Dynamic Adaption of DCF and PCF mode of IEEE 802.11 WLAN

Dissertation

Submitted in partial fulfillment of the requirements
for the degree of

Master of Technology

by

Abhishek Goliya
(Roll No: 01329012)

under the guidance of

Prof. Sridhar Iyer
and
Dr. Leena-Chandran Wadia

Kanwal Rekhi School of Information Technology,
Indian Institute of Technology, Bombay
2003

To my late grandparents
and my family

Abstract

IEEE 802.11 specifies the most famous family of WLANs. It features two basic mode of operation: Distributed Coordinating Function (DCF) and Point Coordinating Function (PCF). Both PCF and DCF mode of IEEE 802.11 do not perform equally well under all traffic scenarios. Their behavior varies depending upon current network size and traffic load. It is useful to use the DCF mode for low traffic and small network size, and the PCF mode for high traffic loads and to reduce contention in large size network. In this thesis, we have designed three protocols to dynamically adapt IEEE 802.11 MAC under varying load. One of them is designed to dynamically switch between either modes. Our Dynamic Switching Protocol (DSP) observes network traffic to decide switching point and switches dynamically to suit current traffic load and network size.

PRRS is our second contribution that aims to reduce polling overheads. A major drawback of polling scheme in PCF, is their inefficiency when only a small number of nodes have data to send. Unsuccessful polling attempts causes unnecessary delays for station with data. We have presented network monitoring based scheme that replaces simple Round Robin scheduling in PCF with our Priority Round Robin Scheduling (PRRS). Result shows considerable increase in throughput especially when small fraction of node has data to transmit.

In addition, we have presented the need to dynamically adapt various configuration parameters in both PCF and DCF. Statically configured values results in degraded performance under varying scenarios .We have showed the performance variation of PCF with PRRS by using different CFP repetition intervals. Our proposed CFP repetition interval adaption algorithm dynamically adjust the value of CFP repetition interval, depending upon last CFP usage.

Acknowledgments

I take this opportunity to express a deep sense of gratitude for **Prof. Sridhar Iyer** and **Dr. Leena-Chandran Wadia**, for providing excellent guidance and unbelievable support . It is because of their constant and general interest and assistance that this project has been successful. I would like to thank the members of the **Mobile Computing Research Group** at KReSIT, namely Satyajit Rai, Srinath Perur, Vijay Rajsinghani, Vikram Jamwal, Deepanshu Shukla and Anupam Goyal for their valuable suggestions and helpful discussions.

I would also like to thank my **family** and **friends** who have been a source of encouragement and inspiration throughout the duration of this project.

Abhishek Goliya
IIT Bombay

Contents

List of Figures

Chapter 1

Introduction: Wireless LANs

1.1 Motivation

Wireless computing is a rapidly emerging technology providing users with network connectivity without being tethered off of a wired network. Wireless local area networks (WLANs), like their wired counterparts, are being developed to provide high bandwidth to users in a limited geographical area. WLANs are being studied as an alternative to the high installation and maintenance costs incurred by traditional additions, deletions, and changes experienced in wired LAN infrastructures. Physical and environmental necessity is another driving factor in favor of WLANs.

The operational environment may not accommodate a wired network, or the network may be temporary and operational for a very short time, making the installation of a wired network impractical. Examples where this is true include ad hoc networking needs such as conference registration centers, campus classrooms, emergency relief centers, and tactical military environments. However, to meet these objectives, the wireless community faces certain challenges and constraints that are not imposed on their wired counterparts.

1.1.1 Why IEEE 802.11 WLAN

IEEE 802.11 standard is one of the prominent wireless local area network standards being adopted as a mature technology. The success of the IEEE 802.11 standard has resulted in the easy availability of commercial hardware and a proliferation of wireless network deployment, in wireless LANs as well as in mobile ad hoc networks. Although IEEE 802.11 is not designed for multihop ad hoc networks, the easy availability has made it, most chosen MAC.

1.2 Need for Specialized Wireless MAC

Existing MAC schemes from wired networks like, CSMA/CD are not directly applicable to wireless medium. In CSMA/CD sender senses the medium to see if it is free. If medium is busy, the sender waits until it is free. If the medium is free, sender starts transmitting data and also continues to listen into the medium. It stops transmission as soon as it detects collision and sends a jam signal. In wired medium, this works because more or less the same signal strength can be assumed all over the wire. If collision occurs somewhere in the wire, everybody will notice it. This assumption gets invalidated in wireless medium, as the signal strength decreases proportionally to the square of distance to the sender.

In wireless medium, sender may apply carrier sense and detect an idle medium. Thus, the sender starts sending, but a collision happens at the receiver due to a second sender. Second sender may or may not be audible to first sender. Hence the sender detects no collision, assumes that data has been transmitted without errors, but actually a collision might have destroyed the data at the receiver.

Besides that, wireless devices are half duplex and battery operated. They are unable to listen to the channel for collision while transmitting data.

1.2.1 Hidden and Exposed Node Problem

The transmission range of stations in wireless network is limited by the transmission power, therefore, all the station in a LAN can not listen to each other. This gives rise to *hidden node* and *exposed node* problem. Consider a scenario shown in Figure 1.1. Transmission range of A reaches B, but not C. The transmission range of C reaches B, but not A. Finally, the transmission range of B reaches both A and C.

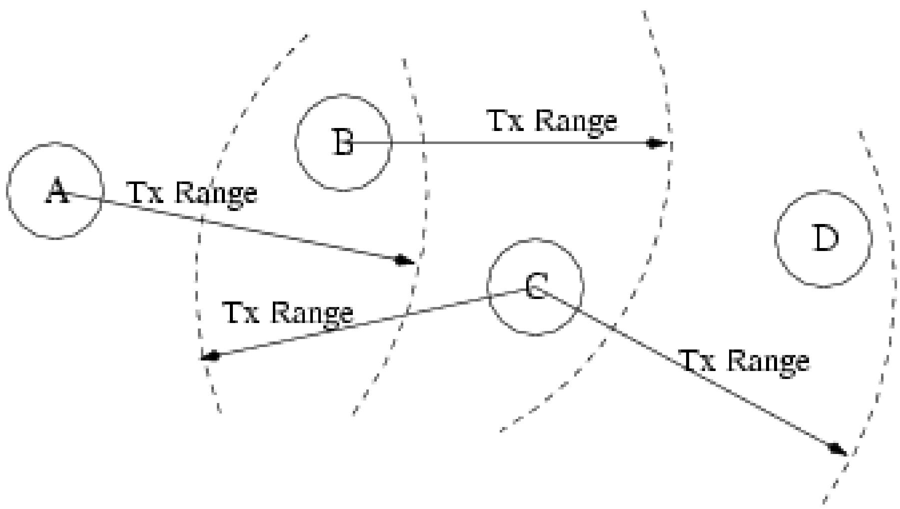


Figure 1.1: Hidden and Exposed Node Scenario

Hence C can listen to B but not A. A start sending to B, C does not hear this transmission, and also wants to send something to B. C senses the medium, medium appears to be free and it starts sending. Hence collision occurs at B. A cannot detect this collision and continues with its transmission. A and C are hidden to each other. This problem is termed as Hidden Node problem.

Hidden terminals cause collision, and Exposed terminals suffer unnecessary delays. Consider the situation that B sends something to A and C wants to transmit data to D. D is not in transmission or interference range of A and B. C senses the medium and finds it busy. Thus, C postpone its transmission. But as A is outside the interference range of C, waiting is not necessary. Collision at B due to C's transmission does not matter as it is too weak to propagate to A. This termed as Exposed Node problem. Node C is exposed to B.

1.3 Challenges in Wireless LANs

Many different and sometimes competing design goals have to be taken into account for WLANs to ensure their commercial success.

- **Global operation:** WLAN products should sell in all countries, therefore, many national and international frequency regulations have to be considered.
- **Low Power:** Devices communicating via a WLAN are typically also wireless devices running on battery power. Hence, WLAN must implement special power saving modes and power management functions.
- **License-free operation:** LAN operators do not want to apply for a special license in order to be able to use the product. Thus, the equipment must operate in a license-free band, such as the 2.4 GHz ISM band.
- **Bandwidth:** Bandwidth is the one of the most scarce resource in wireless networks. The available bandwidth in wireless networks is far less than the wired links.
- **Link Errors:** Channel fading and interference cause link errors and these errors may sometimes be very severe.
- **Robust transmission technology:** Compared to wired counterparts, WLANs operate under difficult conditions. If they use radio transmission, many other electrical devices may interfere.
- **Simplified spontaneous co-operation:** To be useful in practice, WLANs should not require complicated setup routines but should operate spontaneously after power up. Otherwise these LANs would not be useful for supporting e.g., ad hoc meetings, etc.
- **Easy to use:** LANs should not require complex management but rather work on a plug-and-play basis.
- **Protection of investment:** A lot of money has already been invested into wired LANs. Hence new WLANs must protect this investment by being inter operable with the existing networks.
- **Safety and security:** Most important concern is of safety and security. WLANs should be safe to operate, especially regarding low radiation. Furthermore, no users should be able to read personal data during transmission i.e., encryption mechanism should be integrated. The network should also take into account user privacy.
- **Transparency for application:** Existing applications should continue to run over WLANs. The fact of wireless access and mobility should be hidden if not relevant.

1.4 IEEE 802.11 standard

IEEE 802.11 MAC features two mode of operations: Distributed Coordinating Function (DCF) and Point Coordinating Function (PCF). DCF is CSMA/CA based random access protocol that uses random backoff to avoid collision. It uses RTS/CTS exchange mechanism to reserve channel when packet size is above the RTSthreshold. It reduces the hidden terminal effect (section 1.2.1). PCF provide centralized scheduled access to channel. It comprises of chain of contention free period (CFP) and contention period (CP). DCF rules are followed in the CP. In the CFP point coordinator (PC) polls the node one by one and grant access to channel. New stations that need to get enrolled in poll list, send request in CP.

1.5 Problem Statement

Our work aims at optimizing overall performance of IEEE 802.11 MAC. Although we have tried to keep solution robust enough to suit different traffic scenarios, our main focus is on traffic directed towards a central node. Both DCF and PCF do not perform well under all load regime. Each has its own pros and cons depending upon different load condition. When only small number of nodes have data to transmit PCF incurs polling overheads, and at high load DCF performance degrades. We think there is need to dynamically adapt IEEE 802.11 MAC under varying load, such that coexistence of both the modes can be exploited.

Besides that, performance of DCF and PCF depends highly upon their various configuration parameters. Studies shows that good values of these configuration parameters depend upon network load. Statically configured values result in degraded throughput under varying load. So there is need to dynamically adapt these values.

We have proposed learning based protocol to reduce polling overheads in PCF and to dynamically adapt configuration parameters. To exploit better half of both PCF and DCF, we have proposed a protocol to dynamically switch between two modes.

1.6 Thesis outline

In this thesis, in next chapter we start with summarization of existing IEEE 802.11b WLAN standard. We emphasize fully on the MAC sub-layer explaining both DCF and PCF. Chapter 3 discuss in depth view of problems, we attacking and also present theoretical analysis of problem. Moving further, various existing approaches to deal with performance issue are then discussed, along with arguments for the need of our approach.

Following the track, Chapter 4 explains our solutions to problem, termed as Priority Round Robin Scheduling (PRRS) and Dynamic Switching Protocol (DSP). Chapter 5 describes simulation scenario and performance metric used by us. Following this, we analyze our results that lead to further optimization of our solution, PRRS in Chapter 6. Chapter 7 then concludes our work and points to future work to be explored.

Chapter 2

MAC Sublayer in IEEE 802.11

The IEEE standard 802.11 specifies the most famous family of WLANs in which many products are already available. Standard belongs to the group of 802.x LAN standards, e.g., 802.3 Ethernet or 802.5 Token Ring. This means that the standard specifies the physical and medium access layer adapted to the special requirements of wireless LANs, but offers the same interface as the others to higher layers to maintain interoperability.

2.1 Scope and Purpose of IEEE 802.11 standard

The scope of this standard is to develop a medium access control (MAC) and physical layer (PHY) specification for wireless connectivity for fixed, portable, and moving stations within a local area.

The purpose of this standard is to provide wireless connectivity to automatic machinery, equipment, or stations that require rapid deployment, which may be portable or hand-held, or which may be mounted on moving vehicles within a local area. This standard also offers regulatory bodies a means of standardizing access to one or more frequency bands for the purpose of local area communication.

Primary goal of the standard was the specification of a simple and robust WLAN which offers time-bounded and asynchronous services. Furthermore, the MAC layer should be able to operate with the multiple physical layers, each of which exhibits a different medium sense and transmission characteristic. Candidates for physical layers were infrared and spread spectrum radio transmission techniques.

Additionally features of the WLAN should include the support of the power management, the handling of hidden nodes, and the ability to operate worldwide.

2.2 System Architecture

The basic service set (BSS) is the fundamental building block of the IEEE 802.11 architecture. A BSS is defined as a group of stations that are under the direct control of a single coordination function (i.e., a DCF or PCF) which is defined below. The geographical area covered by the BSS is known as the basic service area (BSA), which is analogous to a cell in a cellular communications network. Conceptually, all stations in a BSS can communicate directly with all other stations in a BSS. However, transmission medium degradations due to multipath fading, or interference from nearby BSSs reusing the same physical-layer characteristics (e.g., frequency and spreading code, or hopping pattern), can cause some stations to appear hidden from other stations. An ad hoc network is a deliberate grouping of stations into a single BSS for the purposes of internetworked communications without the aid of an infrastructure network. Figure 2.1 is an illustration of an independent BSS (IBSS), which is the formal name of an ad hoc network in the IEEE 802.11 standard. Any station can establish a direct communications session with any other station in the BSS, without the requirement of channeling all traffic through a centralized access point (AP).

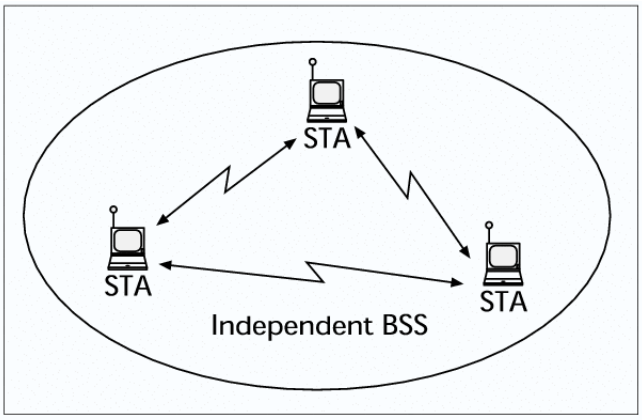


Figure 2.1: Sketch of an ad hoc network

In contrast to the ad hoc network, infrastructure networks are established to provide wireless users with specific services and range extension. Infrastructure networks in the context of IEEE 802.11 are established using APs. The AP is analogous to the base station in a cellular communications network. The AP supports range extension by providing the integration points necessary for network connectivity between multiple BSSs, thus forming an extended service set (ESS). The ESS has the appearance of one large BSS to the logical link control (LLC) sublayer of each station (STA). The ESS consists of multiple BSSs that are integrated together using a common distribution system (DS). The DS can be thought of as a backbone network that is responsible for MAC-level transport of MAC service data units (MSDUs). The DS, as specified by IEEE 802.11, is implementation independent. Therefore, the DS could be a wired IEEE 802.3 Ethernet LAN, IEEE 802.4 token bus LAN, IEEE 802.5 token ring LAN, fiber distributed data interface (FDDI) metropolitan area network (MAN), or another IEEE 802.11 wireless medium. Note that while the DS could physically be the same transmission medium as the BSS, they are logically different, because the DS is solely used as a transport backbone to transfer packets between different BSSs in the ESS. An ESS can also provide gateway access for wireless users into a wired network such as the Internet. This is accomplished via a device known as a portal. The portal is a logical entity that specifies the integration point on the DS where the IEEE 802.11 network integrates with a non-IEEE 802.11 network. If the network is an IEEE 802.X, the portal incorporates functions which are analogous to a bridge; that is, it provides range extension and the translation between different frame formats. Figure 2.2 illustrates a simple ESS developed with two BSSs, a DS, and a portal access to a wired LAN.

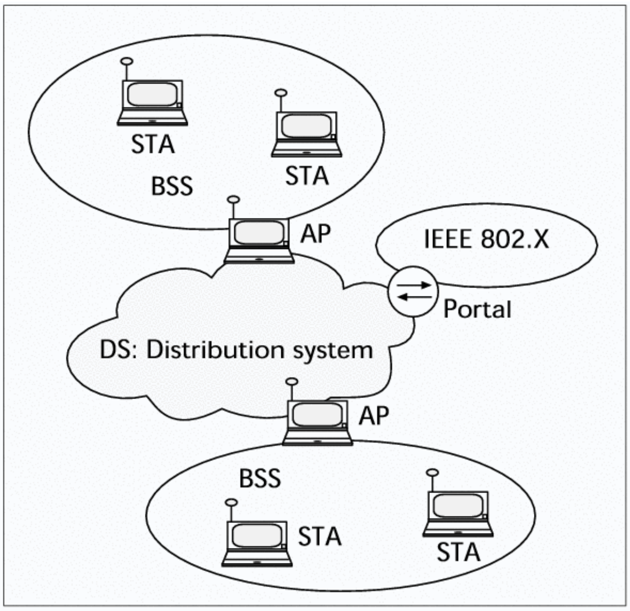


Figure 2.2: Sketch of an infrastructure network

2.3 Medium Access Control Sublayer

The IEEE 802.11 draft standard describes mandatory support for asynchronous data transfer and optional support for distributed time-bounded services (DTBS). Asynchronous data transfer refers to traffic that is relatively insensitive to time delay. Examples of asynchronous data are available bit rate traffic like electronic mail and file transfers. Time-bounded traffic, on the other hand, is traffic that is bounded by specified time delays to achieve an acceptable quality of service (QoS) (e.g., packetized voice and video).

The first scheme, distributed coordination function (DCF), is designed for asynchronous data transport, where all users with data to transmit have an equally fair chance of accessing the network. The point coordination function (PCF) is the second MAC scheme. The PCF is based on polling that is controlled by an access point (AP). The PCF is primarily designed for the transmission of delay-sensitive traffic.

The MAC architecture can be described as shown in Figure 2.3 as providing the PCF through the services of the DCF.

2.4 Distributed Coordination Function

The basic medium access protocol is a DCF that allows for automatic medium sharing between compatible PHYs through the use of CSMA/CA and a random backoff time following a busy medium condition. The DCF shall be implemented in all stations, for use within both IBSS and infrastructure network configuration. It operates solely in the ad hoc network (IBSS) , and either operates solely or coexist with the PCF in an infrastructure network. The MAC architecture is depicted in Figure 2.3, where it is shown that the DCF sits directly on top of the physical layer and supports contention services. Contention services imply that each station with an MSDU queued for transmission must contend for access to the channel and, once the MSDU is transmitted,

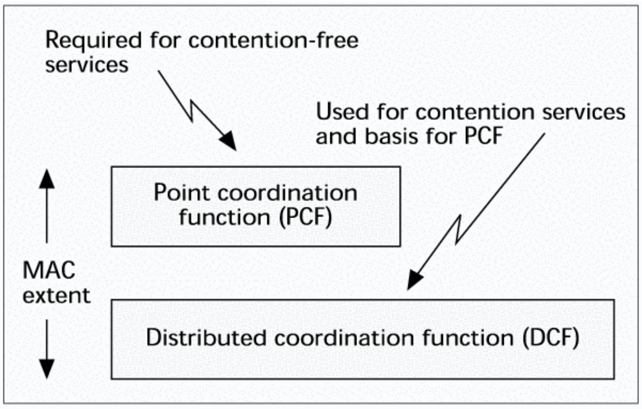


Figure 2.3: MAC Architecture

must recontend for access to the channel for all subsequent frames. Contention services promote fair access to the channel for all stations.

The CSMA/CA protocol is designed to reduce the collision probability between multiple stations accessing a medium, at the point where collisions would most likely occur. Just after the medium becomes idle following a busy medium, is when the highest probability of a collision exists. This is because multiple stations could have been waiting for the medium to become available again. This is the situation that necessitates a random backoff procedure to resolve medium contention conflicts.

Station that needs to transmit data, first sense the carrier. In IEEE 802.11, carrier sensing is performed at both the air interface, referred to as *physical carrier sensing*, and at the MAC sublayer, referred to as *virtual carrier sensing*. Physical carrier sensing detects the presence of other IEEE 802.11 WLAN users by analyzing all detected packets, and also detects activity in the channel via relative signal strength from other sources.

A virtual carrier-sense mechanism shall be provided by the MAC. This mechanism is referred to as the network allocation vector (NAV). The NAV maintains a prediction of future traffic on the medium based on duration information that is announced in RTS/CTS frames prior to the actual exchange of data. The duration information is also available in the MAC headers of other frames sent during the CP. The Duration/ ID field defines the period of time that the medium is to be reserved to transmit the actual data frame and the returning ACK frame. Stations in the BSS use the information in the duration field to adjust their network allocation vector (NAV), which indicates the amount of time that must elapse until the current transmission session is complete and the channel can be sampled again for idle status. The channel is marked busy if either the physical or virtual carrier sensing mechanisms indicate the channel is busy.

Priority access to the wireless medium is controlled through the use of interframe space (IFS) time intervals between the transmission of frames. The IFS intervals are mandatory periods of idle time on the transmission medium. Three IFS intervals are specified in the standard: short IFS (SIFS), point coordination function IFS (PIFS), and DCF-IFS (DIFS). The SIFS interval is the smallest IFS, followed by PIFS and DIFS, respectively. Stations only required to wait a SIFS have priority access over those stations required to wait a PIFS or DIFS before transmitting; therefore, SIFS has the highest-priority access to the communications medium. For the basic access method, when a station senses the channel is idle, the station waits for a DIFS period and samples the channel again. If the channel is still idle, the station transmits an MPDU. The receiving station calculates the checksum and determines whether the packet was received correctly. Upon receipt of a correct packet, the receiving station waits a SIFS interval and transmits a positive acknowledgment frame (ACK) back to the source station, indicating that the transmission was successful. Figure 2.4 is a timing diagram illustrating the successful transmission of a data frame. When the data frame is transmitted, the duration field of the frame is used to let all stations in the BSS know how long the medium will be busy. All stations hearing the data frame adjust their NAV based on the duration field value, which includes the SIFS interval and the ACK following the data frame.

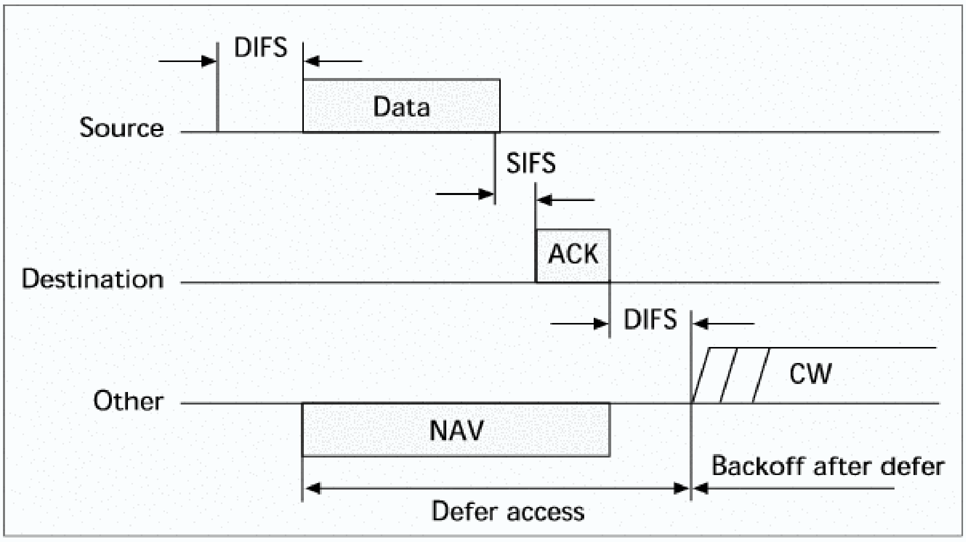


Figure 2.4: Transmission of an MPDU without RTS/CTS

Since a source station in a BSS cannot hear its own transmissions, when a collision occurs, the source continues transmitting the complete MPDU. If the MPDU is large (e.g., 2300 bytes), a lot of channel bandwidth is wasted due to a corrupt MPDU. RTS and CTS control frames can be used by a station to reserve channel bandwidth prior to the transmission of an MPDU and to minimize the amount of bandwidth wasted when collisions occur. RTS and CTS control frames are relatively small (RTS is 20 bytes and CTS is 14 bytes) when compared to the maximum data frame size (2346 bytes). The RTS control frame is first transmitted by the source station (after successfully contending for the channel) with a data or management frame queued for transmission to a specified destination station. All stations in the BSS, hearing the RTS packet, read the duration field and set their NAVs accordingly. The destination station responds to the RTS packet with a CTS packet after an SIFS idle period has elapsed. Stations hearing the CTS packet look at the duration field and again update their NAV. Upon successful reception of the CTS, the source station is virtually assured that the medium is stable and reserved for successful transmission of the MPDU. Note that stations are capable of updating their NAVs based on the RTS from the source station and CTS from the destination station, which helps to combat the hidden terminal problem [1.2.1]. Figure 2.5 illustrates the transmission of an MPDU using the RTS/CTS mechanism. Stations can choose to never use RTS/CTS, use RTS/CTS whenever the MSDU exceeds the value of RTS_Threshold (manageable parameter), or always use RTS/CTS. If a collision occurs with an RTS or CTS MPDU, far less bandwidth is wasted when compared to a large data MPDU. However, for a lightly loaded medium, additional delay is imposed by the overhead of the RTS/CTS frames.

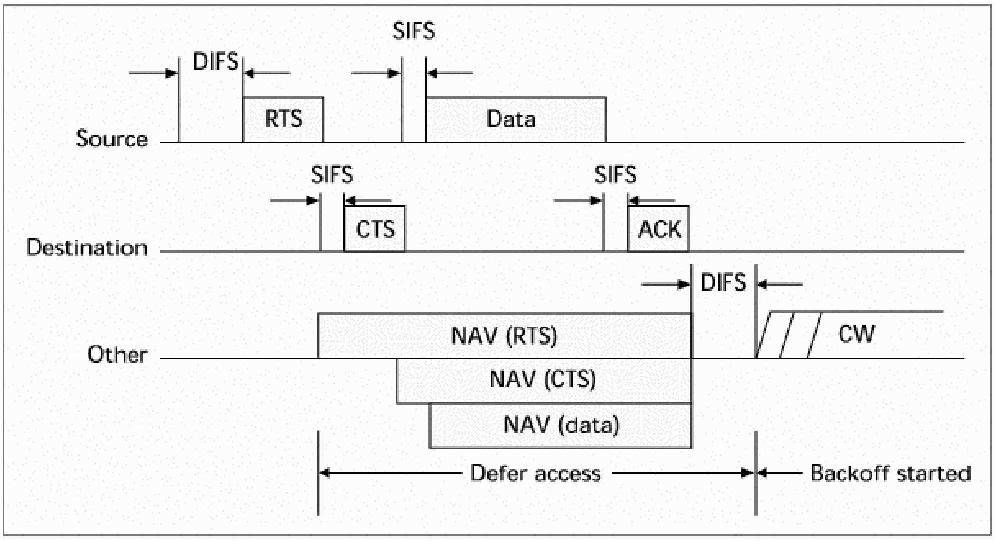


Figure 2.5: Transmission of an MPDU using RTS/CTS

Large MSDUs handed down from the LLC to the MAC may require fragmentation to increase transmission reliability. To determine whether to perform fragmentation, MPDUs are compared to the manageable parameter Fragmentation_Threshold. If the MPDU size exceeds the value of Fragmentation_Threshold, the MSDU is broken into multiple fragments. The resulting MPDUs are of size Fragmentation_Threshold, with exception of the last MPDU, which is of variable size not to exceed Fragmentation_Threshold. When an MSDU is fragmented, all fragments are transmitted sequentially. The channel is not released until the complete MSDU has been transmitted successfully, or the source station fails to receive an acknowledgment for a transmitted fragment. The destination station positively acknowledges each successfully received fragment by sending a DCF ACK back to the source station. The source station maintains control of the channel throughout the transmission of the MSDU by waiting only an SIFS period after receiving an ACK and transmitting the next fragment. When an ACK is not received for a previously transmitted frame, the source station halts transmission and recontends for the channel. Upon gaining access to the channel, the source starts transmitting with the last unacknowledged fragment.

If RTS and CTS are used, only the first fragment is sent using the handshaking mechanism. The duration value of RTS and CTS only accounts for the transmission of the first fragment through the receipt of its ACK. Stations in the BSS thereafter maintain their NAV by extracting the duration information from all subsequent fragments. The collision avoidance portion of CSMA/CA is performed through a random backoff procedure. If a station with a frame to transmit initially senses the channel to be busy; then the station waits until the channel becomes idle for a DIFS period, and then computes a random backoff time. For IEEE 802.11, time is slotted in time periods that correspond to a Slot_Time. Unlike slotted Aloha, where the slot time is equal to the transmission time of one packet, the Slot_Time used in IEEE 802.11 is much smaller than an MPDU and is used to define the IFS intervals and determine the backoff time for stations in the CP. The Slot_Time is different for each physical layer implementation. The random backoff time is an integer value that corresponds to a number of time slots. Initially, the station computes a backoff time in the range equals size of contention window (CW). After the medium becomes idle after a DIFS period, stations decrement their backoff timer until the medium becomes busy again or the timer reaches zero. If the timer has not reached zero and the medium becomes busy, the station freezes its timer. When the timer is finally decremented to zero, the station transmits its frame. If two or more stations decrement to zero at the same time, a collision will occur, and each station will have to generate a new backoff time but for each retransmission attempt, station doubles its contention window. The advantage of this channel access method is that it promotes fairness among stations, but its weakness is that it probably could not support DTBS. Fairness is maintained because each station must recontend for the channel after every transmission of an MSDU. All stations have equal probability of gaining access to the channel after each DIFS interval. Time-bounded services typically support applications such as packetized voice or video that must be maintained with a specified minimum delay. With DCF, there is no mechanism to guarantee minimum delay to stations supporting time-bounded services.

2.5 Point Coordination Function

The PCF provides contention-free frame transfer. The PC shall reside in the AP. It is an option for an AP to be able to become the PC. All stations inherently obey the medium access rules of the PCF, because these rules are based on the DCF, and they set their NAV at the beginning of each CFP. The operating characteristics of the PCF are such that all stations are able to operate properly in the presence of a BSS in which a PC is operating, and, if associated with a point-coordinated BSS, are able to receive all frames sent under PCF control. It is also an option for a station to be able to respond to a contention-free poll (CF-Poll) received from a PC. A station that is able to respond to CF-Polls is referred to as being CF-Pollable, and may request to be polled by an active PC. CF-Pollable stations and the PC do not use RTS/CTS in the CFP. When polled by the PC, a CFPollable station may transmit only one MPDU, which can be to any destination (not just to the PC), and may piggyback the acknowledgment of a frame received from the PC using particular data frame subtypes for this transmission. If the data frame is not in turn acknowledged, the CF-Pollable station shall not retransmit the frame unless it is polled again by the PC, or it decides to retransmit during the CP. If the addressed recipient of a CF transmission is not CF-Pollable, that station acknowledges the transmission using the DCF acknowledgment rules, and the PC retains control of the medium. A PC may use contention-free frame transfer solely for delivery of frames to stations, and never to poll non-CF-Pollable stations.

The PCF controls frame transfers during a CFP. The CFP shall alternate with a CP, when the DCF controls frame transfers, as shown in Figure 2.6.Combined CFP and CP is termed as one **superframe**. Each CFP shall begin with a Beacon frame that contains a DTIM element (hereafter referred to as a DTIM). The CFPs shall occur at a defined repetition rate, which shall be synchronized with the beacon interval as specified in the following paragraphs.

The PC generates CFPs at the contention-free repetition rate (CFPRate), which is defined as a number of DTIM intervals. The PC shall determine the CFPRate to use from the CFPRate parameter in the CF Parameter Set. This value, in units of DTIM intervals, shall be communicated to other stations in the BSS in the CFPPeriod field of the CF Parameter Set element of Beacon frames.

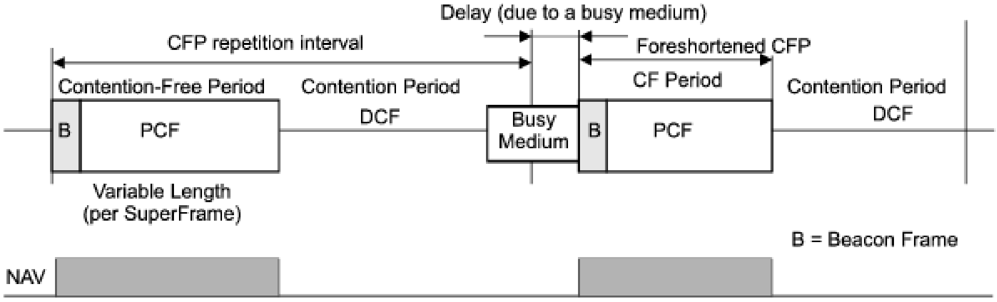


Figure 2.6: Superframe CFP/CP alternation

The length of the CFP is controlled by the PC, with maximum duration specified by the value of the CFPMaxDuration Parameter in the CF Parameter Set at the PC. The PC may terminate any CFP at or before the aCFPMaxDuration, based on available traffic and size of the polling list.

The contention-free transfer protocol is based on a polling scheme controlled by a PC operating at the AP of the BSS. The PC gains control of the medium at the beginning of the CFP and attempts to maintain control for the entire CFP by waiting a shorter time between transmissions than the stations using the DCF access procedure. All stations in the BSS (other than the PC) set their NAVs to the CFPMaxDuration value at the nominal start time of each CFP. This prevents most contention by preventing non-polled transmissions by stations whether or not they are CF-Pollable.The PC shall transmit a CF-End or CF-End+ACK frame at the end of each CFP. A station that receives either of these frames, from any BSS, shall reset its NAV.

At the nominal start of the CFP, the PC senses the medium. If the medium remains idle for a PIFS interval, the PC transmits a beacon frame to initiate the CFP. The PC starts CF transmission a SIFS interval after the beacon frame is transmitted by sending a CF-Poll (no data), Data, or Data+CF-Poll frame. The PC can immediately terminate the CFP by transmitting a CF-End frame, which is common if the network is lightly loaded and the PC has no traffic buffered. If a CF-aware station receives a CF-Poll (no data) frame from the PC, the station can respond to the PC after a SIFS idle period, with a CF-ACK (no data) or a Data + CF-ACK frame. If the PC receives a Data + CFAck frame from a station, the PC can send a Data + CFACK + CF-Poll frame to a different station, where the CF-ACK portion of the frame is used to acknowledge receipt of the previous data frame. The ability to combine polling and acknowledgment frames with data frames, transmitted between stations and the PC, was designed to improve efficiency. If the PC transmits a CF-Poll (no data) frame and the destination station does not have a data frame to transmit, the station sends a Null Function (no data) frame back to the PC. Figure 2.7 illustrates the transmission of frames between the PC and a station, and vice versa. If the PC fails to receive an ACK for a transmitted data frame, the PC waits a PIFS interval and continues transmitting to the next station in the polling list. After receiving the poll from the PC, as described above, the station may choose to transmit a frame to another station in the BSS. When the destination station receives the frame, a DCF ACK is returned to the source station, and the PC waits a PIFS interval following the ACK frame before transmitting any additional frames.

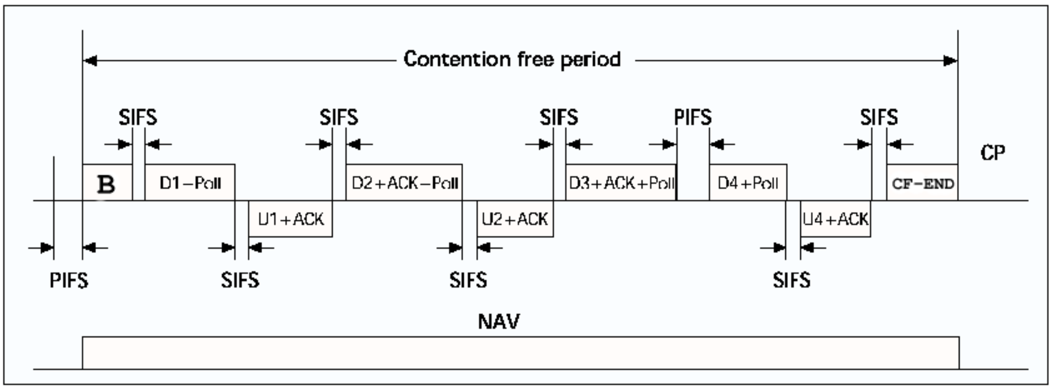


Figure 2.7: PCF PC-to-station frame transmission

2.5.1 Polling List processing and update procedure

The PC shall send a CF-Poll to at least one station during each CFP when there are entries in the polling list. During each CFP, the PC shall issue polls to a subset of the stations on the polling list in order by ascending association id value.

While time remains in the CFP, all CF frames have been delivered, and all stations on the polling list have been polled, the PC may generate one or more CF-Polls to any stations on the polling list. While time remains in the CFP, all CF frames have been delivered, and all stations on the polling list have been polled, the PC may send data or management frames to any stations.

A station indicates its CF-Pollability using the CF-Pollable subfield of the Capability Information field of Association Request and Reassociation Request frames. If a station desires to change the PC s record of CFPollability, that station shall perform a reassociation. During association, a CF-Pollable station may also request to be placed on the polling list for the duration of its association, by setting the CF-Poll Request subfield in the Capability Information field. If a CF-Pollable station desires never to be placed on the polling list, that station shall perform Association with CF-Pollable subfield false Never being polled is useful for CF-Pollable stations that normally use power-save mode.

We have focused more on PCF mode than DCF mode. Standard specifies simple round robin scheduling (RRS). In next chapter we would discuss overheads involved in RRS especially when few nodes have data to send. We then discuss the need for dynamic switching between PCF and DCF mode to take advantage of better half of both the modes. In standard we have seen, how different parameters effect the functionality of both modes. In chapter 6 we will discuss the issue of configuring CFP repetition interval.

Chapter 3

Problem Analysis and Related Work

In this work we have focused on three problem areas that lead to performance degradation in IEEE 802.11 WLAN. One major problem with PCF that gets highlighted when small fraction of associated and pollable nodes in BSS are sending data, and rest are silent. This results in significant polling overheads that wastes the scare channel bandwidth. Second problem area is to define protocol for dynamic switching between two modes. Mean packet delays can be reduced by having DCF when less nodes have pending data and PCF otherwise. Third problem area is statically configured configuration parameters of PCF and DCF. Depending on the network configuration, the standard can operate far from the theoretical throughput limits.

3.1 Need for Switching between PCF and DCF

The DCF mode of IEEE 802.11 exerts a CSMA/CA approach, which is in fact a 1-persistent random access protocol with delay. Random access protocol works satisfactorily as long as network size is limited. Here by network size we mean number of node that have pending data in BSS, i.e. in transmission range of central node. Load is defined as total bits transmitted by all stations in BSS per second. As network expands, competition for accessing shared wireless channel increases. This results in throughput degradation and more delay because of more collision and increased time spent for negotiating channel access. We need ordered way to schedule the channel access at high loads.

IEEE 802.11 provide another more organized way to grant channel access called PCF. But better management always poses some overheads that become prominent under low load scenarios. Similar story appears here. DCF whose performance degrades at high load and in big size network, provide lesser delays at low load. On counter side, scheduled MAC like PCF with centralized control better utilize resources at high load and in large network. But when few nodes have data to send PCF perform worse than DCF because of scheduling overhead in PCF (section 3.2). Graph * shown in figure 3.1 presents goodput and delay at different load. PCF starts with slightly high delay, but it remains low and constant up to 80% goodput. In DCF beyond 60% load the delay increases exponentially. We think dynamic switching between them will increase the channel capacity and offer lower delays.

*Graph by Wolisz, et. al [6]

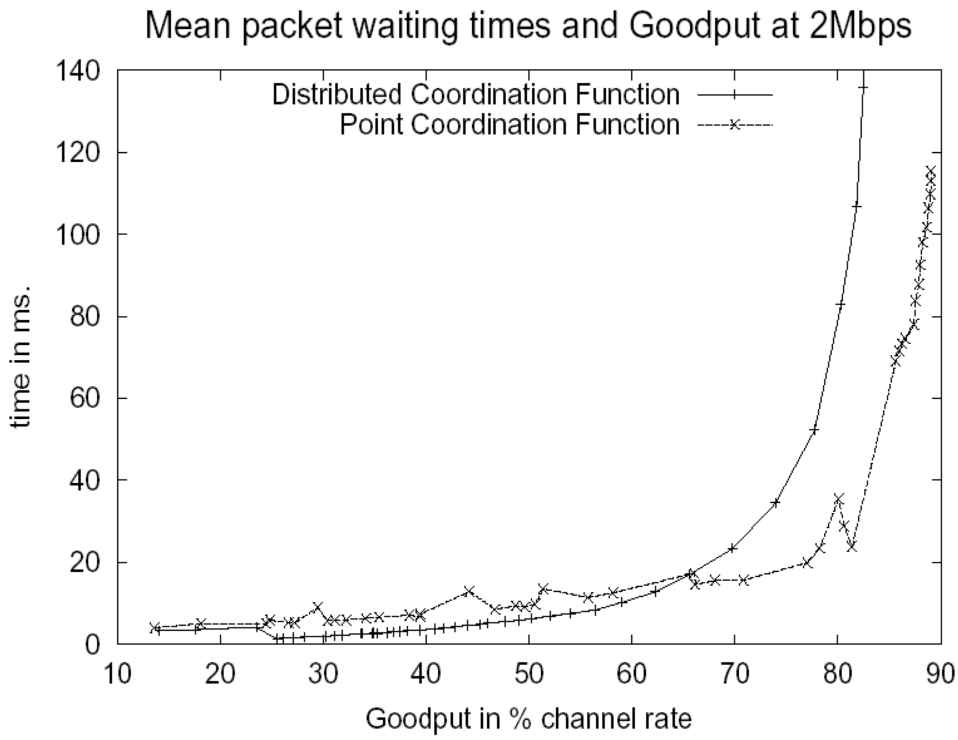


Figure 3.1: Comparison of mean packet waiting and goodput between DCF and PCF at 2 Mbps. 15 Nodes,1 PC and 1500 bytes packet size.

3.2 Polling overheads

In PCF PC is a central coordinator that schedules channel access for all other pollable stations in CFP. PC maintains list of pollable nodes in BSS. At beginning of CFP it polls all stations in Round Robin fashion. Nodes receiving poll respond back, either by transmitting data or null data frame. If station has no pending data then it sends Null frame else data frame. If station fails to do either then its result in poll time out at PC and PC resumes polling. When most stations have pending data, sequential polling provide ordered channel access and reduces collisions. But when few stations have pending data and rest are silent, this polling mechanism becomes significant overhead. It adds unnecessary delay for stations with data, due to unsuccessful poll attempts for stations, with no pending data. Hence resulting in throughput degradation. Graph in figure 3.2 compares the overall throughput of network with 32 and 64 nodes having 16 nodes that have data to transmit. Effect of polling overhead is clearly visible.

3.2.1 Theoretical analysis

Whenever PC polls to stations having no pending data, it adds to overhead, the time to poll, the time to respond with Null frame, and the time taken by protocol for transmission. This overhead gets highlighted when we have more such nodes. If n is total number of nodes in BSS and p fraction of nodes have pending data then polling overhead O_p is

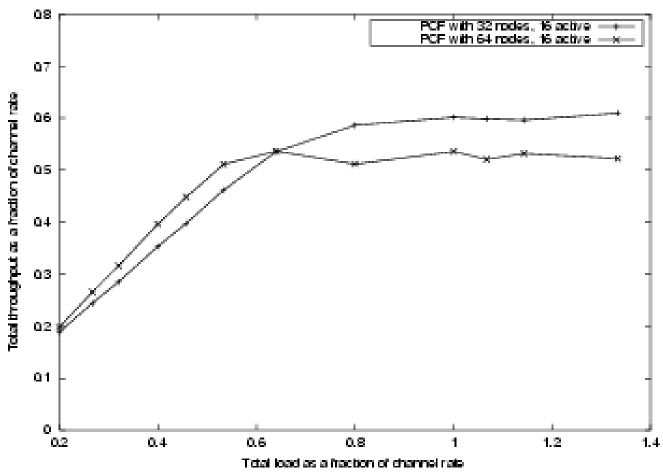


Figure 3.2: Effect of polling overhead on network throughput

Parameter	Symbol	Value
SIFS Interval	$SIFS$	$10\mu s$
Channel B/w	b_w	2Mbps
CF-Poll size	$Size_{Poll}$	20 bytes
Ack Size	$Size_{Ack}$	14 bytes
Null Frame Size	$Size_{Null}$	34 bytes
Time to send poll	T_{Poll}	$Size_{Poll} \times 8/b_w$
Time to send Null Frame	T_{Null}	$Size_{Null} \times 8/b_w$
Time to send Data	T_{Data}	$P_{size} \times 8/b_w$
Time to send Ack	T_{Ack}	$Size_{Ack} \times 8/b_w$

Table 3.1: 802.11b Default parameters

$$O_p = p * n * T_{PollFail}$$

where $T_{PollFail}$ is worst case time * overhead due to unsuccessful poll attempt given as

$$T_{PollFail} = T_{Poll} + SIFS + T_{Null} + SIFS$$

where T_{Poll} and T_{Null} is time to send poll and send Null data frame respectively. If we assume mean packet size be P_{size} then time for successful poll attempt is given by $T_{PollSuccess}$.

$$T_{PollSuccess} = T_{Poll} + SIFS + T_{Data} + SIFS + T_{Ack} + SIFS$$

Now we can compute percentage overhead as ratio of $T_{PollFail}$ and CFP duration. Default values for IEEE 802.11b parameters are depicted in table 3.1. Polling overhead is presented as percentage of CFP get wasted in unsuccessful polling attempts. Assume single complete round of polling. Table 3.2 shows polling overheads in different scenarios

*Frame type CFPoll+DATA+ACK can reduce this using different values of p and P_{size}. Regime where performance can be improved by reducing polling overhead is quite clear. As mean packet size reduces and number of nodes sending null frame increases, the effect of overhead heightens. With increase in total number of nodes in BSS, overall throughput degradation will also increase.

		Percentage of active Nodes			
		12.5%	25%	50%	75%
Packet size in bytes	300	52.37%	32.03%	13.57%	4.97%
	500	41.77%	23.51%	9.29%	3.29%
	1000	27.74%	13.97%	5.19%	1.78%
	1500	20.76%	10.09%	3.60%	1.22%

Table 3.2: Percentage polling overheads with active nodes percentage 10,25, 50, and 75 and packet size (bytes) 300,500,100,and 1500

3.3 Need for Dynamic Tuning Configuration parameter

Both DCF and PCF has lot of configuration parameters that directly affects their functioning. Good values of these parameter is highly dependent on network size and traffic load. Besides network load, traffic model and mobility pattern also play significant role in deciding its performance, as these factors too affect configuration parameters. Table 3.3 list some main parameters that need to accurately configured.

It is not always possible to accurately approximate the network size and load and then statically configured various configuration parameters. Even if network administrator has certain approximation, even then its true that the approximation doesn't hold around the clock. So we though that there is a need to have some intelligent system that learns network continuously and dynamically reconfigures the parameters.

Parameter	Meaning	Effect
RTS_threshold	MPDU size above which RTS/CTS is used	Directly monitor effect of hidden nodes. Value depend upon number of competing stations
CW_min	Minimum number of backoff slots	Directly control number of collision, Value depend upon number of competing stations
aFragmentation Threshold	Size above which MPDU need to be fragmented	Decides probability of successful transmission
CFP repetition interval	Decides CFP and CP duration	Depend on number of nodes to service and directly affect delays

Table 3.3: Configuration parameters

3.4 Tuning of DCF

DCF mode of operation has been studied more than PCF. Frederico, et. al [12] concluded that appropriate tuning of the backoff algorithm can drive the DCF mode operation close to the theoretical throughput limit. They adopted p-persistent backoff algorithm to show that it is possible, by observing the network status, to estimate the average backoff window size that maximizes the throughput.

Wolisz,et. [20, 21] al have identified the backoff strategy and RTS/CTS message exchange as the elements with crucial impact on the network performance. They showed via simulation that it is reasonable only to use RTS/CTS mechanism when load is high. They extensively studied the effect of hidden terminals (section 1.2.1) and need of RTS/CTS using different packet sizes and varying number of stations. Chayya, et. al [13] also concluded same result regarding the use of RTS/CTS mechanism. They also studied the effect of station location on probability of success.

3.5 Solution strategy

Our solution aims to fine tune existing IEEE 802.11 LAN such that it shows better performance results under all load regime. We tried to adhere to existing standard and suggest minimal changes to meet our goals. As standard allows coexistence of both functionality, so our effort aims to devise solution that includes better half of both functionalities. We have used a network monitoring layer at the PC that learns network and tries to approximate current network load. Using information generated by monitoring layer, we have designed simple protocols. First step towards our goal is PRRS priority round robin scheduling that replaces simple round robin scheduling in PCF. Tuning further, we devise a algorithm for dynamically configuring CFP repetition interval. Next step is to dynamically switch between DCF and PCF mode and exploit there coexistence power. We have provided DSP, a dynamic switching protocol to switch between either modes.

3.6 Related Work

To the best of our knowledge, there is very little literature , that deals with polling overhead in PCF. People have mentioned the problem earlier and there also exist some proposed solution to deal with it. Besides that people have depicted the importance of polling for providing QoS service in IEEE 802.11 WLAN. New upcoming standard for QoS, IEEE 802.11e [17] defines intelligent polling based HCF * that assign time bounded transmission opportunities to all stations. Visser,et. al [19] have studied the impact of superframe length (CFP+CP), on the voice transmission over IEEE 802.11 in PCF mode.

3.6.1 Signaling Polling Information

Wolisz, et al [6] have identified the somewhat similar problem and proposed solutions based on implicit and explicit signaling. According to them mobile nodes have to provide the AP with suitable polling information, to reduce polling overhead. Two method proposed by them for signaling polling information is:

* built on top of Enhanced DCF

1. **Explicit signaling:** Node signal its transmission request in the contention period of the superframe by means of a dedicated short signaling frame.
2. **Implicit signaling:** Node indicates by means of the recently transmitted frame that there are more packets pending.

First approach added additional overhead of signaling packet and second one requires existing frame structure to be modified.

3.6.2 STRP: Efficient Polling MAC

Oran sharon and Eitan Altman [15] suggest STRP, a efficient polling MAC which exploits the capture phenomena and enables simultaneous polling and transmission of information packet. Their proposal tries to overcome the inefficiency encountered (unsuccessful poll attempts) when only part of the stations have packets to transmit by taking advantage of the capture phenomena in radio channel.

Simultaneous Transmission and Response Polling(STRP) utilizes three logical ring at the BS. First *System Ring* contains all the stations in the BSS, which are arranged in cyclic. Other two logical rings, *Active ring* and *Idle ring* contains stations that notified BSS that they have packets to transmit and and those that notified BSS that they do not have packets to transmit respectively. Order of stations in logical rings is defined by system ring.

BS enables a station in the active ring to transmit data and polls a station in the idle ring by the same control packet denoted by *Query/Transmit (Q/T)*. It contains identity of the station(I) in the active ring as well as identity of station(J) in the idle ring. When I receive Q/T packet it begins transmission as per predefined rules. When J receives the same packet, it waits for predefined time units and then begins to transmit a *Jam* signal by a weaker signal than I, if it has packet(s) to transmit and wants to join the active ring. Jam from J collides with I's transmission but since J transmits the *Jam* by a weaker signal, by the capture effect, BS succeeds in receiving the packet from I. I while transmitting packet also explicit signals the BS if it wants to stay in the active ring. If I does not signals BS that it wants to stay in active list, then it is shifted to idle list. After detecting the end of packet from I, the BS continues to listen to the upstream link. If it detects the Jam signal from J then J is moved to the active list. J continuously transmit the Jam signal until it hears the next control packets from the BS. Hence signal control packet serve multiple purpose like, enables station in the active ring to transmit, polls station in the idle ring, and may also give indication to some station in the idle ring to end the jam signal. Figure 3.3 shows how the BS transmits to the stations in the logical ring.

Although approach succeeds in reducing in the polling overhead caused by stations having no pending data to transmit, but it adds lots of prerequisite. To implement similar approach in existing PCF, need to change the existing frame structure of control packets or to add new control packet. Most important prerequisite is that the above approach requires two separate transmission links in the system, an *Upstream* and a *Downstream*, capable of working simultaneously. And of course it needs to implement capture phenomena.

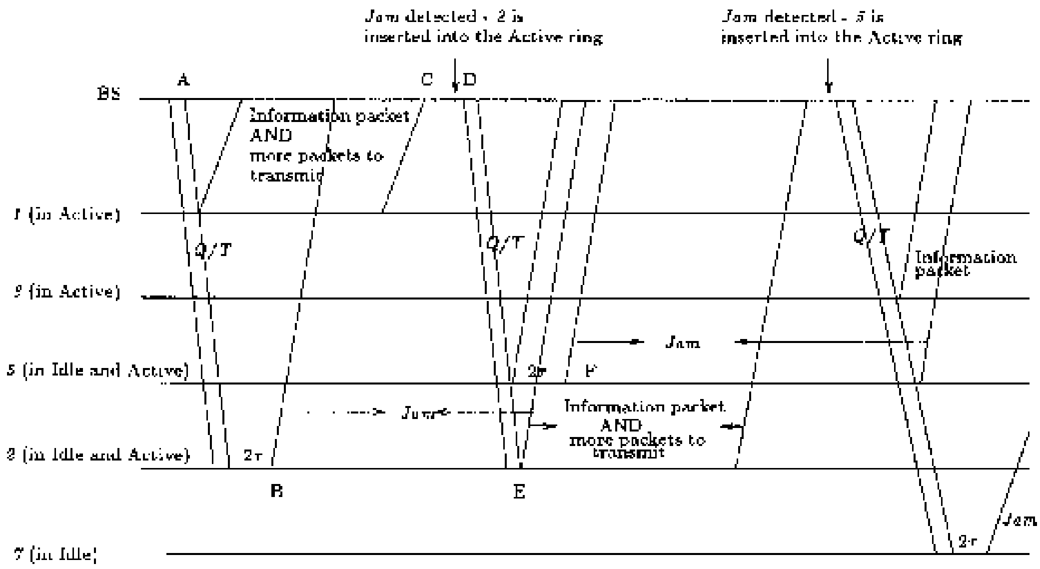


Figure 3.3: A Transmission pattern and ring updation

3.6.3 DDRR: Distributed Deficit Round Robin Scheduling

Ranasinghe, et, al [16] propose DDRR instead of basic round robin scheduling in PCF to provide QoS. In brief the question they are trying to answer is "How many voice and video connection can be accommodated in an 802.11 WLAN satisfying the imposed QoS requirements under different scheduling schemes". Work done doesn't have direct relationship with the problem we focusing. Their basic intention is to add fairness and increase capacity of WLAN. They use explicit signaling mechanism to decide whether to poll once, more times , or skip the station Finally they showed the impact of different polling strategy on capacity of IEEE 802.11 WLAN.

In DDRR, a Deficit counter (DC) is assigned to every station in the poll list. If the value of the i^th deficit counter (DC_i) is positive then scheduler allow i^th station to send packet. After transmission DC_i is decremented by length of packet. If DC_i is still positive and more data flag is set then scheduler continues to poll the same station. If DC_i is negative then scheduler does not send poll request until next CFP. In next CFP its deficit counter is incremented by its quantum Q_i.

3.6.4 How our approach is different

Our aim is to design an approach that completely follows the standard and require minimal modification to current implementation. Only signaling based approaches [6] are applicable to IEEE 802.11 and aims to reduce polling overhead. They require modification in current frame format and adds overhead of explicit signaling packet. We have proposed network monitoring based solution that observes and learns nodes behavior. We require to change the functionality of only PC. Other nodes in network remain unaffected and are not aware of any changes proposed by us.

3.7 Application in Ad hoc Networks

Closed-User-Group Multihop Ad Hoc networks have an increasing role to play in the future networks. Well known examples are military networks, disaster management networks, tourist information center, inquiry booth, etc. Such networks will almost certainly have one or more command and control centers and traffic will be skewed towards them i.e., most nodes will send traffic to the command/control center. Such a traffic pattern has not yet been studied in literature. Problems in DCF gets aggravated in such traffic pattern.

Our focus is on such networks. Central commander node that can be a access point, is surrounded by other nodes that communicate with it. Nodes are in the range of access point but they may not be in range of other nodes in network.

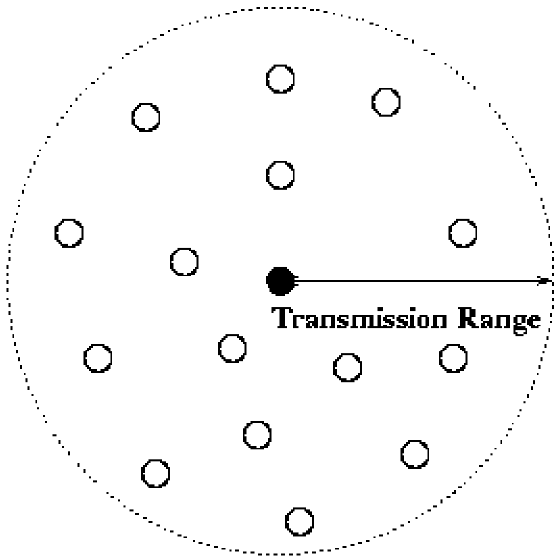


Figure 3.4: Central Node Scenario

Chapter 4

Optimizing PCF mode of IEEE 802.11 MAC

4.1 Why PCF

In recent years there has been an increasing trend towards personal computers and workstations becoming portable and mobile. People need the same service quality as in wired network. Future demands support of voice and other real time traffic. We believe PCF will better satisfy the future needs. Existing studies shows the PCF ability to provide better quality of service and support of voice and real time traffic. Upcoming standard for QoS in IEEE 802.11 MAC, 802.11e [17] also justify our keen interest in optimizing PCF. Malathi,et. al [18] discuss the support of voice services via PCF mode. We are not focusing on QoS issues, support of voice and real time data, etc. We have proposed generalised improvement in PCF that we believe will enhance existing IEEE 802.11 mac.

4.2 Solution Overview

As we said earlier in section 3.5 our proposed solutions are based on a network monitoring layer. We have only modified the PC functionality, rest nodes work as usual. At present network monitoring layer does very simple job of classifying nodes as active node and passive node on the basis of observed traffic. Figure 4.1 shows solution model at PC. We start with explaining Priority Round Robin Scheduling (PRRS) that aims to reduce polling overheads. PRRS replaces simple round robin scheduling in PCF with priority round robin scheduling. On observing results of PRRS, we design a protocol to further enhanced its performance by dynamically adapting CFP repetition interval. We have discussed CFP adaption algorithm after showing the simulation results of PRRS, in chapter 6.1. Dynamic Switching Protocol (DSP) is our next proposed protocol that aims at exploiting coexistence power of PCF and DCF and merges better half of both modes. We have suggested various criteria to decide switching point for dynamically switching between two modes PCF and DCF.

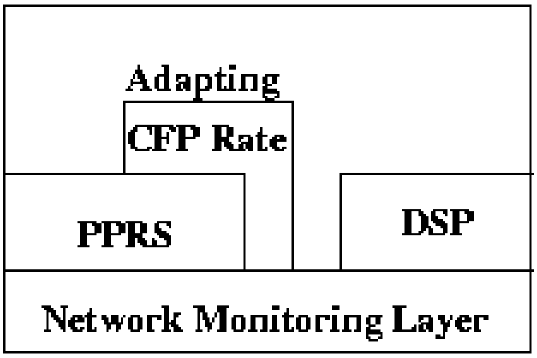


Figure 4.1: Solution model at PC

4.3 Network Monitoring Layer

Network Monitor which is PC or access point plays a vital role in both approaches. All stations in BSS need to first associate with PC. PC maintains circular list of all stations associated with it. Root of cause of degraded throughput at low load, when few nodes * are speaking, is unsuccessful polling attempts. To avoid such unsuccessful polling attempt one needs to have information whether node next in poll list would have data to transmit or not. Our network monitor PC attempts to do so. It classifies node into distinct categories namely:

- **Active Node :** Node having high probability to transmit data when being polled.
- **Passive Node:** Node having low probability to transmit data when being polled.

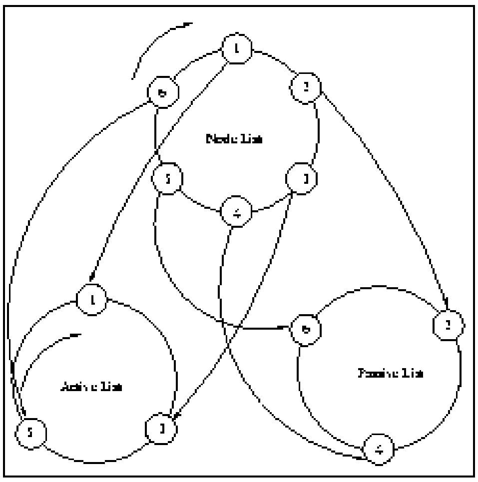


Figure 4.2: Example of stations/nodes in Active and Passive list. The order is imposed by the Node list.

We termed circular list of all associated nodes as node list. We define two logical list Active and Passive list. Active list contains all active nodes and passive list contains

*Word station, and node are used interchangeably.

all passive nodes. Consider figure and assume that stations 1,3, and 5 are classified as active nodes and may have packet to send. These stations belong to the Active list. Stations 2,4, and 6 are idle nodes and so they belong to passive list. The relative order in each list is the same as in the node list.

$$NodeList = ActiveList \cup PassiveList$$

$$ActiveList \cap PassiveList = \Phi$$

Initially we place all the nodes in active list.

4.3.1 Node List Management

In contention period(CP) of super frame, PC learns node's activity and at the beginning of each polling round in the next CFP, PC decides which new nodes to be added to active list. Node list management in contention free period (CFP) is shown in Figure4.3 Every station on active list retain its position in the list if it transmits data in response to poll. If station does not respond to poll or send null frame then it is termed as passive node and placed in passive list. Active node i retains its position as it sends data frame in response to poll frame, indicating higher probability of having more data. But node j is moved to passive list, as its null response reduces its priority.

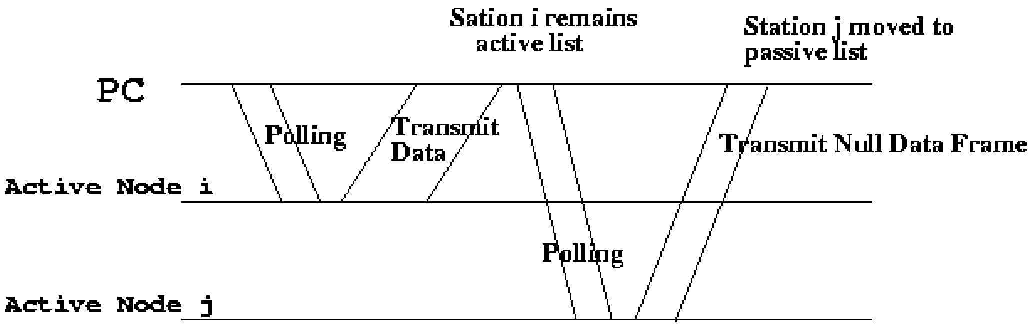


Figure 4.3: Transmission sequence and list updation in CFP

PC eavesdrops every packet transmitted in the CP and keeps count of number of data packets transmitted in CP. Count keeps importance in our DSP approach and its use will be discussed later. It examines MAC header to determine frame control type and subtype. If it is a RTS packet or data packet then PC marks source node as Listen node. Heuristic behind this is very simple. Node transmitting data or RTS packet has higher probability of sending more data if polled in CFP. In contrast node that remain silent during entire CP has lesser probability to have data. Obviously second part of heuristic does hold if CP period is small or node may not get opportunity to transmit or collision occurs, etc. We will discuss this issue in detail later.

Node in passive list remains there if it remain silent in entire CP period. If node in passive is marked Listen then it is placed in active list for next round of polling.

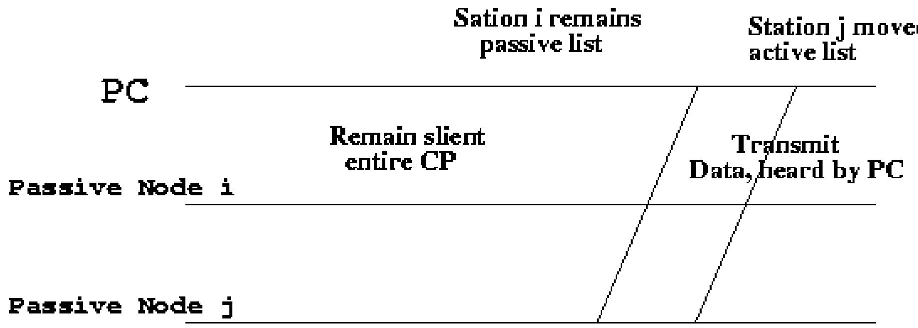


Figure 4.4: Transmission sequence and list updation in CP

4.4 PRRS-Priority Round Robin Scheduling

Instead of simple Round Robin scheme, we now define priority scheduling that uses two priority class. All nodes in active list posses same priority. Hence we poll them in simple round robin fashion. Nodes in Passive list are assigned low priority. Our existing PRRS does not schedule them at all.

PRRS suffers from unbounded waiting time. Waiting time can be bounded to certain extent if we use service differentiation feature in CP. Amad et. al [11] have suggested a simple scheme for service differentiation in DCF, based on:

- Different backoff increase function
- Different DIFS interval
- Different maximum frame length

Deng, et. al [14] have also proposed simple scheme based on the shorter IFS and the shorter random backoff time. If station is not polled in CFP and has data to transmit then it could have priority access to channel. As a result, in CP station that want them to be placed in poll list would have always high priority to access channel than station already in poll list.

Alternative way to place upper bound on waiting time is to use Bilevel Feedback Scheduling (BFS) . All nodes in active list are polled in Round Robin fashion. Nodes in Passive list are not polled at all. But in each CFP period we increase the priority of nodes in passive list by threshold value and when their priority reaches the level of active list then they are shifted to active list. We could also have different increment rates for different nodes depending upon their recent activity. By having different increment rates for different nodes, we are actually converting BFS to Multilevel Feedback Scheduling (MFS). We need to decide appropriate threshold value that balances polling overhead and waiting time.

Algorithm 1: Priority Round Robin Scheduling

$Active\_list \leftarrow \forall_{0 \geq i \leq N}$ add n_i where n_i is associated with PC and pollable
$Passive\_list \leftarrow$
if CFP and PC **then**
  for $n_i \in Active\_list$ **do**
    Poll n_i
    if Recv Null Data Frame or Poll time out **then**
      $Active\_list \leftarrow Active\_List - n_i$
      $Passive\_list \leftarrow Passive\_List + n_i$
Ensure: Nodes position in logical lists match their corresponding position in node list
    end if{Actual Implementation may have single node list with boolean parameter mark_don't_poll associated with each node. Set mark_don't_poll to true }
  end for
else if CP and PC **then**
  if Listen Packet **then**
    if Packet.subtype = MAC_Subtype_RTS or Packet.subtype = MAC_Subtype_DATA **then**
      $source\_node \leftarrow Packet.source$
      $Active\_list \leftarrow Active\_List + source\_node$
      $Passive\_list \leftarrow Passive\_List - source\_node$
    end if
  end if{Set mark_don't_poll to false }
end if

4.5 DSP-Dynamic Switching Protocol

Dynamic switching protocol is defined to exploit the better half of both DCF and PCF modes. Section 3.2 clears the need of switching protocol. At the broad level, we can say that in small sized network it is better to use the DCF, otherwise use the PCF. We define network size, not as the total number of nodes in BSS but as the number of active nodes. We have network monitoring layer at PC that attempts to approximate the network size. But it serves our purpose till, we follow the PCF access mode. So we need to have some extra mechanism in our existing network monitoring layer to approximate the network size in DCF mode.

4.6 Learning In DCF

Recalling our basic learning mechanism (section 4.3.1), PC classifies the node as an active node, if it successfully responds to the poll or sends RTS/DATA packet in CP. If node doesn't have data to send, when polled then it is transferred to the passive list. Criteria of classifying nodes as an active node is directly applicable. But in DCF mode, for how much time it should be kept in the active list. We need to move the nodes from the active list after bounded interval.

So, in DCF we need to define more criteria for node list management. Consider a situation, node A has data to send. As per DCF rules, it waits for DIFS time and then takes random backoff. Value for random backoff always lies in the range 0 to the size of current contention window (CW). If random backoff does not decrement to zero and medium becomes busy then node freezes the backoff and resumes it later. Considering this fact, We can say that if node has data to transmit then it will eventually get chance to transmit after fixed bounded number of transmissions that equals to backoff value. In worst case,the number of transmissions that can happen before the node gets it turn (after taking random backoff) is the size of its current contention window. Therefore, it can be said that if node does not speak (means send RTS/DATA) for bounded number of transmissions of DATA packets (size of contention window) then node have less probability to have data.

Now problem boils down to determine the average size of minimum contention window. We can assume that minimum CW size and maximum CW size is configured to be constant. Size of the current contention window gets doubled, every time an unsuccessful attempt to transmit an MPDU occurs. It happens till the CW reaches the its maximum defined value. After every successful transmission of MSDU or after fixed number of retransmission trials* , the value of CW is reset to minimum contention window size.

Let p be the probability of unsuccessful attempt to transmit MPDU, CW_{min} be the minimum contention window size and CW_{max} be the maximum contention window size. Each transmission attempt is independent of all previous attempts. Therefore, average contention window size is given by:

$$CW_{avg} = CW_{min} + p \times 2 \times CW_{min} + p^2 \times 4 \times CW_{min} + \quad \dots \quad + p^{level} \times CW_{max}$$

where *level* equals $\log_2 CW_{max} \quad - \quad \log_2 CW_{min}$.

We now extend our node list management protocol to incorporate above criteria of classifying nodes as an active or a passive. We believe that if node has pending data

* when (STA long retry count)SLRC reaches aLongRetryLimit, or when (STA short retry count) SSRC reaches dot11ShortRetryLimit.

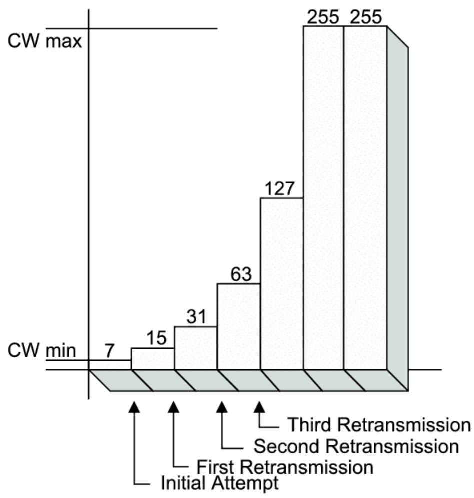


Figure 4.5: An example of exponential increase of CW

then it will contend for medium and would eventually get chance. to transmit. If node has pending data and is contending for medium then it has high probability to get the chance with in CW_{avg} transmissions. If any active station remains silent for cw_{avg} transmissions of MPDU in DCF then it is moved to passive list.

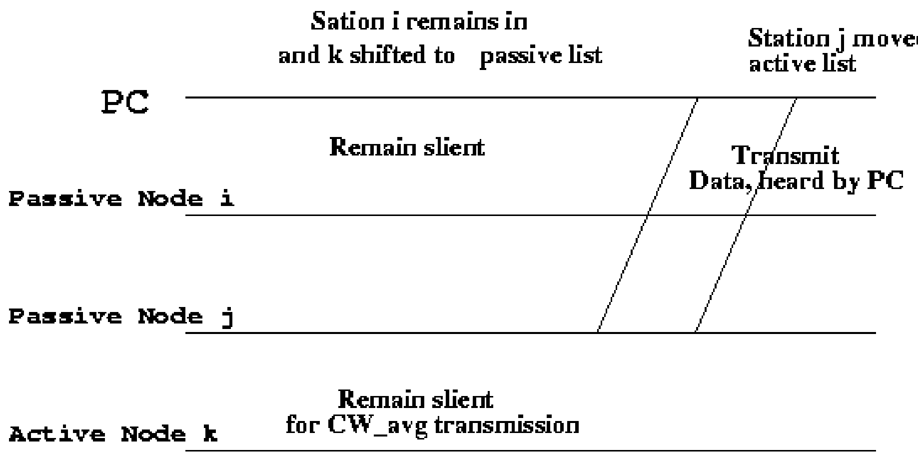


Figure 4.6: Extending learning to DCF

4.7 When to Switch

Switching point plays a vital role in DSP. It is very difficult to precisely answer the question that "What should be the optimum point for switching and how to measure that point?". We have depicted certain factors that may give some approximation of switching point. We will first discuss switching from PCF to DCF and then continued with switching from DCF to PCF.

In PCF mode PC being central coordinator, makes the designing of the switching protocol little bit easier. It can be assumed that PC can hear every other node in its

BSS. By two ways we can approximate network load and size. First by keeping track of number of active and passive nodes in the network (refer section 4.3.1). We can also approximate network load by keeping track of CFP utilization. Details of how to approximate load by CFP utilization can be found in section 6.1. There, we have used this criteria to dynamically adapt CFP repetition interval. We can also use both approaches simultaneously to precise our decision. Here we have used only our first basic learning approach.

In DCF mode, things becomes more complicated. How will keep track of node's activities, or how many node should do this, Who will be next PC, etc. To simplify things here, we have used the restricted version of this protocol. We require one fixed pre configured node to act as a PC. PC is made responsible for the network monitoring even in DCF. Different criteria for approximating load suggested by us are listed below:

- **Number of Active Nodes:** Section 4.6 explains in detail, how to figure number of active nodes in DCF. But question rises what should be the number for switching point. Wolisz, et. al have come with graph (Figure 4.7) that shows the switching point at different load. If we assume network on the average 50-70% loaded then switching point lies in range 10-15. Obviously we need to determine dynamically load approximation to maximize performance.

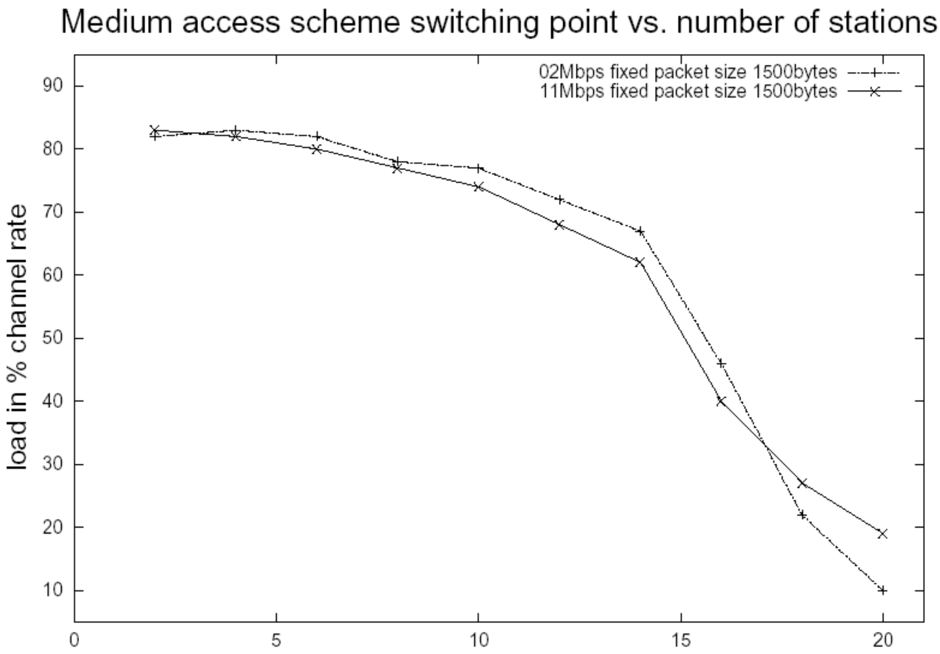


Figure 4.7: Extending learning to DCF

- **Network utilization** By our assumption, PC can hear all traffic in BSS, so it can actually calculate the network utilization percentage to dynamically approximate the network load.

- **RTS Failure** Number of RTS failure which means fail to receive CTS in response to RTS, gives us clear indication of contention in the network. If RTS failure is high, means more contending stations, then it is high time for the DCF to be replaced with PCF.

- **Number of Collision** Assuming center position for PCF, it can hear collision in its BSS. So we can use the fact to approximate load on the basis of number of collision.

Although we have depicted so many criteria for approximating network load, but still we are not in position to replace all fuzzy decision boundaries with actual switching points. Lots of simulation results and actual experimental data need to be generated, in order come up with good switching points. We decided to stick with graph (Figure 4.7) and to use only single criteria of number of active nodes.

4.8 Restricted DSP

Restricted version of DSP, imposes centralization of monitoring and decision making regarding switching point. Just as the PCF, DSP also needs the central coordinator termed as PC. PC can regarded as a head of network that decides fate of other station in BSS. All station in BSS needs to be associated with PC just as they do in PCF. This gives us the fair indication of total number of nodes in BSS. Station can change its association parameters by reassociation request and can also deassociate with deassociation request. Procedure for all these request is same as defined in PCF mode of IEEE 802.11 standard.

Initially network starts in PCF mode. Station associates with PC. PC starts monitoring network and decides whether to continue with PCF or switch to DCF. Beacon ‘transmissions are scheduled by PC at fixed regular interval.

In PCF mode decision of whether to switch to DCF or not is made at the time of sending beacon that starts CFP. So after every CFP repetition interval, PC can decide to switch between either modes.

4.8.1 Switching PCF to DCF

Basic intuition behind the protocol is to exploit the virtual carrier sense at the stations in BSS. PC announces CFP by setting the CFP MaxDuration field in the beacon. Stations in BSS, on the receipt of beacon set their NAV according to the CFP MaxDuration value. This action prevents stations from taking control of the medium during the CFP. If we can avoid setting of NAV at stations then stations can successfully clear virtual carrier sense. Eventually they can take control of medium through DCF access rules. Hence the network would behave as if, it is in DCF mode. If PC decides to switch to DCF then following steps are taken:

- Transmit beacon as it do in normal fashion, but with different set of CF parameter.
- Figure 4.8 shows CF parameter set in beacon frame. If PC instead of announcing CFP, want to switch to DCF then set CFMaxDuration, CF DurRemaining and CFP count to non zero high value.
- PC set its CFP count to infinite, set *mode* * variable DCF and set boolean variable *cfp* * false.
- PC now doesn’t transmit CFPoll, CF-END, etc.

* Both variables mode and cfp are required for bookkeeping, not a standard parameter.

	Element ID	Length	CFP Count	CFP Period	CFP MaxDuration (TU)	CFP DurRemaining (TU)
Octets:	1	1	1	1	2	2

Figure 4.8: CF parameter set in beacon frame

4.8.2 Switching DCF to PCF

Switching from DCF to PCF uses the same concept of setting NAV that disables the stations to take over medium at the beginning of CFP period. But certain things need to be changed to enable PC and to declare CFP as soon as possible whenever it wants to switch. In DCF beacon transmission is distributed but in our DSP we have priorities PC, such that it is whole and sole responsible for beacon transmission at fixed interval.

Every beacon transmitted in DCF by PC always contains values zero for CFP MaxDuration and CFP DurRemaining. To priorities PC in DCF for transmitting beacons, we have used IFS smaller than DIFS, i.e., PIFS. After PC senses medium free, it needs to wait for only PIFS time instead of DIFS time for other stations. PC decides whether to switch to PCF or not at the time to send beacon. If PC decides to switch to PCF then following steps are taken:

- Transmit beacon containing normal values
- PC set its CFP count to its original value(in PCF), set *mode* variable PCF and set boolean variable cfp true.
- PC now resume its normal functioning, same that it do in PCF after announcing CFP begin.

Bookkeeping helps PC to keep track of its current state. These variables are later by it decide its functioning.

4.9 Distributed DSP protocol

Distributed version of protocol is designed, keeping in view ad hoc networks. What we believe that the future advanced IEEE 802.11 LAN card will have both functionality built in and user will have choice to switch it on or off. Infrastructure need of access point will not be any more. As in ad hoc networks nodes are able to move freely and network size varies a lot. So we believe dynamic switching between either modes would give better performance result and would increase network capacity.

Each node will now monitor the network to keep track of its active neighbours. When number of active neighbours reaches a switching point, then nodes would start to form cluster and runs distributed leader selection algorithm to elect PC among them. That node will then switch on its PCF mode and would take the role of point coordinator. After wards our restricted DSP protocol would take care.

To make whole picture realistic, there exist lots of challenges that needs to be resolved. Our DSP is first step towards it. Dynamic PC coordinator selection requires a efficient leader selection algorithm that imposes very little overhead. Beside that there are many security issue regarding association, deassociation with old and new PC, that needs to be resolved.

Chapter 5

Simulation Results

Our simulations are done using the public domain network simulator NS-2 (2.1b8)[3]. Support for wireless simulations in ns was added as a part of the CMU Monarch project [4]. Support for the PCF mode of IEEE 802.11 already exists. [2]. PCF patch added by Lindgren, et. simulates only limited PCF features. We have simply extended some feature of existing PCF patch like:

- We added the support for Null data frame that need to be sent in response to poll, if station have no pending data. Previously it was resolved via poll timed out at PC.
- We added support for sending broadcast packet in CFP. Existing implementation simply drops such packet in CFP

Support for association, deassociation and reassociation still have not been added. Presently nodes need to be associated through tcl script. Since we assume nodes remain in range of PC all the time, therefore static association simply serves our purpose.

5.1 Simulation Setup

Our studies are confined to a single cell of radius 240m, slight less than the transmission range of central coordinator. Conceptually every station in region can communicate directly with central node. However, transmission medium degradations due to multipath fading, or interference from nearby BSSs reusing the same physical-layer characteristics can cause some stations to appear hidden from other stations. In our simulations we are working with only one BSS, a clean channel without errors and fading effects etc., so all stations can indeed communicate directly with PC.

We have used the default values for all the physical and MAC layer parameters. The number of stations other than PC in circular cell is varied from 8 to 64 asynchronous data user. Nodes are placed randomly around PC. All our runs are averaged over ten such random placements. At stations, we attached a cbr source that simulates arrival of frames for transmission at constant rate. Packet size is kept constant at 500 bytes for most simulations, except when throughput is studied as a function of packet size. The choice of 500 bytes as a packet size worth studying is motivated by the fact that we consider messaging applications to be appropriate for wireless networks.

Parameter	Value
Transmission Power	281.8mW
Transmission Range	250m
Slot Time	$20\mu s$
SIFS	$10\mu s$
Channel Bandwidth	2Mbps
Number of Stations	Varied from 8 to 64
Central Coordinator	1
Packet Size	500 bytes
RTS/CTS threshold	250 bytes
Fragmentation threshold	2346 bytes
CW_Min	31
CW_Max	1024
CFP repetition interval	Varied from 50 to 400 TUs
Time Unit (TU)	1024μs

Table 5.1: Simulation Parameters

5.2 Simulation Results of PRRS

We simulated PRRS for 300 seconds using different traffic loads and network size. Nodes placement is total random. In all simulations for PRRS, we have changed the sources dynamically. Consider graph in figure 5.1, we have three different set of 8 sources. Source nodes of each set send packets for 100 seconds. All 8 cbr sources start and end at same time.

5.2.1 Throughput

We define throughput as the total number of bits per second passed up from the MAC sublayer at each destination. Then we present it as a fraction of channel bandwidth. So the throughput what we measure here is actually the goodput, because control frames, management frames,routing packets, header size, etc. are not counted. Similarly, we offered load as the average number of bits per second of actual data offered to the MAC sublayer at each source. It is then represented as a fraction of channel bandwidth.

Figure 5.1 shows PRRS result with 25% node active and total 32 nodes in BSS. Highest throughput achieved by PRRS is $0.637534 \times 2000000 = 1275068$ bps and with RRS is $0.585006 \times 2000000 = 1170012$ bps. Around 10% increase in goodput is achieved. Considering the same ratio of active nodes and total nodes, with 64 nodes (figure 5.2 Throughput difference is highest when offered load is 80%. Throughput achieved by PCF with RRS at 80% offered load is $0.511695 \times 2000000 = 1023390$ bps and by PCF with PRRS is $0.592234 \times 2000000 = 1184460$ bps. Throughput improved by 15.7% approximately.

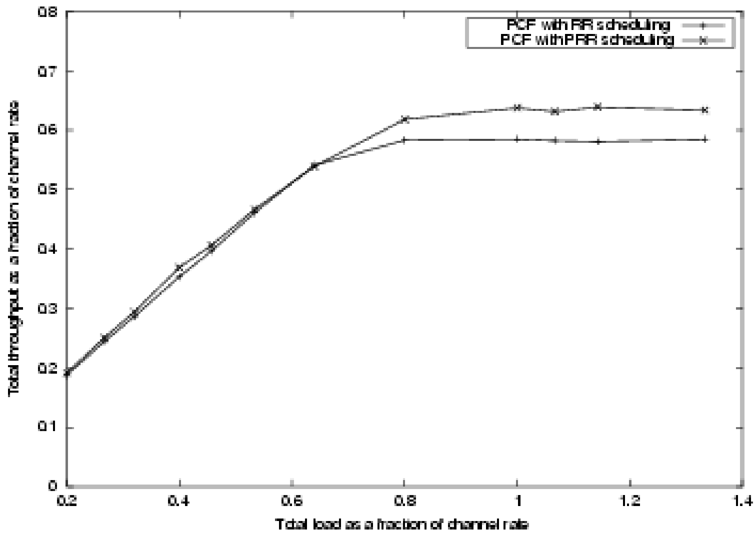


Figure 5.1: Throughput Comparison between PCF with PRRS and non optimized PCF with RRS. Total node =32 and 25% active node

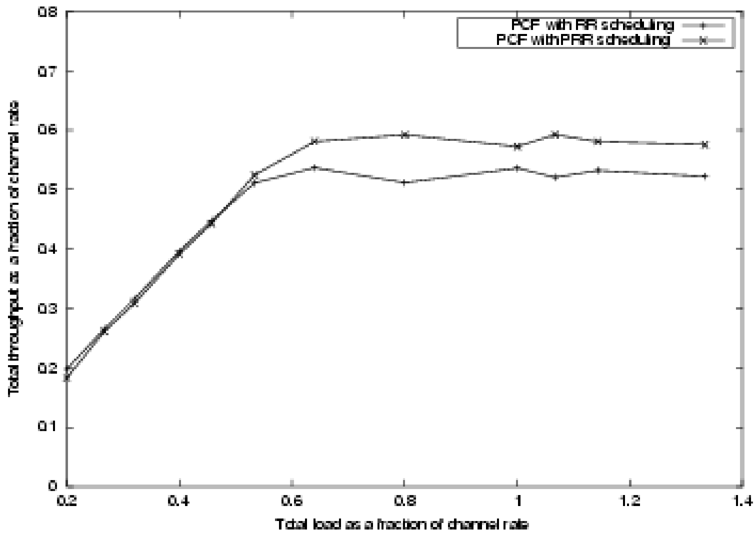
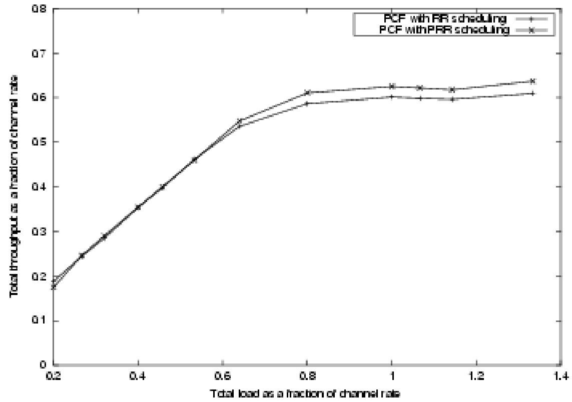
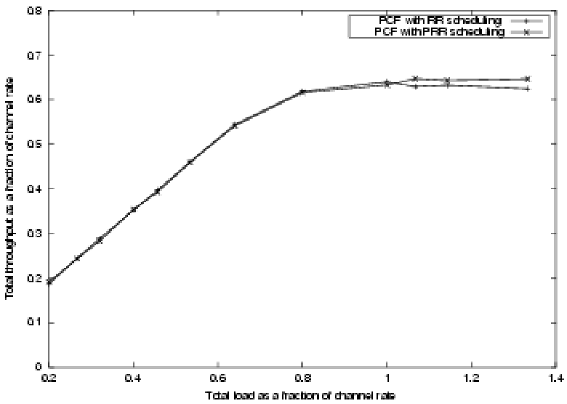


Figure 5.2: Throughput Comparison between PCF with PRRS and non optimized PCF with RRS. Total node =64 and 25% active node

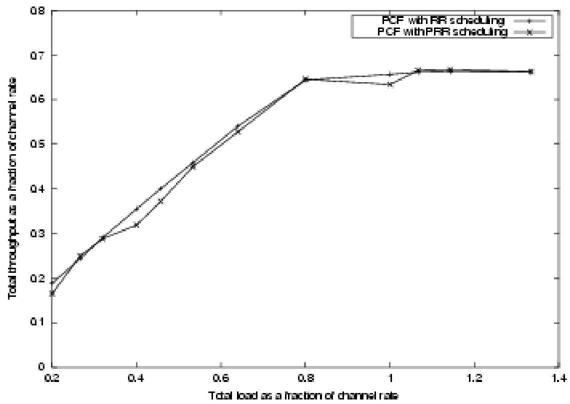
Figure 5.3 shows the throughput comparison with 32 nodes and among them 50% 75%100% active. As number of active nodes increases throughput difference decreases. Figure 5.4 shows the throughput comparison with 64 nodes and among them 50% and 75% active.



(a) 50% Active nodes

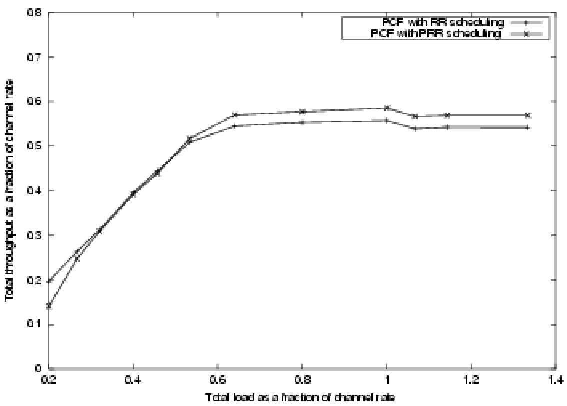


(b) 75% Active nodes

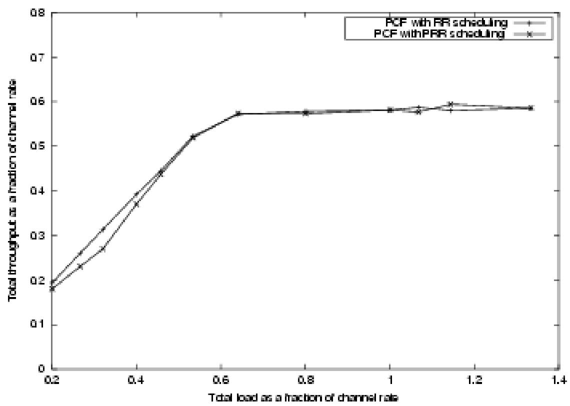


(c) 100% Active nodes

Figure 5.3: Throughput comparison of PRRS against non optimized PCF with 32 nodes



(a) 50% Active nodes



(b) 75% Active nodes

Figure 5.4: Throughput comparison of PRRS against non optimized PCF with 64 nodes

5.2.2 Delay

Delay here is measured as end to end delay at agent layer. We have used DSDV as a routing protocol. Since DSDV [1] is proactive and our runs are limited to single hop and do not involves mobility, so routing overhead can be assumed to be constant. Hence measurement of agent layer end to end delay is justifiable and it can be said that differences are significantly due to MAC performance.

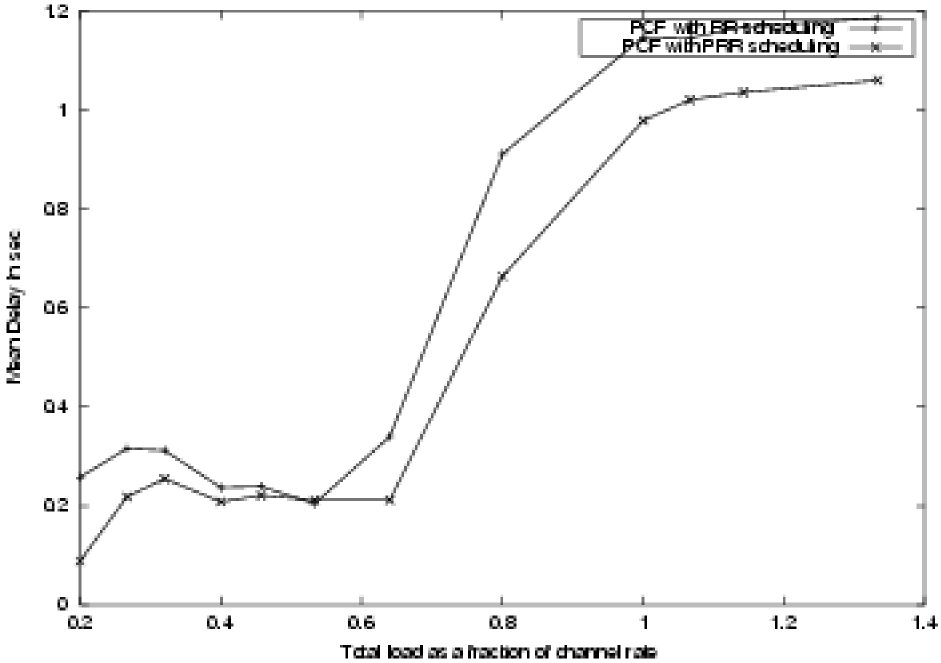
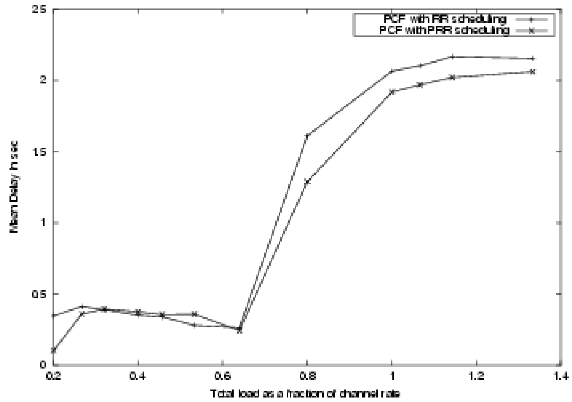


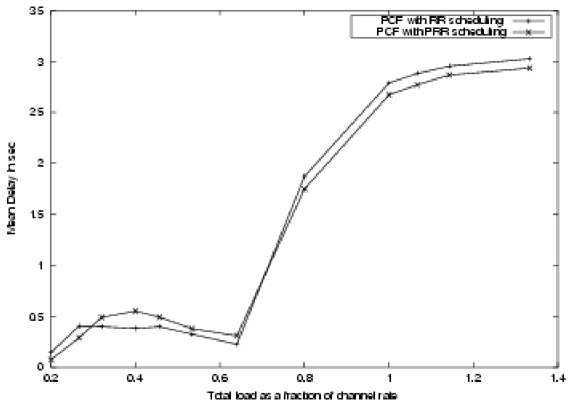
Figure 5.5: Average Delay comparison between PCF with PRRS and PCF with RRS. Total node =32 and 25% active node

Figure 5.5 shows PRRS delay result with 25% node active and total 32 nodes in BSS. Mean delays have reduced as unsuccessful polling attempt has reduced. As a result, active nodes get next chance to transmit early. But if we observe delay graph (Figure 5.6) with 50%, 75%, and 100% nodes active, delay values for PRRS increases and also it becomes more than RRS at some points.

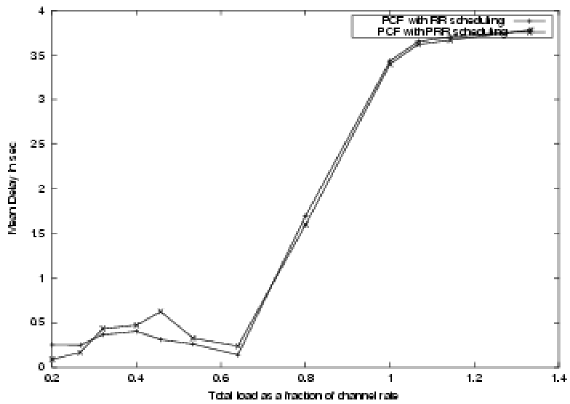
We think reason behind this is the problem that we stated in section 4.4. Competent nodes may not be getting chance to send data in CP as result they are not added to poll list. This delays their chance to send till next CFP or CP. Figure 5.7 compares mean packet delays with 64 nodes and 25% and 50% nodes active.



(a) 50% Active nodes

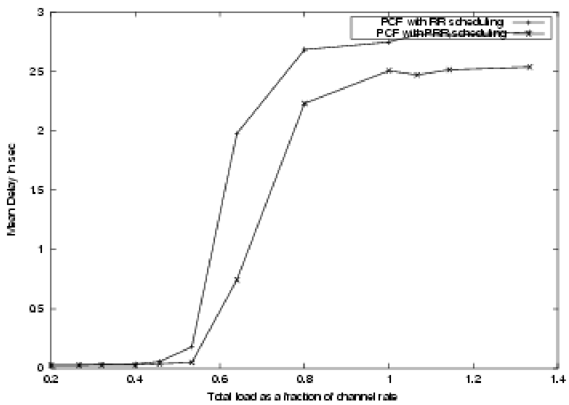


(b) 75% Active nodes

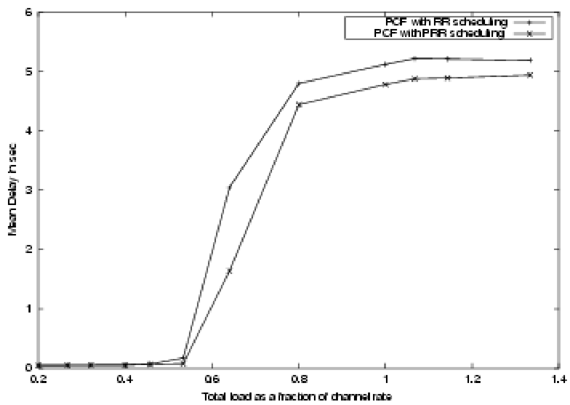


(c) 100% Active nodes

Figure 5.6: Delay Graphs with 32 nodes and among them 50%, 75%, and 100% active.



(a) 25% Active nodes



(b) 50% Active nodes

Figure 5.7: Delay Graphs with 64 nodes and among them 25% and 50% active

5.3 Simulation Results of DSP

We simulated DSP for 350 seconds. We changed the traffic pattern for DSP simulation. Previously all cbr connection starts and ends at same instant of time but now we have introduced 1ms time gap between starting time of cbr connections. We have placed nodes randomly and averaged the reading over 10 such random placements. Switching point used by us is:

- DCF till number of active nodes less than equal to 10.
- PCF when number of active nodes exceeds 10.

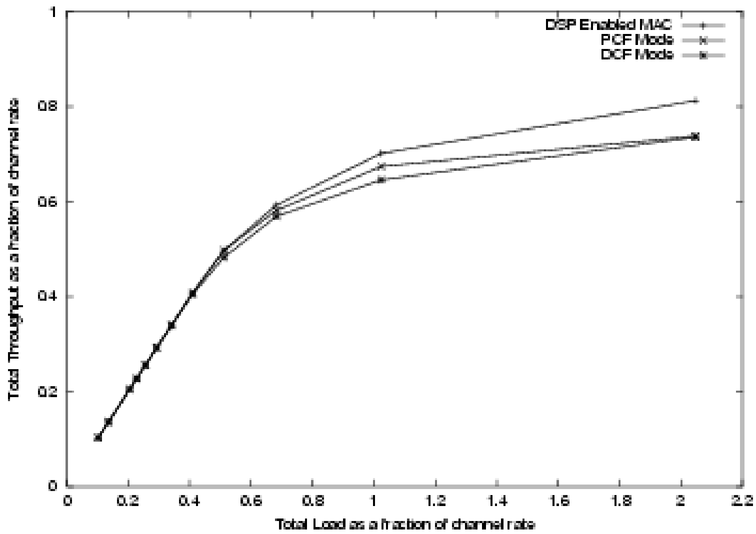


Figure 5.8: Throughput comparison of DSP with PCF and DCF

Graph shown in figure 5.8 shows the throughput comparison of DSP with PCF and DCF mode. We simulated with 16 nodes in BSS. Number of active nodes is varied form 4 to 16. Traffic load is defined as

- From 0 to 50 sec, we have 4 cbr connections
- From 51 to 100 sec, we have 8 cbr connections
- From 101 to 150 sec, we have 12 cbr connections
- From 151 to 200 sec, we have 16 cbr connections
- From 201 to 250 sec, we have 11 cbr connections
- From 251 to 300 sec, we have 7 cbr connections
- From 301 to 351 sec, we have 4 cbr connections

On x-axis we have varied packet inter arrival time from 200 ms to 10 ms. Result shows the throughput improvements on using DSP. Graph in figure 5.9 shows the delay curve for DSP. DSP offers lower mean packet delays than both PCF and DCF.

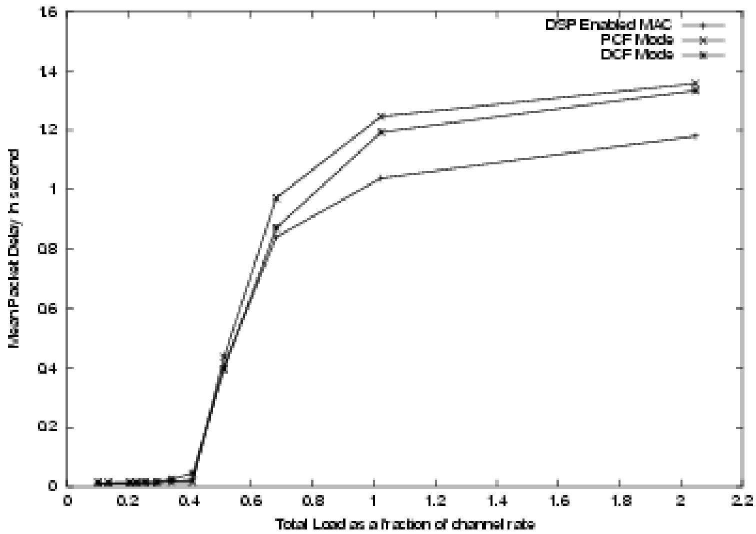


Figure 5.9: Delay comparison of DSP with PCF and DCF

5.4 Result Summary

PRRS shows better results than PCF with RRS, especially when less than 75% node have pending data to send. It suffers from higher delays when percentage of active nodes reaches 75% and more. We have discussed possible reason for this in section 4.4. DSP requires extensive experimentations. We have seen improvements in both throughput and mean delay.

We have used different CFP repetition intervals, while experimenting with PRRS. We varied the parameter in accordance with number of nodes in network. In next chapter we will show effect of this parameter on PCF performance.

Chapter 6

Adapting CFP under varying Load

In this chapter we concentrate on the dynamic adaption of CFP repetition interval. Basic intuition behind the use of different CFP repetition interval in our simulations (Section 5.2) for different number of nodes in network,is to poll nodes in CFP exactly once and to have enough CP period left for the PC to learn nodes activity. So that more more competing station can be added to active list. Big CFP repetition interval implies large CFP_Max duration. After single round of polling, PC transmits CF_End and starts the CP that lasts till the next CFP. Large CP, no doubt gives the PC more time to examine nodes activity but network remains in DCF mode for that longer period.

Similarly small CFP repetition interval implies small CFP_Max duration. So, it may happen that PC may not be able to poll all active nodes at least once. CP period is also less because whole CFP_max may be utilized. This reduces the chance for competing stations to get enrolled in active list. Hence stations are polled after longer delays. We statically calculated CFP repetition interval for our runs. Let total number of nodes in network be 32 and mean packet size be 500 bytes. Time require to poll all stations assuming each will have data to send is:-

$$\begin{aligned} Total\_Time &= 32 * T_Poll \\ &= 32 * 3SIFS + T_CFpoll + T_Data \\ &= 32 * 2.5TUs \\ &= 80TUs \end{aligned}$$

It takes 80 TUs to poll all 32 stations. Hence, we took CFP repetition interval as 100 TUs for such scenario.

In short, arbitrary chosen value for CFP repetition interval may deviate performance far from achievable Effect of CFP repetition interval on throughput can be properly analysed from graphs. Simulation setup is same as we described in section 5.1.Graph 6.1 shows that with 32 active nodes better throughput is achieved when CFP repetition interval is 100 TUs. At peak load story is slightly different. CFP repetition interval equals 50 TUs gives better throughput than rest. We think more throughput value may be because, nodes always have data to send, so low CFP repetition interval like 50 TUs allows to poll at lower delays. But we think that it may cause unfairness with other stations. Since network remains in CP for less time, so other competent nodes get less time to get enrolled in active list while stations in active list get polls at lower delay.

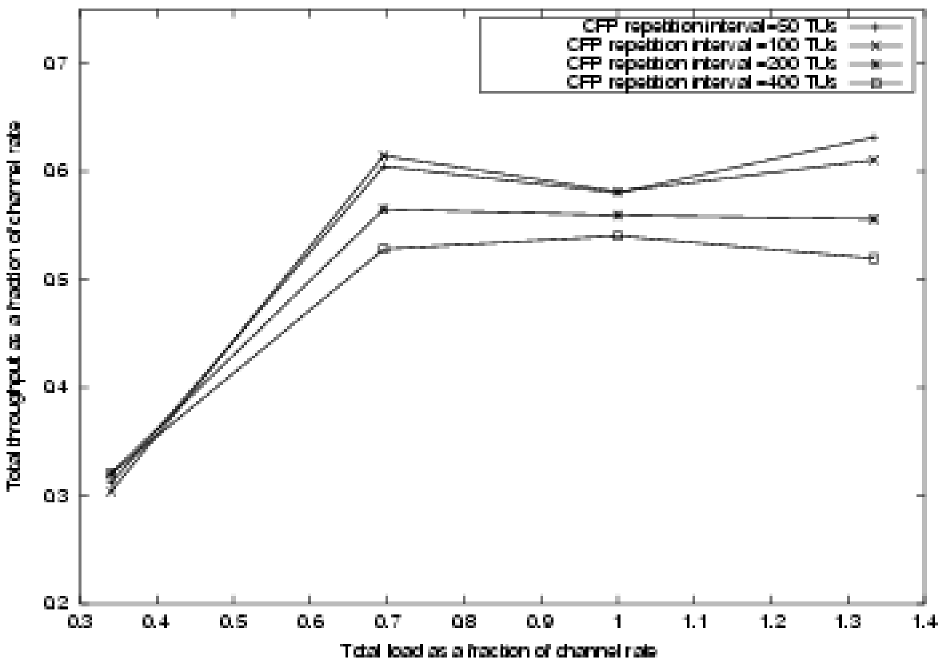
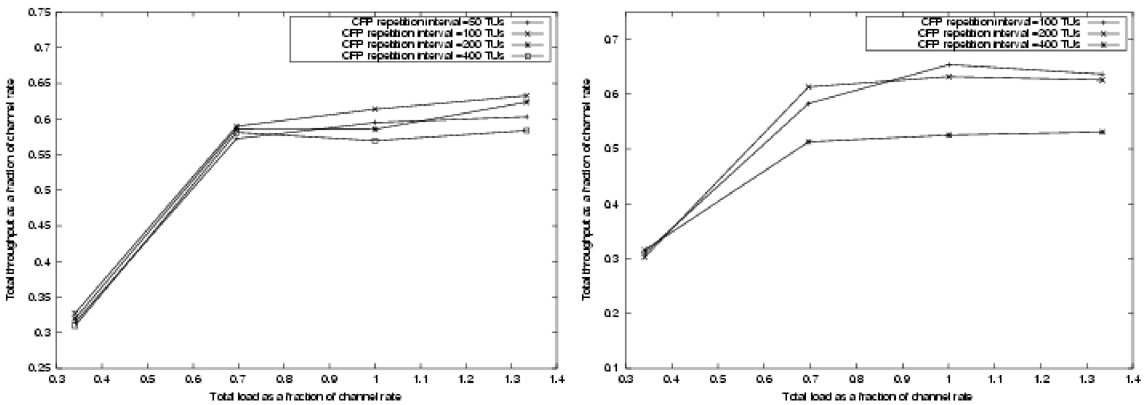


Figure 6.1: Throughput in PCF with PRRS for different CFP repetition interval with 32 active nodes

We need to noted that there is high variation in throughput by changing just single configuration parameter (CFP repetition interval). At 70% load, throughput acheived with CFP repetition interval equals 100 TUs is approximately 61.5% and with 400 TUs it is only 52.8%. This clearly justifies our argument of dynamic adaption.

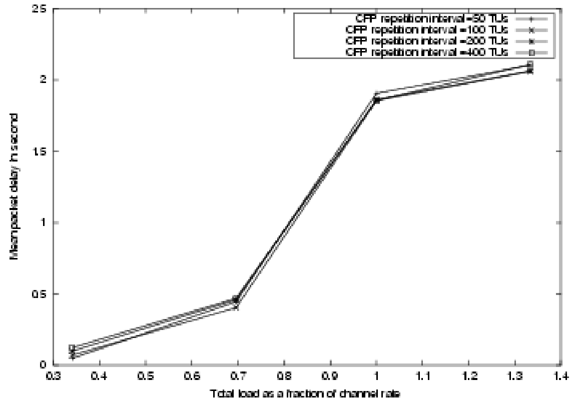


(a) 16 Active nodes

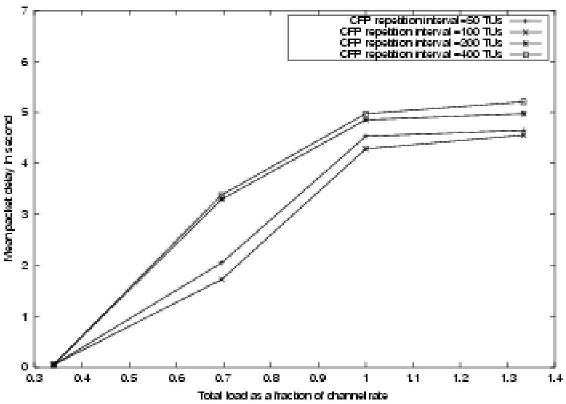
(b) 64 Active nodes

Figure 6.2: Throughput in PCF with PRRS for different CFP repetition interval with 8,16 and 64 active nodes.

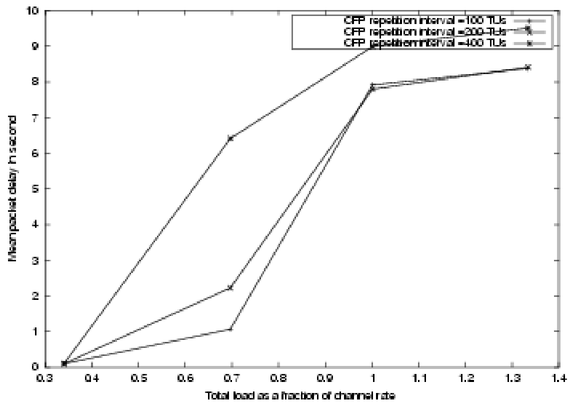
As the number of active nodes changes, better CFP repetition interval value varies. Graphs in figure 6.2 shows the dependence of good value on the number of active nodes. Same arguments holds for the delay values. Graph in figure 6.3 shows the delay curves



(a) 16 Active nodes



(b) 32 Active nodes



(c) 64 Active nodes

Figure 6.3: Mean packet delays in PCF with PRRS for different CFP repetition interval with 8,16 and 64 active nodes.

for different number of active nodes.

6.1 CFP Adaption

Standard states that each beacon transmitted during CFP or starts CFP, always contains information regarding the remaining CFP duration or CFP max duration respectively. Stations hearing beacon, set their NAV according to announced CFP duration. We utilize this fact to further optimize PRRS and dynamically adapt CFP repetition interval under varying load. Our solution to dynamically adapt CFP repetition interval is also applicable to PCF with RRS.

Although we have notion of number of active nodes in network but assumption of average packet size may not be correct. So statically calculated CFP repetition interval may not be optimal. We take different path of measuring CFP utilization to define new CFP repetition interval. Basic purpose is to keep to keep the network in CFP for maximum possible time and to ensure fairness for passive nodes by having enough CP length and CP repetition interval.

Algorithm 2: CFP Adaption

```
PollsToSend ← Number of station in active list {Decrement PollsToSend by one before polling a node}
CFP_Start ← Clock.time at which beacon transmission starts CFP
CFP_end ← Clock.time at which CF_END transmitted
temp ← CFP_end − CFP_Start × 100/CFP_Max_Duration
CFP_utilization ← 0.8 × CFP_utilization + 0.2 × Current_CFP_utilization {Averaging CFP_utilization }
if CFP_utilization ≤ 50 then
    CFP_repetition_count ← ⌊CFP_repetition_count/2⌋
Ensure: CFP_repetition_interval ≥ MinCFP_repetition_interval
Ensure: CFP_repetition_interval is multiple of beacon interval
else if CFP_utilization ≥ 80 or PollsToSend ≠ 0 then
    Increment CFP_repetition_count {CFP repetition interval is defined as CFP_repetition_count × beacon_interval. Increment will add beacon interval to it }
else
    Decrement CFP_repetition_count
end if
```

We poll all active nodes once and declare CFEnd. CFP utilization is calculated as percentage of CFP max duration utilized in polling i.e., time spent between CFP_Begin and CF_End. If PC scans the poll list completely i.e., if we observe that the CFP is underutilized then we increase CFP repetition interval. Similarly, if CFP max duration exhausts and PC fails to complete the entire poll list then we decrease the CFP repetition interval.

CFP repetition interval is defined as a multiple of beacon interval. Our CFP adaption algorithm 2 includes both exponential and linear updates, depending upon percentage of CFP utilized. By linear update, we mean addition or subtraction of beacon interval. By exponential update, we mean division by two. Every update ensures that new value is also a multiple of the beacon interval and it does not cross the minimum boundary defined by the standard. Decision for new value is not entirely based on current observation. To avoid frequent change, we use weighted average technique with 80% weightage to averaged reading and 20% weightage to current one.

$$New\_CFP\_utilize = 0.8 \times Old\_CFP\_utilize + 0.2 \times Current\_utilization$$

We define CFP repetition interval as $CFP\_repetition\_count \times beacon\_interval$. If utilization is $U\%$ then new CFP repetition count C is defined as

$$C = \begin{cases} \lfloor C/2 \rfloor & U \leq 50\% \\ C+1 & U \geq 80\% \text{ or CFP ends with more nodes to poll} \\ C-1 & \text{else} \end{cases}$$

6.2 Simulation Results

We have used same simulation scenario as used in section 5.2. Figure 6.4 shows the performance of CFP adaption enabled PCF. It adapts very well at moderate load, and the delay characteristics (fig. 6.5) have also improved.

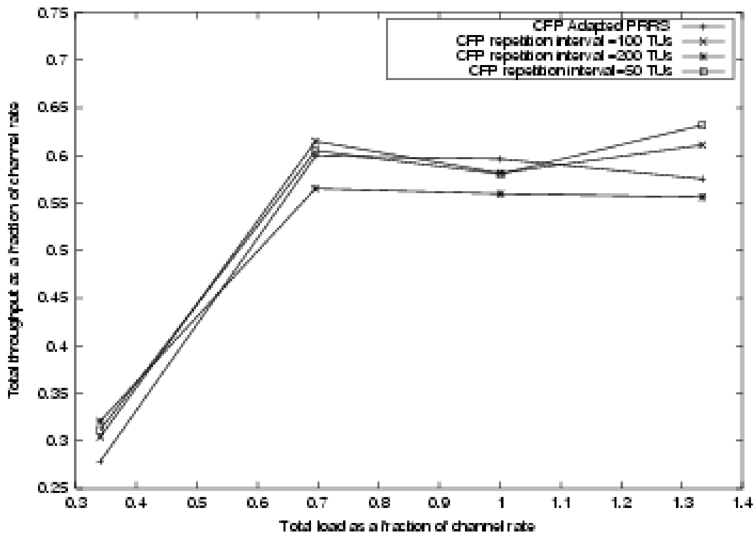


Figure 6.4: Throughput comparison of PCF with PRRS for different CFP repetition interval with CFP adaption enabled PCF with PRRS

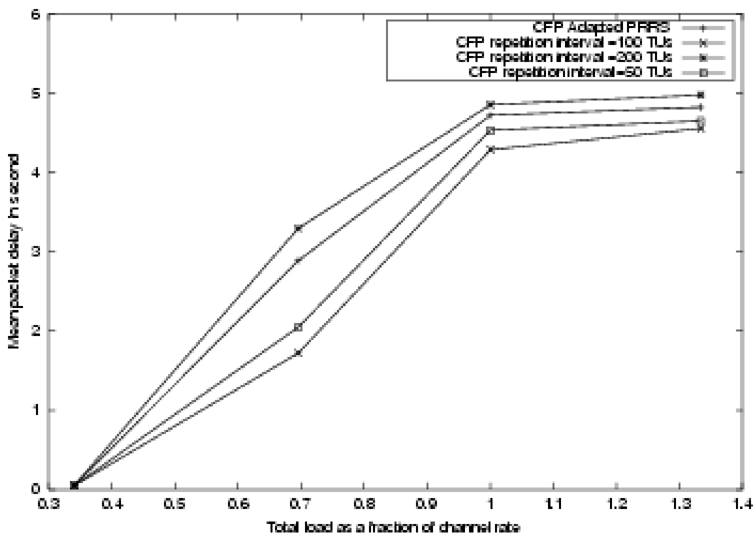
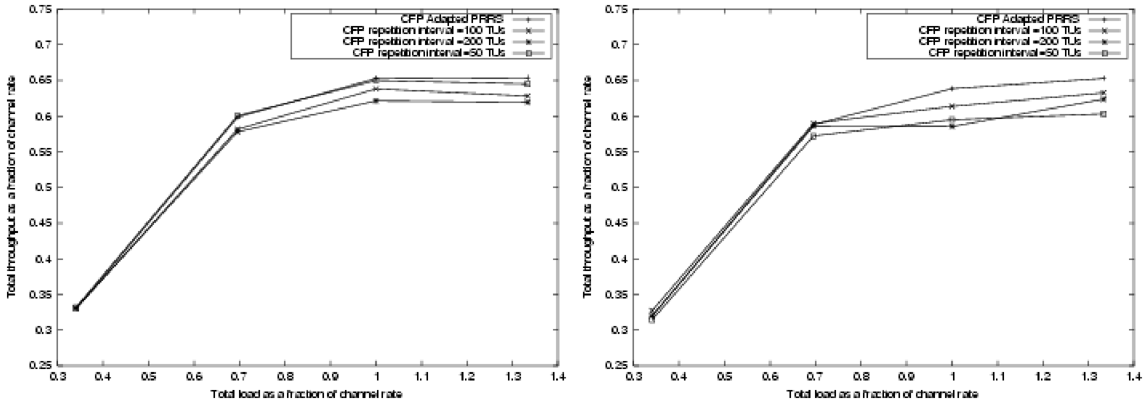


Figure 6.5: Delay characteristic of PCF with PRRS for different CFP repetition interval with CFP adaption enabled PCF with PRRS

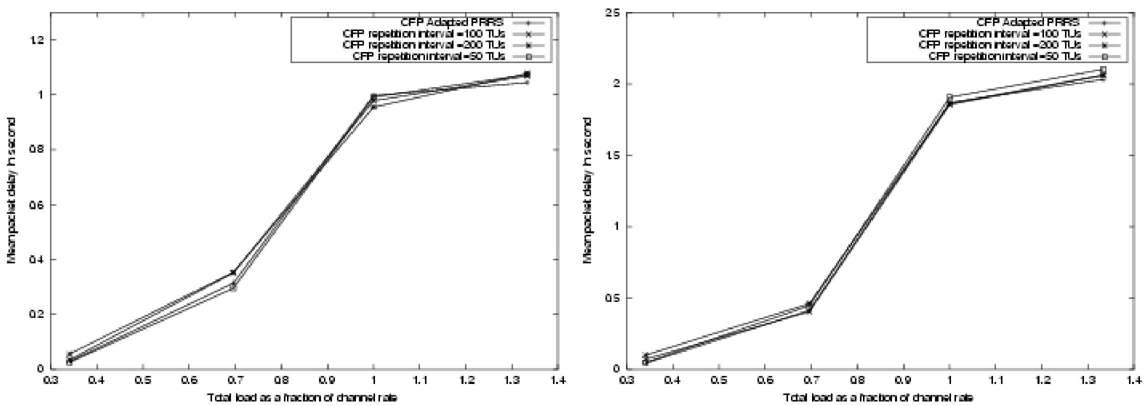
Figure 6.6 shows results with 8 and 16 nodes active. Their delay characteristic are shown in figure 6.7



(a) 8 Active nodes

(b) 16 Active nodes

Figure 6.6: Throughput comparison of PCF with PRRS for different CFP repetition interval with CFP adaption enabled PCF with PRRS, having 8 and 16 active nodes.



(a) 8 Active nodes

(b) 16 Active nodes

Figure 6.7: Delay characteristic of PCF with PRRS for different CFP repetition interval with CFP adaption enabled PCF with PRRS, having 8 and 16 active nodes.

Chapter 7

Conclusion and Future Research

7.1 Conclusion

IEEE 802.11 MAC needs dynamic adaption to enhance its performance. Static configured MAC performance deviates a lot from achievable limit. **We have suggested a network monitoring based approaches to approximate the network size and load and dynamically adapt MAC**. Our approaches add very little overhead and strictly follows the standard, without demanding any change in existing frame formats and access procedures. The best thing about our approaches is that, they add just one additional network monitoring layer at access point (PC) and rest all stations functionality remain unchanged.

PRRS that replaces simple round robin scheduling in PCF, significantly overcomes the efficiency of the polling schemes especially when small fraction of stations have data to transmit and when the traffic load is moderate. We have achieved around 10% to 15% improvement in throughput. By reducing unsuccessful polling attempts when few nodes in BSS have data to transmit, it reduces mean packet delays. This makes it more suitable for handling real time data and multimedia traffic.

DSP that defines protocol for dynamic switching between PCF and DCF, opens a new door to exploit coexistence of DCF and PCF mode and to mix better half of both the modes. We have also provided various ways to approximate size and traffic load, for defining ideal switching point. Our idea of distributed DSP would increase the network capacity and enhance performance in an ad hoc networks.

We have showed the need for dynamic adaption of CFP repetition interval for ensuring both better throughput and the fairness. Around 10-20% throughput variation is observed by using different configuration. Our **CFP Adaption** protocol successfully adapt CFP rate to suit current network load. CFP adapted PCF has achieved performance almost close to or even better than statically configured PCF.

7.2 Future Research

Our current version of *PRRS* introduces slight more delays for some nodes, when number of active nodes * approaches total number of nodes. We need to implement service differentiation mechanism to priorities nodes that have not been polled in CFP, to send data packet in CP. There is need to explore alternative Bilevel feedback scheduling policy (Section 4.4). Whether Bilevel feedback scheduling is sufficient, when we need multiple

*nodes have data to transmit

feedback scheduling, how to adjust number of levels in feedback scheduling dynamically depending upon current traffic load and network size, etc., are still open for research and further experimentation.

Restricted version of *DSP* needs better approximation of traffic load to define switching point. We have suggested various alternative for that, but there is need to extensively explore these options and do rigorous experiments. Designing the distributed version of DSP is a great challenge in itself. It require robust protocol for clustering of nodes and PC selection. Many security related issues * need to be resolved for deploying distributed DSP.

CFP Adaption algorithm is just a first step towards dynamic adaption of configuration parameters. Algorithm needs to be refined further. Using the network monitoring layer, we need to design protocol for adapting other configuration parameters like minimum CW size, RTS/CTS threshold,etc.

*Regarding association and deassociation with newly selected PC

Bibliography

[1] Charles E. Perkins, Pravin Bhagwat, *Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers*, ACM SIGCOMM Computer Communication Review, v.24 n.4, p.234-244, Oct. 1994.

[2] Anders Lindgren and Andreas Almquist, *Implementation of the PCF mode of IEEE 802.11 for the NS-2 Network Simulator*, http://www.sm.luth.se/ dugdale/index/software.shtml

[3] The VINT project, *NS notes and documentation*, editors Kevin Fall and Kannan Varadhan, http://www.isi.edu/nsnam/ns

[4] Josh Broch, David A. Maltz David B. Johnson,Yih-Chun Hu and Jorjeta Jetcheva, *A Performance Comparison of Multi-hop Wireless ad hoc Networking Routing Protocols*, Proceedings of Mobicom,October 1998.

[5] Shugong Xu, Tarek Saadawi *Does IEEE 802.11 MAC Protocol Work Well in Multihop Wireless Ad Hoc Networks?*, IEEE Communications Magazine, p.130-137, June 2001.

[6] Andreas Kopsel, Jean-Pierre Ebert, and Adam Wolisz, *A Performance Comparison of Point and Distributed Coordination Function of an IEEE 802.11 WLAN in the presence of Real-Time Requirements*, Proc. of 7th Intl. Workshop on Mobile Multimedia Communications (MoMuC2000), October 23-26, 2000.

[7] Leena Chandran-Wadia, Shruti M. and Sridhar Iyer. *Throughput Performance of the Distributed and Point Coordination Functions of an IEEE 802.11 Wireless LAN* , ICCC'2002, Aug. 11-4, Mumbai, India

[8] Brian P. Crowe, I. Widjaja, J. Kim, P. Sakai, *IEEE 802.11 Wireless Local Area Networks*, IEEE Communications Magazine, September 1997.

[9] Vaduvur Bharghavan, Alan Demers, Scott Shenker and Lixia Zhang. *MACAW: a media access protocol for wireless LAN's* in Proc. ACM SIGCOMM '94, pp. 212-25, London, UK, Aug. 31–Sept. 2, 1994.

[10] IEEE Std. 802.11, *Wireless LAN Media Access Control (MAC) and Physical Layer (PHY) Specifications*, 1999.

[11] I. Aad and C. Castelluccia. *Differentiation mechanisms for IEEE 802.11* In Proceedings of IEEE INFOCOM 01,Anchorage, Alaska, April 2001.

[12] Frederico Calì, Marco Conti, and Enrico Gregori *Dynamic tuning of the IEEE 802.11 protocol to achieve a theoretical throughput limit*, IEEE /ACM Transactions on Networking, 8(6):785-799, 2000.

[13] Harshal S. Chhaya and Sanjay Gupta. *Performance modeling of asynchronous data transfer methods of IEEE 802.11 mac protocol*, Wireless Networks,3(3):217-234, 1997.

[14] J. Deng and R. Chang. *"A priority scheme for IEEE 802.11 DCF access method"* IEICE Trans. Comm. E82-B, (1): 96-102, Jan. 1999.

[15] Oran Sharon, Eitan Altman *An efficient polling MAC for Wireless LANs* IEEE/ACM Transactions on Networking 9(4):439-451, (2001).

[16] Ravindra S. Ranasinghe, Lachlan L. H. Andrew, and David Everitt, *Impact of polling strategy on capacity of 802.11 based wireless multimedia LANs.* In Proc. IEEE Int. Conf. On Networks ICON,Brisbane, Australia, 1999.

[17] P. May, O. Klein, G. Hiertz, S. Mangold, S. Choi and L. Stibor, *IEEE 802.11e wireless LAN for quality of service (invited paper)*, In Proceedings of the European Wireless, volume 1, pages 32-39, Florence, Italy, February 2002.

[18] Malathi Veeraraghavan, Nabeel Cocker, and Tim Moors: *Support of voice services in IEEE 802.11 wireless LANs.* INFOCOM 2001 : 488-497

[19] Matthijs A. Visser and Magda El Zarki *Voice and data transmission over an 802.11 wireless network.* Proceedings of the 6th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC'95, pages 648-652, Sept. 27-29 1995.

[20] J.Weinmiller, H.Woesner, J. P. Ebert, and A.Wolisz. *Analyzing and tuning the Distributed Coordination Function in the IEEE 802.11 DFWNAC draft standard.* MASCOT,San Jose, CA, Feb. 1996.

[21] Jost Weinmiller, Morten Schlger, Andreas Festag, and Adam Wolisz. *Performance study of access control in wireless LANS IEEE 802.11 DFWMAC and ETSI RES 10 Hiperlan*, Mobile Networks and Applications, 2(1):55-67, 1997.