 2 Mbps. The results are summarized in Table 5.4. Similar to the previous simulation results, our scheduling algorithm has a smaller dropping rate and provides higher throughput for MPEG-4 TSs. For VoIP TSs, the dropping rate is almost the same for the three scheduling algorithms.

5.2.3 Scenario 3

In this scenario, we set $T_{cp} = 0$ and place five QSTAs in the network. Among three of the five QSTAs, each has two two-state TSs, called TS1 and TS2. The transition probabilities p and q are set to 0.8 and 0.7, respectively, for both

Table 5.3: The TSPECs of MPEG TSs in scenario two.

TSPEC (MPEG-4)	Star War V	Jurassic Park I	Jurassic Park II
Mean data rate	256 kb/s	256 kb/s	256 kb/s
Delay bound	40 <i>ms</i>	40 <i>ms</i>	40 <i>ms</i>
Nominal MSDU size	1067.2 octets	1067.2 octets	1067.2 octets
Maximum MSDU size	1024 octets	1024 octets	1024 octets
Maximum Burst size	7353 octets	9973 octets	11573 octets
Peak data rate	1.76 Mb/s	2.39 Mb/s	2.78 Mb/s
User priority	5	5	5



Table 5.4: Throughput, dropping rate, and access delay in scenario two.

	Traffic Type	TGe	SETT-EDD	TXOP/DC
Throughput	MPEG-4	0.67 b/s	0.79 b/s	0.88 b/s
	VoIP	0.19 b/s	0.19 b/s	0.19 b/s
	Poisson Traffic	0.85 b/s	0.84 b/s	0.86 b/s
Dropping rate	MPEG-4	29.2 %	18.6 %	10.5 %
	VoIP	≈ 0 %	≈ 0 %	≈ 0 %
	Poisson Traffic	28.4 %	29.1 %	27.6 %
Delay	MPEG-4	16.9 <i>ms</i>	17.2 <i>ms</i>	24.0 <i>ms</i>
	VoIP	19.1 <i>ms</i>	18.7 <i>ms</i>	26.7 <i>ms</i>
	Poisson Traffic	151.3 <i>ms</i>	158.5 <i>ms</i>	165.3 <i>ms</i>

Table 5.5: TSPECs of TS1 and TS2 in scenario three.

Parameters	TS1	TS2
Mean data rate	128 kb/s	384 kb/s
Delay bound	20 <i>ms</i>	30 or 60 <i>ms</i>
Nominal MSDU size	1024 octets	1024 octets
Maximum MSDU size	1024 octets	1024 octets
Maximum burst size	9216 octets	14336 octets
Peak data rate	3 Mb/s	1.5 Mb/s
p	0.8	0.8
q	0.7	0.7

Table 5.6: Throughput, dropping rate, and access delay for different traffic sources in scenario three when the delay bound of TS2 is 30 ms.

		TGe	SETT-EDD	TXOP/DC
Throughput	TS1	0.19 b/s	0.31 b/s	0.36 b/s
	TS2	0.41 b/s	0.75 b/s	0.79 b/s
Dropping rate	TS1	58.8 %	34.1 %	18.1 %
	TS2	56.8 %	24.0 %	20.3 %
	Average	51.1 %	27.8 %	17.6 %
Delay	TS1	10.3 <i>ms</i>	8.0 <i>ms</i>	8.4 <i>ms</i>
	TS2	15.9 <i>ms</i>	12.9 <i>ms</i>	14.7 <i>ms</i>

TS1 and TS2. The delay bound for TS1 is 20 ms, and the bound for TS2 is 30 ms or 60 ms. The other parameters are shown in Table 5.5. For the rest of the two QSTAs, each has a Poisson traffic source which is transmitted according to EDCA rules.

Table 5.6 and Table 5.7 show the simulation results when the delay bounds of TS2 are set to 30 ms and 60 ms, respectively. The results show that when the delay bound of TS2 is increased, the dropping rates for both TS1 and TS2 decrease. This also increases the throughput as well as the access delay of TS2.

Table 5.7: Throughput, dropping rate, and access delay for different traffic sources in scenario three when the delay bound of TS2 is 60 ms.

		TGe	SETT-EDD	TXOP/DC
Throughput	TS1	0.23 b/s	0.32 b/s	0.39 b/s
	TS2	0.71 b/s	0.76 b/s	0.82 b/s
Dropping rate	TS1	50.6 %	30.4 %	17.8 %
	TS2	27.2 %	20.3 %	15.1 %
	Average	34.6 %	23.5 %	16.1 %
Delay	TS1	9.7 <i>ms</i>	8.1 <i>ms</i>	6.67 <i>ms</i>
	TS2	25.6 <i>ms</i>	17.1 <i>ms</i>	29.1 <i>ms</i>

5.3 Concluding Remarks

In this chapter, we have proposed a scheduling algorithm called TXOP/DC for IEEE 802.11e WLANs. The scheduling algorithm imposes a deadline constraint on each TXOP to restrict a station to transmit only frames before the deadlines constraint. It also utilizes the information of data rates measured by stations to reduce frame dropping and decrease polling overheads. Simulation results show that our proposed algorithm has a smaller frame dropping rate and a higher channel throughput.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

In this dissertation, we investigate the performance issue and the QoS issue of WLANs. In Chapter 3, we have proposed a new MCB algorithm which explores the possibility of using multiple backoff chains to make stations to adapt to different congestion levels. With the capability of switching to different backoff chains, MCB offers higher throughput than the existing protocols, such as GDCF, IEEE 802.11, MILD, EIED, and LILD, and still provides fair access to the wireless channel.

In Chapter 4 we propose a jamming-based MAC (JMAC) protocol which is free from both the hidden terminal problem and the erroneous reservation problem. JMAC separates source stations' traffic from destination stations' traffic into different channels, and explicitly signals the channel status by jamming the channels. We also discuss how to choose a proper ratio of bandwidth for the two channels. It is shown that, the optimal ratio changes with data frame sizes so there is no global optimal value for all sizes. But the values in the range of 0.7 and 0.8 would be a good approximation for frame sizes ranging from 128 to 2048 bytes. It is also shown that channel division incurs

some cost in terms of the frame transmission time; however, from simulation results, the advantages of being free from the hidden terminal problem and the erroneous reservation problem, and the benefit of more concurrent transmissions in JMAC can compensate the cost and provide higher throughput than IEEE 802.11.

In Chapter 5 we have proposed a scheduling algorithm for IEEE 802.11e WLANs. The scheduling algorithm adds a deadline constraint on each TXOP to restrict frames to be transmitted in the TXOP. Simulation results show that our proposed algorithm has a smaller dropping rate than the simple scheduler of IEEE 802.11 and SETT-EDD. This smaller dropping rate also increases the network throughput but a higher access delay due to less frames being dropped.

6.2 Future Work

The study of wireless channel allocation starts from the performance issue to the QoS issue. More and more attention is shifted to the QoS-related topics. The following lists some possible research directions:

- The EDCA of IEEE 802.11e is a random access scheme. Due to its efficiency and simplicity compared to HCCA, there have been many studies of transporting real-time traffic by EDCA. However, the drawback of EDCA is its high dropping rate when the network is in high traffic load. Therefore, research on connection admission control and enhancement of the channel access of EDCA could be directed to future work.
- The VoIP has become one of most popular applications. How to support VoIP service in WLANs is a good research direction. This includes the studies of tuning MAC parameters and provide enhancement on MAC to achieve better performance.

- WLANs have been widely deployed in public places such as schools, airports, railway stations, cafes, megastores, etc.. In such public environments, there are different types of users who have different bandwidth requirements accessing the networks. It is desirable to classify them and allocate bandwidth according to their requirements or importance. One of the solutions to this application is hierarchical link-sharing [30, 31] which has been extensively studied for wired networks. How to support link-sharing in WLANs will be an interesting future work.



Bibliography

- [1] N. Abramson, “The ALOHA system - Another alternative for computer communication,” in *1970 Fall Joint Comput. Conf., AFIPS Conference Proceedings*, 1970, pp. 281–285.
- [2] F. A. Tobagi and L. Kleinrock, “Packet switching in radio channels: Part I - Carrier sense multiple-access modes and their throughput-delay characteristics,” *IEEE Transactions on Communications*, vol. com-23, no. 12, pp. 1400–1415, 1975.
- [3] IEEE P802.11 TGe, “Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Medium Access Control (MAC) Enhancements for Quality of Service (QoS),” in *802.11e/D13.0*, January 2005.
- [4] S. Mangold, S. Choi, G. R. HierTZ, O. Klein, and B. Walke, “Analysis of IEEE 802.11e for QoS support in wireless LANs,” in *IEEE Wireless Communications*, December 2003, pp. 40–50.
- [5] Y. Xiao, H. Li, and S. Choi, “IEEE 802.11e: QoS provisioning at the MAC layer,” in *IEEE Wireless Communications*, June 2004, pp. 36–43.
- [6] V. Bharghavan, “MACAW: A media access protocol for wireless LAN’s,” in *ACM SIGCOMM*, 1994, pp. 212–225.

- [7] G. Bianchi, L. Frata, and M. Oliveri, "Performance evaluation and enhancement of the CSMA/CA MAC protocol for 802.11 wireless LANs," in *IEEE Symposium on Personal Indoor and Mobile Radio Communications*, Oct. 1996, pp. 392–396.
- [8] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 3, pp. 535–547, 2000.
- [9] F. Cali, M. Conti, and E. Gregori, "IEEE 802.11 protocol: Design and performance evaluation of an adaptive backoff mechanism," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 9, pp. 1774–1786, 2000.
- [10] M. Natkaniec and A. R. Pach, "An analysis of the backoff mechanism used in IEEE 802.11 Networks," in *IEEE Symposium on Computers and Communications*, July 2000, pp. 444–449.
- [11] F. Cali, M. Conti, and E. Gregori, "Dynamic tuning of the IEEE 802.11 protocol to achieve a theoretical throughput limit," *IEEE/ACM Transactions on Networking*, vol. 8, no. 6, pp. 785–799, 2000.
- [12] Z. J. Haas and J. Deng, "On optimizing the backoff interval for random access schemes," *IEEE Transactions on Communications*, vol. 51, pp. 2081–2090, Dec. 2003.
- [13] D. Raychaudhuri and K. Joseph, "Analysis of the stability and performance of exponential backoff," in *IEEE WCNC*, vol. 3, March 2003, pp. 1754–1759.
- [14] N.-O. Song, B.-J. Kwak, J. Song, and L. E. Miller, "Enhancement of IEEE 802.11 distributed coordination function with exponential increase exponential decrease backoff algorithm," in *IEEE VTC 2003-Spring*, vol. 4, April 2003, pp. 2775–2778.

- [15] G. Bianchi and I. Tinnirello, “Kalman filter estimation of the number of competing terminals in an IEEE 802.11 network ,” in *IEEE INFOCOM*, 2003, pp. 844–852.
- [16] Y. Kwon, Y. Fang, and H. Latchman, “Design of MAC protocols with fast collision resolution for wireless local area networks,” *IEEE Transactions on Wireless Communications*, vol. 3, pp. 793–807, 2004.
- [17] Y. Wang and B. Bensaou, “Achieving fairness in IEEE 802.11 DFWMAC with variable packet lengths,” in *IEEE GLOBECOM*, vol. 6, 2001, pp. 3588–3593.
- [18] P. M. Soni and A. Chockalingam, “Analysis of link-layer backoff schemes on point-to-point markov fading links,” *IEEE Transactions on Communications*, vol. 51, no. 1, pp. 29–32, 2003.
- [19] V. Vitsas, “Throughput analysis of linear backoff scheme in wireless LANs,” *IEEE Electronic Letters*, vol. 39, no. 1, pp. 99–100, 2003.
- [20] C. Wang, B. Li, and L. Li, “A new collision resolution mechanism to enhance the performance of IEEE 802.11 DCF,” *IEEE Transactions on Vehicular Technology*, vol. 53, no. 4, pp. 1235–1246, July 2004.
- [21] F. A. Tobagi and L. Kleinrock, “Packet switching in radio channels: Part II - The hidden terminal problem in carrier sense multiple-access modes and the busy-tone solution,” *IEEE Transactions on Communications*, vol. com-23, no. 12, pp. 1417–1433, 1975.
- [22] C. Wu and V. O. K. Li, “Receiver-initiated busy-tone multiple access in packet radio networks,” in *Proceedings of the ACM workshop on Frontiers in computer communications technology*, pp. 336–342.

- [23] J. Deng and Z. J. Haas, "Dual busy tone multiple access (DBTMA): A new medium access control for packet radio networks," in *IEEE ICUPC*, 1998, pp. 973–977.
- [24] Z. J. Haas and J. Deng, "Dual busy tone multiple access (DBTMA) - performance evaluation," in *IEEE VTC*, 1999, pp. 314–319.
- [25] P. Karn, "MACA - A new channel access method for packet radio," in *ARRL CRRL Amateur Radio 9th Computer Networking Conference*, April 1990, pp. 134–140.
- [26] F. Talucci, M. Gerla, and L. Fratta, "MACABI (MACA by invitation): A receiver oriented access protocol for wireless multiple networks," in *PIMRC*, 1994, pp. 1–4.
- [27] A. Grilo, M. Macedo, and M. Nunes, "A scheduling algorithm for QoS support in IEEE 802.11e networks," *IEEE Wireless Communications*, vol. 10, pp. 36–43, June 2003.
- [28] R. Jain, D. Chiu, and W. Hawe, "A quantitative measure of fairness and discrimination for resource allocation in shared computer systems," in *DEC Research Report TR-301*, 1984.
- [29] "Video traces for network performance evaluation," <http://trace.eas.asu.edu/>.
- [30] J. Bennett and H. Zhang, "Hierarchical packet fair queuing algorithms," *IEEE/ACM Transactions on Networking*, vol. 5, no. 5, pp. 675–689, 1997.
- [31] S. Floyd and V. Jacobson, "Link-sharing and resource management models for packet networks," *IEEE/ACM Transactions on Networking*, vol. 3, no. 4, pp. 365–386, 1995.