PRIORITY AND FAIR SCHEDULING IN WIRELESS LOCAL AREA NETWORKS

A Thesis

by

ANURAG DUGAR

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

August 2001

Major Subject: Computer Engineering

PRIORITY AND FAIR SCHEDULING IN WIRELESS LOCAL AREA NETWORKS

A Thesis

by

ANURAG DUGAR

Submitted to Texas A&M University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Approved as to style and content by:

Nitin H. Vaidya
(Co-Chair of Committee)

A. L. Narasimha Reddy
(Co-Chair of Committee)

Gwan Seung Choi
(Member)

Chanan Singh
(Head of Department)

August 2001

Major Subject: Computer Engineering

ABSTRACT

Priority and Fair Scheduling in Wireless Local Area Networks. (August 2001)

Anurag Dugar, B.E., Karnataka Regional Engineering College, India

Co–Chairs of Advisory Committee: Dr. Nitin H. Vaidya
Dr. A. L. Narasimha Reddy

In recent years, fair scheduling and quality of service (QoS) in Wireless Local Area Networks have received significant attention from the networking research community. This thesis presents a distributed Medium Access Control (MAC) protocol for priority and fair scheduling in a Wireless Local Area Network. The proposed protocol supports multiple priorities, as well as a mechanism (using weights) for controlling the way bandwidth is shared by flows within a given priority level.

To Nani, Mummy and Papa

ACKNOWLEDGMENTS

I would like to thank my advisor and co-chair Dr. Vaidya for all his guidance, encouragement and valuable comments. His enthusiasm towards research and teaching has been a great source of inspiration. He has been a great mentor and has always been there to offer me advice regarding academics and career. Without his help and support, my graduate school experience would not have been so fulfilling.

I would like to thank my co-chair Dr. Reddy for all his guidance and time. He has always been very supportive and helpful.

I would like to thank Dr. Choi for being part of my advisory committee.

I would like to thank the members of the Mobile Computing and Networking group for the numerous discussions we had in different areas of Wireless and Mobile ad hoc Networks. Being a part of this group has greatly enhanced my knowledge and has opened up my mind to various research possibilities in the area of Wireless and Mobile ad hoc Networks.

I would like to thank NSF and Microsoft Research for supporting me financially to carry out research.

I would like to thank Tammy Carda for all her help and support during my stay at Texas A&M.

I would like to thank my parents and my sisters for their constant support and encouragement. Their belief in my abilities has always inspired me to do my best.

TABLE OF CONTENTS

CHAPTER Page

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Fairness is an important issue when accessing a shared wireless channel. With *fair scheduling*, different flows wishing to share the wireless channel can be allocated bandwidth in proportion of their “weights”. Also, with the increasing use of wireless local area networks (LANs) , there is a need for supporting multiple priority levels and Quality of Service (QoS) in the wireless LANs. The scheme proposed in this thesis tries to address this need by supporting multiple priority levels along with weighted fair sharing of a wireless channel. The proposed scheme is fully distributed in nature and is implemented without using a central coordinator to arbitrate medium access.

This thesis is organized as follows. Chapter 2 describes the existing wireless LAN standards, QoS in Wireless LANs, related work and the motivation behind the proposed scheme. Chapter 3 describes the proposed scheme in detail. Chapter 4 describes some optimizations in the proposed scheme to improve the performance. Chapter 5 presents the simulation model and performance evaluation. Chapter 6 presents an alternative considered for the proposed scheme and discusses its merits and demerits viz a viz the proposed scheme. Lastly, Chapter 7 discusses the conclusions and the scope for future work.

The journal model is *IEEE Transactions on Automatic Control.*

CHAPTER II

BACKGROUND

A. Wireless Local Area Network Standards

Wireless networks have gained widespread acceptance in recent years. Efforts have been going on to standardize the Physical layer and the Medium Access Control (MAC) layer for Wireless LANs in order to enable multi-vendor inter-operability and facilitate the availability of commercial products based on these standards. Currently there are two existing standards for Wireless Local Area networks (Wireless LANs):

- IEEE 802.11: Defined by the IEEE 802 LAN working group.
- HIPERLAN: Defined by the European Telecommunication Standards Institute (ETSI)

1. IEEE 802.11 Access Mechanism

There are two access schemes used in IEEE 802.11[11]:

- Distributed Coordination Function (DCF)
- Point Coordination Function (PCF)

DCF uses a contention based access mechanism. In DCF, the medium access is controlled by using a CSMA/CA scheme called DFWMAC (Distributed Foundation Wireless MAC). A station, that intends to transmit, first senses the medium. If the medium is found to be idle for a time period of *distributed inter frame spacing* (DIFS) duration, the station proceeds with its transmission. However, if the station senses the medium as busy, it waits till the end of the ongoing transmission, then waits for the medium to be idle for DIFS duration, and then randomly selects a backoff interval from its contention window. While the medium is idle, this backoff interval is decremented after each slot time[1]. If the medium becomes busy while the backoff interval is non-zero, then the backoff interval is frozen until the medium is found to be idle again for DIFS duration. Eventually, when the backoff interval reaches zero, this station can start its own transmission. This access mechanism can be extended by the RTS/CTS (Ready to send/Clear to send) message exchange in order to guarantee collision free transmission of the data packets.

PCF, on the other hand, uses a contention free access mechanism. In PCF, the medium access is controlled by a central coordinator. The central coordinator polls the stations that are on its polling list and allows them contention free access to the medium.

2. Hiperlan Access Mechanism

In Hiperlan[5], the channel access mechanism is based on channel sensing and a contention resolution scheme called Elimination Yield - Non-preemptive Priority Multiple Access (EY-NPMA). A station seeking access to the channel, listens to the channel for a certain time period (1700 bit-periods). If the channel is found to be idle during this period, then the station is allowed to start transmission of the data frame. On the other hand, if the channel is found to be busy during this period, then the station waits till the end of the ongoing transmission. Synchronization is performed at the end of the current transmission interval and a channel access cycle is initiated by all the stations seeking access. The channel access cycle is split into 3 phases: Priority resolution, Elimination and the Yield phase.

The aim of the Priority resolution phase is to allow only the highest priority

[1]Slot time is a fixed duration of time defined in the IEEE 802.11 standard.

stations, among the contending ones, to proceed to the next phase. In Hiperlan, a user priority level and a lifetime is assigned to each packet. The actual channel priority level for a packet is determined by its user priority level and its residual lifetime. The lower the priority level, the higher is the priority. The Priority resolution phase consists of a number of slots. In this phase, each station that has a packet with a channel access priority level m, senses the channel for the first m slots. If the channel is idle during this interval, then the node transmits a burst in the m+1-th slot and moves onto the next phase, else it drops out of contention and waits for the next channel access cycle. In the Elimination phase, every surviving station transmits a burst of a random length, bounded and defined by a certain discrete probability distribution. After this the station listens to the channel for an Elimination Survival verification Period (ESVP). If another station sends a burst of a longer duration, i.e. the station senses a transmission during ESVP, the station drops out of contention and waits for the next channel access cycle, else it moves onto the Yield phase. In the Yield phase, every station listens to the channel for a Yield period of random duration. If a station hears a transmission while it is listening, it drops out of contention and waits for the next channel access cycle, else it transmits the data packet immediately after the Yield period.

B. Priority and Fair Scheduling in Wireless LANs

1. Existing Standards

Fairness and Quality of Service (QoS) in wireless LANs have received significant attention from the networking research community in recent years. Among the existing standards, IEEE 802.11 can support priority and fair scheduling using the Point Coordination Function (PCF). In PCF, a centralized controller has control over the network resources and can allocate them to the users as per the QoS agreements. But PCF has certain inherent disadvantages because of its centralized nature. In a centralized approach, if the coordinator fails, then no communication can take place among the nodes until the failure of the coordinator is detected and a new coordinator is elected. Also, if a node is not able to communicate with a coordinator, then it would be isolated from all other nodes in the wireless LAN. Distributed MAC protocols are better suited for Wireless LANs wherein the nodes maybe battery-powered and more susceptible to failure. Though the Distributed Coordination Function (DCF) in the IEEE 802.11 standard can arbitrate access to the channel in a distributed manner, it does not have any support for priority or fair scheduling. Hiperlan [5], which is a wireless LAN standard defined by ETSI, uses a distributed scheme to support priority based scheduling and time bounded services. But it does not support a weighted fair allocation of the channel bandwidth to the users belonging to a particular priority level.

2. Related Work

Much research has been performed on *fair queuing* algorithms for achieving a *fair* allocation of bandwidth on a shared link (for e.g. [6, 9, 10, 13, 14, 16]) There has been some work on achieving fairness using distributed MAC protocols for wireless networks (for e.g. [7, 18, 21, 22]). One recent related work [1] proposed a protocol to achieve weighted fairness where the nodes having different weights are allocated bandwidth in proportion to their weights. However, this protocol can exhibit short-term unfairness for some nodes when their transmissions collide. There has also been work on distributed protocols that take *priorities* into account when performing medium access control and can support real-time applications [4, 8, 15, 12]. Some interesting work on distributed scheduling algorithms for QoS support and real-time traffic on a wireless LAN has also been performed [3, 5, 17]. But none of these schemes supports multiple priorities along with weighted fairness.

C. Motivation

Fairness is an important issue when accessing a shared wireless channel. With fair scheduling, different flows wishing to share the wireless channel can be allocated bandwidth in proportion of their *weights*. Also, with the increasing use of wireless LANs, there is a need for supporting multiple priorities in order to rpovide a variable quality of service to the users in a wireless LAN. In the past, various schemes for wireless LANs have been proposed that focus on either fair scheduling or priority scheduling, but not both. This thesis tries to address both of these issues by presenting an innovative scheme to support multiple priority levels along with weighted fair sharing of a wireless channel. The proposed scheme [19, 20] borrows some of the ideas from Hiperlan [5] and Distributed Fair Scheduling scheme proposed in [1].

CHAPTER III

PROPOSED SCHEME

The channel access mechanism in the proposed scheme [19, 20] is based on carrier sensing and a contention resolution scheme. The contention resolution scheme proposed in this paper borrows some of the ideas from HIPERLAN [5] and Distributed Fair Scheduling (DFS) scheme proposed in a recent related work[1].

A. Basic Idea Behind the Proposed Scheme

The objective of the proposed scheme is to support multiple priorities in a wireless LAN, as well as to provide a mechanism (using weights) for controlling the way bandwidth is shared by flows within a given priority level. In order to achieve this objective, the contention resolution for the shared channel is broken up into a number of phases. Every node in the wireless LAN is assigned a priority and a weight. Each backlogged node chooses a backoff interval using a technique which attemps to emulate Self-Clocked Fair Queuing (SCFQ) [24]. The backoff interval is chosen so that it is proportional to the finish tag of the backlogged packet. Each backlogged node in the system then senses the channel. If the channel is found to be idle for a certain predefined duration, then the backlogged nodes initiate a contention resolution cycle. During each phase in the contention resolution cycle, a backlogged node senses the channel for a certain *waiting time*. If the backlogged node finds the channel to be idle during this *waiting time*, then that node transmits a burst (or RTS/DATA in case of the last phase) to assert itself as the winner of that phase and moves on to the next phase. All other backlogged nodes who listen to a burst during their waiting time, drop out of contention and wait for the next contention resolution cycle. During the first phase in the contention resolution cycle, the backlogged nodes try to resolve the priorities in a distributed manner. In this phase, the waiting time for each backlogged node is chosen to be inversely proportional to its priority - the higher the priority, the shorter is the waiting time. At the end of this phase, all the nodes having the highest priority among all the backlogged nodes would move on to the next phase. These nodes then try to select a node among themselves having a packet with the minimum finish tag in a distributed manner. The waiting time for each backlogged node in this phase is proportional to its backoff interval which inturn is proportional to its finish tag - the smaller the finish tag, the shorter is the waiting time. The node(s) having a packet with the minimum finish tag would emerge as the winner(s) and initiate a transmission cycle by sending a RTS (or a DATA) packet. In the proposed scheme we present an innovative technique to compress the *waiting time* during the phase following the priority resolution phase (i.e. the time spent in counting down the backoff interval). We also present an innovative contention resolution scheme that ensures that the colliding nodes within a given priority level get access priority over all other backlogged nodes within the same priority level.

B. Detailed Description of the Proposed Scheme

Every backlogged node in the system first senses the channel. If the channel is found to be idle for an "inter-round spacing" (*irs*), say, M slots, then the backlogged nodes initiate a contention resolution cycle. In order to ensure that a newly backlogged node does not enter an ongoing contention resolution cycle, the *irs* duration is chosen to be greater than the maximum possible idle time (or the *waiting time*) during a contention resolution cycle.

In the proposed approach, every node in the wireless LAN is assigned a priority and a weight. Every backlogged node first chooses a backoff interval and then determines a tuple of the form (p, c, n, $d_{(n-1)}$,..., d_0) which is used for contention resolution. These are described in more detail in the following subsections.

1. Choosing a Backoff Interval

This scheme borrows the idea of picking a suitable backoff interval from DFS [1]. The essential idea is to pick a backoff interval proportional to the finish tag of the packet. This is achieved by picking the backoff interval as a function of length of the packet to be transmitted, and weight of the flow to which the packet belongs. The proposed technique attempts to emulate the Self-Clocked Fair Queuing (SCFQ) [24] in a distributed manner so as to transmit the packet with the minimum finish tag first and achieve a weighted fair allocation of bandwidth. Specifically, a backlogged node i picks a backoff interval B_i for its k-th packet, P_i^k, as a function of its weight, ϕ_i, and packet length L_i^k, as follows :

$$B_i = \left\lfloor Scaling\_Factor * \frac{L_i^k}{\phi_i} \right\rfloor \tag{3.1}$$

An appropriate choice of the *Scaling_Factor* allows us to choose a suitable scale for the backoff interval. To reduce the possibility of collisions the B_i value chosen above is randomized as follows:

$$B_i = \lfloor \rho * B_i \rfloor \tag{3.2}$$

where ρ is a random variable uniformly distributed in the range [0.9,1.1].

2. Contention Resolution

Once a backlogged node has chosen a backoff interval, it breaks it down into a base-N representation. For example, if a node had chosen a backoff interval equal to 33 (in decimal or base-10 notation) and the base was chosen to be 2, then that node would break the backoff interval into its base-2 representation as 100001. The backlogged node then constructs a tuple of the form (p,c,n,$d_{(n-1)}$,...,d_0). The elements of the tuple are assigned the following meaning:

- *p*: This is the most significant element in the tuple and denotes priority of the node. With an inter-round spacing *irs* of M slots, our approach can support up to M priority levels ranging from 0 to M-1. The lower the priority level, the higher is the priority.

- *c*: This element represents *collision status* of a node. Every node i maintains a collision counter $ccntr_i$. This counter is incremented on every collision suffered by a node and is reset to 0 after every successful transmission attempt by the node. If the collision counter for a node is greater than 0, then c is set to zero; else it is set to 1.

- *n*: This element denotes the number of digits in the backoff interval's representation in base-N. For the above mentioned example, the number of digits in the backoff interval, i.e. n, would be equal to 6.

- d_i: denotes the i-th digit (0-th digit being the least significant) in the base-N representation of the chosen backoff interval.

Whenever a backlogged node finds the channel to be idle for *irs* (i.e. inter-round spacing) duration, it initiates a contention resolution by transmitting a burst[1]. It then uses its tuple to contend for the channel during the contention resolution cycle. Each element in this tuple signifies a phase in the contention resolution cycle. Only those nodes who win the first phase, get to go onto the second phase and so on until the

[1]Burst is a control frame which is transmitted just to occupy the channel so that the other nodes listening to the channel can sense the channel as busy.

last phase, after which we will have potential winner(s) of the contention resolution cycle. During the i-th phase, the i-th digit, say t_i, in the tuple (p,c,n,$d_{(n-1)}$,...,d_0) is used as follows. In the i-th phase, a node listens to the channel for t_i slots. If it hears a transmission while it is listening, it drops out of contention. Otherwise, it transmits a *burst* after listening for t_i slots and goes into the next phase. All the nodes which make a transition from i-th phase to the $(i+1)$-th phase have identical most significant i elements in their tuples.

3. Transmission and Recalculation of Backoff Interval

The winner of the contention resolution cycle, node i, transmits a Request-to-Send (RTS) packet (similar to the Distributed Coordination Function in the IEEE 802.11 standard). If the receiver is willing to accept the data packet, it responds by sending a Clear-to-Send (CTS) packet on receipt of the RTS. When the sender gets the CTS, it sends the data packet and piggybacks its backoff interval B_i and its priority level on the data packet. A node j belonging to the same priority level as node i and which had contended with node i, recalculates its backoff interval on listening to the transmission as follows:

$$\begin{aligned} B_j &= B_j - B_i \qquad && if B_j \geq B_i \\ &= 0 && otherwise \end{aligned} \tag{3.3}$$

Node j then reconstructs the tuple on the basis of this new, smaller backoff interval. Now, when node j finds the channel idle for *irs* period, it initiates a new contention resolution cycle and uses the reconstructed tuple during this cycle. This procedure is designed to ensure that there is a fair sharing of bandwidth among flows within each priority class.

4. Collision Resolution

Every node i maintains a collision counter, $ccntr_i$. This counter, $ccntr_i$, is incremented by 1 on every collision suffered by node i and is reset to 0 after every successful transmission by node i. Now if two or more contending nodes belonging to the winning priority level had chosen the least backoff interval, they all would emerge as winners of the contention stage and would end up in collision. Since these colliding nodes were winners of the contention stage, they should get priority access over all other nodes within the same priority level. This is accomplished in the following manner: suppose Node i sends a RTS and doesn't get back a CTS within the timeout duration. It then assumes that its RTS collided with a transmission from some other node. It then takes the following actions:

- Node i increments $ccntr_i$ by 1.
- Node i chooses a new B_i uniformly distributed in $[1, 2^{ccntr_i - 1} * K]$, where K is a constant parameter.
- Node i constructs a new tuple on the basis of this new B_i
- Node i waits for the channel to be idle for inter-round spacing (*irs*) period and then initiates a contention resolution cycle using the new tuple.
- When node i gets to transmit, it piggybacks its original backoff interval and not the new backoff interval that it had chosen after suffering a collision. Also, when a node with a non-zero collision counter has to recalculate its backoff interval on hearing a Data transmission, it subtracts the piggybacked backoff interval from its original backoff interval and not the new backoff interval. Since the new backoff interval, which is used for contention resolution, is not modified, such nodes do not have to reconstruct the tuple after recalculation. The recalculated

backoff interval is then piggybacked whenever such nodes get to transmit the Data packet.

Let us assume that node i's priority class was the highest during the next contention resolution cycle. Now all the backlogged nodes within this class would check their *collision status*. Since node i has a non-zero collision counter, its *collision status* would be zero (as already explained while defining the elements of the tuple). Hence, node i would transmit a burst without waiting for any slot and then move onto the next element in the tuple. But for all other backlogged nodes within the same class who have not suffered a collision, *collision status* would be set to 1. Hence, all these nodes would have to wait for one slot before they can transmit a burst and move onto the next element in the tuple. While these nodes would be listening to the channel for one slot, they would hear the burst transmitted by node i and would subsequently drop out of contention. Hence node i would get access priority over all these nodes.

CHAPTER IV

SELECTION OF PARAMETERS AND PERFORMANCE OPTIMIZATION

A. Selecting the Optimum Base

Recall that, in the scheme proposed in Chapter 3, the backoff interval is broken down into its base-N representation in order to construct a tuple to be used for contention resolution. In the proposed scheme, there are 2 causes for overhead : overhead involved in resolving the contention among flows, and the RTS-CTS overhead to ensure collision free transmission of data. The overhead involved in resolving the contention among flows is determined by the inter-round spacing *irs*, the tuple length and the tuple elements, which are, in turn, directly related to the base chosen for breaking down the backoff interval. Let n_{max} be the number of digits in the base-N representation of the maximum possible value of the backoff interval. The maximum possible idle time, T_{max}, during a contention resolution cycle is then given by maximum(N-1, n_{max}). If N is greater than n_{max}, then T_{max} is equal to N-1 slots, else it is equal to n_{max} slots.

Now, in order to ensure that a newly backlogged node does not enter an ongoing contention resolution cycle, the *irs* (i.e. inter-round spacing) duration is chosen to be greater than T_{max} (i.e. *irs* is equal to N slots if N is greater than n_{max}, else it is equal to n_{max}+1 slots) . Also, the tuple length (related to the number of phases in the contention resolution cycle), and the average duration of each phase is governed by the base chosen. Our goal was to select an optimum base in order to minimize the overhead, and hence maximize the aggregate throughput. Figure 1 presents the simulation results for the aggregate throughput achieved as a function of the chosen base. The number of nodes was 32 and the *Scaling_Factor* was 0.02 in these simulations.

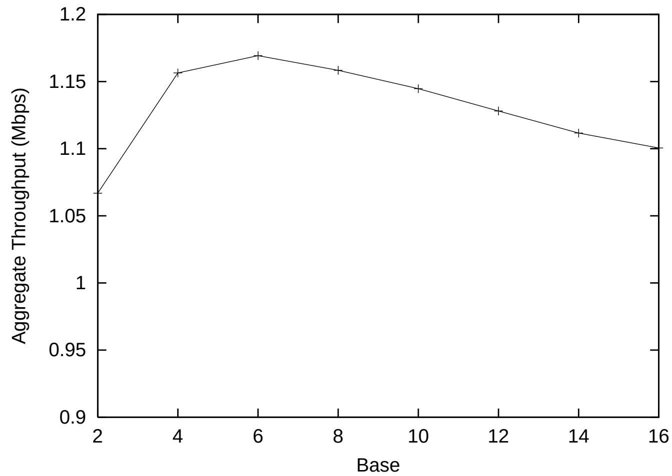


Fig. 1. Aggregate Throughput vs Base Value.

As can be seen from the results, the performance degrades when the base is chosen to be either too small or too large. When the base is chosen to be too small, it results in shorter average phase durations. However, it drives up the tuple length (i.e. the number of phases during a contention resolution cycle) and n_{max} (which in turn drives up the *irs* duration). On the other hand, if the base is chosen to be too large, it results in shorter tuple length, and hence fewer phases during each contention resolution cycle. However, it drives up the average duration of each phase and the *irs* duration. For the chosen parameters, base-6 was found to be optimum and was used for all other results presented in this paper.

B. Optimizing the Performance

The contention resolution mechanism in the proposed scheme was slightly modified in order to reduce the overhead and optimize the performance. We do away with the collision status (second element in the tuple) and assign access priorities to the colliding nodes in a more efficient manner. As discussed earlier, every node i maintains a collision counter $ccntr_i$ which is incremented on every collision suffered by a node and is reset to 0 after every successful transmission attempt by the node. When the nodes reach the second phase of contention resolution (i.e. after priority resolution), they check their collision counter. If the collision counter for a node is greater than 0, then it transmits a burst and then moves on to the next element in the tuple, else it moves on to the next element in the tuple without transmitting a burst. Say node i had suffered a collision and was the highest priority node during the next contention resolution cycle. Since node i has a non-zero collision counter, it would transmit a burst and move on the next element in the tuple. But all other backlogged nodes within the same class would move on to the next element (n, i.e the number of digits in their backoff interval, which is guaranteed to be non-zero) without transmitting a burst. While these nodes would be listening to the channel for n slots, they would hear the burst transmitted by node i and would subsequently drop out of contention. Hence node i would get access priority over all these nodes.

CHAPTER V

SIMULATION MODEL AND PERFORMANCE EVALUATION

In this chapter, we present performance evaluation results for the proposed scheme. The scheme was implemented using ns-2 simulator [2]. The ns-2 simulator includes a module to simulate the Distributed Coordination Function (DCF) function in IEEE 802.11. We modified this module to simulate the proposed scheme. The backoff interval was bound to a maximum value of 8192. The channel bandwidth was taken to be 2 Mbps.

We compare the proposed scheme with the Distributed Coordination Function (DCF) in the IEEE 802.11 standard [11] and the Distributed Fair Scheduling (DFS) scheme proposed in [1] . In our simulation model, if we have a LAN with n nodes, we set up $n/2$ flows (n is always chosen to be an even number) – flow i is set up from node $2i$ to node $2i+1$ (the nodes are numbered 0 through $n-1$). The choice of the destination nodes for the flows is arbitrary, and any destination could have been chosen for each flow without affecting the results.

A. Fair Scheduling

The figures presented in the following subsections consider the case where the $n/2$ flows (in case of a LAN with n nodes) have identical weights and priorities – the chosen weight for each flow is $2/n$ (this choice is arbitrary, and similar results hold for other choices too) and the priority of each node is P0. All the flows are generating traffic at the same rate and are always backlogged. Also, the aggregate demand by the flows exceeds the channel capacity. Later, in section B, we consider the case when there are multiple priority flows present in the LAN.

1. Comparison of Proposed Scheme with IEEE 802.11 Distributed Coordination Function (DCF)

In the following figures we compare the aggregate throughput and the fairness index achieved by the proposed scheme and 802.11 as a function of the number of nodes on the wireless LAN . The *Scaling_Factor* for the proposed scheme was chosen to be 0.02 in these simulations.

- Fairness Index

  For environments where all flows are always backlogged, we evaluate a *fairness index* [23] as follows, where T_f denotes throughput of flow f, and ϕ_f denotes weight of flow f.

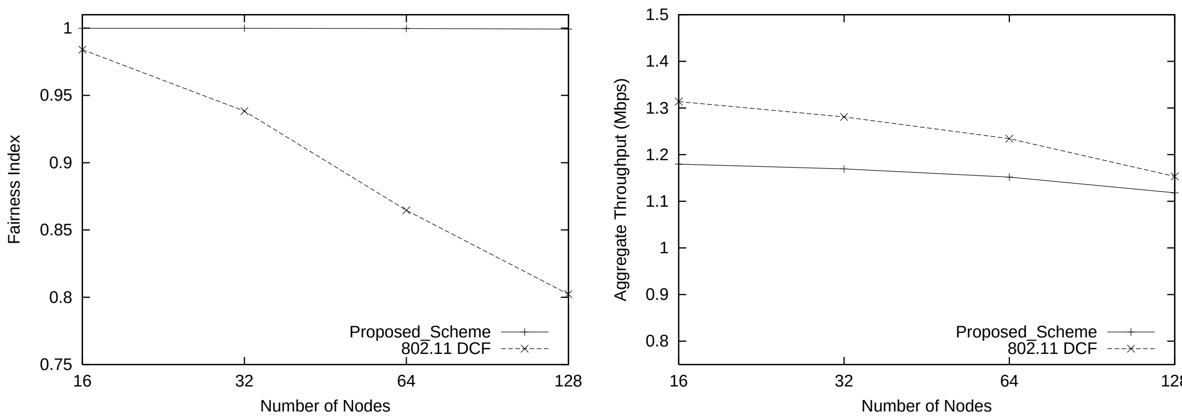
$$\text{Fairness index} = \frac{\left(\sum_f T_f/\phi_f\right)^2}{\text{number of flows} * \sum_f (T_f/\phi_f)^2}$$

  The higher the value of the fairness index, the more is the fairness.

  As can be seen from Figure 2(a), the fairness index achieved by IEEE 802.11 DCF degrades as the number of nodes on the wireless LAN is increased. This is because of the fact that as the number of nodes, or the load, on the wireless LAN is increased, there is an increased likelihood of collisions. Since the colliding nodes were winners of the contention over other nodes, they should get prior access over other nodes after suffering a collision. However, in case of IEEE 802.11 DCF, the colliding nodes invoke binary exponential backoff (i.e. they pick up a larger backoff interval after suffering a collision) and hence do not get prior access over other nodes. This results in unfairness towards the colliding nodes. On the other hand, in the proposed scheme, the colliding nodes get prior access over all other backlogged nodes within the same priority class. Hence, the proposed scheme is able to achieve a consistently high fairness index (almost close to 1) even when the number of nodes scales up.

- Aggregate Throughput

  As can be seen from Figure 2(b), IEEE 802.11 tends to achieve a higher aggregate throughput – this illustrates the trade-off between throughput and fairness. The proposed scheme improves fairness, but possibly at the cost of lower aggregate throughput (due to longer durations of time spent on channel access protocol).



(a) Fairness Index (b) Aggregate Throughput

Fig. 2. Comparison of IEEE 802.11 and the Proposed Scheme.

- Impact of Transmission errors on Fairness

  In a wireless LAN, a packet may be lost because of either a collision or transmission errors. Wireless channels are prone to errors due to many factors such as attenuation, multipath fading, noise, co-channel interference, and node mobility. Since in a wireless LAN it is difficult for a sender to distinguish between looses due to errors or losses due to collisions , presence of errors can cause difficulty in a fair allocation of the channel bandwidth. In case of IEEE 802.11 DCF, whenever there is a packet loss, the sender assumes that the loss is due to a collison with another transmission. It then invokes an exponential backoff mechanism in order to lower the contention on the channel. As a result, the error-prone nodes are unnecessarily penalized and do not get their expected share of the bandwidth. In the proposed scheme too, any loss is assumed to be due to a collison with another transmission. However, unlike IEEE 802.11 DCF, the proposed scheme ensures that the nodes which have suffered a collision get prior access over other nodes within the same priority class. A node lagging due to wireless errors can reclaim the lost bandwidth by getting prior access over other nodes within the same priority class. Hence, the proposed scheme can achieve a fair allocation of the channel bandwidth even in the presence of transmission errors.

2. Comparison of the Proposed Scheme with DFS

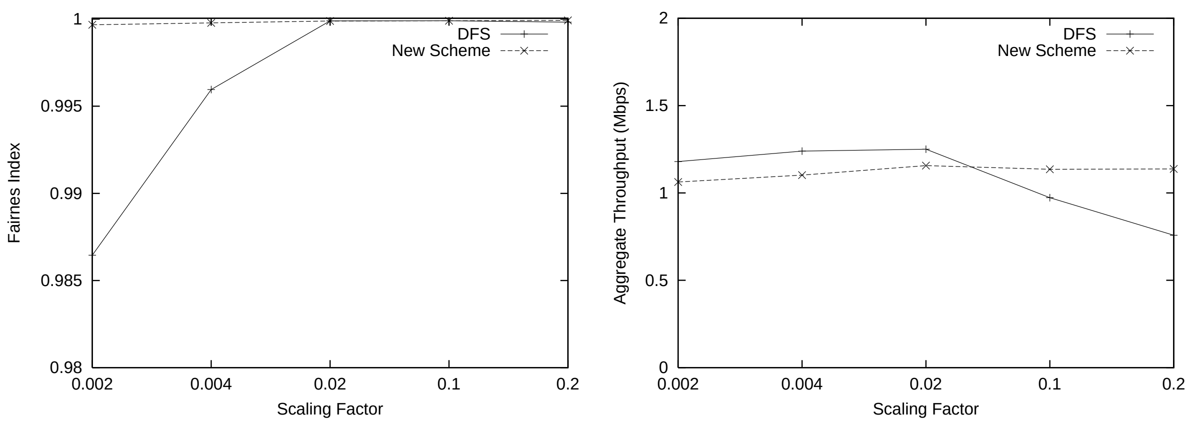
We now compare aggregate throughput and fairness index achieved by the proposed scheme and DFS [1] as a function of the *Scaling_Factor*. The wireless LAN had 32 nodes in these simulations.

- Fairness Index

  As can be seen in Figure 3(a), when the *Scaling_Factor* is very small, the fairness index degrades in case of DFS. In DFS, when the *Scaling_Factor* is very small, there is a high likelihood of collisions which can potentially exhibit unfairness towards the colliding nodes because of access priority reversals. On the other hand, in the proposed scheme, the colliding nodes within a class get access priority over other backlogged nodes within the same class. As a result, the proposed scheme achieves a high fairness index even for very small *Scaling_Factors*. As for higher *Scaling_Factors*, they result in an increase in fairness, and both DFS and the proposed scheme do well in such cases. This figure illustrates that unlike DFS, the proposed scheme is relatively insensitive to the choice of the *Scaling_Factor*.

- Aggregate Throughput

  As can be seen in Figure 3(b), as the *Scaling_Factor* is increased, the aggregate throughput achieved by DFS declines sharply. The nodes in both DFS and the proposed scheme pick backoff interval in the same manner. An increase in the *Scaling_Factor* results in a proportional increase in the backoff interval. However, in DFS, large backoff intervals lead to a greater overhead and reduced aggregate throughput. On the other hand, in the proposed scheme, the aggreagate throughput achieved by the proposed scheme drops relatively slowly as the backoff interval is increased. This implies that in the proposed scheme there is only a slight increase in the overhead with the scaling factor even though it results in large backoff intervals. This is because of the fact that our scheme reduces the time spent in counting down backoff interval by breaking it down into base- N elements and then using these elements during contention resolution cycle. On the other hand, in DFS, the nodes have to countdown the backoff intervals to 0 (as in IEEE 802.11) before they can transmit. This leads to long durations of idle time when the backoff intervals are large in case of DFS. Another observation that can be made from this fact is that the proposed scheme would do much better than DFS in scenarios where different flows have different weights and the only flows that are backlogged have smaller weights. Smaller weights would result in larger backoff intervals in both DFS and the proposed scheme. However, in DFS, this would lead to a much greater increase in the overhead as compared to the proposed scheme.



(a) Fairness Index (b) Aggregate Throughput

Fig. 3. Comparison of DFS and the Proposed Scheme.

As can be seen from Figure 3, the performance of DFS is very sensitive to changes in the scaling factor. Whereas the proposed scheme achieves a fairly good performance over a wide range of scaling factors. When the number of nodes in the LAN is large, we would like to choose a larger scaling factor to reduce the probability of collisions and maintain fairness. At the same time we would like to maximize the aggregate throughput achievable. Under such circumstances, the proposed scheme offers much more flexibilty as compared to DFS in choosing the scaling factor to achieve the desired objectives.

- Effect of Packet Size

Figure 4 displays the effect of the packet size on the aggregate throughput achieved by DFS and the proposed scheme. In the simulations for this result, there were 32 nodes in the wireless LAN and the *Scaling_Factor* was chosen to be 0.1 for both DFS and the proposed scheme. As mentioned earlier, in the proposed scheme, the total overhead consists of contention resolution cycle overhead and the RTS-CTS overhead. In case of DFS, the total overhead consists of the time spent in counting down the backoff interval and the RTS-CTS overhead. As the packet size is increased, the RTS-CTS overhead is amortized over the packet size. Also, larger packets mean lesser contention resolution cycles in the proposed scheme, and hence reduced overhead. However, an increase in the packet size results in a proportional increase in the backoff interval. As already explained, an increase in the backoff interval results in a much greater increase in the overhead in case of DFS as compared to the proposed scheme. Hence, though the performance of both DFS and the proposed scheme improves with an increase in the packet size, the effect is much more pronounced in case of the proposed scheme.

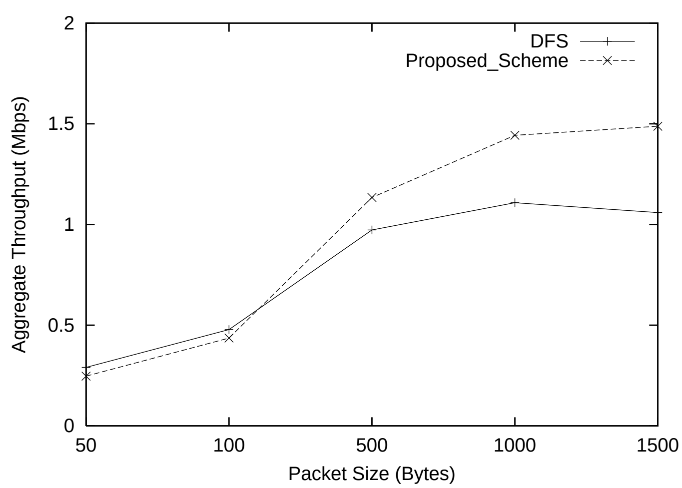


Fig. 4. Effect of Packet Size on DFS and the Proposed Scheme.

B. Priority Scheduling and Weighted Fairness

The simulation model in this case had 20 nodes (i.e., 10 flows) in the wireless LAN. 5 of these flows (0,1,2,3 and 4) were of priority level P0 and the other 5 (5,6,7,8 and 9) were of priority level P1 (P0 being the higher priority). All the high priority flows had a weight 1. The low priority flows had the weights 0.2, 0.4, 0.6, 0.8 and 1 respectively. The 5 high priority flows were generating CBR traffic at the rate of 0.05Mbps, 0.1Mbps, 0.15 Mbps, 0.2Mbps and 0.25Mbps respectively. The rates were chosen so that the aggregate demand by the high priority flows was less than the link capacity. The low priority flows were generating CBR traffic at the rate of 0.5 Mbps each. The packet size was fixed and was identical for all flows in the simulations reported here. As can be seen from the Figure 5, since the aggregate demand by the high priority flows is below the channel capacity, each high priority flow gets a share of the bandwidth equal to its demand irrespective of its weight. The bandwidth leftover by the high priority flows is allocated to the low priority flows in proportion of their weights.

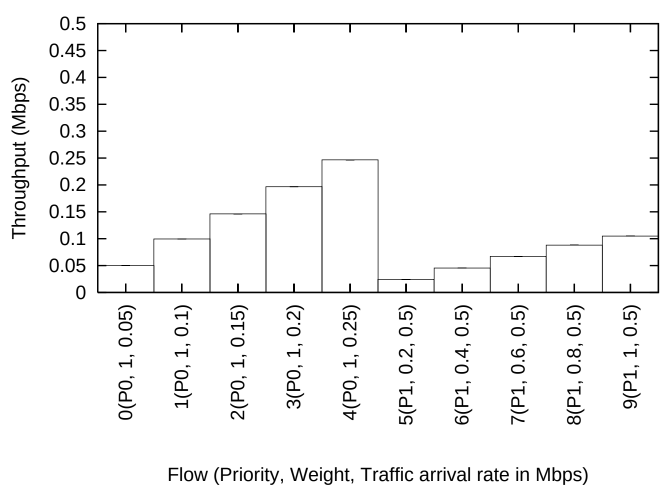


Fig. 5. Priority and Weighted Fair Scheduling by the Proposed Scheme.

However, if the aggregate demand of the high priority flows exceeds the channel capacity then it is desirable that there be a weighted fair allocation of the bandwidth to these high priority flows. The results presented in Figure 6 present the case where there were 10 nodes (i.e. 5 flows) in the wireless LAN. All the flows had the same priority, P0, but different weights. The weights of the flows were 1/2, 1/4, 1/6, 1/8, and 1/10 respectively. Each of these flows was generating CBR traffic at the same rate and the aggregate demand exceeded the channel capacity. The packet size was fixed and was identical for all flows in the simulations reported here. As can be seen from Figure 6, the (throughput/weight) ratio remains more or less constant implying that each flow is getting a bandwidth allocation in proportion to its weight.

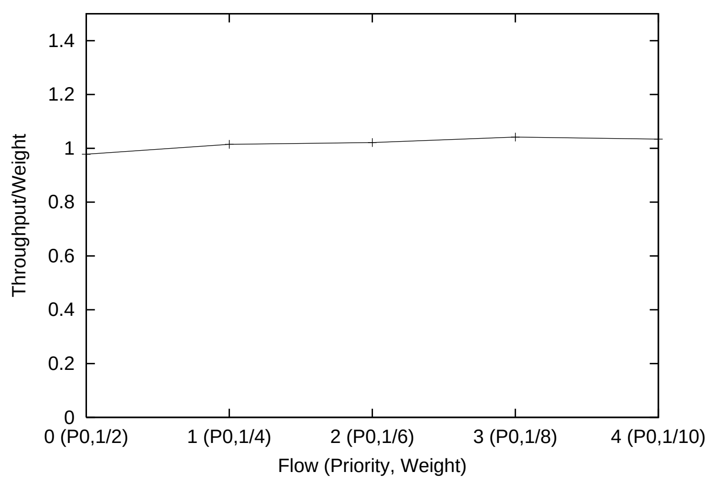


Fig. 6. Throughput/Weight Achieved by Flows Having Different Weights.

CHAPTER VI

ALTERNATIVE SCHEME FOR CONTENTION RESOLUTION

In this chapter we consider an alternative to the scheme proposed in Chapter III and discuss its merits and demerits viz a viz the Proposed Scheme.

A. Alternative Scheme

In the scheme proposed in Chapter III, the tuple is constructed in such a manner that the smaller a tuple element, the higher are its chances of winning the cooresponding phase during a contention resolution cycle. During each phase in the contention resolution cycle, a contending node waits for a certain number of slots equal to the corresponding tuple element before transmitting a burst. If it hears a transmission while it is waiting, it drops out of contention and waits for the next contention resolution cycle.

As an alternative, the tuple could have been constructed such that the larger a tuple element, the higher are its chances of winning the corresponding phase during a contention resolution cycle. In this scheme, during each phase in the contention resolution cycle, a contending node would transmit a burst for a certain number of slots equal to the corresponding tuple element and then listen for a slot (referred to as the observation slot). If it hears a transmission during this observation slot, it drops out of contention and waits for the next contention resolution cycle.

1. Proposed Scheme vs Alternative Scheme

In the alternative scheme, we could use a smaller inter-round spacing (*irs*) because of the fact that the idle time during a contention resolution cycle would be bound by the duration of the observation slot. However, in the alternative scheme, the duration of each phase during a contention resolution cycle would be determined by who has the longest burst. Whereas, in the scheme proposed in Chapter III, the duration of each phase in a contention resolution cycle is determined by who has the shortest waiting time. Hence, the phase durations tend to be much shorter in the scheme proposed in Chapter III as compared to the alternative scheme. Intutively speaking, the proposed scheme would result in much lesser contention resolution overhead as compared to the alternative scheme. Another fact that makes the alternative scheme less attractive is the power consumption involved. Since the alternative scheme involves transmission of long bursts during each phase in a contention resolution, it is not suitable for wirless networks with battery powered nodes.

CHAPTER VII

CONCLUSIONS AND FUTURE WORK

This thesis proposes a distributed Medium Access Control (MAC) protocol for supporting multiple priorities and weighted fair scheduling in a wireless LAN, combining the techniques used in HIPERLAN [5] and DFS [1]. Performance results show that the proposed protocol can support multiple priorities and can allocate bandwidth to the flows within the same class in proportion of their weights. The ideas presented in this thesis can be used to support Quality of Service (QoS) and real time applications in a wireless LAN.

As part of on-going research and future work, we are considering adaptive mechanisms to dynamically choose the appropriate parameters for the proposed scheme. For example, we are considering dynamic adjustment of the the *Scaling_Factor* as a function of the load on the network. When the load on the network is low, the nodes can use a smaller *Scaling_Factor* in order to reduce the overhead. As the load on the network is increased and the nodes detect losses due to collision, they can increase the *Scaling_Factor* to reduce the probability of collisons. Similarly, the weights and priorities can also be adjusted dynamically. For example, the weight of a node can be made proportional to the size of its pending queue and the priority can be decided on a per packet basis as a function of the residual life time of the packet. Future work can also focus on extending the ideas presented here to multi-hop wireless networks.

REFERENCES

[1] N.H. Vaidya, P. Bahl, and S. Gupta, "Distributed Fair Scheduling in a Wireless LAN," in *Sixth Annual International Conference on Mobile Computing and Networking*, Boston, August 2000.

[2] K. Fall and K. Varadhan, "*ns* Notes and documentation," Tech. Rep., VINT Project, UC-Berkeley and LBNL, 1997.

[3] J. Deng and R.S. Chang, "A Priority Scheme for IEEE 802.11 DCF Access Method," IEICE Trans. Commun., vol. E82-B, no. 1, pp. 96-102, January 1999.

[4] K. Ramamritham and W. Zhao, "Virtual time CSMA protocols for hard real-time communication," *IEEE Transactions on Software Engineering*, vol. 13, no. 8, pp. 938-952, 1987.

[5] G. Anastasi, L. Lenzini, and E. Mingozzi, "Stability and Performance Analysis of HIPERLAN", in *Proc. IEEE Infocom*, pp. 134-141, 1998.

[6] J. C. R. Bennett and H. Zhang, "Wf2q: Worst-case fair weighted fair queueing," in *INFOCOM*, pp. 120-128, March 1996.

[7] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, "MACAW: A media access protocol for wireless LANs," in *ACM SIGCOMM*, pp. 212–225, August 1994.

[8] G. L. Choudhury and S. S. Rappaport, "Priority access schemes using CSMA-CD," *IEEE Transactions on Communications*, vol. COM-33, pp. 620–626, July 1985.

[9] A. Demers, S. Keshav, and S. Shenker, "Analysis and simulation of a fair queueing algorithm," in *Proc. SIGCOMM*, pp. 1-12, September 1989.

[10] P. Goyal, H. M. Vin, and H. Cheng, "Start-time fair queueing: A scheduling algorithm for integrated services packet switching networks," *IEEE/ACM Transactions on Networking*, vol. 5, pp. 690–704, October 1997.

[11] IEEE, "IEEE std 802.11 - wireless LAN medium access control (MAC) and physical layer (PHY) specifications," http://www.manta.ieee.org/groups/802/11, 1997.

[12] "P802.1p/D5 Standard for Local and Metropolitan Area Networks - Supplement to Media Access Control (MAC) bridges: Traffic class expediting and dynamic multicast filtering," http://www.manta.ieee.org/groups/802/11, 1997.

[13] S. Keshav, "On the efficient implementation of fair queueing," *Journal of Internetworking: Research and Experience*, vol. 2, pp. 57–73, September 1991.

[14] 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 Transactions on Networking*, vol. 1, pp. 344-357, June 1993.

[15] S. M. Sharrock and D. H. Du, "Efficient CSMA/CD-based protocols for multiple priority classes," *IEEE Transactions on Computers*, vol. 38, pp. 943–954, July 1989.

[16] M. Shreedhar and G. Varghese, "Efficient fair queuing using deficit round robin," in *SIGCOMM'95*, pp. 231-242, Cambridge, MA, 1995.

[17] J. L. Sobrinho and A. S. Krishnakumar, "Real-time traffic over the IEEE 802.11 medium access control layer," *Bell Labs Technical Journal*, pp. 172-187, Autumn 1996.

[18] N. H. Vaidya and P. Bahl, "Fair scheduling in broadcast environments," Tech. Rep. MSR-TR-99-61, Microsoft Research, August 1999.

[19] A. Dugar, N. H. Vaidya and P. Bahl, "Priority and Fair Scheduling in a Wireless LAN," Tech. Rep. TR01-006, Computer Science Department, Texas A&M University, April 2001.

[20] A. Dugar, N. H. Vaidya and P. Bahl, "Priority and fair scheduling in a Wireless LAN," to appear in *IEEE MILCOM*, VA, October 2001.

[21] H. Luo, S. Lu, and V. Bharghavan, "A new model for packet scheduling in Multihop Wireless Networks," in *ACM MOBICOM*, August 2000.

[22] H. Luo and S. Lu, "A self-coordinating approach to distributed Fair Queueing in Ad Hoc Wireless Networks," in *IEEE INFOCOM* , Anchorage, AK, April 26, 2001.

[23] R. Jain, G. Babic, B. Nagendra, and C. Lam, "Fairness, call establishment latency and other performance metrics," Tech. Rep. ATM_Forum/96-1173, ATM Forum Document, August 1996.

[24] P. Goyal, H.M. Vin, and H. Cheng, "Start-time fair queuing: A scheduling algorithm for integrated services packet switching networks," in *proc. IEEE/ACM Transactions on Networking*, vol. 5, pp. 690-704, October 1997.

VITA

Anurag Dugar received his Bachelor of Engineering degree in Electronics and Communications Engineering from Karnataka Regional Engineering College, Surathkal, India. He joined the graduate program in Electrical Engineering at Texas A&M University in August 1999. His research has been in the area of Fairness and Quality of Service (QoS) in Wireless and Mobile Ad Hoc Networks. He can be reached through the Department of Electrical Engineering, Texas A&M University, College Station, TX 77843.

The typist for this thesis was Anurag Dugar.