ON QUALITY OF SERVICE AND FAIRNESS OF MEDIUM ACCESS CONTROL IN WIRELESS AD-HOC NETWORKS

WANG, YU

NATIONAL UNIVERSITY OF SINGAPORE

2000

ON QUALITY OF SERVICE AND FAIRNESS OF MEDIUM ACCESS CONTROL IN WIRELESS AD-HOC NETWORKS

WANG, YU

(*B.Eng., Shanghai Jiaotong University, P.R.China*)

A THESIS SUBMITTED

FOR THE DEGREE OF MASTER OF ENGINEERING

DEPARTMENT OF ELECTRICAL ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2000

Acknowledgments

I would like to express my deep gratitude to my supervisor, Dr. Brahim Bensaou for his valuable guidance, support, encouragement, advice and comments during my research in the labyrinth of wireless networks. I would also like to thank him for giving me the opportunity to present our paper at an international conference, which is a valuable experience for me.

I am very grateful to my co-supervisor, A/Prof. Chi Chung Ko for his constant support and encouragement to my research and valuable advice on how to conduct research effectively and make impressive presentations.

I would also like to thank the Digital and Systems Applications Laboratory and Centre for Wireless Communications for providing me with all the necessary computing and communication facilities and congenial environments to carry out my research.

At last, I would like to thank the National University of Singapore for the Research Scholarship which makes this thesis work possible.

Summary

An ad-hoc network is a dynamic multi-hop wireless network deployable without the aid of any pre-existing network infrastructure or centralized administration. It can be rapidly deployed on demand and thus is very attractive in many military and civilian applications.

An efficient and effective medium access control (MAC) protocol with which mobile stations can share a common broadcast channel is essential in an ad-hoc network. A well-designed MAC protocol should achieve reasonable throughput as the channel is a scarce resource. Sometimes, prioritized access is required for service differentiation for mixed types of traffic, such as packetized voice and data in a disaster relief environment. In some other environments such as collaborative computing, fair access to the channel is also very important. Existing MAC protocols either do not support prioritized access or do not solve fairness problems well in multi-hop environment. This and in general quality of service (QoS) support issue in ad-hoc networks motivate the design of new MAC protocols for such networks.

In this thesis, we propose two MAC protocols that can support either prioritized or fair access. One is called priority based multiple access (PriMA) protocol, which takes into account the QoS requirements of the packets queued in stations to accord each station prioritized access to the channel dynamically. It can support service differentiation for real time and non-real time traffic and be integrated into an overall QoS architecture for an ad-hoc network. The other is called fair share based multiple access (FSMA) protocol. We first define a new metric to measure fairness and then propose the FSMA protocol which can provide each station with prescribed fair share regardless of its location. The FSMA protocol is meant to solve the fairness problem that can appear in IEEE 802.11 based collaborative computing environments. Both protocols' performance is evaluated against alternative MAC protocols via computer simulations. Included results confirm that these two protocols are effective in achieving their design goals and thus can cater to the special needs of wireless ad-hoc networks.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

With the increasing affordability of laptop computers and wireless data communication devices, wireless communications between mobile users are becoming more and more popular. Sometimes, when there is no communication infrastructure such as base stations and mobile switching centres that make up today's cellular networks or the existing infrastructure is expensive or inconvenient to use, people can set up networks of their own on the fly, which are so called “ad-hoc networks”. Due to the limited transmission range of mobile stations, ad-hoc networks are usually multi-hop and cooperative, which requires that each participating mobile station sometimes act as a router, forwarding packets for other stations.

Some advantages of ad-hoc networks include:

On demand setup No *a priori* plan is needed.

Fault tolerance Single point of failure does not exist and mobile stations can automatically find alternative paths to communicate with others.

Unconstrained connectivity If two stations are close enough, they can then form an instantaneous link without further ado.

As ad-hoc networks can be installed quickly in emergency or some other special situations and are self-configurable, they are very attractive in many applications, such as battlefield communications, disaster recovery (fire, earthquake) and collaborative computing during special events (concerts, festivals and etc.). There has been a growing interest in ad-hoc networks in recent years [1–5]. For example, the Internet Engineering Task Force (IETF) has set up the mobile ad-hoc networks (MANET) working group [3, 6] to develop and evolve MANET routing specifications and introduce them to the Internet Standards track [7–12]. Recently, the Swiss Federal Institute of Technology (EPFL) has launched an up to ten-year project TERMINODES [13, 14] whose goal is to design self-operated mobile communication systems and thus empower citizens with direct communication facilities. Research projects such as Monarch [1] in Carnegie Mellon University and WINGs [15] in University of California at Santa Cruz are also notable research initiatives in ad-hoc networks.

Much of the research work has been focused on designing distributed routing and medium access control (MAC) protocols for ad-hoc networks. In this thesis, we will focus on this latter aspect of ad-hoc networks and more importantly on QoS enabled MAC protocol design.

1.2 Problem Statement

As there is no central entity to do channel planning and allocation, all stations in an ad-hoc network should share a common broadcast channel to transmit and receive packets. Multiple simultaneous transmissions can result in garbled data, making communication impossible. A MAC protocol moderates stations' access to the shared channel by defining rules for each station so that they can communicate with each other in an orderly and efficient manner.

A well designed MAC protocol should exhibit the following properties:

Throughput Throughput can be defined as the fraction of the channel capacity used for data transmission. As the wireless channel is a scarce resource, throughput is one of the foremost design considerations. However, throughput should not be the sole target of a MAC protocol. Otherwise the easiest way to optimize the total throughput in a shared channel is just to eliminate sharing and turn the channel to a pair of source and destination stations exclusively. Obviously, this contradicts the objective of networking for resource sharing and effective communications.

Fairness of access In an ad-hoc network, due to lack of central administration and partially-connected network topology, contention is not homogeneous. Some stations may be at a disadvantage in accessing the shared channel and suffer severe throughput degradation when the load to the channel is high. A well designed MAC protocol should avoid such situations.

Priority support Despite limited bandwidth, ad-hoc networks are still expected to operate in diversified application scenarios and may be required to support prioritized traffic. For example, in disaster relief situations, sometimes packetized voice should get through earlier than other data packets. Additionally, for an ad-hoc network to function properly, packets for setting up and maintaining routes should have a higher priority than other packets. Here, we should note that, priority support cannot be achieved by just scheduling packets in single station, say moving an urgent packet to the head of its queue, as this station still needs to compete with other stations. Therefore, priority support is desirable so that a station queued with the most urgent packet can have higher priority than other stations in accessing the shared channel.

Routing support It has been discussed that packets for setting up and maintaining routes should have a higher priority and a well designed MAC protocol should meet this requirement. Additionally, unlike traditional IP routers in wireline networks, any station that acts as a router in ad-hoc networks usually has only one network interface, and there are no separate point-to-point links for it to route packets or exchange routing information. All the traffic should go through the shared medium or channel. According to some recently proposed ad-hoc routing protocols [16–18], when an ad-hoc network converges to a certain degree, some stations that behave as cluster heads [17] or belong to the core of the routing structure [18] will handle more traffic that transits through them. Obviously, in this case, these routers require higher priorities and/or higher throughput than other stations. This factor should also be considered when designing MAC protocols for ad-hoc networks.

Some of the characteristics may not be compatible with others and certain properties are achieved at the expense of others in some design tradeoffs. For example, due to the dynamic nature of ad-hoc networks, any station may act as a router or a normal host alternately. Therefore, stations should have dynamic priorities instead of static priorities, which cannot be known *a priori*. A MAC protocol that arbitrates stations' access to the shared channel according to their priorities may sometimes violate the goal of short-term fairness, because some stations may be deprived of the opportunity to access the channel temporarily when other stations have high priority packets and need to send them out in a burst. From another point of view, if a MAC protocol tries to ensure stations' fair access to the channel, it will not be able to allow some stations with urgent packets to gain prioritized access to the channel as it may violate the fairness goal.

Therefore, we argue that priority-based access and fair-based access are two conflicting goals that are hard to reconcile in one MAC protocol as it requires different approaches to achieve either goal.

Based on above reasons, our goal is to design *separate* MAC protocols that can achieve reasonable throughput and provide better support for routing, while at the same time stations can have either priority-based access, or fair-based access to the shared channel according to the different application scenarios.

1.3 Thesis Contribution

The thesis proposes two MAC protocols for ad-hoc networks. One is called priority based multiple access (PriMA) protocol [19, 20] which takes into account the QoS requirements of the packets queued in stations and can provide each station with priority-based access to the channel. The prioritized access is useful to achieve service differentiation when there are mixed types of traffic, such as packetized voice and data in disaster relief scenarios. The other is called fair share based multiple access (FSMA) protocol [21] which allows the stations to access the channel according to the prescribed fair share. It is useful to ease the fairness problem that results from stations' different locations in ad-hoc networks. This protocol may be useful in collaborative computing during special events, where each participant can gain fair share to make the collaboration function properly.

1.4 Thesis Organization

The remainder of the thesis is organized as follows. Chapter 2 investigates in detail the characteristics of ad-hoc networks that challenge MAC protocol design, and then gives a review on existing MAC protocols that may be used in ad-hoc networks. Chapter 3 describes PriMA protocol for ad-hoc networks with mixed traffic and evaluates its performance against existing MAC protocols. Chapter 4 describes FSMA protocol for ad-hoc networks where cooperative computing is dominant and evaluates its performance against alternative MAC protocols. Chapter 5 summarizes the work presented in the thesis and raises some points of interest and directions for future work.

Chapter 2

Literature Review

There have been a plethora of MAC protocols proposed for different kinds of wireless networks in the past. However, many of them cannot be used in ad-hoc networks as these protocols depend on some features that ad-hoc networks lack. Section 2.1 investigates the special characteristics of ad-hoc networks that make them different from other kinds of networks. Sections 2.2 and 2.3 give a review on past MAC protocols that may be used in ad-hoc networks, from simple pioneering protocols such as ALOHA to advanced protocols such as the distributed foundation wireless MAC (DFWMAC) protocol in IEEE 802.11 working group's recommended standard for wireless local area networks (LANs). In sections 2.4 and 2.5, we review some protocols that target either fairness of access or real-time traffic support. Section 2.6 summarizes this chapter.

2.1 Ad-hoc Networks Close-up

When it comes down to designing a MAC protocol for ad-hoc networks, we should note that ad-hoc networks lack many features that are present in traditional wireline and wireless infrastructure networks and should not be relied upon. Below are some of the difficulties that impose great challenges on MAC protocol design.

No central administration A central station that can be reached by all stations, such as a base station (BS) or access point (AP) in cellular networks or wireless infrastructure LANs, is not available or cannot be depended upon in an ad-hoc network. Due to the dynamic nature of ad-hoc networks, it is difficult for a station to maintain up-to-date global information and reliance on such information for scheduling stations' access can make MAC protocols much more complex and less robust. Therefore, those centralized MAC protocols either cannot be used or are not favoured in ad-hoc networks.

"Hidden terminal" problem Due to the limited transmission range of mobile stations, multiple transmitters within range of the same receiver may not know one another's transmissions, and thus in effect are "hidden" from one another. For example, in Figure 2.1, station A is station C's hidden terminal and vice versa. If there is no means for coordination between hidden stations such as A and C, they may transmit at around the same time and result in collisions at the receiver B. This problem was first investigated by Tobagi et al. in [22] which showed that a hidden terminal could degrade throughput significantly. The hidden terminal problem is common in ad-hoc networks due

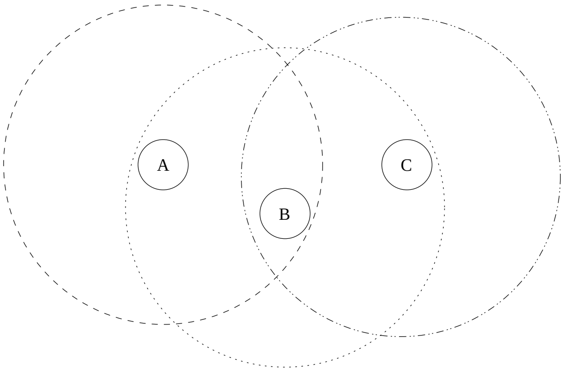


Figure 2.1: "Hidden terminal" problem

to their multi-hop characteristics and should not be ignored when designing MAC protocols.

Single channel operation Multiple channel operations usually require separate radio transceivers which may be economically infeasible at present. Additionally, there is no central scheduler to assign channels. So we assume all stations have to transmit and receive in the same channel and will consider only single-channel MAC protocols thereafter. One consequence of this is that duplexing in MAC layer is done using time division duplex (TDD).

No collision detection capability Due to the "near-far effect", a mobile station cannot know whether its transmission is successful by just listening to the channel as the signal from other stations will be drowned in its own transmitted signal. Therefore, unlike their wired peers in Ethernet, mobile stations do not have collision detection (CD) capability and depend on positive acknowledgment from receiving stations to ascertain that a previous transmission is successful.

No global synchronization The distributed nature of ad-hoc networks makes it difficult and costly to achieve global synchronization and operate in slotted mode. Therefore, time division multiple access (TDMA) based protocols cannot be used despite their great flexibility.

With such limitations, only MAC protocols that are distributed (no assumption of the availability of a central node), use a single channel and require no global clock for synchronization are considered in this thesis.

2.2 Pioneering Protocols

ALOHA [23] is the earliest random access MAC protocol for packet radio networks. Any station can transmit a data packet at any time regardless of the state of the wireless channel. After sending a data packet, a station will back off for a random amount of time if it does not receive an acknowledgment packet in due time. As the vulnerability period (the period of time during which a data packet may encounter interference at the intended destination) is twice the packet's transmission time, the chance of collisions is high and the maximum channel throughput is very low (about 18% of the channel capacity with a separate channel for acknowledgment).

Kleinrock et al. [24] introduced the carrier sense multiple access (CSMA) protocols to reduce the high packet loss rates in ALOHA. CSMA protocols require a station to listen to the channel before transmitting a data packet. If a station senses the channel busy, it should wait until the channel becomes clear. After that, it can either begin transmission immediately (1-persistent CSMA), or wait for a random amount of time (non-persistent CSMA), or transmit with probability p (p-persistent CSMA). Kleinrock et al. showed that CSMA protocols can boost channel throughput for a single-hop network where all stations are within transmission range of one another and propagation delay is much less than packet transmission time.

However, Tobagi et al. [22] showed that CSMA protocols suffer from "hidden terminal" problem, which can degrade the throughput significantly, as sensing the channel at the transmitter does not offer any information about the state of the channel at the receiver. This implies that the vulnerability period may be as long as that of ALOHA in an ad-hoc network with hidden terminals and CSMA's performance can degrade to that of ALOHA.

To address the "hidden terminal" problem, Karn [25] proposed a new MAC protocol called multiple access with collision avoidance (MACA) which formed the basis of some other more sophisticated protocols. In MACA, carrier sense is discarded totally. Instead, short request-to-send (RTS) and clear-to-send (CTS) packets should be first successfully exchanged between a pair of source and destination stations before actual data packet transmission can begin. Other stations that overhear RTS and/or CTS packets should defer their access to the channel until ongoing data packet transmission finishes. Here, RTS and CTS packets are used as control packets to reserve the channel around the receiver and therefore can alleviate the hidden terminal problem.

2.3 Throughput Enhanced Protocols

2.3.1 Multiple Access Collision Avoidance for Wireless

As MACA was originally proposed for amateur packet radio use and left many details unspecified, Bharghavan et al. [26] refined and extended it to MACA for wireless (MACAW) for indoor wireless LANs.

The authors showed that hidden terminals and binary exponential backoff can also lead to fairness problem, which means that some stations may suffer severe throughput degradation when the channel load is high. According to the binary exponential backoff algorithm, a station's contention window[1] size will be doubled after each unsuccessful transmission until a prescribed maximum value is reached and will return to a minimum value if a data packet is successfully transmitted. Therefore a station that has last succeeded in transmitting a data packet will enjoy a much smaller contention window, thus statistically shorter backoff timer than other stations that have failed. When the channel load is high, these other stations may suffer excessive access delay and severe degradation in throughput.

To solve the problem, the authors proposed the MACAW protocol which includes a different backoff algorithm called multiplicative increase linear decrease (MILD) and a backoff copy scheme. In MACAW, each station sets its current backoff timer in the packet to be sent, other stations that receive the packet copy the backoff timer to themselves. In this way, all the stations within communication

---

[1] The backoff timer is generated from a uniform distribution which ranges from 0 to the size of the current contention window.

reach will use the same backoff timer to compete for access to the channel. Additionally, instead of resetting the contention window to the minimum value after each successful transmission, a station will decrease its contention window linearly. Otherwise, with the new backoff copy scheme, other stations near the station that last succeeds will use the same small contention window size and collisions will be high. With these modifications, MACAW can alleviate the fairness problem and increase throughput.

When reliable delivery of data is needed, MACAW includes additional control packets. One is link level acknowledgment which enables fast recovery and thus can increase transport level throughput in an error prone wireless channel. Another is the data sending (DS) packet that a station sends before sending a data packet. DS packet informs other stations about the existence and the length of the following data packet, so that they can compete more effectively. Thus the final enhanced handshake between a pair of sender and receiver is RTS-CTS-DS-DATA-ACK.

In addition, "per stream" fairness was also introduced in MACAW. It means that each stream that originates from either the same station or different stations should be treated equally and given comparable share of the channel capacity. This is different from "per station" fairness which accords channel capacity to individual stations instead of individual streams. For multiple streams that originate from a station, MACAW keeps separate queues for each stream and runs backoff algorithms independently for each stream to achieve "per stream" fairness.

However, MACAW still left some problems unsolved. For example, in the simple configuration shown in Figure 2.2, station A has packets for station B and station

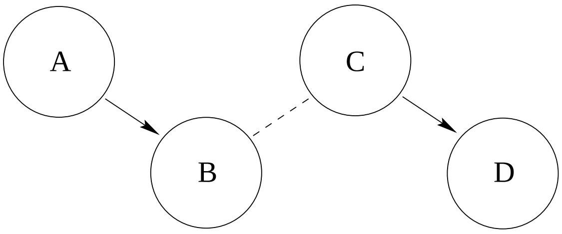


Figure 2.2: Fairness problem

C has packets for station D. Stations A and C are both greedy sources, which means that they always have packets for stations B and D respectively. Station C's transmission request (RTS packet) can always be received free from collisions by D while A's RTS may collide with C's concurrent transmission at B, which results in no response from B. Therefore station C usually succeeds in transmitting its packets and station A will suffer severe degradation in throughput. Another problem for MACAW is that the backoff copy scheme only works when congestion is homogeneous, that is, each station has similar number of competing neighbors. This is not generally the case in ad-hoc networks due to their multi-hop characteristics. For more details about these problems, readers can refer to reference [26] whose discussion is quite comprehensive.

2.3.2 Floor Acquisition Multiple Access

The protocols described so far that utilize RTS/CTS handshake do not use carrier sensing, or more accurately, they use "packet sensing" only. This means that a station only reacts to a complete, interference-free packet and ignores any other type of channel activities. It is still possible that data packets collide with each other or with control packets[2]. Fullmer et al. [27–31] unified the basic schemes used in many MAC protocols for carrier sensing and collision avoidance into a new channel access discipline called floor acquisition multiple access (FAMA).

Evaluation of FAMA shows that combination of carrier sensing and RTS-CTS exchange can improve throughput significantly. In addition, by enforcing that CTS packets are sufficiently longer than RTS packets and with sufficiently long waiting time after the channel state transits from busy to idle, data packet collision can be avoided and the hidden terminal problem can be solved completely. However, FAMA does not include link-level acknowledgment which seems to be a design flaw. Although FAMA can guarantee that data packets are transmitted free of collision, other factors such as interference, fades and multipath effects can render data packet corrupted or even lost. Transport layer recovery may be costly, as shown in [32], and fast link-level acknowledgment is beneficial to increase transport layer throughput. Therefore, it is necessary to implement at least MAC-level acknowledgment in FAMA even if full MAC-level recovery may not be required by upper layers.

2.3.3 Distributed Foundation Wireless MAC Protocol

IEEE 802.11 working group also proposed a similar MAC protocol called distributed foundation wireless medium access control (DFWMAC) [33] for wireless LANs, which in essence is a carrier sense multiple access with collision avoidance

---

[2] Here we refer to RTS, CTS, ACK packets as control packets to differentiate them from actual data packets.

(CSMA/CA) protocol. DFWMAC includes two coordination functions, namely distributed coordination function (DCF) and point coordination function (PCF), for ad-hoc and infrastructure wireless LANs respectively. As PCF depends on a fixed station called access point (AP) as a central coordinator to operate, which is not possible in ad-hoc networks, we will not consider it thereafter.

DFWMAC DCF provides basic and RTS/CTS access method. The basic access method is just non-persistent CSMA with acknowledgment. The RTS/CTS access method combines RTS/CTS packet exchange between a pair of sender and receiver with carrier sensing. It uses a 4-way dialog, namely RTS-CTS-DATA-ACK, to combat the hidden terminal problem and errors in wireless channel. RTS and CTS packets include in their headers a field called network allocation vector (NAV). It is used to inform neighboring stations who overhear the RTS and/or the CTS packets how long they should defer access to the channel.

Since the inception of IEEE 802.11 working group, there has been a lot of literature on the performance evaluation of IEEE 802.11 wireless LANs [34–45]. Chhaya et al. [35, 46] have showed that the performance of the RTS/CTS access method degrades much slower than the basic access method when the number of hidden terminals is large, or the offered load is significantly larger than the channel capacity; therefore, the RTS/CTS method is more robust to fluctuations in parameter values which are common in ad-hoc networks.

However, DFWMAC DCF still suffers from the fairness problem which was first investigated in MACAW [26]. This is due to the binary exponential backoff algorithm used in DFWMAC DCF for collision resolution, which we have explained previously in section 2.3.1. Additionally, DFWMAC DCF is not immune to the hidden terminal problem as it does not meet the conditions of RTS/CTS packet size relationship and waiting time enforced by FAMA and data packets may still suffer collisions.

2.4 Fairness Based Protocols

Bharghavan et al. [26] pioneered the work in addressing fairness problems in wireless LANs and proposed the MACAW protocol. However, MACAW still left some problems unsolved, as discussed previously in section 2.3.1.

More recently, Ozugur et al. [47, 48] proposed a p_{ij}-persistent CSMA based backoff algorithm to achieve fairness. This paper first defined the fairness index to be the ratio of maximum link throughput to minimum link throughput and the fairness goal is to minimize the fairness index. Then it proposed that each station calculated a link access probability p_{ij} for each of its links based on the number of connections from itself and its neighbors (connection based), or based on the average contention period of its and other stations' individual links (time based). Whenever its backoff period ends, station i will send RTS packet to j with probability p_{ij} or back off again with probability $1-p_{ij}$. The proposed scheme relies on periodic broadcast packets in the time-based approach or on aperiodic broadcast packets in the connection-based approach whenever the network topology changes. This paper also investigated the effects of combining contention window exchange with either connection-based or time-based approach.

However, none of the schemes can achieve the best results for all network configurations investigated in [48] and sometimes the best results are achieved when these schemes are in fact not used. In addition, broadcast packets are unreliable to disseminate information to neighbors. As the RTS/CTS access method cannot be used and no acknowledgment packet can be sent in this case, no one can ensure if its broadcast packets can be delivered to all of its neighbors, which makes the performance of this method tightly coupled to the successful dissemination of the information in the network.

In the ongoing research work of Vaidya et al. [49], the authors identified the difficulties in defining fairness itself in multi-hop networks and defined a generalized resource sharing (GRS) algorithm which needs further investigation as it relies on sorting all the active flows in the networks which in turn requires availability of global information. In addition, a distributed fair scheduling algorithm was also proposed to achieve fairness on local area networks (one hop) and its performance was investigated.

2.5 Quality of Service (QoS) Targeted Protocols

None of the aforementioned MAC protocols takes any packet level QoS parameters such as packet loss ratio, packet delay, etc. into account when scheduling access to the channel among stations. They can support only best effort delivery service, and thus limit the applications in ad-hoc networks.

Therefore, recently, there has been a surge in modifying and extending equalitybased medium access schemes to support priority-based access when the packets queued at stations have different QoS requirements. Although this research effort does not compete with the effort deployed in the wireless LANs (infrastructure based), some researchers have been working on QoS support for distributed wireless networks [50–52].

In [50, 51], stations with real-time packets in the queue would jam the channel with black bursts (BBs) whose length is proportional to the delay incurred. The station that sends the longest BBs wins access to the channel and can transmit its packet thereafter. However, this approach fails when hidden terminals exist as those hidden terminals may have experienced the same delay and each BB contention period is not guaranteed to produce a unique winner, thus real-time data packets will still suffer from collisions.

In [52], group allocation multiple access (GAMA) protocol was proposed for scheduling real-time and datagram traffic in a single-hop wireless ad-hoc LAN. GAMA includes a contention period during which stations can send requests to join a transmission group and a contention-free period during which the stations in the transmission group take turns to transmit packets. This approach does not work well if hidden terminals exist. This can be explained as follows. On the one hand, if hidden terminals do not join the transmission group that they may interfere with, GAMA cannot ensure that data packets are transmitted free from collision. On the other hand, if hidden terminals do join the transmission group to avoid collisions, following the same logic, then all the other stations in the network have to join the same transmission group one by one. It is very difficult to maintain the global group due to the dynamic nature of ad-hoc networks. In addition, we cannot benefit from spatial reuse of the spectrum. Therefore, these protocols that were originally proposed for wireless LANs are not directly applicable to multi-hop ad-hoc networks.

Elimination-yield non-preemptive priority multiple access (EY-NPMA) is the channel access method used in HIPERLAN (high performance radio local area network) systems [53–56] deployed in Europe, which are high speed wireless LANs that support mixed types of traffic.

EY-NPMA supports five different access priorities. According to the EY-NPMA, channel access cycle includes four phases, namely priority resolution, elimination, yield and transmission. In the priority resolution phase, stations that have the highest access priority transmit first and enter the next phase. In the elimination phase, these stations transmit a geometrically distributed random number of slots. Those stations that transmit last win and enter the next phase. In the yield phase, the survived stations back off for a random amount of time. Those stations that recover the earliest from backoff procedure begin transmission. If these transmitting stations do not receive acknowledgment packets in due time, they will attempt to retransmit in the next channel access cycle.

Although EY-NPMA supports different access priorities, it adapts very badly in ad-hoc networks. First it cannot cope well with hidden terminals and second it relies on synchronization of all stations to start a cycle, which is known to be very difficult in distributed systems in general. Performance evaluation on HIPERLAN can be seen in [40, 57–59]. Actually, Apostolas et al. [59] have shown that dynamic

TDMA is better than EY-NPMA if global synchronization is available.

2.6 Summary

We have given a brief review on existing MAC protocols that may be used in ad-hoc networks. Protocols that combine RTS/CTS exchange with carrier sense are shown to achieve better throughput than those only using either RTS/CTS or carrier sense in ad-hoc networks where hidden terminal problem is common. Some protocols address fairness in their design in addition to the predominant metric – throughput. They can alleviate the fairness problem but still leave some issues unaddressed. There have also been some protocols designed to support real time traffic, or more generally, prioritized traffic. However, they require hidden terminal free environment to operate correctly, therefore they are not readily applicable to ad-hoc networks.

Chapter 3

Priority Based Multiple Access (PriMA)

In this chapter, we propose a new protocol called priority based multiple access (PriMA) to support stations' differentiated access priorities to the channel. It implements MAC-level acknowledgment the same as in IEEE 802.11 DFWMAC DCF protocol [33] while adopting the collision-free data transmission characteristics of FAMA-NCS [29], a variant of FAMA protocols for ad-hoc networks. More importantly, it implements a novel distributed scheduling algorithm which gives stations a dynamic-priority based access to the channel by taking into account the packet-level QoS requirements.

The remainder of the chapter is organized as follows. Section 3.1 describes PriMA protocol in detail. Section 3.2 compares by simulation the performance of FAMA-NCS, IEEE 802.11 DFWMAC DCF and PriMA protocols. Section 3.3 summarizes this chapter.

3.1 PriMA Protocol

3.1.1 Overview

PriMA protocol requires a station that wishes to send a data packet to acquire the channel before transmitting the packet. The channel is acquired by establishing an RTS-CTS dialog between the sender and the receiver. Although multiple control packets may collide with each other, data packets are always sent free from collisions. This is achieved by the enforced requirements of size relationship between RTS and CTS as well as different periods a station should wait after receiving a packet or sensing the channel busy. Fullmer et al. have given detailed description about this in [28]. They have notably shown that, as long as the length of the CTS packet is sufficiently longer than that of the RTS packet, CTS can act as a jamming signal to prevent other stations from transmission. Additionally, stations that hear the channel busy should wait long enough for the possible data transmission to go on unobstructed. PriMA's trivial modification to FAMA-NCS is to add the reply of an acknowledgment packet after successful reception of a data packet. This is also the common case in some other MAC protocols such as DFWMAC DCF protocol. The extension of MAC-level acknowledgment still ensures data packets free from collisions.

The most important feature of PriMA is that the access to the channel is based on the priorities of the packets queued in stations. This is achieved by the three times calculated according to the QoS requirements of the data packets queued in each station. The first one is the *access time* that a station should wait after the channel becomes idle before transmitting an RTS packet. The access time is similar to the distributed interframe space (DIFS) in DFWMAC and the silent period of channel in FAMA-NCS. However, unlike the fixed length of DIFS and the random length of the silent period in FAMA-NCS, the access time is dynamically adjusted based on the QoS requirements of the packets. The second one is the *delay time* carried in each RTS packet. It indicates how long the intended receiver can wait before replying with a CTS packet. Therefore, an earlier sent RTS packet may be preempted by a later sent RTS packet which indicates a higher priority data packet transmission request. The third one is the *backoff time* that a station should wait before retransmission when collision occurs. The backoff time is uniformly distributed, however the upper bounds of the distribution vary among stations. Stations that hold higher priority packets have lower upper bounds. Therefore, they can statistically recover earlier than other stations and bid for the channel again. For stations queued with normal data packets, we just set a large upper bound comparable to that of DFWMAC. Therefore, our backoff scheme is not necessarily more prone to collapse than binary exponential backoff with a fixed upper bound.

3.1.2 Description of PriMA

To simplify our description of PriMA, the processing time and transmit-to-receive turnaround time are ignored. We say a station "detects" collisions if it senses the channel busy without being able to receive any intelligible packet. Following, we define some of the notations used in this section:

$rtPacket$: Data packet with QoS parameters specified.

$nrtPacket$: Data packet without any QoS parameters specified.

T_d : Maximum one-hop channel propagation delay.

T_{type} : Time to transmit a packet of type *type* where *type* can be either RTS, CTS, DATA or ACK.

T_{delay} : Allowed delay time before a reply to an RTS packet is sent. This is a value carried in the RTS packet's header.

T_{left} : Time left for an $rtPacket$ to be delivered; when T_{left} is less than a threshold, the corresponding $rtPacket$ will be discarded.

T_{max} : Maximum time to complete one successful RTS-CTS-DATA-ACK transmission.

T_{access} : Time during which the channel should be sensed idle before transmitting an RTS packet.

T_{defer} : The longest time that a station should defer access to the channel when "detecting" collisions. It equals $T_{data} + 3 * T_d$.

T_{unit} : Time used as a scaling factor to map packet delay requirement to backoff time.

N_{lost} : Number of packets discarded during a session.

N_{sent} : Number of packets sent during a session.

PLR : Packet loss ratio (QoS requirement), obtained from upper layers (e.g., session management layer).

Figure 3.1 shows how PriMA operates. In this figure, stations A and C are hidden from each other and station B is a neighbor of both A and C. As in all other collision avoidance multiple access techniques, PriMA uses the RTS-CTS combination to implement collision avoidance. Additionally, to ensure the problem of hidden terminals is addressed adequately, PriMA imposes FAMA's conditions on the sizes of the RTS and CTS packet [28]:

- $T_{rts} > 2 * T_d$
- $T_{cts} > T_{rts} + 2 * T_d$

and stations that sense the channel busy should wait for T_{defer} to let the possible ongoing data transmission to finish unobstructed. FAMA's rationale in imposing these conditions works as follows. If C did not hear A's RTS and sends an RTS packet that at most collides with the CTS reply from B to A, the condition ensures that C hears the trailer of the CTS and thus C would abort any transmission and enter the backoff procedure. This ensures that the data packet from A to B is transmitted collision free. Note that it is not necessary to adopt these restrictions (enforced in FAMA) in PriMA to provide soft QoS. Nevertheless, we believe that it does not make sense to address the problem of QoS (be it soft QoS) without addressing first the hidden terminal problem, especially when the offered traffic load can be very high. PriMA's approach to providing QoS can also be applied to IEEE 802.11's DFWMAC protocol.

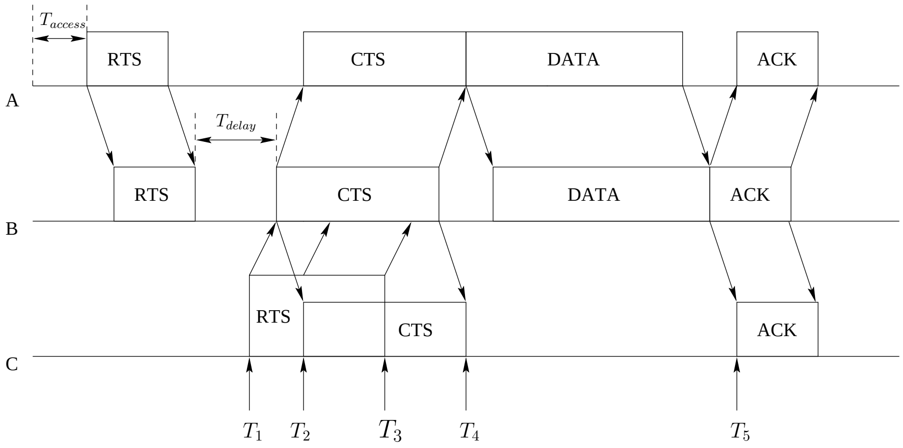


Figure 3.1: PriMA illustration

In PriMA, any station that has a packet to send should wait until it senses the channel idle for a certain amount of time, called *access time*, before transmitting an RTS request. The access time calculation, as shown in Algorithm 3.1, is based on the required QoS and the perceived QoS.

Algorithm 3.1 Access time calculation

```
if (nrtPacket or rtPacket with $T_{left} > 1$ sec) then
    $T_{access} = 2 * T_d$
else
    if ($T_{left} > 0$) then
        $T_{access} = T_{left} - (N_{lost} - PLR * N_{sent}) * T_{max}$
        if ($T_{access} < 0$) then
            $T_{access} = 0$
        else
            $T_{access} = 2 * T_d / |\log_2 (T_{access}/2)|$
        end if
    end if
end if
```

Note that in order to reduce the waste of bandwidth due to this access time, time-sensitive packets which have a time to live of more than one second, compete at the same priority level as the time insensitive packets. These time-sensitive packets would gain higher priorities only when they become more urgent. Note that

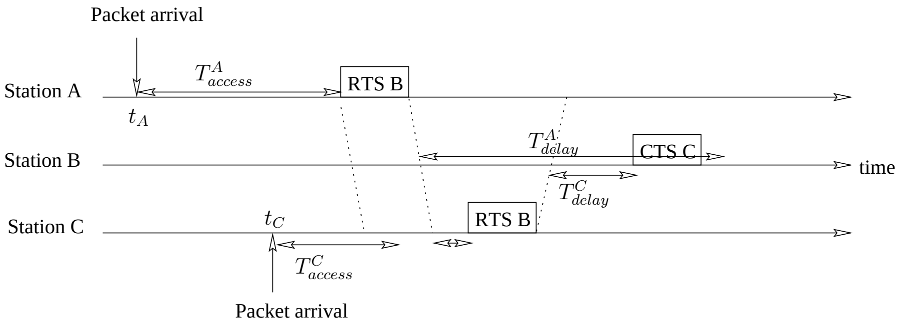


Figure 3.2: Illustration of preemption in PriMA

initially, at the packet's first attempt, T_{left} is initialized to a value proportional to the maximum tolerable delay of the packet. Also, throughout the algorithm, T_{left} decreases with the ticks of the clock in the same proportion. The calculation in line 5 of the algorithm shows that the more a station has suffered excess dropped packets ($N_{lost} - PLR * N_{sent}$), the shorter T_{access} will be. This excerpt of pseudo-code clearly shows that when two stations attempt to access the channel simultaneously, T_{access} will differentiate their requests according to their packets' QoS requirement.

When a station succeeds in its access to the channel, it sets a "delay" field in its RTS packet and sends it out. The value of the field is denoted by T_{delay}, which indicates how long the intended receiver can wait before replying with a CTS packet. T_{delay} is calculated as shown in Algorithm 3.2.

Algorithm 3.2 Delay time calculation

```
if (nrtPacket) then
    $T_{delay} = 2 * T_{rts}$
else
    if ($T_{left}/T_{max} > 2$) then
        $T_{delay} = 2 * T_{rts}$
    else
        $T_{delay} = (T_{left}/T_{max}) * T_{rts}$
    end if
end if
```

To illustrate the importance of the delay time, let us consider the example in Figure 3.2 where two stations, A and C, attempt to communicate with station B. We will neglect the propagation delay in this example. C's data packet arrives at the MAC layer at time t_C later than A's data packet who arrives at t_A, however, C's delay target is more stringent than A's. Due to the lack of a global coordinator in the system and the lack of global state information, there is no way for C to succeed in sending an RTS packet to B before A does, since $t_C + T^C_{accces} > t_A + T^A_{access}$. A sets a value T^A_{delay} in its RTS packet which tells B the amount of time it can wait before replying with a CTS. After C hears the end of A's RTS (we will see later the case where A and C are hidden from each other), it will apply the same procedure it uses for the first access, however with only the remaining access time. In other words, C will wait until it senses an idle channel for $T_{left} = t_C + T^C_{accces} - t_A - T^A_{access}$ starting at $t_A + T^A_{access} + T_{RTS}$. If this time left is smaller than T^A_{delay}, i.e., if C's access time expires before B replies with a CTS to A, then C can send to B an RTS that will preempt A's RTS. As a result, although A succeeds in sending the RTS before C, C still has the possibility to preempt A, before A starts transmitting its data packet.

In the calculation of T_{delay} for a real time packet, only T_{left} is used. This is because T_{left} is a composite value affected by both packet delay and packet drop ratio. Thus we can keep the algorithm simple yet effective. Algorithm 3.2 shows that the RTS packet for a data packet that is more delay sensitive will have shorter T_{delay} when the data packet is about to expire. The calculation will permit the packets to gain higher priorities after they are delayed for some time. Stations usually do not reply with CTS immediately after receiving an RTS packet unless required to do so, therefore a station may receive multiple RTS packets in a row. If the later arrived RTS packet requires shorter delay than that of the earlier one, the station can reply to it first. This makes it possible for delay sensitive data packets to preempt other data packets. In addition, T_{delay} does not exceed two times the RTS packet transmission time to minimize the protocol overhead.

It is inevitable that control packets may collide with each other, therefore those stations that detect collisions should also wait for T_{defer} and then back off for a random amount of time $T_{backoff}$, which is calculated as shown in Algorithm 3.3, where $U(0, x)$ is a uniformly distributed random number in the interval $(0, x)$.

Algorithm 3.3 Backoff time calculation

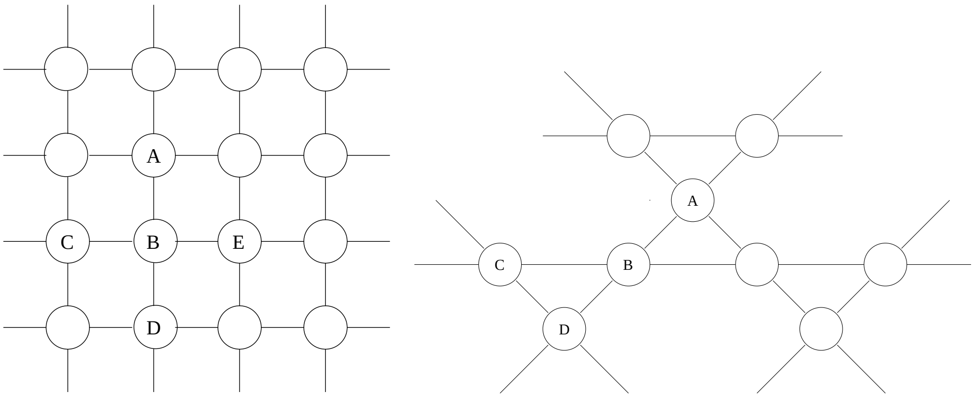
```
if (nrtPacket or rtPacket with $T_{left}/T_{unit} > Maxtimer$) then
    $T_{backoff} = U(0, Maxtimer) * 2T_d$
else
    $T_{backoff} = U(0, T_{left}/T_{unit}) * 2T_d$
end if
```

The value of *Maxtimer* is simply set to a large value, say 800, which is comparable to the maximum timer used in DFWMAC protocol. For example, the IEEE 802.11 wireless LAN standard specification for direct sequence spread spectrum (DSSS) uses a timer (or *contention window*, in 802.11's terminology) ranging from 31 to 1023. In PriMA, $T_{backoff}$ is based on a uniform distribution whose upper bound value varies according to the delay requirement of a data packet. This calculation will statistically give stations that have delay sensitive data packets shorter $T_{backoff}$. Thus, on average, these stations will end the backoff period earlier than other stations and bid for the channel again. To illustrate how Algorithm 3.3 contributes in differentiating the stations, let us return to the example of Figure 3.2. In the case where A and C are hidden from each other. C cannot hear the RTS packet of A and thus after exhausting its timer T^{C}_{access}, it would also send an RTS packet which would collide at B with A's RTS packet. After T^{A}_{delay} (respectively T^{C}_{delay}) station A (respectively station C) notices the collision by timeout, and thus they both apply the above backoff algorithm with their respective QoS targets. In this case, C has higher probability in bidding for the channel before A does, since C draws its uniformly distributed backoff time from a smaller interval than A does, and thus C has higher probability to terminate its backoff before A.

From the above descriptions, we can see that by dynamically adjusting the three times, T_{access}, T_{delay} and $T_{backoff}$, according to the data packets' QoS requirements, PriMA can give stations priority-based access to the channel. At the same time, various waiting time periods are carefully defined to prevent control packets from colliding with data packets. The complete specification for PriMA protocol is shown in Appendix A.

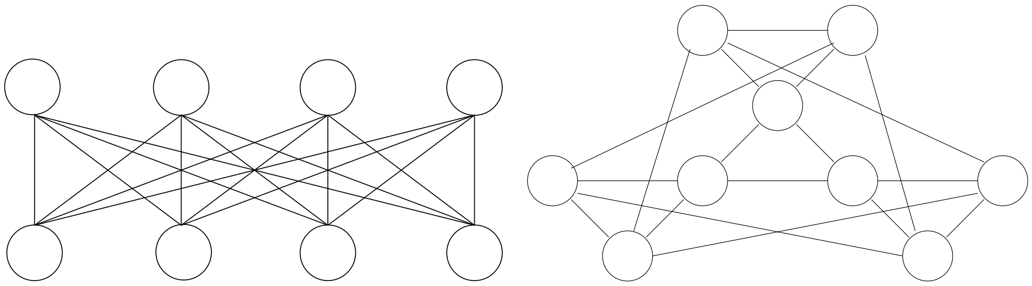
3.2 Simulation Results and Discussions

PriMA is a complex MAC protocol for analytical modeling because of the dynamic nature of the different times, thus in this chapter the performance of PriMA is investigated by simulations. In our experiments, we investigate symmetrical networks where each station has N neighbors and is hidden from Q neighbors of any one of its neighbors, thus each station has the same spatial characteristics and we can



(a) N=4, Q=3 (b) N=4, Q=2

Figure 3.3: Two sample configurations



(a) N=4, Q=3 (b) N=4, Q=2

Figure 3.4: Two sample configurations (collapsed)

ensure that a station's location has no effect on service differentiation. Figure 3.3 shows two sample configurations for $N = 4$, $Q = 2$ and $Q = 3$. For example, for station A in Figure 3.3(a), it has 4 neighbors including station B and is hidden from station B's 3 neighbors: C, D and E. As this graph can grow to infinitely large, we fold (or collapse) it so that the total number of stations in the network is limited while symmetry is maintained. The resultant graphs are shown in Figure 3.4.

We assume a 1Mbps ideal channel with zero preamble and processing overhead. We have performed different sets of simulations with OPNET Modeler/Radio [60].

Protocol	RTS	CTS	DATA	ACK	backoff timer	backoff unit time
FAMA	20-byte	25-byte	500-byte	-	10	160μsec
IEEE 802.11	25-byte	20-byte	500-byte	20-byte	31-1023	6μsec
PriMA	20-byte	25-byte	500-byte	20-byte	800	12μsec

Table 3.1: Protocol configuration parameters

DIFS	SIFS	EIFS
12μsec	0μsec	1.3msec

Table 3.2: Extra configuration parameters for IEEE 802.11

The results are compared with those provided by alternative protocols, namely FAMA-NCS and IEEE 802.11 DFWMAC[1]. Table 3.1 lists the parameters used in the simulation. As we ignore the extra time incurred by hardware and software, the different interframe spaces (IFSs) in IEEE 802.11 are reduced accordingly and they are shown in Table 3.2.

In the first set of simulations, all stations generate Poisson traffic with the same mean rate and all require best-effort delivery service. The simulation measures the throughput of the protocol against the degree of the nodes. The results are shown in Figure 3.5. For comparative purpose, we also show the analytical results for Slotted-ALOHA with separate acknowledgement channel [61] and non-persistent CSMA [22] in these figures. The figures demonstrate that FAMA's performance degrades dramatically when the number of competing stations (including neighbors and hidden terminals) increases. This is due to the ineffectiveness of uniform backoff scheme with small timer in collision resolution used in FAMA. However, it

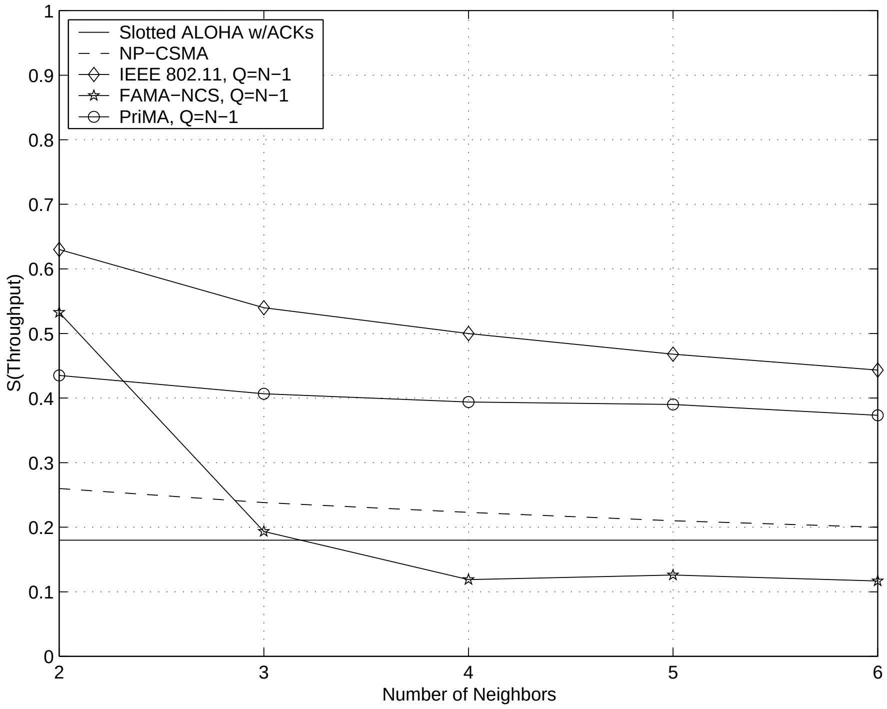
---

[1] We use its specification for DSSS if applicable.

still performs well in other situations because of its immunity to hidden terminal problem. IEEE 802.11 performs quite well in all the situations because of its binary exponential backoff scheme. Although PriMA's performance is not quite outstanding when the number of competing stations is small, however, PriMA does achieve a rather stable throughput and is comparable to IEEE 802.11 when the number of competing stations increases. PriMA, by introducing the different times as well as the MAC acknowledgments in fact sacrifices throughput to achieve different priorities among the stations (which we show later). In addition, large backoff timers will become more effective when the number of competing stations increases, therefore the throughput difference between PriMA and IEEE 802.11 is decreasing. In other words, PriMA captures in fact the effectiveness of FAMA in solving the hidden terminal problem as well as 802.11's ability in sustaining throughput at high loads. It sacrifices however a little throughput by introducing the different times in order to achieve differentiation between the stations as shown in the following.

In the second set of simulations, each station still generates Poisson traffic with the same mean rate. However, we set one station (router) to generate packets that require delay-bounded delivery, while the other stations (hosts) have no special delay requirement. This, in essence, makes the packets from the router have higher priority than packets from the hosts.

This scenario may be applicable in a case when some stations form a group and choose one of them to act as a router for this group. In other words, the router will handle a larger amount of traffic and thus needs some priority to access the



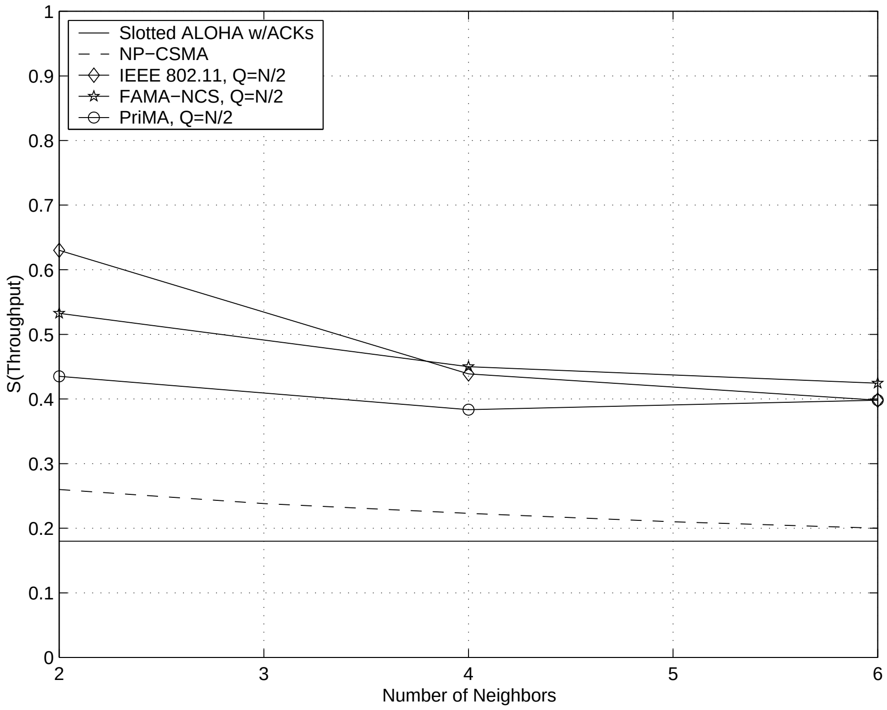
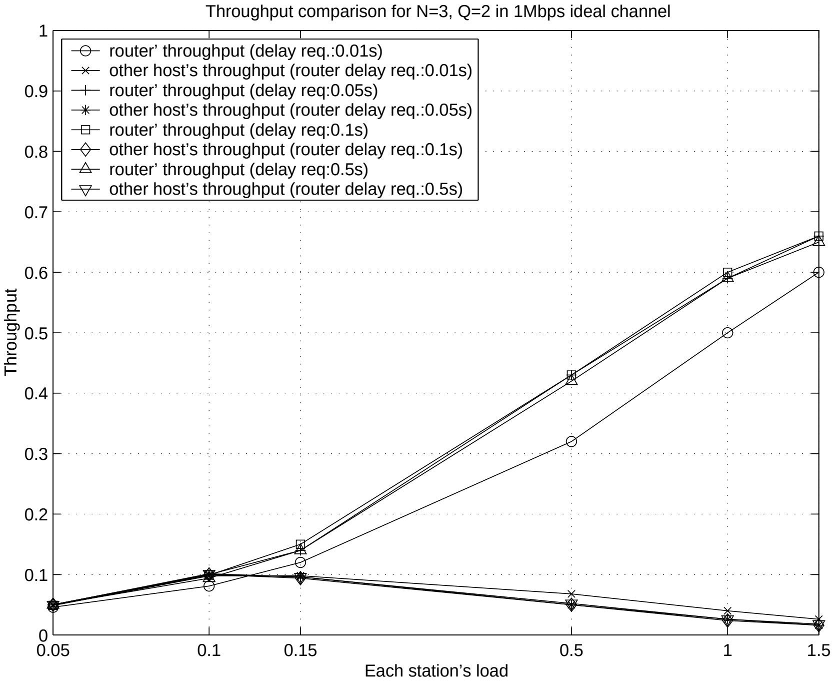


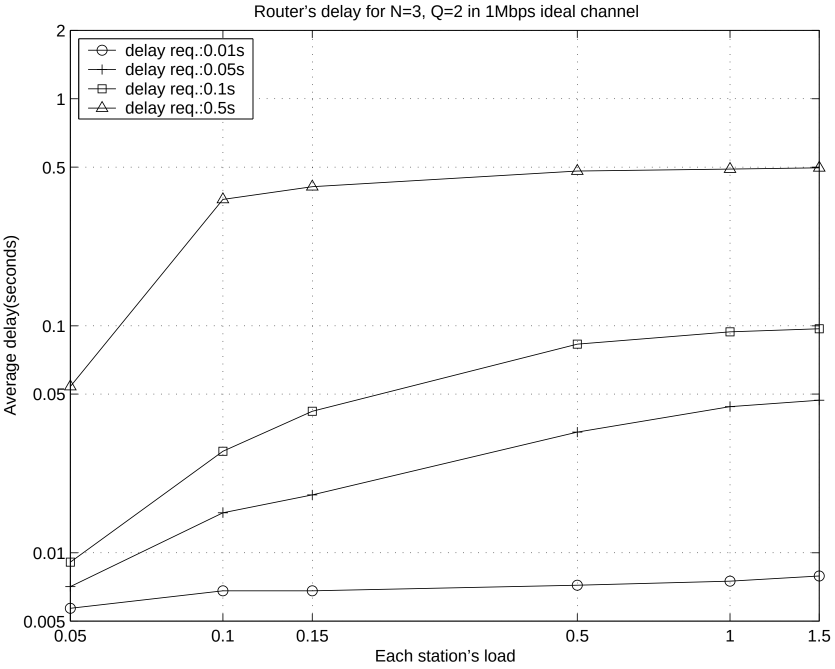
Figure 3.5: Throughput versus node degree

channel. Simulation results are shown in Figure 3.6 and 3.7 with different values of N and Q. The figure clearly shows that the router has higher throughput than other stations despite the fact that they all offer the same traffic load to the channel. Even when the number of competing stations increases, the router can still achieve much higher throughput than other normal stations. This differentiation cannot be achieved by protocols such as FAMA-NCS and IEEE 802.11 DFWMAC, which do not support prioritized channel access.

In the third set of simulations, we select two stations (one high priority router and one low priority router) to generate constant rate packets that require different packet delivery delay bounds. They are two neighboring competing routers, which may correspond to a core router and an edge router in a routing infrastruture. Usually the core router will handle the largest amount of traffic and may require higher priority than the edge router, while the edge router requires higher priority than other normal hosts. We investigate two performance metrics under different scenarios: namely, the packet loss ratio (due to expiration), and the average delay. To get comparative results, we set the packet interarrival time equal to the packet delay requirement, otherwise protocols such as IEEE 802.11 DFWMAC will never keep up with the packet generation rate and eventually will drop nearly all the packets. Simulation results are shown in Figure 3.8. The first three sub-figures (3.8(a), 3.8(b), 3.8(c)) show the packet loss ratio against the offered load. They show that when the offered load increases, the packet drop ratio for IEEE 802.11 increases significantly while PriMA can still maintain low packet drop ratio for the two routers. Figure 3.8(d) shows that the average packet delay achieved by the

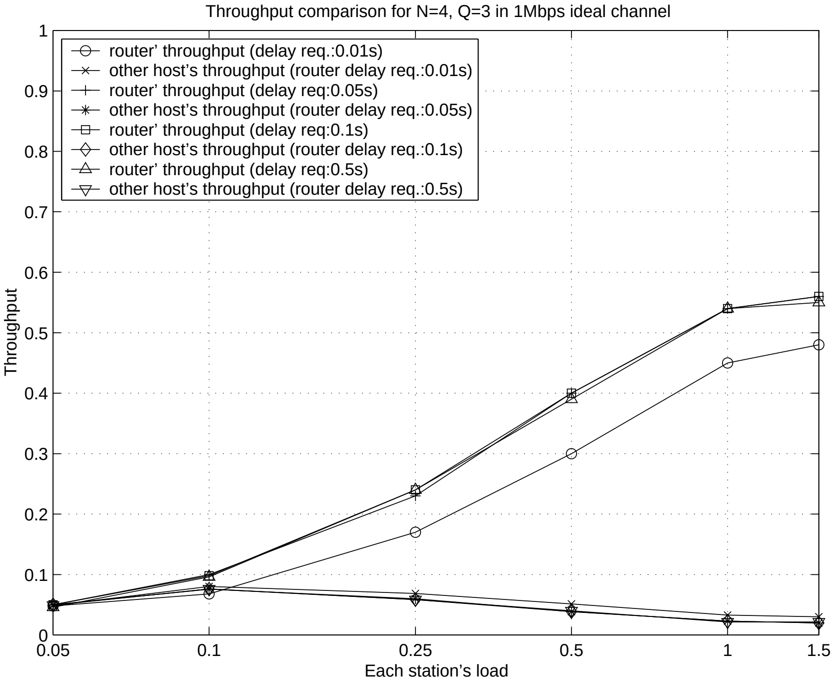


(a)

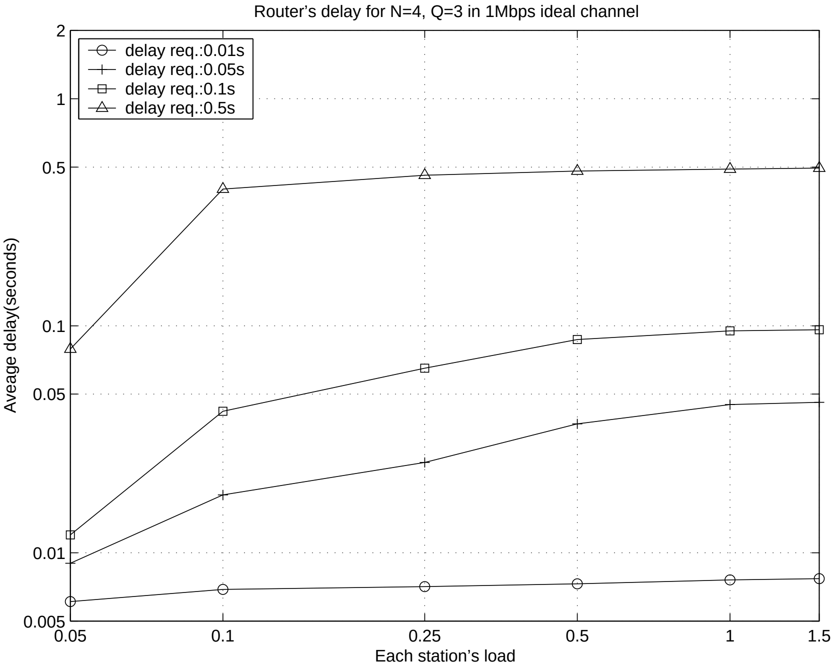


(b)

Figure 3.6: One router configuration (N=3)



(a)



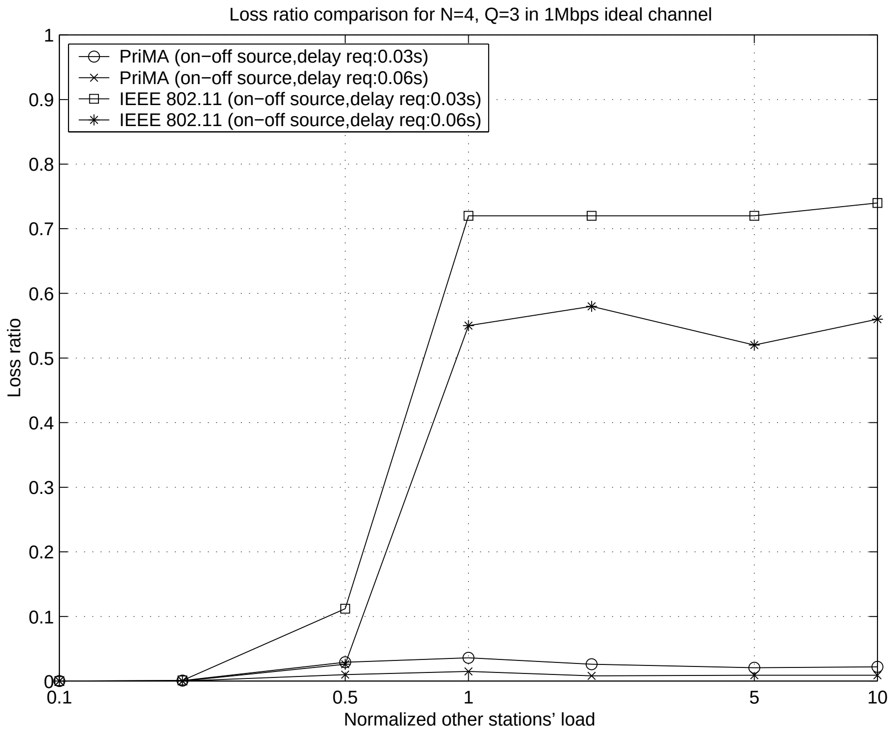
(b)

Figure 3.7: One router configuration (N=4)

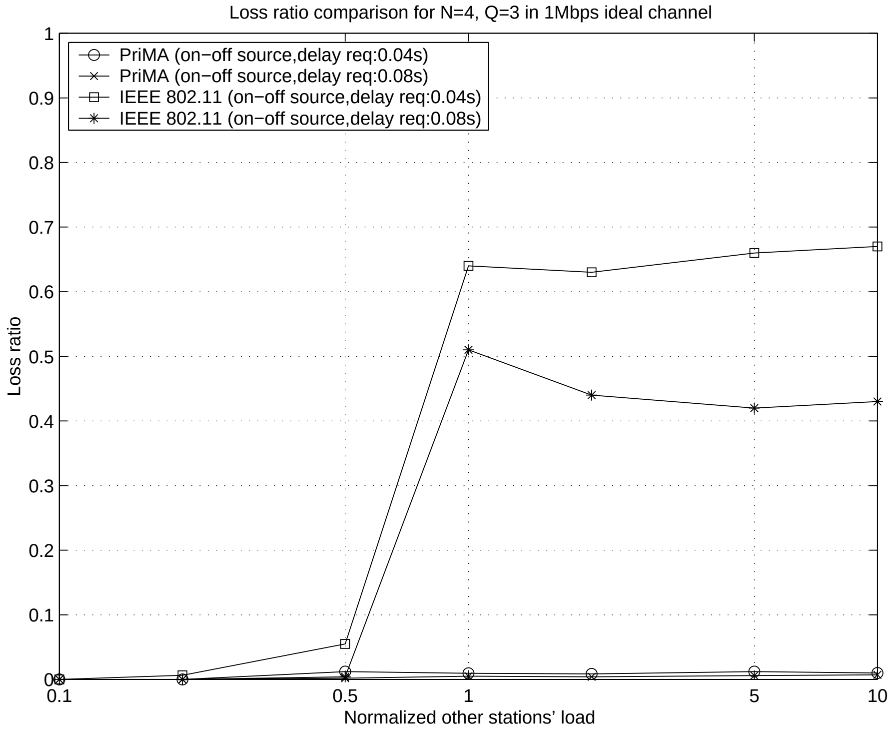
two routers in PriMA. The figure clearly shows that when the traffic load is very low, the packets from all the stations experience the same (insignificant) delay. However, when the traffic load increases, the protocol starts differentiating the stations according to their requirements.

3.3 Summary

PriMA is a new MAC protocol that is specifically designed for ad-hoc networks. It can achieve good throughput in ad-hoc networks where hidden terminal problem is common. Additionally, each station has priority-based access to the channel, thus PriMA can provide elementary QoS support from the bottom up, making it a good choice for supporting higher layer protocols that require QoS. Another benefit of PriMA is that it can provide better support for ad-hoc routing as some packets can be given higher priorities and get delivered earlier than others. Simulation results show that PriMA is an especially promising MAC protocol for ad-hoc networks in presence of heterogeneous traffic and QoS requirements.

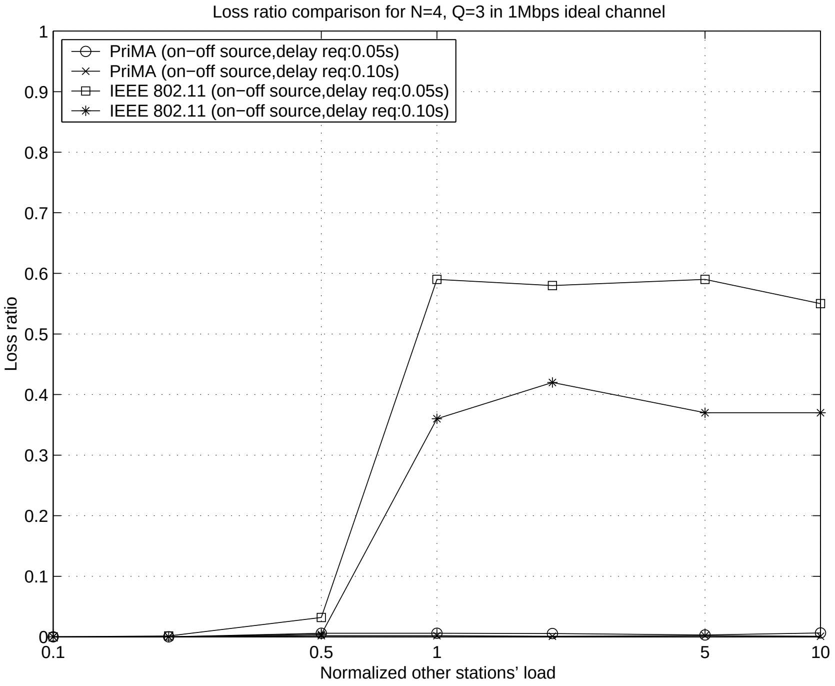


(a)

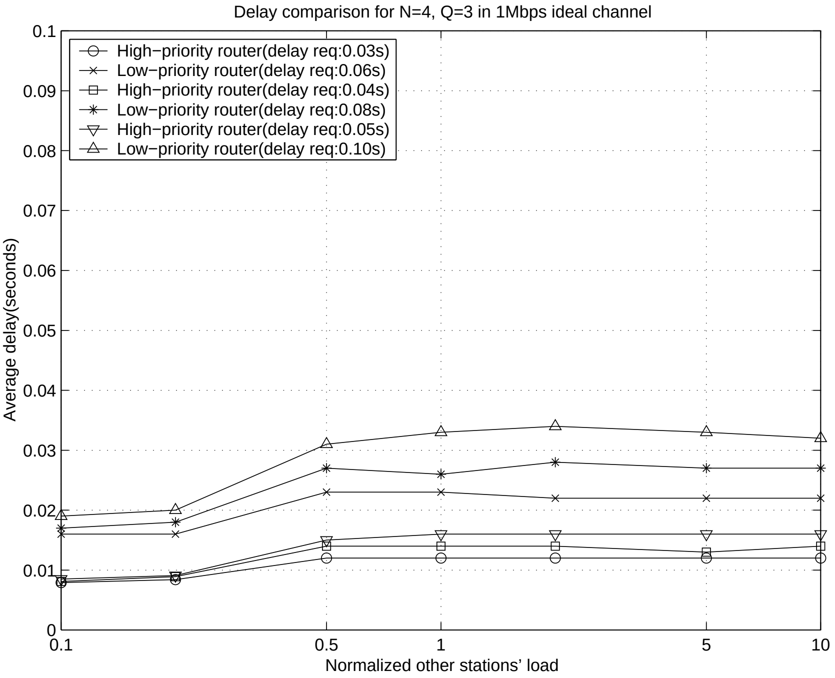


(b)

Figure 3.8: QoS differentiation



(c)



(d)

Figure 3.8: QoS differentiation (contd.)

Chapter 4

Fair Share Based Multiple Access (FSMA)

As we have discussed in section 2.4, there has been growing interest in fairness in MAC protocols for distributed wireless networks. Some ad-hoc solutions have been proposed in [26, 47, 48] to alleviate the fairness problem. However, these schemes do not support variable length packets and thus their applicability is rather limited. The goal of this chapter is to address the fairness problem in a general and more practical approach. Section 4.1 first defines a new metric for measuring fairness and then proposes a different backoff scheme for the IEEE 802.11 DFWMAC DCF protocol. Section 4.2 evaluates the performance of the proposed scheme and compares its performance to those obtained from the original backoff scheme using several ad-hoc network configurations. Section 4.3 summarizes this chapter.

4.1 FSMA Protocol

4.1.1 Fairness Metric

We define fairness in the sense of fair queueing as defined in [62]. To facilitate the discussion, we introduce the following notation:

ϕ_i : A pre-defined fair share that station i should receive. Normally, it should be determined at admission control, i.e. when the node joins the ad-hoc network, and can be readjusted for example when a node becomes a router. How to choose this parameter, how to do admission control and how to adjust the parameter, are still open research problems.

W_i : The actual throughput achieved by station i.

L_i : Station i's offered load.

A fair MAC protocol should have the following properties. When the stations' offered load to the channel is much lower than the channel capacity, each station's request for transmission should be met. This means that for any station i, $W_i = L_i$. When stations' offered load exceeds the channel capacity, each station should be able to get its fair share of the channel, i.e., throughput proportional to ϕ. This means that for any station i and j, $\frac{W_i}{\phi_i} = \frac{W_j}{\phi_j}$. This is the ideal situation. In practice, we want to minimize the value of $|\frac{W_i}{\phi_i} - \frac{W_j}{\phi_j}|$. Instead of working with absolute values, we define the fairness index, FI, to be:

$$FI = max\{\forall i, j : max(\frac{W_i}{\phi_i}, \frac{W_j}{\phi_j})/min(\frac{W_i}{\phi_i}, \frac{W_j}{\phi_j})\}$$

Therefore, our goal becomes the design of a distributed MAC protocol that can minimize *FI* and thus achieve fairness for all the stations in an ad-hoc network.

Here it should be noted that our definition of ϕ_i is somewhat different from that in fair queueing. In fair queueing, there is a central scheduler for all flows and all ϕ_is sum to 1, i.e., there is admission control. In our algorithm, since there is no central scheduler, sum of the ϕ_is of a node and its neighbors may be greater than 1. Thus in this case, since we are trying to quantify *relative* discrepancies in service, the ratio of the two thus *FI* is more appropriate than the difference. This point will be clearer when we describe our algorithm later.

There comes the problem of how to choose ϕ_i for any station i in a distributed network. As mentioned previously, this is an admission control problem which is beyond the scope of this chapter. However, in situations where the ad-hoc network is open to everyone without admission control, which can happen in situations where all the stations are trusted and known not to misbehave, for example in a collaborative computing environment, the following approach can be used. If each station is considered to be a greedy source and wants to get the same share as all other stations as a whole entity, then it can just set $\phi = 0.5$ regardless of the number of its neighbors. As to any station, say i, it requests the same share as all the others in its vicinity. These stations have a total share of $\phi_o = 1 - \phi_i = 0.5$, which equals to this station's share ϕ_i. This can be interpreted as a per-station fairness. If a station has two active links (or streams in MACAW's terminology [26]), which can happen when a station acts as a router in an ad-hoc network, it can instead set ϕ_i

to meet :

$$\frac{\phi_i}{\phi_o} = \frac{\phi_i}{(1-\phi_i)} = \frac{2}{1} \Rightarrow \phi_i \approx 0.67,$$

which simply shows that the station (router) wants to obtain two times as much share of bandwidth as other stations' to function as a router properly. This can be interpreted as MACAW's per-stream fairness. For a router in an ad-hoc network, to gain per-stream fairness is especially beneficial for the proper functioning of the whole network, as the router can still handle more traffic than other normal hosts despite the fact that the router does not have separate links to route packets from one side to the other.

4.1.2 Description of FSMA

To achieve the fairness goal, we propose a different backoff scheme for the original DFWMAC protocol. In this scheme, each station estimates its share and other stations' share of the channel and then adjusts the contention window accordingly. The following notations are used in the fair share estimation algorithm:

W_{ei} : The estimated share of the estimating station itself.

W_{eo} : The estimated share of other stations.

T_{type} : Time to transmit a packet of type *type*.

T_{edata} : When receiving an ACK packet, the estimated transmission time of the corresponding data packet.

Algorithms 4.1 and 4.2 show how the fair share estimation works. The basic idea is that from the point of view of station i, it sees that it is sharing the channel with a group of "belligerent" stations who are competing with it for channel access. Thus each station has just the notion of "myself" and "the others." A station estimates dynamically what throughput it gets and what throughput "others" gets, and then adjusts its contention window according to the fairness index thus defined. In other words the contention window is adjusted in order to equalize the throughput obtained by the different stations. A station can estimate roughly how much bandwidth "others" obtain by looking at the packets in its vicinity and updating its estimation of either its own share or others' share based on the roles of the received packets. For example, if a station receives a CTS packet destined to itself, the station sends a data packet and updates estimation of its own share as the data packet transmission request was originated by itself. Additionally, RTS and CTS packets' transmission time is also counted towards its estimated share because they are used as channel reservation packets which consume channel resource as well.

In fact, it is relatively easy for a station to adjust its estimation whenever it receives an RTS, CTS or DATA packet as these packets carry sufficient information for the station to judge. However, the main difficulty lies in what a station should do when it receives an ACK packet, as the ACK packet does not carry any information about the length of the corresponding DATA packet and thus this station does not know if the RTS/CTS access method is used for the DATA packet when its length is unknown. To solve this problem, we use a simple approach based on the

Algorithm 4.1 Fair share estimation for received packets

```
switch (received packet type) {
case RTS:
    if (destID != localID) {
        $W_{eo}$ += $T_{rts}$; update $T_{edata}$
    } else {
        send CTS packet; $W_{eo}$ += $(T_{rts} + T_{cts})$
    }
case CTS:
    if (destID != localID) {
        $W_{eo}$ += $(T_{rts} + T_{cts})$; update $T_{edata}$
    } else {
        send DATA packet; $W_{ei}$ += $(T_{rts} + T_{cts} + T_{data})$
    }
case DATA:
    if (destID != localID) {
        if ($T_{data} > RTS\_THRESHOLD$) {
            $W_{eo}$ += $(T_{rts} + T_{cts} + T_{data})$
        } else {
            $W_{eo}$ += $T_{data}$
        }
        $T_{edata} = T_{data}$
    } else {
        send ACK packet;
        if ($T_{data} > RTS\_THRESHOLD$) {
            $W_{eo}$ += $(T_{rts} + T_{cts} + T_{data} + T_{ack})$
        } else {
            $W_{eo}$ += $(T_{data} + T_{ack})$
        }
    }
case ACK:
    if (destID != localID) {
        if ($T_{edata} > RTS\_THRESHOLD$) {
            $W_{eo}$ += $(T_{rts} + T_{cts} + T_{edata} + T_{ack})$
        } else {
            $W_{eo}$ += $(T_{edata} + T_{ack})$
        }
    } else {
        if ($T_{data} > RTS\_THRESHOLD$) {
            $W_{ei}$ += $(T_{rts} + T_{cts} + T_{data} + T_{ack})$}
        } else {
            $W_{ei}$ += $(T_{data} + T_{ack})$
        }
    }
}
```

Algorithm 4.2 Fair share estimation when sending packets

```
Whenever sending an RTS packet, $W_{ei}$ += $T_{rts}$.
Whenever sending a DATA packet without using RTS/CTS, $W_{ei}$ += $T_{data}$.
```

reasoning of packets' time affinity, which simply means that the newly received ACK packet destined to "others" may be closely related with the previous received RTS/CTS/DATA packet destined to "others." In DFWMAC, an RTS/CTS packet includes a field in its packet header named Network Allocation Vector (NAV) which notifies other neighboring stations how long they should defer access to the channel such that the following DATA and ACK packet transmissions can be completed without interference from these stations. If a station receives an RTS/CTS packet destined to other stations, it can derive the length of the corresponding DATA packet and set the corresponding transmission time to T_{edata}. If a station receives a DATA packet destined to other stations, it already knows the length of the DATA packet and updates T_{edata}. When the station receives an ACK packet destined to other stations, it can then adjust its estimation based on the value of T_{edata}.

With this estimation, we modify the original binary exponential backoff scheme used in IEEE 802.11 DFWMAC DCF protocol, which can lead to fairness problems as discussed previously in section 2.4. We define the estimated fairness index to be: $FI_e = (\frac{W_{ei}}{\phi_i})/(\frac{W_{eo}}{\phi_o})$ and the adjustment of contention window is shown in Algorithm 4.3.

Algorithm 4.3 shows that if a station estimates that it has got more share than it should get, it will double its contention window size until it reaches the maximum

Algorithm 4.3 Contention window adjustment

```
switch ($FI_e$) {
case $>C$:
    $CW_{new} = min(CW_{new} \times 2, CWMAX)$
case $(1/C, C)$:
    $CW_{new} = CW_{old}$
case $<1/C$:
    $CW_{new} = max(CW_{old}\ /\ 2, CWMIN)$
}
```

value ($CWMAX$) so that its neighbors can have more chances to recover earlier from backoff procedure and win access to the channel and vice versa. If a station estimates that it has got only its fair share, it will hold onto its current contention window size. In this algorithm, C is a constant used to adjust the adaptability of the algorithm. The smaller the value of C, the more aggressively is the contention window size adjusted and vice versa. However, the choice of C is rather limited. For example, if we choose $C = 2$, stations would not change their contention windows when estimated FI is between $(0.5, 2)$ and the probability of collision may be high when the number of competing stations is large and/or the load to the channel is high. Influence of C will be investigated later.

4.2 Simulation Results and Discussions

As our algorithm has been designed with the following in mind:

- Support for both fixed and variable packet lengths (MACAW and p_{ij}-persistent CSMA do not support variable packet length);

- Support both per-station fairness and per-stream fairness;
- The parameter C is of interest and its effects need investigation;
- Tradeoff throughput for fairness, thus need to compare throughput and fairness.

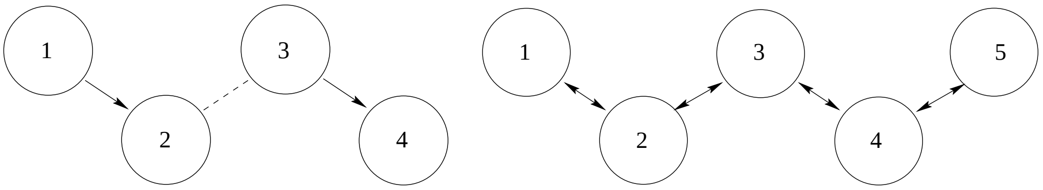
Therefore we need to investigate the algorithm completely to make sure that our algorithm performs up to our expectation. Extensive simulation results are presented in this section.

In our experiments, we investigate some configurations of wireless ad-hoc networks used in MACAW [26] and p_{ij}-persistent CSMA [48]. These are the 4-station, 5-station and 6-station scenarios. They are shown in Figure 4.1, where arrow lines indicate that there is traffic between stations and dashed lines indicate that the stations are within communication reach of each other but no traffic flows between them.

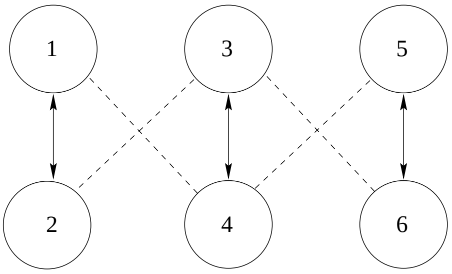
We assume a 1Mbps ideal channel with zero preamble and processing overhead and a propagation delay of about 6μ seconds. We have performed different sets of simulations with OPNET Modeler/Radio and we compare our results with the original IEEE 802.11 DFWMAC DCF protocol[1]. We investigate the performance of the system for both fixed packet length and variable packet length.

---

[1] We use the specification for direct sequence spread spectrum (DSSS) where applicable.



(a) 4-station (b) 5-station



(c) 6-station

Figure 4.1: Network configurations

4.2.1 Fixed Packet Length

Table 4.1 lists the parameters used to generate the simulations results. As we ignore the extra time incurred by hardware and software, the different interframe spaces (IFSs) in IEEE 802.11 are reduced accordingly and they are also shown in Table 4.1. Unless otherwise specified, all stations use fair share $\phi = 0.5$.

RTS	CTS	DATA	ACK
25-byte	20-byte	500-byte	20-byte

CWMIN	CWMAX	C	backoff unit time
31	1023	1.1	6μsec

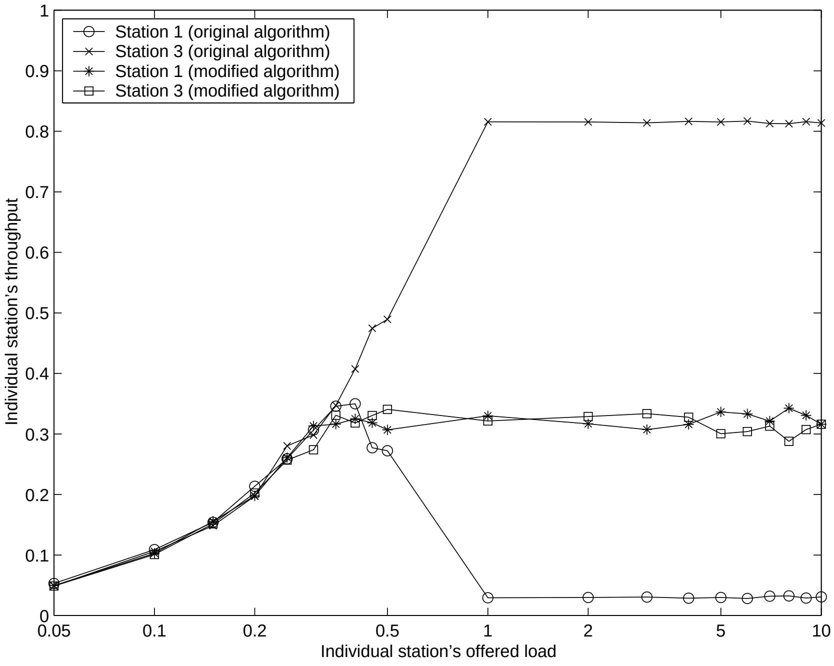
DIFS	SIFS	EIFS
12μsec	0μsec	1.3msec

Table 4.1: Protocol configuration parameters

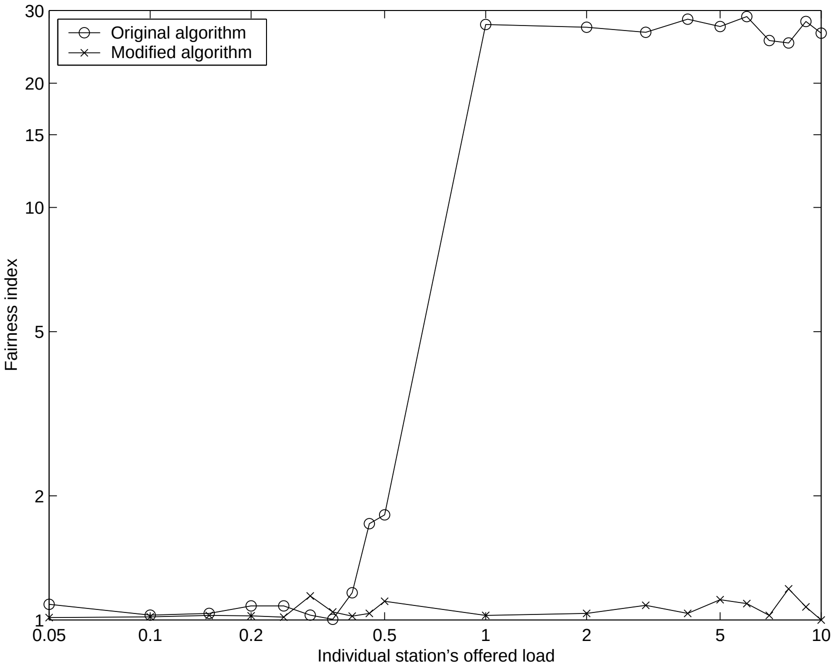
In the 4-station scenario, station 1 and 3 generate Poisson traffic with the same mean rate, and results are shown in Figure 4.2. Figure 4.2 shows that DFWMAC will have serious fairness problem when the offered load is high enough, which can be explained as follows. Most of the time station 1's transmission may coincide with station 3's transmission as they are hidden from each other. Station 2 will not be able to receive station 1's packet due to station 3's concurrent transmission. However, station 4 can still receive station 3's transmission successfully and reply to station 3 thereafter. According to the binary exponential backoff (BEB) scheme used in DFWMAC, a station's contention window size will be doubled after each unsuccessful transmission and will return to the minimum value only if a data packet is successfully transmitted. Therefore station 3 usually enjoys a much smaller contention window and thus a statistically shorter backoff timer than station 1. When the load is high, station 3 will *capture* the channel eventually. In our backoff scheme, if station 3 overhears a few packets transmitted from station 2 (in this case, either CTS or ACK packet), its estimation will show that it has obtained more bandwidth share than it should and will increase its contention window size[2] accordingly. With the ever increase of station 3's contention window, station 1 will get more chances to transmit packets to station 2. In the end, station 1's throughput can be balanced with station 3's throughput, so this scheme can achieve far better fairness than the BEB scheme. It should also be noted that in MACAW, even after it has augmented the handshake to RRTS-RTS-CTS-DS-DATA-ACK (RRTS stands for Request for RTS packet) with MILD backoff algorithm and backoff copy

---

[2] Here station 3 does not even need to know the fact that station 1 has packets for station 2.



(a)



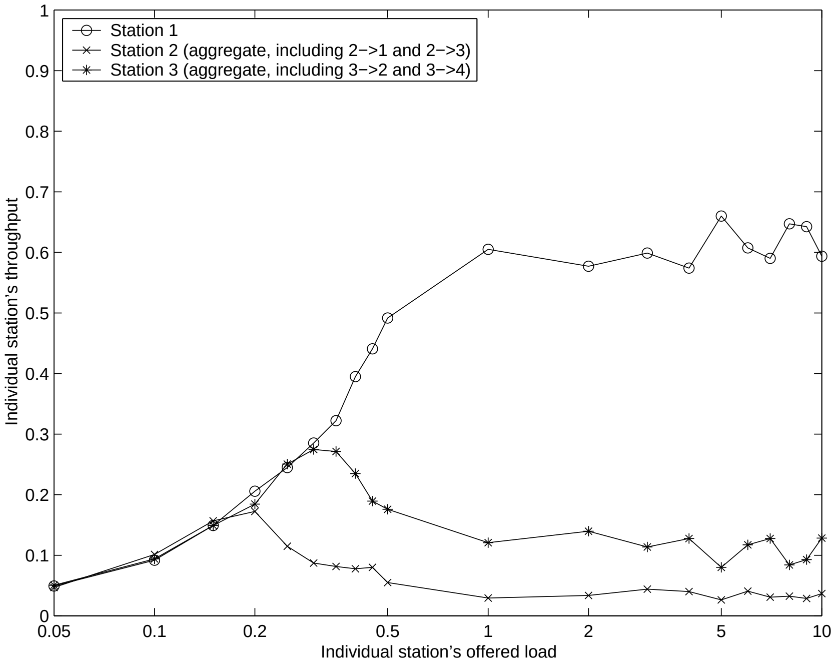
(b)

Figure 4.2: (a) Station throughput, (b) fairness index versus station offered load for the 4-station scenario.

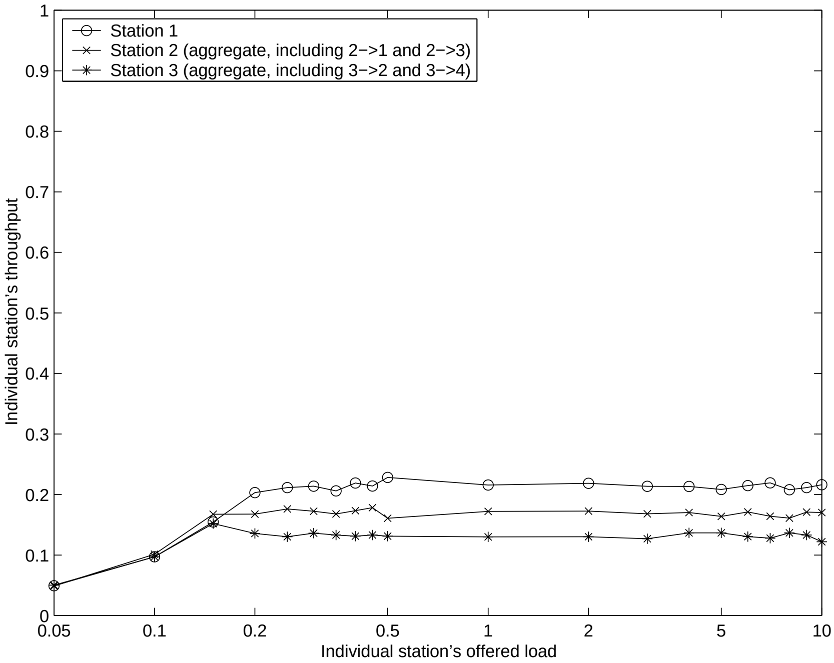
scheme, it still cannot solve the fairness problem that appears in this 4-station scenario.

In the 5-station scenario, we investigate two cases. In the first case, each station generates Poisson traffic with the same mean rate. For station 2, 3 and 4, each has two active links to its neighbors. For each packet that is generated, each of these stations will randomly choose a neighbor as its destination. For this case, we consider per-station fairness only and aggregate the two links' throughput as the corresponding station's throughput. The results are shown in Figure 4.3. Due to symmetry, we show results for stations 1, 2 and 3 only. In this case, edge stations (1 and 5) face less congestion and their packets are easier to get through. As the binary exponential backoff always favors the last succeeding station, the edge stations will get much higher throughput than other stations. Our scheme works much better to achieve fairness because stations 1 and 5 will yield the channel to other stations when they estimate that they have obtained more share than what they should get.

In the second case, each station generates Poisson traffic for each link with the same mean rate. Therefore, stations 2, 3 and 4, require equal fair share for each of their links. We experiment with two situations. One is that stations 2, 3 and 4 still set $\phi = 0.5$, the other is that these stations set $\phi = 0.67$ which indicates that they require two times the share of other stations as each of them has two active links. The results are shown in Figure 4.4. The figure shows that even if stations 2, 3 and 4 do not increase their ϕ, the modified algorithm can still achieve much better fairness than DFWMAC. When they do increase the fair share ϕ to that of

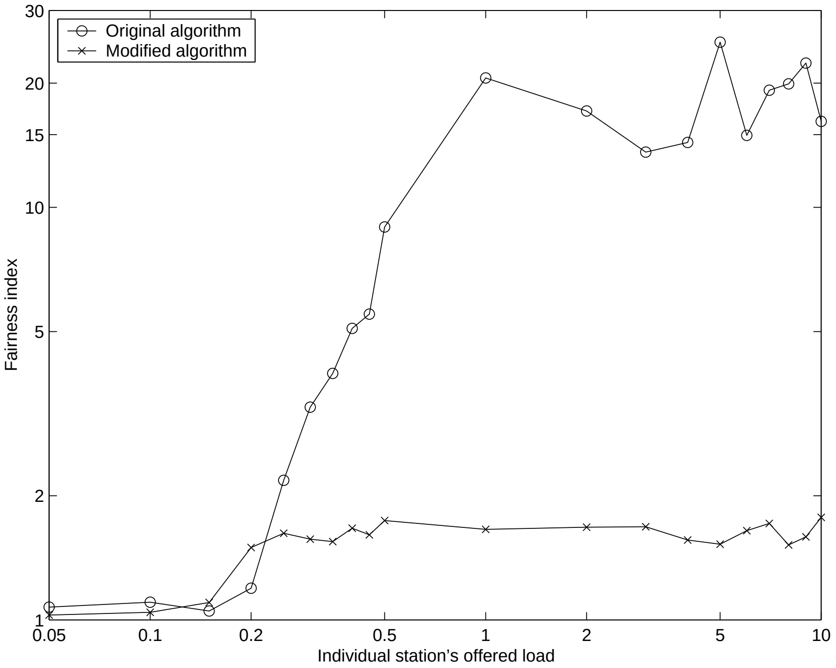


(a)



(b)

Figure 4.3: (a) Station throughput (original algorithm), (b) station throughput (modified algorithm), (c) fairness index versus station offered load for the 5-station scenario.



(b)

Figure 4.3: (a) Station throughput (original algorithm), (b) station throughput (modified algorithm), (c) fairness index versus station offered load for the 5-station scenario. (contd.)

a router (0.67), the fairness can be further improved.

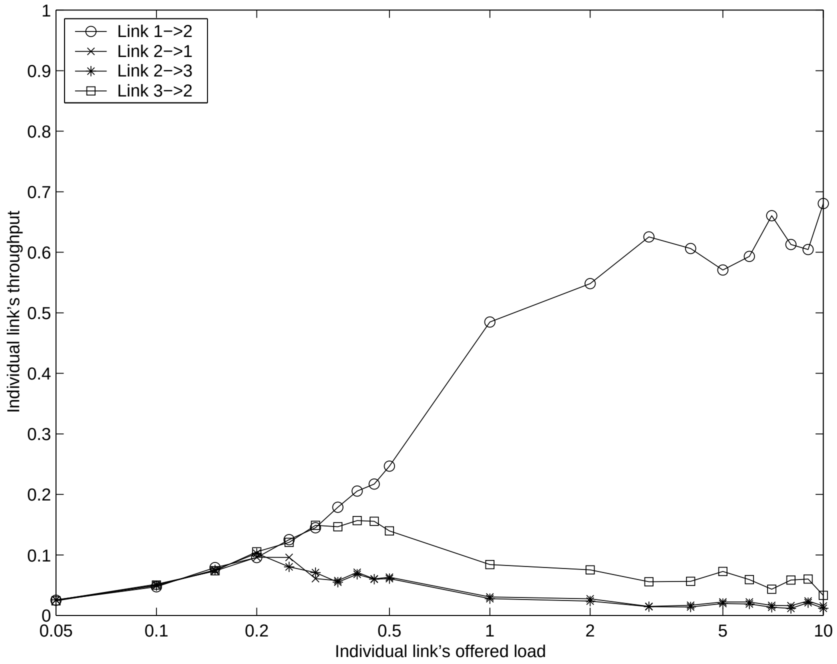
In the 6-station scenario, each station generates Poisson traffic with the same mean rate and results are shown in Figure 4.5. Due to symmetry, we show the results for station 1 and 3 only. As there may be concurrent transmissions between two pairs of edge stations (station 1 and 2, station 5 and 6), inner stations 3 and 4 suffer severe degradation in throughput as in the case of the original DFWMAC protocol. Our estimation becomes somewhat inaccurate in this case because some of these concurrent transmissions between edge stations may be interpreted as noise by inner stations and will not be counted in the fair share estimation[3]. However, our approach can still achieve far better fairness than DFWMAC.

All the simulations show that we tradeoff some throughput for fairness. As our approach in fact encourages stations to participate in fair competition, some channel bandwidth is lost due to the fact that more chances are given to those stations in unfavorable locations to compete for the channel and thus possibility of collisions increases.

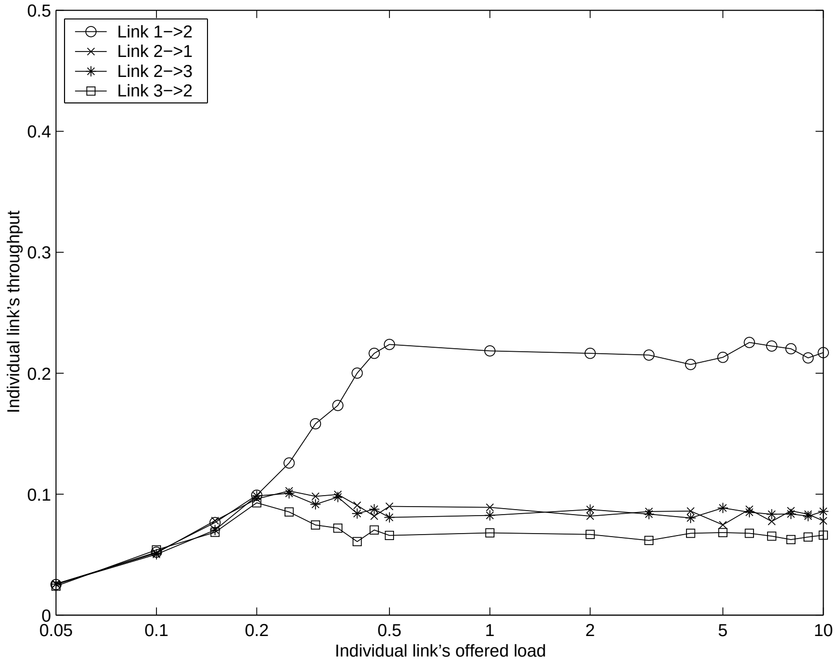
Next, let us investigate how C influences the fair index and throughput and provide some insight on the choice of C. Figure 4.6 shows that C has little influence on the fairness index and throughput in the 4-station scenario. In fact, only two stations are actively competing with each other in the 4-station scenario, therefore collision resolution is relatively easy even though they are hidden from each other.

---

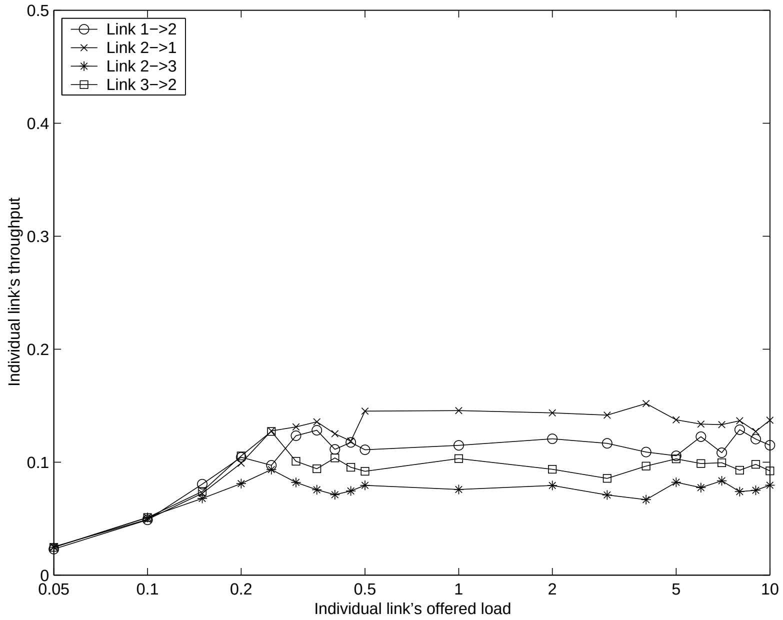
[3] An anonymous reviewer of [21] suggested the use of carrier sensing instead of packet sensing to tackle the problem. However, this approach is not accurate as well as no station knows if the noisy period of the channel should be added to its own share or other stations' share. In addition, this approach requires constant monitoring of the channel and thus is not economical from a power standpoint.


Link 1−>2
Link 2−>1
Link 2−>3
Link 3−>2
Individual link's throughput
Individual link's offered load
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0.05
0.1
0.2
0.5
1
2
5
10

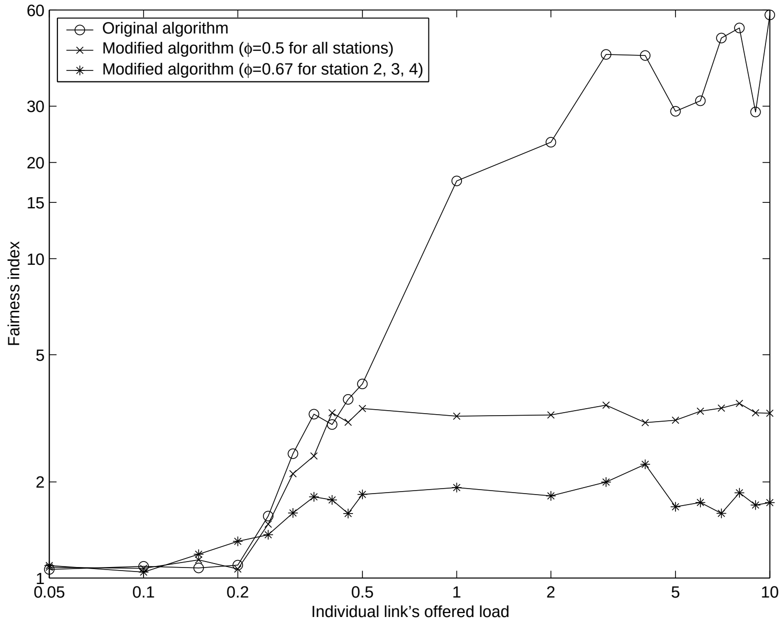
(a)


Link 1−>2
Link 2−>1
Link 2−>3
Link 3−>2
Individual link's throughput
Individual link's offered load
0
0.1
0.2
0.3
0.4
0.5
0.05
0.1
0.2
0.5
1
2
5
10

(b)

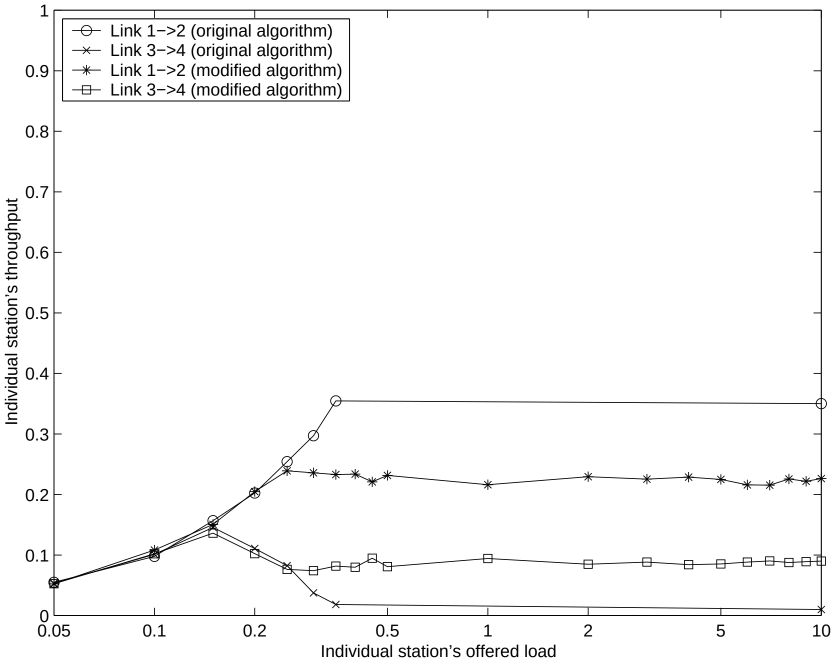


(c)

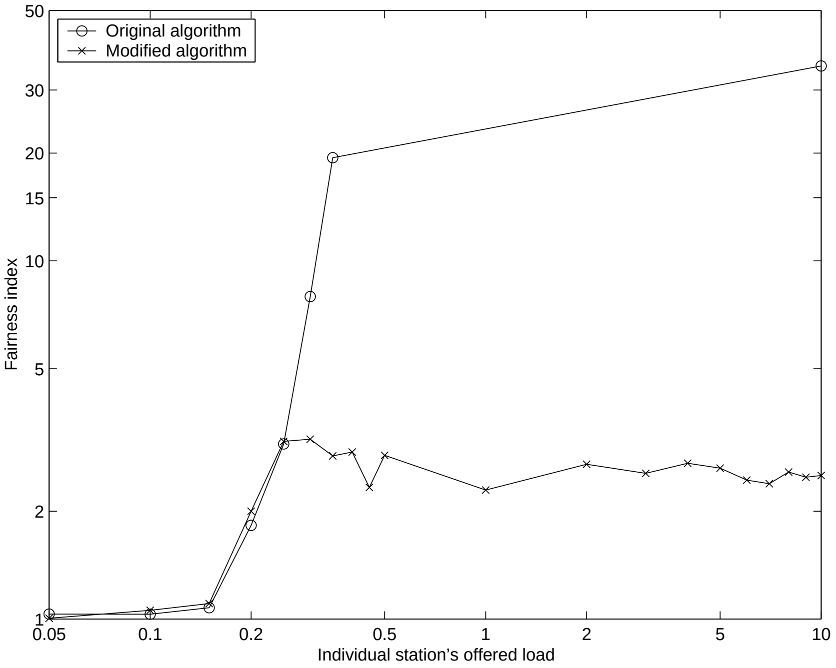


(d)

Figure 4.4: (a) Link throughput (original algorithm), (b) link throughput (modified algorithm, $\phi = 0.5$ for all), (c) link throughput (modified algorithm, $\phi = 0.67$ for stations 2, 3 and 4), (d) fairness index versus station offered load for the 5-station scenario.



(a)

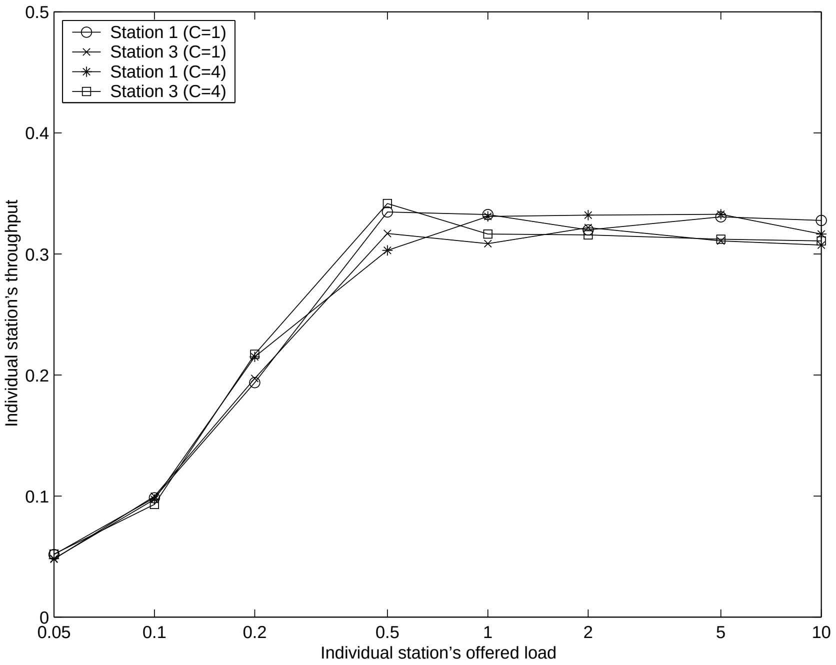


(b)

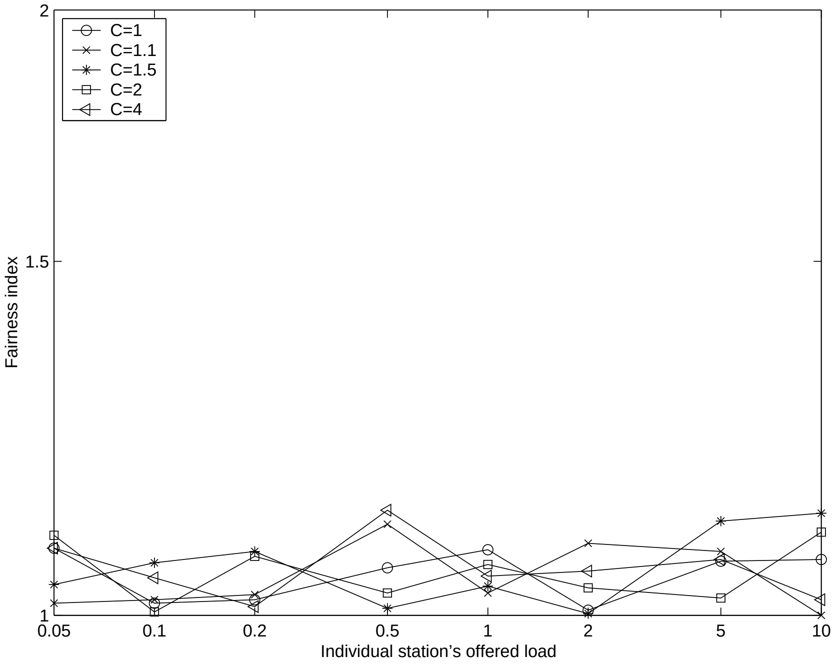
Figure 4.5: (a) Station throughput, (b) fairness index versus station offered load for the 6-station scenario.

Figure 4.7 shows how C influences per-station fairness in the 5-station scenario. In this case, each station generates Poisson traffic with the same rate. The difference is that for stations 2, 3 and 4, each chooses randomly one of its two neighbors as a packet's destination. It shows that when $C = 2$ or $C = 4$, achieving stable throughput is actually difficult when the load to the channel is high enough. The estimation-based algorithm becomes unstable when C increases and the fairness index fluctuates. In addition, as a station does not update its contention window size if its estimated fair index is within range of $(1/C, C)$, large values of C may lead to the station being stuck for a very long time with a small contention window. In this case, the probability of collisions is very high, resulting consequently in a low throughput.

Figure 4.8 and 4.9 show how C influences per-stream fairness for the 5-station scenario. Figure 4.8 refers to the case when all stations choose the same $\phi = 0.5$ while Figure 4.9 refers to the case when stations 2, 3 and 4 choose $\phi = 0.67$ if they want their two links to be treated equally. We see once again that large values of C result in unstability and low throughput when the load is high. When we compare the aggregate throughput of stations 1, 2 and 3 with different ϕ and C as shown in Figure 4.10, we have an interesting finding that if we choose C properly, increasing ϕ for stations 2 and 3 does not decrease the aggregate throughput of stations 1, 2 and 3 as the loss in station 1's throughput is compensated by the increase in stations 2 and 3's throughput. This is a very desirable feature of the estimation based algorithm.

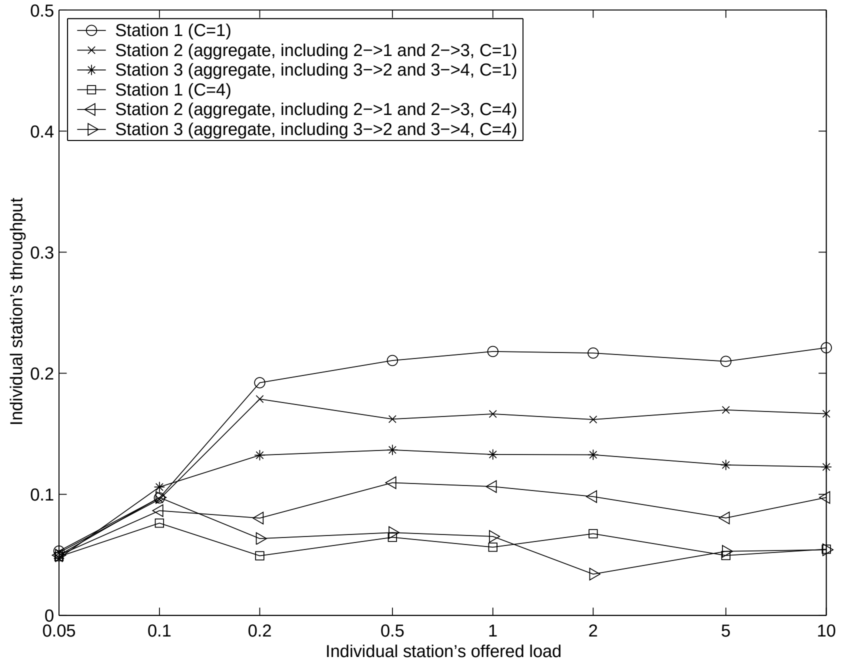


(a) Throughput

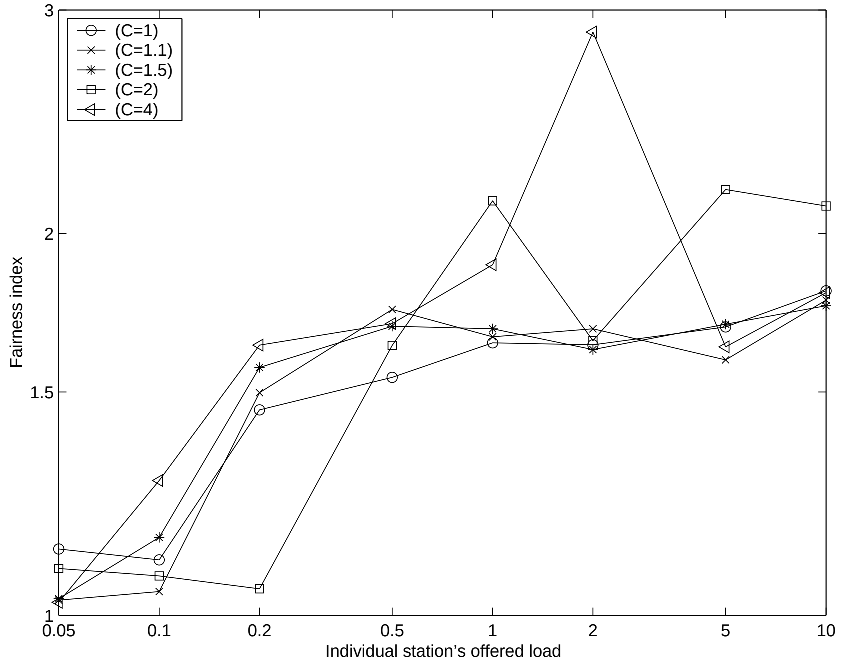


(b) Fair index

Figure 4.6: C's influence on the 4-station scenario

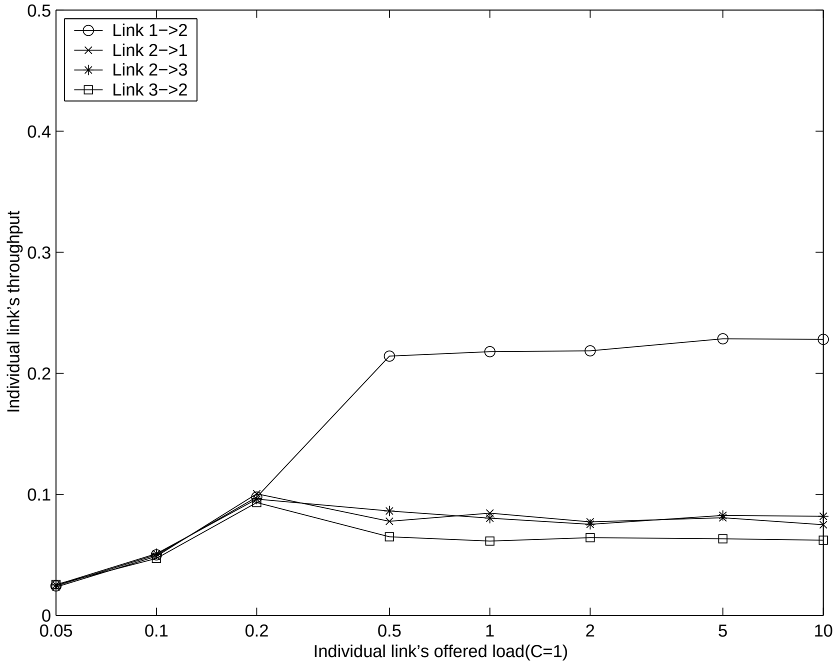


(a) Throughput

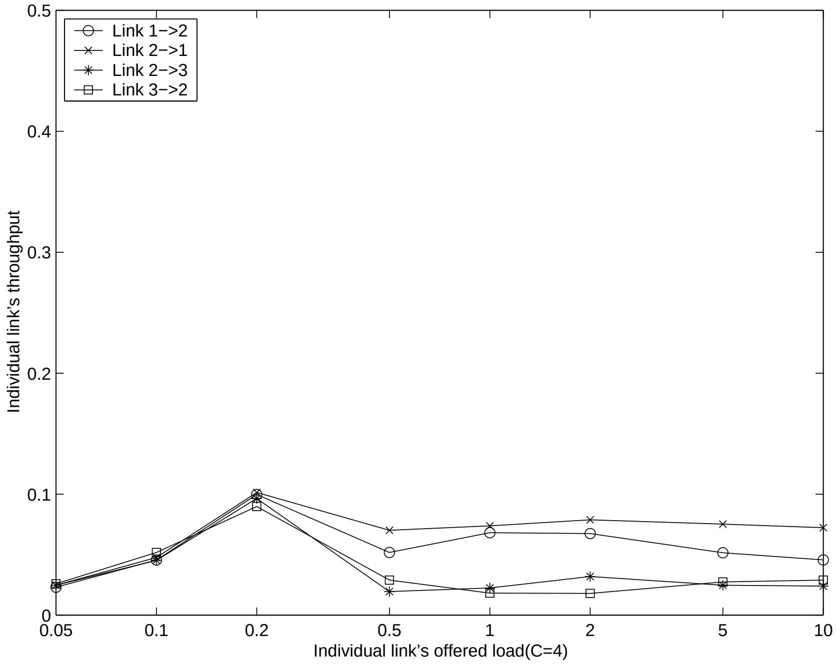


(b) Fair index

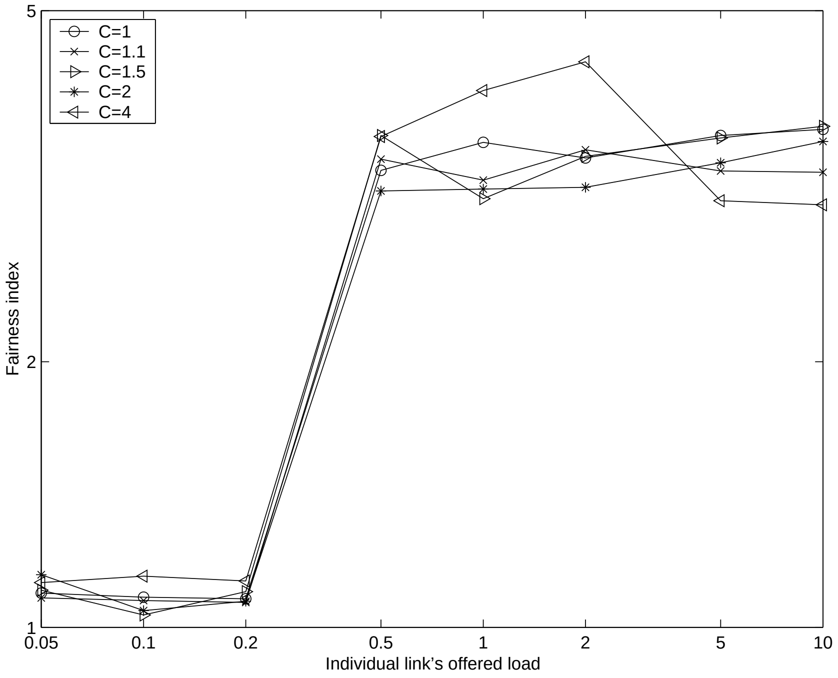
Figure 4.7: C's influence on the 5-station scenario (per station fairness)



(a) Throughput(C=1)

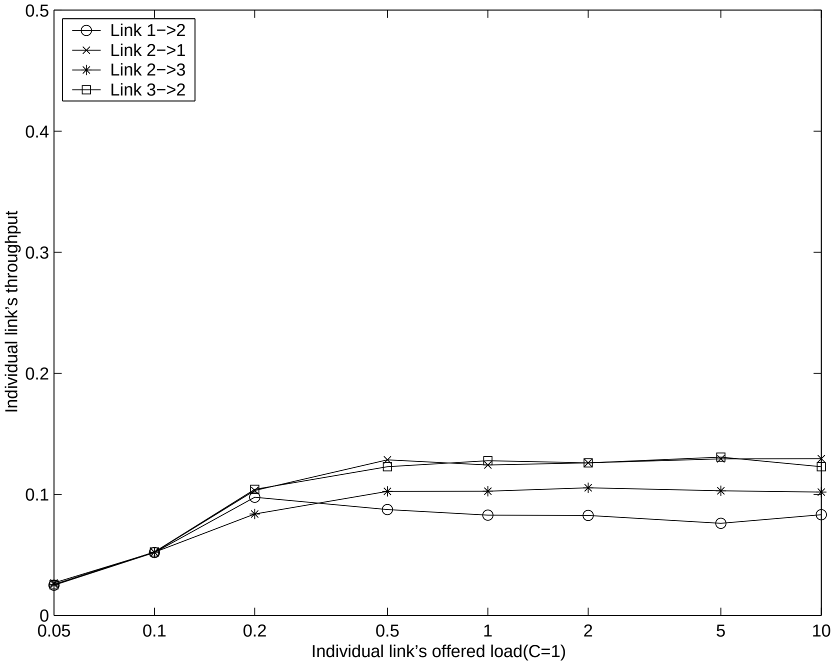


(b) Throughput(C=4)

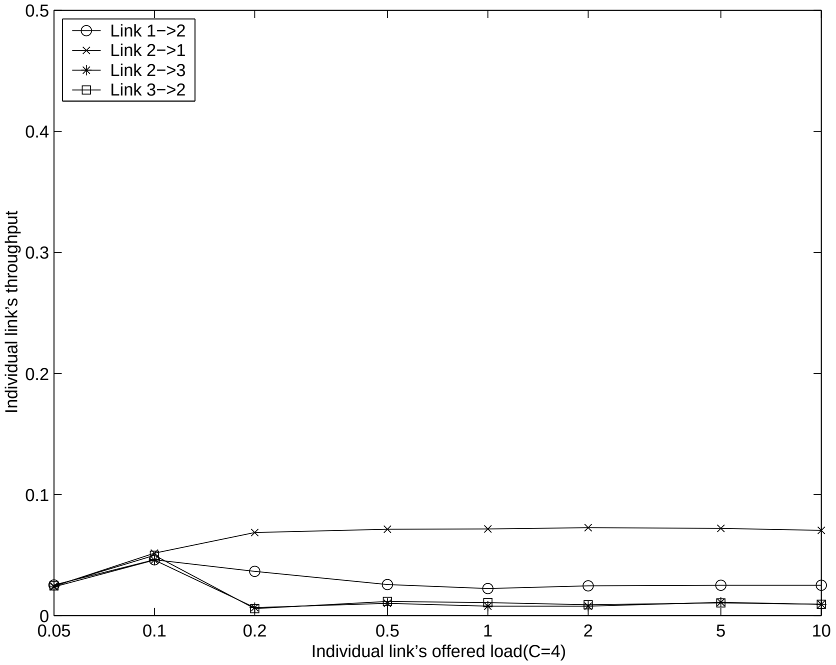


(c) Fair index

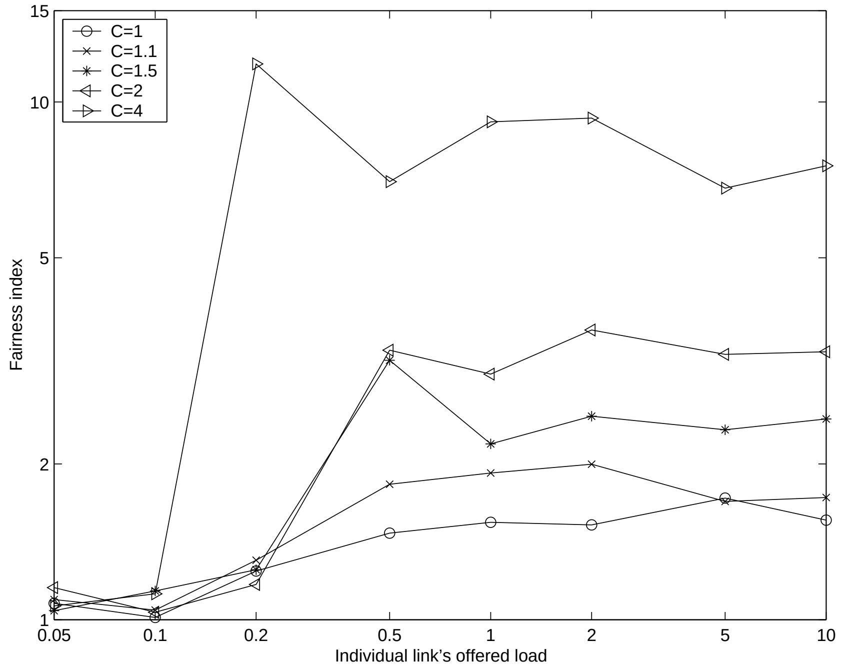
Figure 4.8: C's influence on the 5-station scenario (per stream fairness, $\phi = 0.5$ for all)



(a) Throughput(C=1)



(b) Throughput(C=4)



(c) Fair index

Figure 4.9: C's influence on the 5-station scenario (per stream fairness, $\phi = 0.67$ for stations 2, 3 and 4)

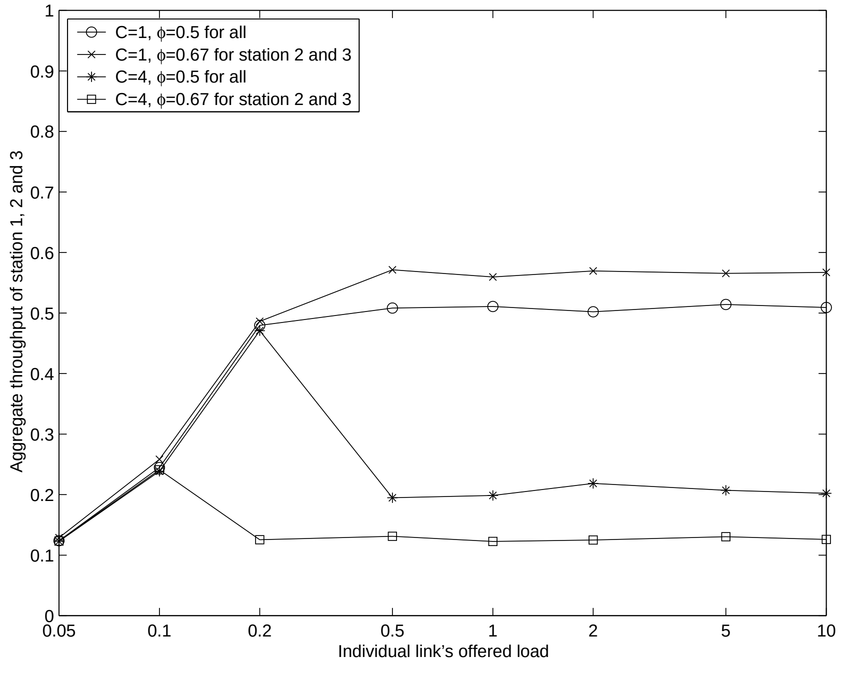


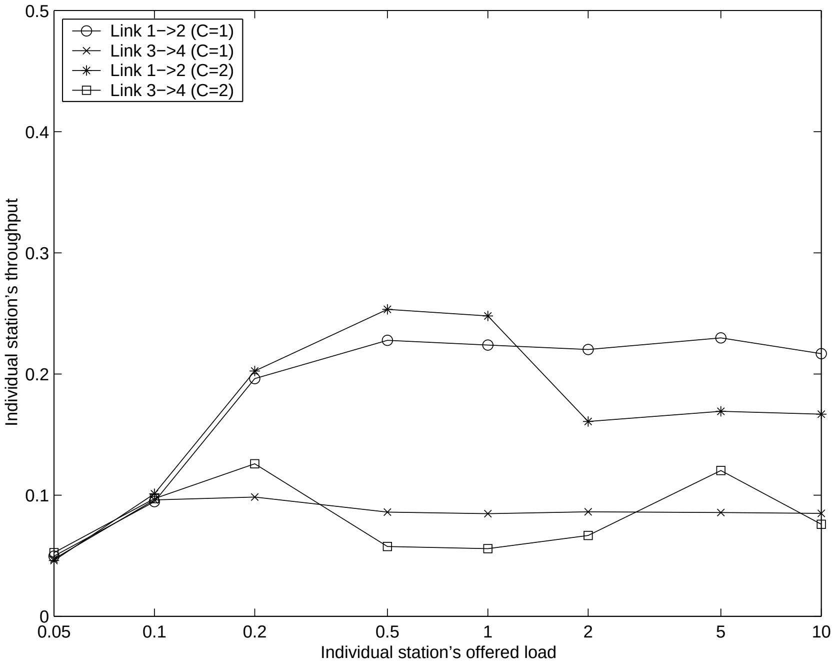
Figure 4.10: Aggregate throughput comparison

Figure 4.11 shows that the influence of C to fairness index and throughput is consistent in the 6-station scenario. Besides, as stations in this network (6-station) have more stations to compete with than in the previous configurations (i.e., station 3 faces competitions from 3 neighbors and 2 hidden terminals), the estimation based algorithm becomes unstable when $C \geq 1.5$. From these sets of simulations, we can conclude that smaller Cs are usually favored as a station can act more aggressively to changes in its estimated fair index and it is unlikely that a station is stuck in a small contention window which can have negative effects on throughput. Generally speaking, C can be chosen to be equal to 1 or a little fraction above to achieve optimal results.

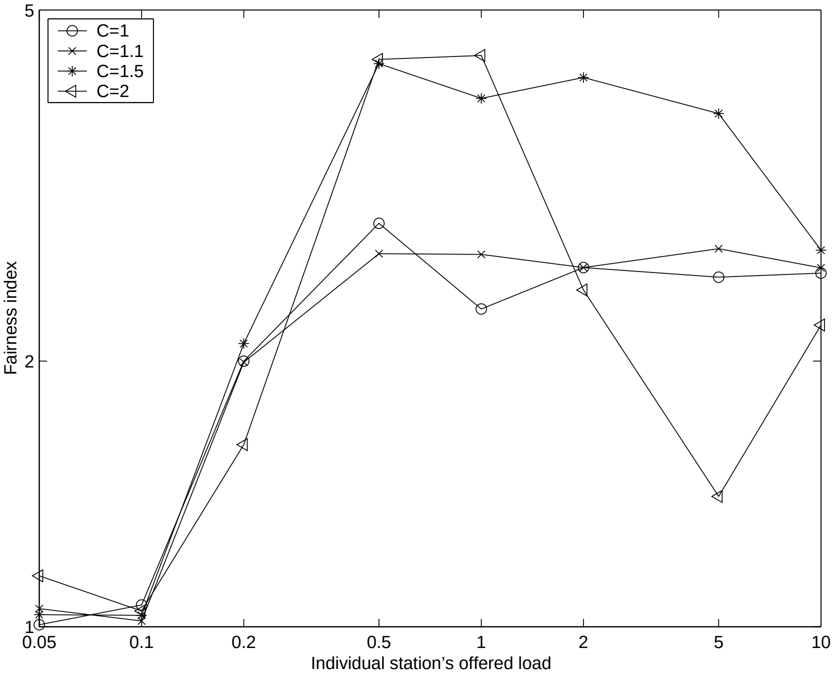
4.2.2 Variable Packet Length

For the variable packet length case, we choose two types of data packet. One is short data packet with total length of 50 bytes. The other is long data packet with total length of 500 bytes. For the short data packets, the RTS/CTS access method is not used. These two types may represent for example Telnet and FTP traffic commonly carried in IP based networks. The data packet generator at each station generates a short packet or a long one with equal probability. According to the insight provided by previous subsection, we choose $C = 1$. Other simulation parameters remain unchanged.

The results for the 4-station scenario are shown in Figure 4.12. For the 5-station scenario, we still investigate two cases. One is per-stream fairness whose results are shown in Figure 4.13. The other is per-station fairness whose results are shown



(a) Throughput



(b) Fair index

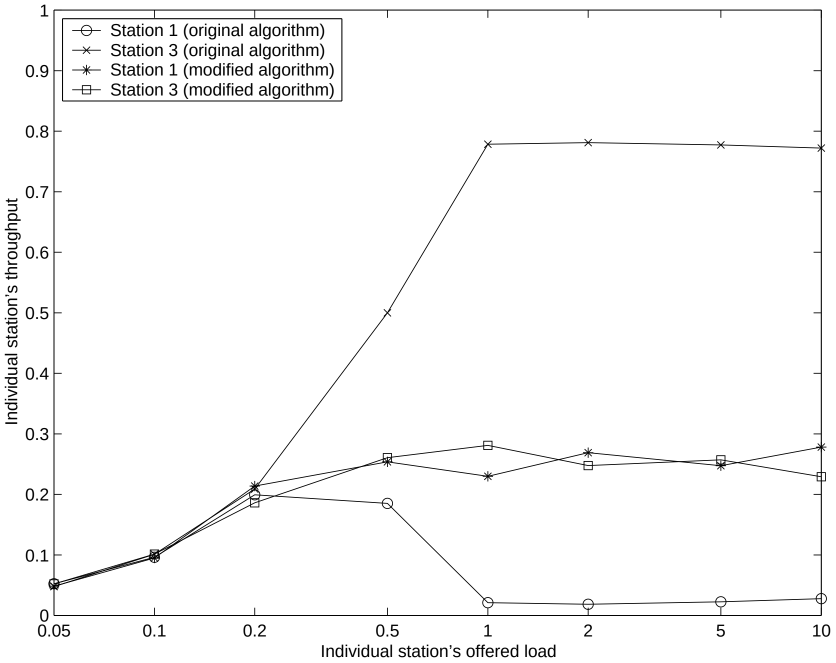
Figure 4.11: *C*'s influence on the 6-station scenario

in Figure 4.14. The results for 6-station scenario is shown in Figure 4.15. All these results show that the fairness problem sometimes is more severe with BEB compared with the case for fixed packet length while the fair share based backoff scheme can still achieve far better fairness than BEB.

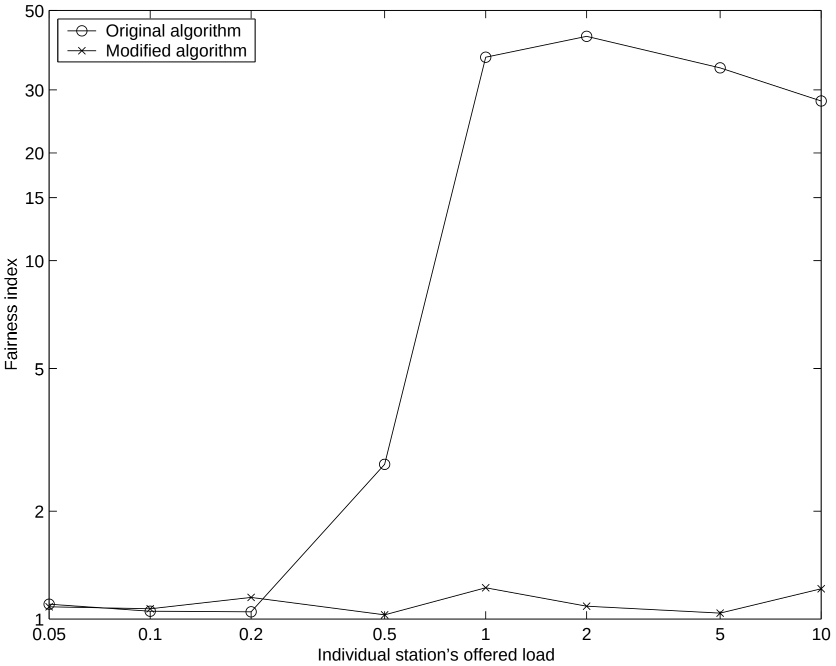
The influence of C is also investigated. The results for the 4-station scenario are shown in Figure 4.16. The results for the 5-station scenario are shown in Figure 4.17, 4.18 and 4.19. The results for the 6-station scenario are shown in Figure 4.20. Again, small C is favored in all these cases as stations can be more aggressive in adjusting their contention window and avoid being stuck in a small contention window for a long time.

4.3 Summary

In this chapter, we defined a new fairness metric for wireless ad-hoc networks incorporating both the idea of per-station and per-stream (or per-link) fairness and pointed out that the target to achieve fairness is to minimize the fairness index thus defined. We then proposed a different backoff scheme for IEEE 802.11 DFWMAC DCF protocol, where each station adjusts its contention window according to the estimated shares that other stations and itself have obtained. The new scheme can support the case for variable packet length. Simulation results show that this scheme can achieve far better fairness than the original backoff scheme of DFWMAC, despite the fact that it sacrifices some throughput. As this scheme does not assume any knowledge about the network's topology and thus does not

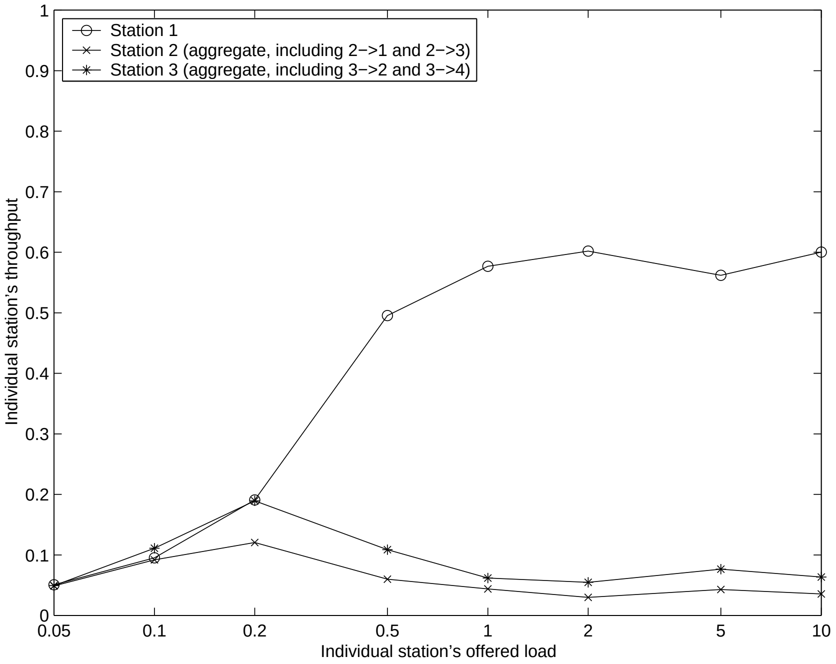


(a)

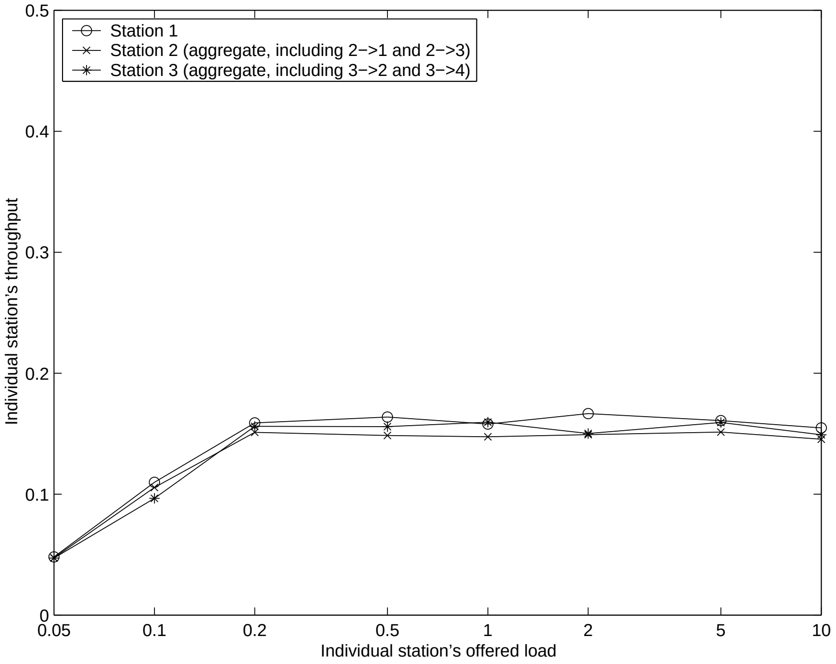


(b)

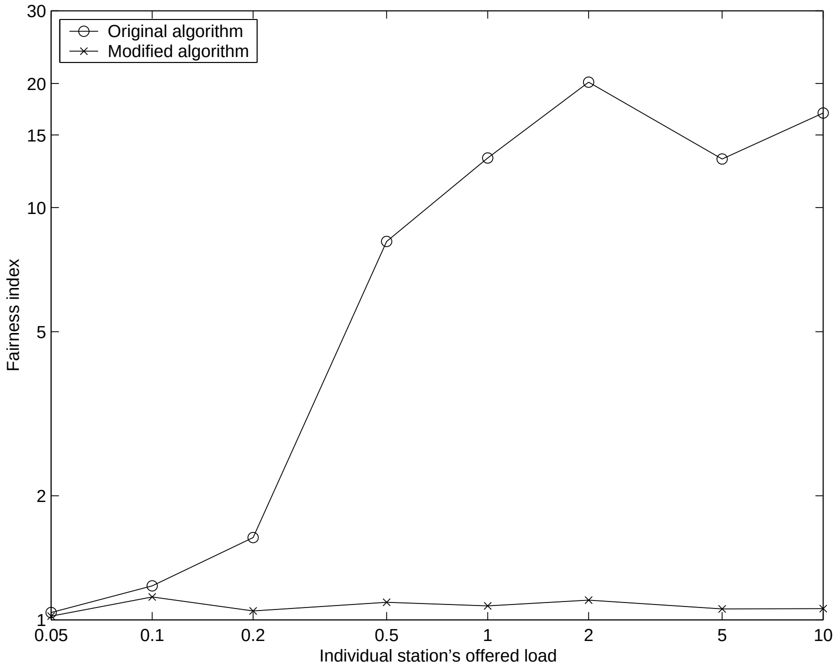
Figure 4.12: (a) Station throughput, (b) fairness index versus station offered load for the 4-station scenario – variable packet length.


Station 1
Station 2 (aggregate, including 2−>1 and 2−>3)
Station 3 (aggregate, including 3−>2 and 3−>4)
Individual station's throughput
Individual station's offered load
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0.05
0.1
0.2
0.5
1
2
5
10

(a)

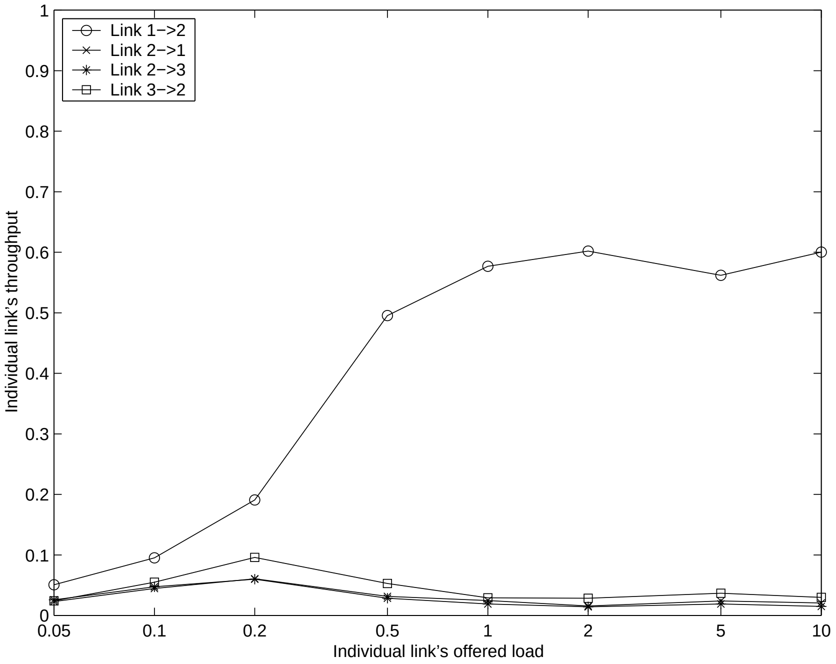

Station 1
Station 2 (aggregate, including 2−>1 and 2−>3)
Station 3 (aggregate, including 3−>2 and 3−>4)
Individual station's throughput
Individual station's offered load
0
0.1
0.2
0.3
0.4
0.5
0.05
0.1
0.2
0.5
1
2
5
10

(b)

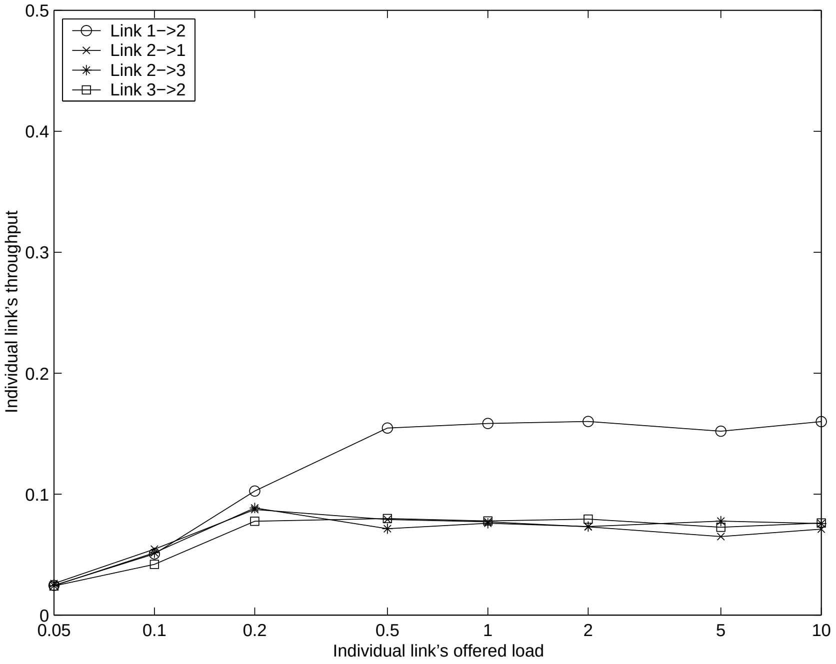


(c)

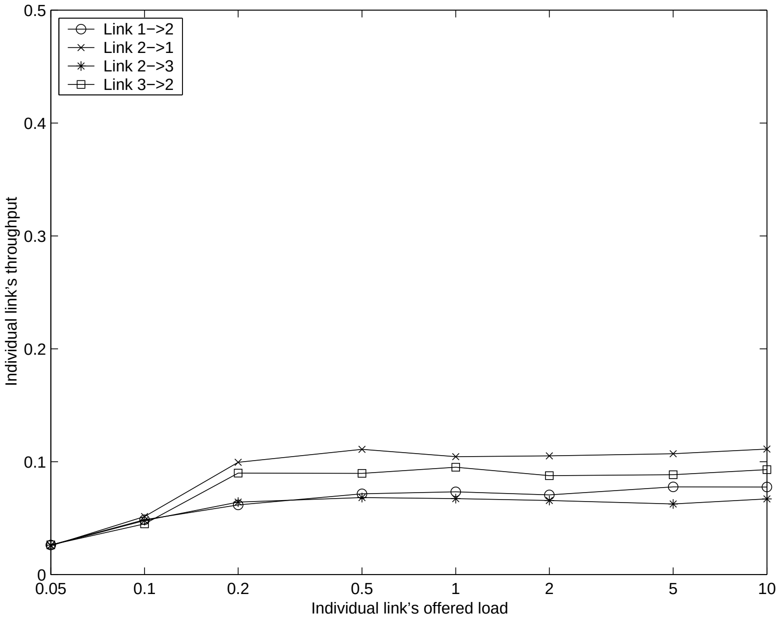
Figure 4.13: (a) Station throughput (original algorithm), (b) station throughput (modified algorithm), (c) fairness index versus station offered load for the 5-station scenario – variable packet length.


Link 1−>2
Link 2−>1
Link 2−>3
Link 3−>2
Individual link's throughput
Individual link's offered load
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0.05
0.1
0.2
0.5
1
2
5
10

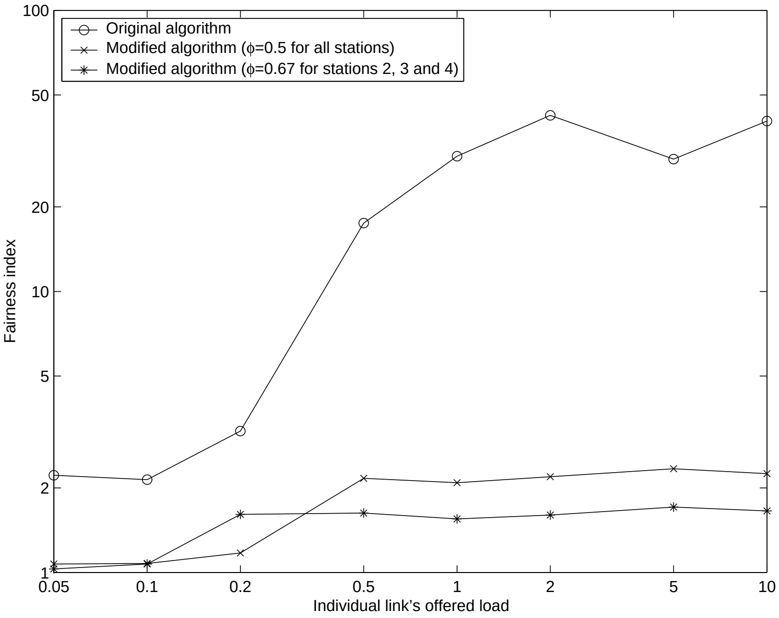
(a)


Link 1−>2
Link 2−>1
Link 2−>3
Link 3−>2
Individual link's throughput
Individual link's offered load
0
0.1
0.2
0.3
0.4
0.5
0.05
0.1
0.2
0.5
1
2
5
10

(b)



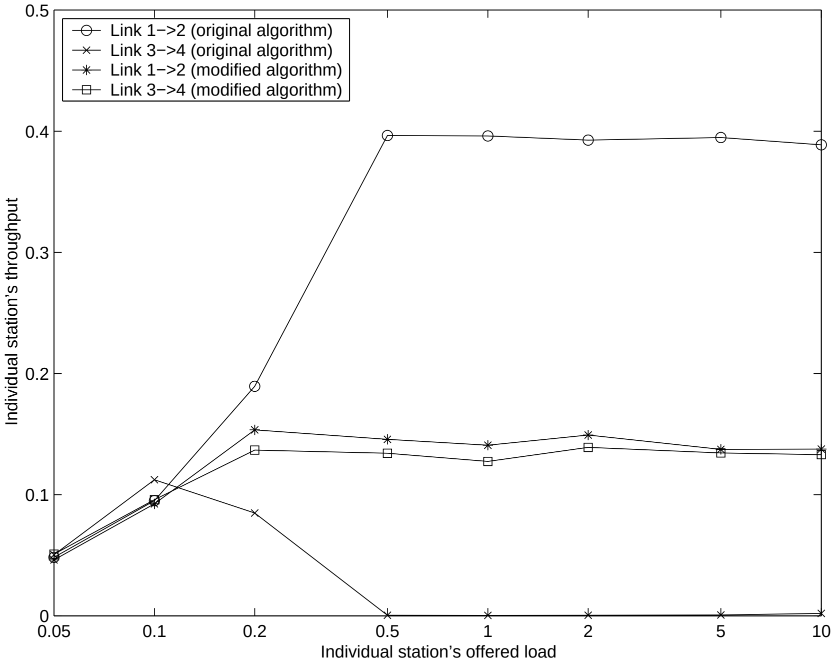
(c)



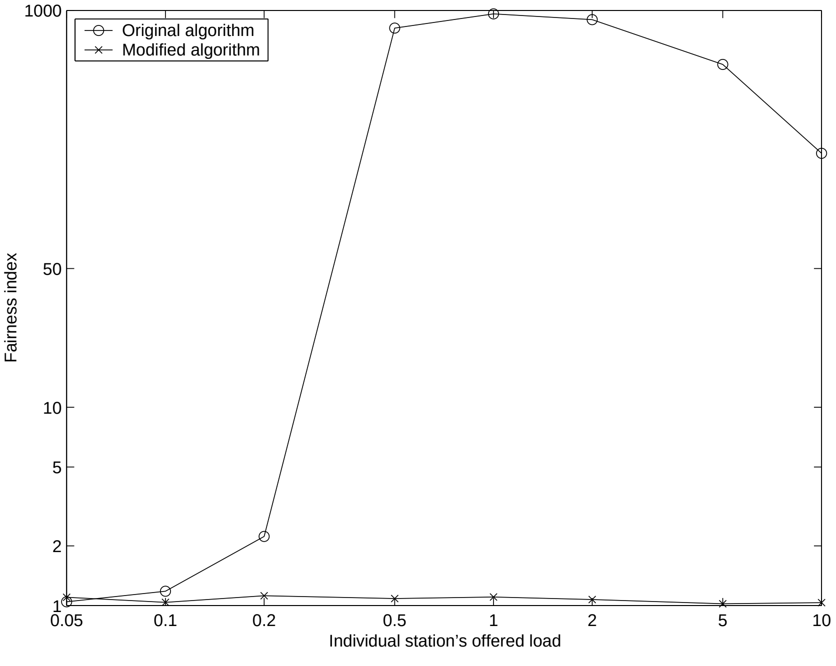
(d)

Figure 4.14: (a) Link throughput (original algorithm), (b) link throughput (modified algorithm, $\phi = 0.5$ for all), (c) link throughput (modified algorithm, $\phi = 0.67$ for stations 2, 3 and 4), (d) fairness index versus station offered load for the 5-station scenario – variable packet length.

.

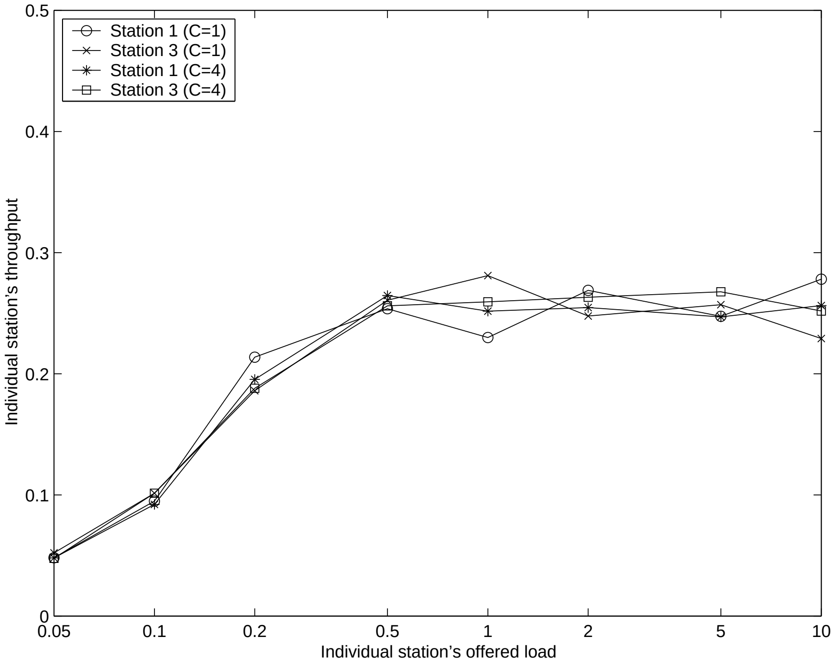


(a)

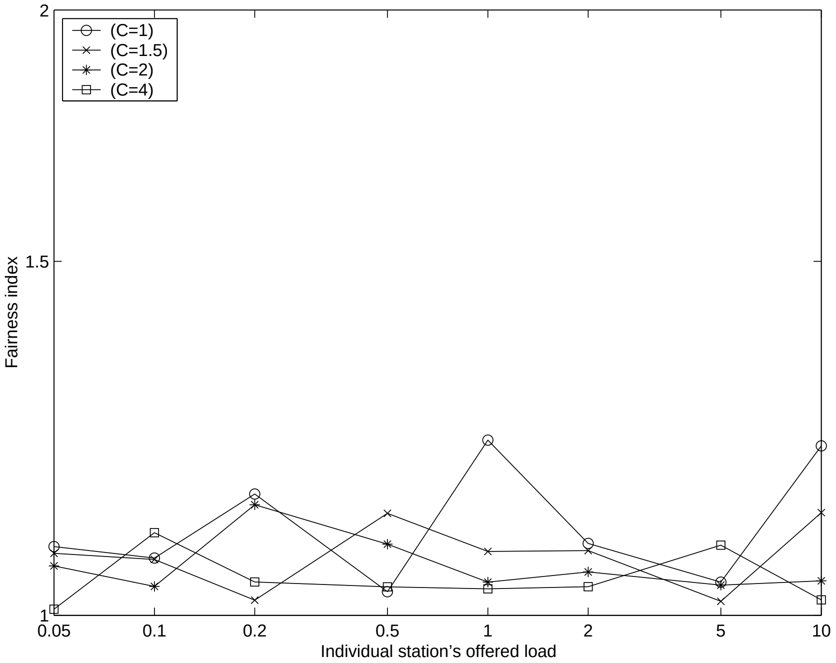


(b)

Figure 4.15: (a) Station throughput, (b) fairness index versus station offered load for the 6-station scenario – variable packet length.

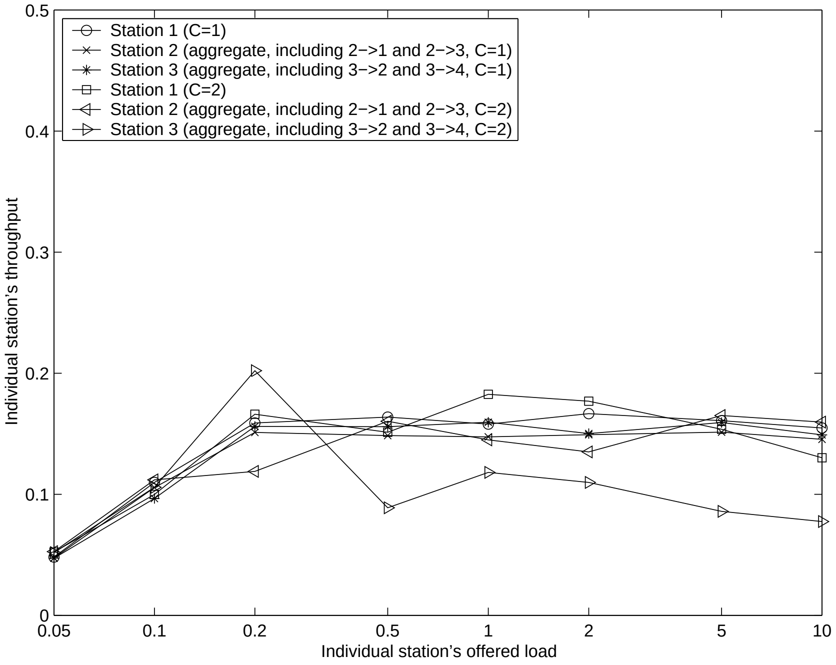


(a) Throughput

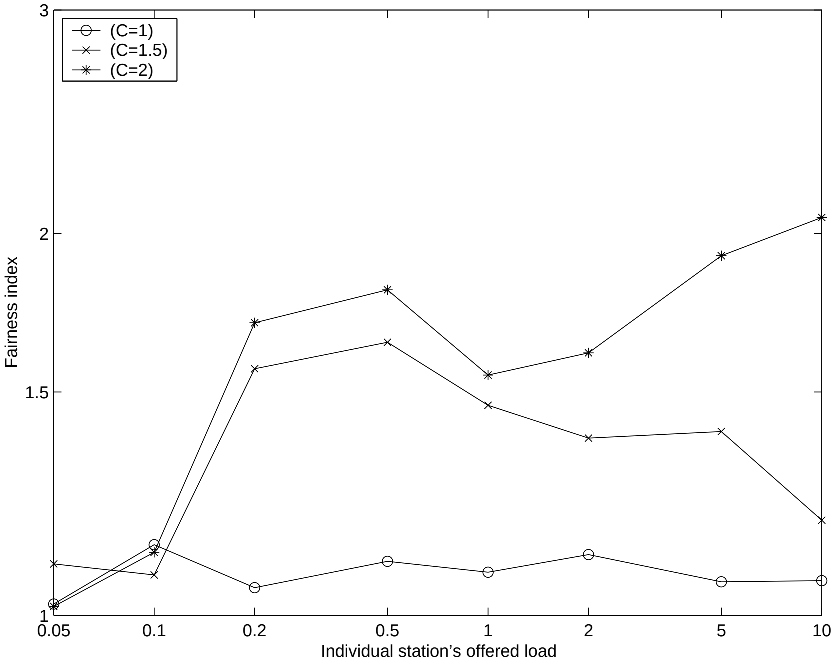


(b) Fair index

Figure 4.16: C's influence on the 4-station scenario

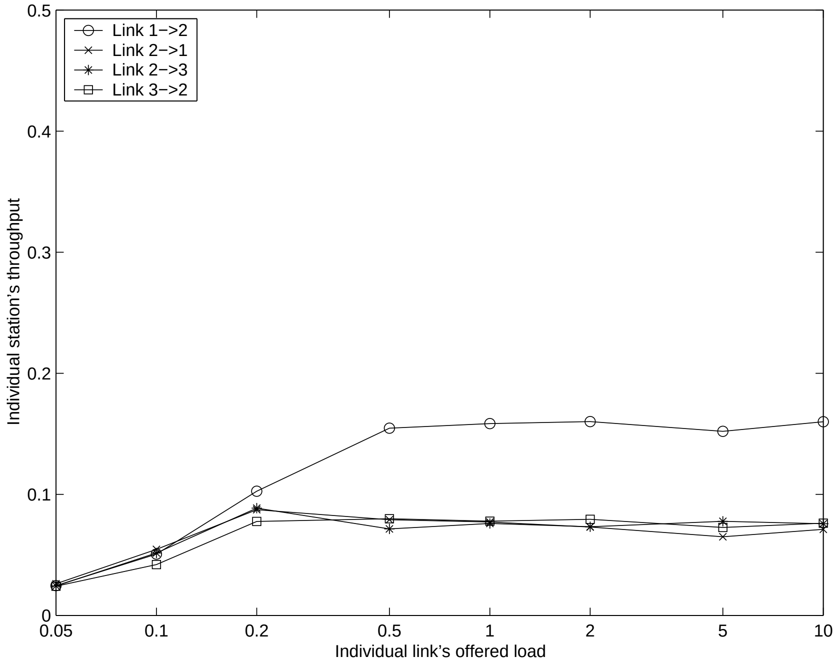


(a) Throughput

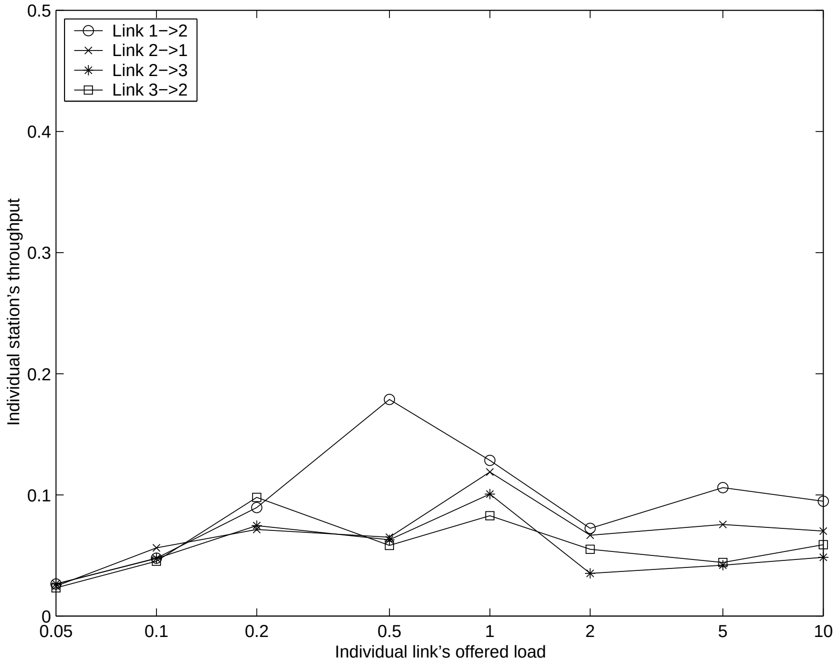


(b) Fair index

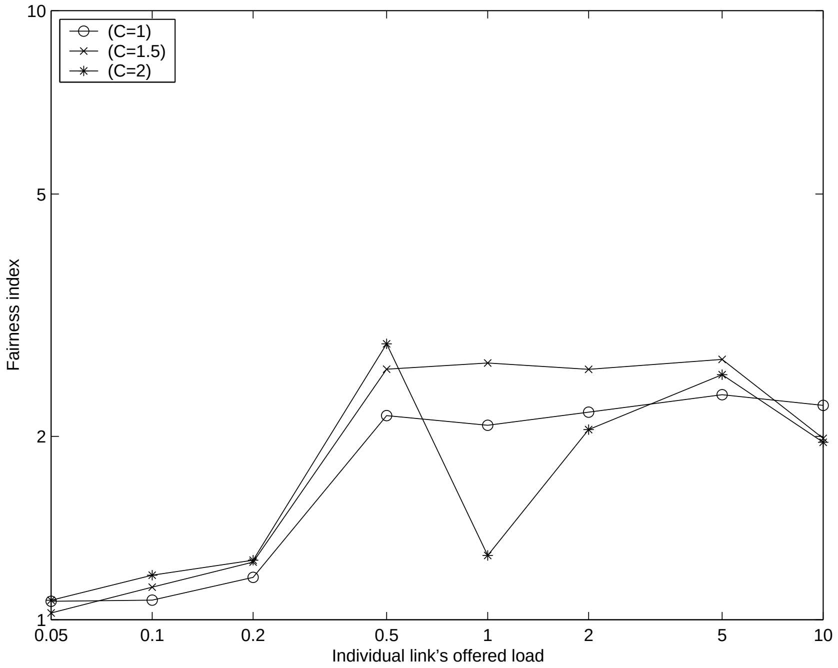
Figure 4.17: C's influence on the 5-station scenario (per station fairness)



(a) Throughput(C=1)

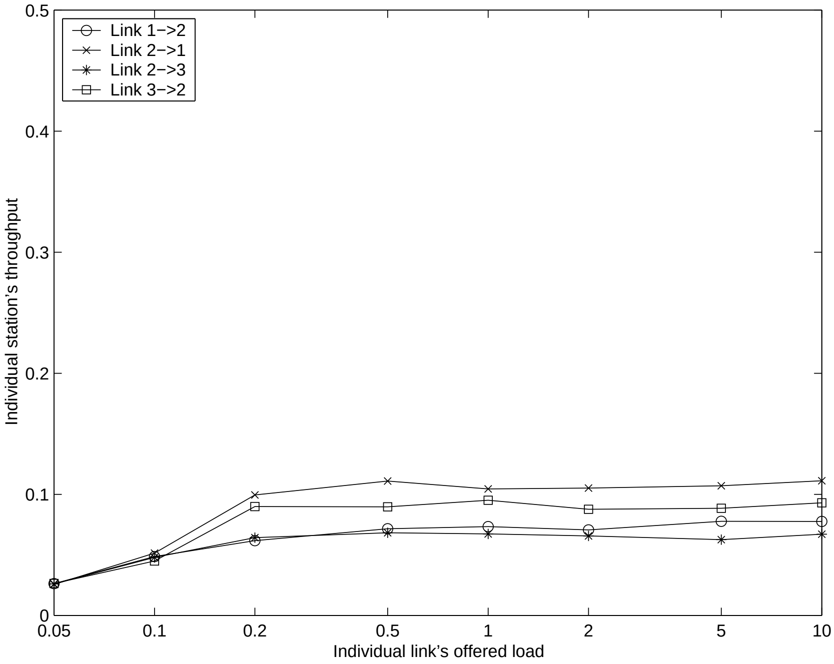


(b) Throughput(C=2)

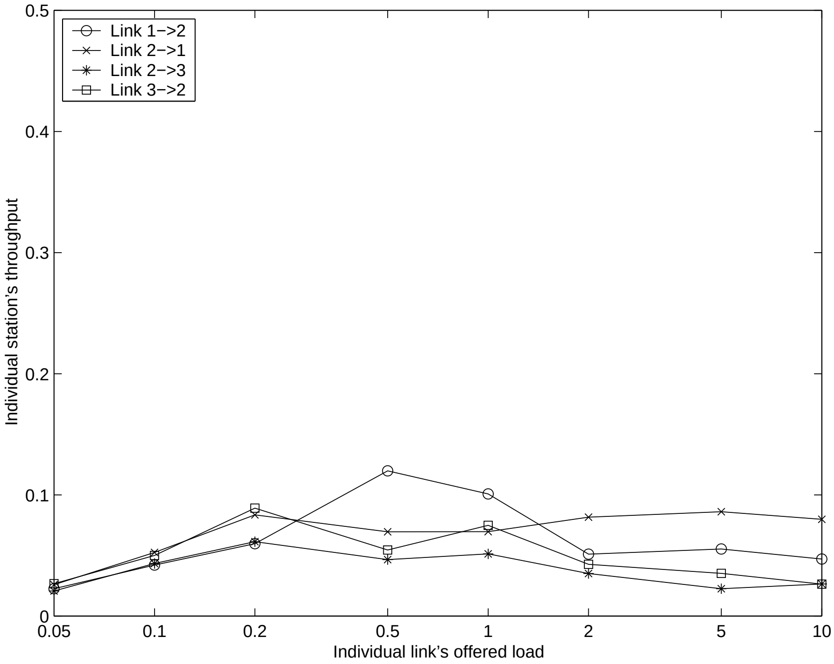


(c) Fair index

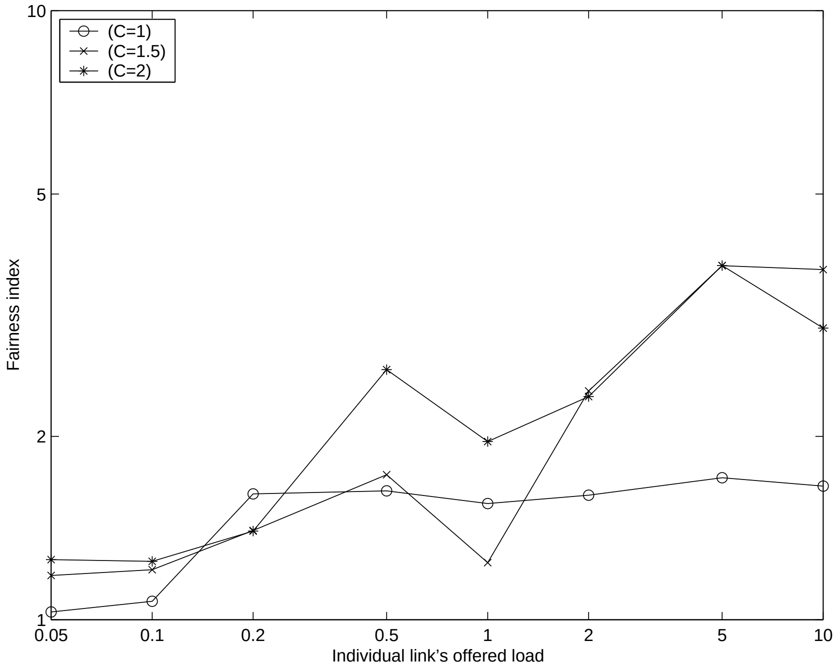
Figure 4.18: C's influence on the 5-station scenario (per stream fairness, $\phi = 0.5$ for all)



(a) Throughput(C=1)

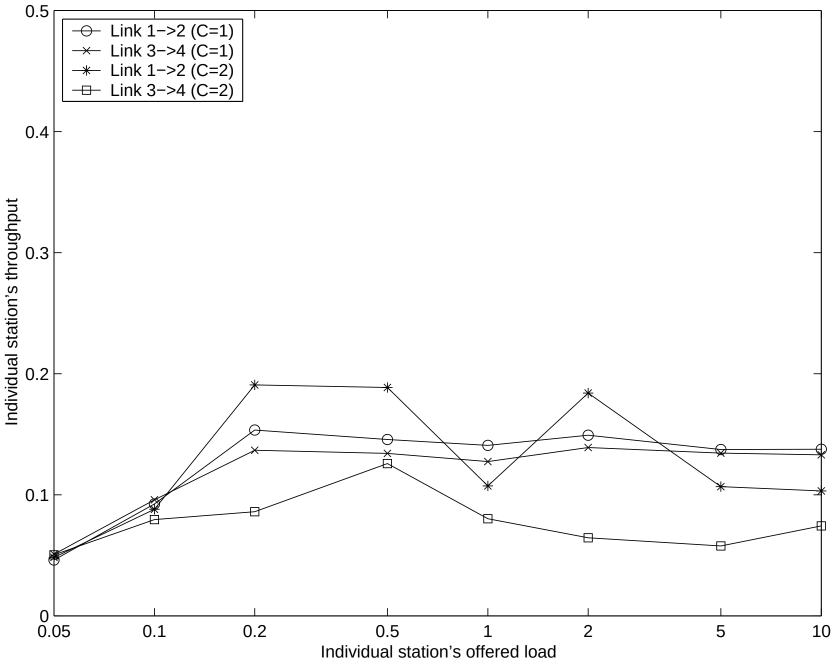


(b) Throughput(C=2)

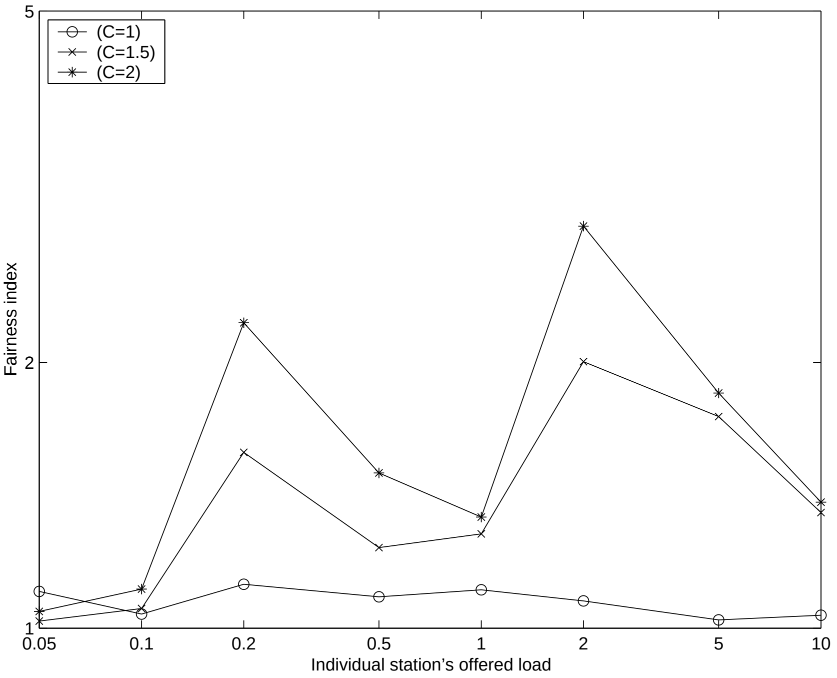


(c) Fair index

Figure 4.19: C's influence on the 5-station scenario (per stream fairness, $\phi = 0.67$ for stations 2, 3 and 4)



(a) Throughput



(b) Fair index

Figure 4.20: C's influence on the 6-station scenario

require broadcast packets to disseminate information among stations, it is very simple and can be easily overlaid in the existing DFWMAC protocol.

Chapter 5

Conclusion

In this thesis, the characteristics of ad-hoc networks have been investigated in detail and the challenges imposed on MAC protocol design have been identified. Existing MAC protocols are inadequate to cater to the special needs of ad-hoc networks and thus we have proposed two MAC protocols specifically designed for ad-hoc networks.

One is the so-called priority based multiple access (PriMA) protocol, the other is the fair share based multiple access (FSMA) protocol. As these two protocols combine carrier sensing with optional RTS/CTS exchange to combat the hidden terminal problem, they can achieve good throughput. Additionally, both of them can provide better support for ad-hoc routing. In PriMA, stations that act in router's capacity can set a finite delay requirement for their packets (e.g. packets carrying routing information) to gain higher priority in accessing the channel and to have their packets delivered before others. In FSMA, routers can set a larger pre-defined fair share, which will enable them to send more packets than other normal stations if they want to.

The PriMA protocol can also be incorporated in an overall QoS architecture which can make use of PriMA's support for mixed types of traffic while the FSMA protocol provides a simple and effective solution to the fairness problem encountered in most existing MAC protocols. Both protocols' performances have been evaluated using computer simulations and results have shown that they are promising MAC protocol candidates for ad-hoc networks.

Future investigation will concentrate on the interaction between the MAC layer and the network layer, notably by trying to identify how routing protocols may be modified to exploit the special services provided by the PriMA and FSMA protocols.

Bibliography

[1] D. B. Johnson and D. A. Maltz, "Protocols for Adaptive Wireless and Mobile Networking," *IEEE Personal Communications*, vol. 3, pp. 34–42, Feb. 1996.

[2] C.-K. Toh, *Wireless ATM and Ad-Hoc Networks.* Kluwer Academic Publishers, 1997.

[3] "Mobile Ad-hoc Networks (manet) Charter." `http://www.ietf.org/html.charters/manet-charter.html`.

[4] Z. J. Haas and S. Tabrizi, "On Some Challenges and Design Choices in Ad-hoc Communications," *Proc. of IEEE MILCOM '98*, vol. 1, pp. 187–192, 1998.

[5] M. Corson, J. Macker, and G. Cirincione, "Internet-based Mobile Ad Hoc Networking," *IEEE Internet Computing*, vol. 3, no. 4, pp. 63–70, 1999.

[6] "NRL Mobile Ad Hoc Networking Page." `http://tonnant.itd.nrl.navy.mil/manet/manet_home.html`.

[7] J. P. Macker and M. S. Corson, "Mobile Ad Hoc Networking and the IETF," *ACM Mobile Computing and Communications Review*, vol. 2, no. 1, 1998.

[8] J. P. Macker and M. S. Corson, “Mobile Ad Hoc Networking and the IETF,” *ACM Mobile Computing and Communications Review*, vol. 2, no. 2, 1998.

[9] J. P. Macker and M. S. Corson, “Mobile Ad Hoc Networking and the IETF,” *ACM Mobile Computing and Communications Review*, vol. 2, no. 3, 1998.

[10] J. P. Macker and M. S. Corson, “Mobile Ad Hoc Networking and the IETF,” *ACM Mobile Computing and Communications Review*, vol. 2, no. 4, 1998.

[11] J. P. Macker and M. S. Corson, “Mobile Ad Hoc Networking and the IETF,” *ACM Mobile Computing and Communications Review*, vol. 3, no. 1, 1999.

[12] J. P. Macker and M. S. Corson, “Mobile Ad Hoc Networking and the IETF,” *ACM Mobile Computing and Communications Review*, vol. 3, no. 2, 1999.

[13] J.-P. Hubaux, J.-Y. Le Boudec, S. Giordano, and M. Hamdi, “The Terminode Project: Towards Mobile Ad-Hoc WANs,” in *IEEE International Workshop on Mobile Multimedia Communications (MoMuC ’99)*, pp. 124–128, 1999.

[14] “Terminode: Mobile Communication and Information Systems.” `http://www.terminodes.org/`.

[15] J. J. Garcia-Luna-Aceves, C. Fullmer, E. Madruga, D. Beyer, and T. Frivold, “Wireless Internet Gateways (WINGs),” in *Proc. of IEEE MILCOM ’97*, Nov. 1997.

[16] Z. Haas and M. Pearlman, “The Zone Routing Protocol (ZRP) for Ad Hoc Networks.” draft-ietf-manet-zone-zrp-02.txt, June 1999. Work in progress.

[17] M. L. Jiang, J. Y. Li, and Y. C. Tay, "Cluster Based Routing Protocol(CBRP) Functional Specification." draft-ietf-manet-cbrp-spec-01.txt, Aug. 1999. Work in progress.

[18] R. Sivakumar, P. Sinha, and V. Bharghavan, "CEDAR: A Core-Extraction Distributed Ad Hoc Routing Algorithm," *IEEE Journal on Selected Areas in Communications, Special Issue on Ad Hoc Networks*, vol. 17, pp. 1454–65, Aug. 1999.

[19] Y. Wang and B. Bensaou, "Priority Based Multiple Access for Service Differentiation in Wireless Ad-Hoc Networks," in *Mobile and Wireless Communication Networks*, pp. 14–31, Springer-Verlag, 2000. Series Lecture Notes in Computer Science Vol. 1818.

[20] B. Bensaou and Y. Wang, "Distributed Scheduler in Carrier Sense Multiple Access with Collision Avoidance Networks," 1999. Submitted through CWC for a US patent.

[21] B. Bensaou, Y. Wang, and C. C. Ko, "Fair Medium Access in 802.11 Based Wireless Ad-Hoc Networks," in *First Annual IEEE & ACM International Workshop on Mobile Ad Hoc Networking and Computing*, Aug. 2000.

[22] 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 Trans. on Communications*, vol. 23, no. 12, pp. 1417–1433, 1975.

[23] N. Abramson, "The ALOHA System - Another Alternative for Computer Communications," in *Proc. of the Fall Joint Computer Conference*, pp. 281–285, 1970.

[24] L. Kleinrock and F. A. Tobagi, "Packet Switching in Radio Channels: Part I - Carrier Sense Multiple-access Modes and Their Throughput-delay Characteristics," *IEEE Trans. on Communications*, vol. 23, no. 12, pp. 1400–1416, 1975.

[25] P. Karn, "MACA - a New Channel Access Method for Packet Radio," in *ARRL/CRRL Amateur Radio 9th Computer Networking Conference*, pp. 134–140, ARRL, 1990.

[26] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, "MACAW: A Media Access Protocol for Wireless LANs," in *Proc. of ACM SIGCOMM '94*, 1994.

[27] C. L. Fullmer and J. J. Garcia-Luna-Aceves, "Floor Acquisition Multiple Access (FAMA) for Packet-Radio Networks," in *Proc. of ACM SIGCOMM '95*, 1995.

[28] C. L. Fullmer and J. J. Garcia-Luna-Aceves, "Solutions to Hidden Terminal Problems in Wireless Networks," in *Proc. of ACM SIGCOMM '97*, 1997.

[29] J. J. Garcia-Luna-Aceves and C. L. Fullmer, "Performance of Floor Acquisition Multiple Access in Ad-Hoc Networks," in *Proc. of 3rd IEEE Symposium on Computers and Communications*, 1998.

[30] C. L. Fullmer, *Collision Avoidance Techniques for Packet-Radio Networks.* PhD thesis, University of California at Santa Cruz, June 1998.

[31] J. J. Garcia-Luna-Aceves and C. L. Fullmer, “Floor Acquisition Multiple Access (FAMA) in Single-channel Wireless Networks,” *ACM/Baltzer Mobile Networks and Applications*, vol. 4, no. 3, pp. 157–174, 1999.

[32] K. Tang and M. Gerla, “Fair Sharing of MAC under TCP in Wireless Ad Hoc Networks,” in *Proceedings of IEEE MMT’99, Venice, Italy*, Oct. 1999.

[33] IEEE Computer Society LAN MAN Standards Committee, ed., *IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.* IEEE Std 802.11-1997, The Institute of Electrical and Electronics Engineers, New York, 1997.

[34] K. C. Chen, “Medium Access Control of Wireless LANs for Mobile Computing,” in *IEEE Networks*, pp. 50–63, 1994.

[35] H. S. Chhaya and S. Gupta, “Throughput and Fairness Properties of Asynchronous Data Transfer Methods in the IEEE 802.11 MAC Protocol,” in *Proc. of PIMRC ’95*, 1995.

[36] G. Bianchi, L. Fratta, and M. Oliveri, “Performance Evaluation and Enhancement of the CSMA/CA MAC Protocol for 802.11 Wireless LANs,” in *Proc. of PIMRC ’96*, pp. 392–396, 1996.

[37] B. P. Crow, I. Widjaja, L. G. Kim, and P. T. Sakai, "IEEE 802.11 Wireless Local Area Networks," *IEEE Communications Magazine*, vol. 35, pp. 116–126, Sept. 1997.

[38] I. Widjaja, B. P. Crow, L. G. Kim, and P. T. Sakai, "Investigation of the IEEE 802.11 Medium Access Control (MAC) Sublayer Functions," *Proc. of IEEE INFOCOM '97*, vol. 1, pp. 126–133, 1997.

[39] B. P. Crow, I. Widjaja, L. G. Kim, and P. T. Sakai, "Performance of IEEE 802.11 Wireless Local Area Networks," *Proc. of SPIE Conference on Digital Backbone and Fast Packet Technologies*, Nov. 1996.

[40] J. Weinmiller, M. Schläger, A. Festag, and A. Wolisz, "Performance Study of Access Control in Wireless LANs – IEEE 802.11 DFWMAC and ETSI RES 10 Hiperlan," *ACM/Baltzer Mobile Networks and Applications*, vol. 2, no. 1, pp. 55–67, 1997.

[41] J. Weinmiller, H. Woesner, and A. Wolisz, "Analyzing and Improving the IEEE 802.11 MAC Protocol for Wireless LANs," *Proc. of MASCOTS '96*, pp. 200–206, Feb. 1996.

[42] S. Khurana, A. Kahol, S. Gupta, and P. Srimani, "Performance Evaluation Of Distributed Co-ordination Function for IEEE 802.11 Wireless LAN Protocol in Presence of Mobile and Hidden Terminals," *Proc. of MASCOTS '99*, pp. 40–47, 1999.

[43] Z. Hadzi-Velkov and L. Gavrilovska, "Influence of Burst Noise Channel and Hidden Terminals over the IEEE 802.11 Wireless LANs," *Proc. of IEEE VTC '99*, vol. 5, pp. 2641–2645, 1999.

[44] Z. Hadzi-Velkov and L. Gavrilovska, "Performance of the IEEE 802.11 Wireless LANs and Influence of Hidden Terminals," in *4th International Conference on Telecommunications in Modern Satellite, Cable and Broadcasting Services*, pp. 102–105, 1999.

[45] Y. C. Tay and K. C. Chua, "A Capacity Analysis for the IEEE 802.11 MAC Protocol." ACM/Baltzer Wireless Networks, to appear.

[46] H. S. Chhaya, "Performance Evaluation of the IEEE 802.11 MAC protocol for Wireless LANs," Master's thesis, Illinois Institute of Technology, May 1996.

[47] T. Ozugur, M. Naghshineh, P. Kermani, C. M. Olsen, B. Rezvani, and J. A. Copeland, "Balanced Media Access Methods for Wireless Networks," in *Proc. of ACM/IEEE MOBICOM '98*, pp. 21–32, Oct. 1998.

[48] T. Ozugur, M. Naghshineh, P. Kermani, and J. A. Copeland, "Fair Media Access for Wireless LANs," in *Proc. of IEEE GLOBECOM '99*, Dec. 1999.

[49] N. H. Vaidya and P. Bahl, "Fair Scheduling in Broadcast Environments," Tech. Rep. MSR-TR-99-61, Microsoft Research, Dec. 1999.

[50] J. L. Sobrinho and A. S. Krishnakumar, "Real-time Traffic Over the IEEE 802.11 Medium Access Control Layer," *Bell Labs Technical Journal*, vol. 1, no. 2, pp. 172–187, 1996.

[51] J. L. Sobrinho and A. S. Krishnakumar, "Quality-of-Service in Ad Hoc Carrier Sense Multiple Access Wireless Networks," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 8, pp. 1353–1368, 1999.

[52] A. Muir and J. J. Garcia-Luna-Aceves, "Supporting Real-time Multimedia Traffic in a Wireless LAN," in *Proc. SPIE Multimedia Computing and Networking*, pp. 41–54, 1997.

[53] ETSI Secretariat, ed., *Radio Equipment and Systems (RES); HIgh PErformance Radio Local Area Network (HIPERLAN) Type 1; Functional Specification.* ETS 350 652 Edition 1, European Telecommunications Systems Institute, Oct. 1996.

[54] ETSI Secretariat, ed., *Radio Equipment and Systems (RES); HIgh PErformance Radio Local Area Network (HIPERLAN) Type 1; Functional Specification.* ETS 350 652/A1 Edition 1, European Telecommunications Systems Institute, May 1997.

[55] ETSI Secretariat, ed., *Radio Equipment and Systems (RES); HIgh PErformance Radio Local Area Network (HIPERLAN) Type 1; Functional Specification.* ETS 350 652/A2 Edition 1, European Telecommunications Systems Institute, June 1998.

[56] ETSI Secretariat, ed., *Broadband Radio Access Networks (BRAN); HIgh PErformance Radio Local Area Network (HIPERLAN) Type 1; Functional Speci-*

fication. ETS 350 652 Version 1.2.1, European Telecommunications Systems Institute, July 1998.

[57] I. Vukovic, "HIPERLAN Type 1: Performance Analysis of the Channel Access Control Protocol," in *Proc. of IEEE VTC '98*, pp. 377–381, 1998.

[58] I. Vukovic, "Analysis of the Hidden Node Problem in HIPERLAN," in *Proc. of IEEE ICC '99*, pp. 1234–1238, 1999.

[59] C. Apostolas, R. Tafazolli, and B. Evans, "Comparison Between Elimination Yield Non Pre-emptive Priority Multiple Access (EY-NPMA) and Dynamic TDMA (D-TDMA)," in *Proc. of PIMRC '96*, pp. 663–667, 1996.

[60] "OPNET Technologies, Inc.." `http://www.opnet.com/`.

[61] F. A. Tobagi and L. Kleinrock, "The Effect of Acknowledgment Traffic on the Capacity of Packet-Switched Radio Channels," *IEEE Trans. on Communications*, vol. 26, no. 6, pp. 815–826, 1978.

[62] A. K. Parekh and R. G. Gallager, "A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: the Single Node Case," *IEEE/ACM Trans. on Networking*, vol. 1, pp. 344–357, 1993.

Appendix A

PriMA Protocol Specification

This appendix specifies PriMA protocol in pseudo-code, which closely matches OPNET Modeler/Radio's Proto-C programming paradigm.

A.1 Variable Definitions

CS Carrier Sensed

T_d Maximum one-hop channel propagation delay

T_{type} Time to transmit a packet of type *type* where *type* can be either RTS, CTS, DATA or ACK

T_{delay} Allowed delay time before a reply to an RTS packet is sent. This is a value carried in the RTS packet's header.

T_{left} Time left for a $rtPacket$ to be delivered; when T_{left} is less than a threshold, the corresponding $rtPacket$ will be discarded.

T_{max} Maximum time to complete one successful RTS-CTS-DATA-ACK transmission

T_{access} Time required to sense the channel idle before transmitting an RTS packet

T_{defer} : The longest time that a station should defer access to the channel when "detecting" collisions.

T_{unit} : Time used as a scaling factor to map packet delay requirement to backoff time

N_{lost} Number of packets discarded during a session

N_{sent} Number of packets sent during a session

PLR Packet loss ratio (QoS requirement), obtained from upper layers (e.g., session management layer).

$rtPacket$ Data packet with QoS parameters specified

$nrtPacket$ Data packet without any QoS parameters specified

A.2 Requirements and Assumptions

- $T_{rts} > 2 * T_d$
- $T_{cts} > T_{rts} + 2 * T_d$
- $T_{max} = T_{rts} + 2 * T_{rts} + T_{cts} + T_{data} + T_{ack} + 4 * T_d$
- $T_{defer} = T_{data} + 3 * T_d$
- $Maxtimer = 800$
- $T_{unit} = 20 * T_d$

A.3 Algorithm

```
proc Init()
{
    Timer ← $T_{defer}$
    Set PersistentWait Flag
    Reason = INIT
    call Dispatch()
}

proc Dispatch()
{
```

```
    while (No event and no local packet) Wait
    if (Packet received) call ProcPacket()
    if (CS) call ProcChannel()
    if (Timeout) call ProcTimeout()
    if (Local packet) Enqueue packet
    if (No event and queue not empty)
        call SchedAccess()
    call Dispatch()
}

proc SchedAccess()
{
    if (nrtPacket or rtPacket with $T_{left} > 1$ sec)
        $T_{access} = 2 * T_d$
    else {
        if ($T_{left} > 0$)
            $T_{access} = T_{left} - (N_{lost} - PLR * N_{sent}) * T_{max}$
        if ($T_{access} < 0$) $T_{access} = 0$
        else $T_{access} = 2 * T_d / |\log_2(T_{access}/2)|$
    }
    Timer ← $T_{access}$
    Reason = ACCESS
    call Dispatch()
}

proc SchedBackoff()
{
    if (nrtPacket or rtPacket with $T_{left}/T_{unit} > Maxtimer$)
        $T_{backoff} = U(0, Maxtimer) * 2T_d$
    else {
        $T_{backoff} = U(0, T_{left}/T_{unit}) * 2T_d$
        }
    }
    Timer ← $T_{backoff}$
    Reason = BACKOFF
    call Dispatch()
}

proc ProcPacket()
{
    switch (Packet type) {
    case RTS:
        if (PersistentWait flag set) Restore previous timer
        else Timer ← min($T_{delay}$ in this RTS, time left to send
            pending CTS (if any) by either local or other stations)
```

```
    case CTS:
        if (destID == localID) {
            Send data packet
            Timer ← $2 * T_d$
            Reason = WAITREPLY
        } else {
            Timer ← $T_{defer}$
            Reason = WAITOTHERS
            Set PersistentWait flag
        }
    case DATA:
        if (destID == localID) {
            Pass packet to upper layer
            Send ACK packet
            Timer ← $2 * T_d$
            Reason = WAITREPLY
        } else {
            Timer ← $T_{ack} + 2 * T_d$
            Reason = WAITOTHERS
            Set PersistentWait flag
        }
    case ACK:
        if (destID == localID) Remove local packet from queue
    }
    call Dispatch()
}

proc ProcChannel()
{
    Wait channel to become idle
    Timer ← $T_{defer}$
    Reason = WAITOTHERS
    Set PersistentWait flag
    call Dispatch()
}

proc ProcTimeout()
{
    switch (Timeout reason) {
    case INIT:
        Clear PersistentWait flag
    case ACCESS:
        if (nrtPacket) $T_{delay} = 2 * T_{rts}$
        else {
            if ($T_{left}/T_{max} > 2$) $T_{delay} = 2 * T_{rts}$
```

```
            else $T_{delay} = (T_{left}/T_{max}) * T_{rts}$
        }
        Set $T_{delay}$ in RTS packet and send it out
        Timer $\leftarrow T_{delay} + 2 * T_d$
        Reason = WAITREPLY
    case WAITREPLY:
        if (Wait for CTS or ACK packet)
            call SchedBackoff()
    case PENDINGCTS:
        if (my turn to send CTS packet) {
            Send CTS packet
            Timer $\leftarrow 2 * T_d$
            Reason = WAITREPLY
        } else {
            Timer $\leftarrow T_{cts} + 2 * T_d$
            Reason = WAITOTHERS
            Set PersistentWait flag
        }
    case WAITOTHERS:
        Clear PersistentWait flag
        if (Not queue empty) call SchedBackoff()
    }
    call Dispatch()
}
```