Performance Comparison of Medium Access Control in Wireless LAN IEEE-802.11 and HIPERLAN

Wenping Liu

A Thesis

in

The department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of Master of Applied Science (Electrical Engineering) at

Concordia University

Montreal, Quebec, Canada

March, 2005

Your file Votre référence
ISBN: 0-494-04384-9
Our file Notre référence
ISBN: 0-494-04384-9

ABSTRACT

Performance Comparison of Medium Access Control

in Wireless LAN IEEE-802.11 and HIPERLAN

Wenping Liu

There are two principal wireless LAN standards: IEEE 802.11 and ETSI HIPERLAN, which define the specification of physical and media access control layer. WLAN medium access control protocols describe the rules for orderly access to the shared wireless medium and play a crucial role in the efficient and fair sharing of scarce wireless bandwidth resource. IEEE 802.11 MAC layer employs CSMA/CA protocol to access medium, whereas ETSI HIPERLAN uses EY-NPMA protocol to control the medium access.

In this thesis, the two standards are introduced first. Then, the simulation models are presented for various network architectures and environmental parameters in order to obtain comparable performance results regarding the buffer size, the packet lifetime period and the traffic load of terminal stations. After that, the simulation results are presented for different simulation models with analyzing and evaluating the performance for both standards in different criteria and various input parameters.

ACKNOWLEDGEMENTS

I would like to thank Dr. A. K. Elhakeem for his continuous guidance and suggestions on my research. He always helped me to approach the problems from different perspectives and to define the scope of what I needed to accomplish. He always pointed out my mistakes and suggested solutions to the problems I encountered. His suggestions and observations were extremely helpful throughout this thesis.

I would like to express my appreciation to my wife for her unlimited encouragement and support.

Finally, I want to acknowledge the Help Desk of the Electrical and Computer Science of Concordia University for their help on the simulation programming.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ACK	Acknowledgement
AP	Access Point
BSA	Basic Service Area
BSS	Basic Service Set
CAC	Channel Access Control
CSMA/CA	Carrier-Sense Multiple Access with Collision Avoidance
CSMA/CD	Carrier-Sense Multiple Access with Collision Detection
CTS	Clear To Send
CW	Contend Window
DBPSK	Differential Binary Phase Shift Keying
DCF	Distributed Coordination Function
DIFS	DCF Inter-Frame Space
DLL	Data Link Layer
DQPSK	Differential Quadrature Phase Shift Keying
DSSS	Direct Sequence Spread Spectrum
ESS	Extended Service Set
ETSI	European Telecommunications Standards Institute
EY-NPMA	Elimination-Yield, Non-Preemptive Multiple Access
FHSS	Frequency Hopping Spread Spectrum
GFSK	Gaussian Frequency Shift Keying
HBR	High Bit Rate

HIPERLAN	HIgh PErformance Radio Local Area Network
HCPDU	HIPERLAN CAC PDU
HMPDU	HIPERLAN MAC PDU
IBSS	Independent BSS
IFS	Inter-Frame Space
ISM	Industrial, Scientific, and Medical
LAN	Local Area Network
LBR	Low Bit Rate
LLC	Logic Link Control
MAC	Medium Access Control
MPDU	MAC layer Protocol Data Unit
MSDU	MAC layer Service Data Unit
NAV	Network Allocation Vector
OFDM	Orthogonal Frequency Division Multiplexing
OSI	Open System Interconnection
PC	Point Coordinator
PCF	Point Coordination Function
PHY	Physical layer
PIFS	PCF Inter-Frame Space
RTS	Ready To Send
SIFS	Short Inter-Frame Space
TTL	Time To Live
WLAN	Wireless local area network

Chapter 1

Introduction

Wireless LANs (WLAN) have rapidly become a major growth point for the network industry in recent years and are expected to grow more in the upcoming years. They are used as an extension of the wired network with a wireless last link to attach a large number of mobile terminals [6]. There are two main wireless LAN standards: IEEE 802.11 and HIPERLAN. IEEE 802.11 wireless LAN standard was developed by the IEEE 802.11 work group covers wireless networks for portable, moving or fixed stations.

The HIPERLAN standard was developed by the European Telecommunications Standards Institute (ETSI) [12]. There are two primary versions of HIPERLAN, namely, HIPERLAN/1 and HIPERLAN/2; however, they are dissimilar. In this thesis, we only focus on HIPERLAN/1.

The objective of this thesis is to compare the performance on medium access control layer for these two WLAN standards under different network environment through the simulations using computer software.

1.1 Introduction to IEEE 802.11

IEEE 802.11 standard [21] is a younger member of the IEEE 802 family; it defines the functionality and procedures which must be provided within the medium access control (MAC) and physical (PHY) layers, as defined in IEEE 802, Overview and Architecture for Local and Metropolitan Area Networks. The IEEE 802.11 defines MAC and PHY layers interface with the Logical Link Control (LLC) layer and the higher layers of a standard IEEE 802, so wireless devices can use the same LLC developed for other IEEE 802-compliant systems [22]. Thus, IEEE 802.11 standard can achieve full functionality for the upper layers without worrying about the quite significant differences between a network based on a reliable cable and another one using the air.

1.1.1 IEEE 802.11 Network Structures

The basic element of the IEEE 802.11 network architecture [21] is called the Basic Service Set (BSS). The BSS is defined as a group of stations (wireless nodes) which are located within a general limited physical area within which each station (STA) is theoretically capable of communicating with every other STA (assuming an ideal environment with no communication barriers, physical or otherwise), and the group of stations is directly controlled by a single coordination function. This coordination function can be the distributed coordination function (DCF) or the point coordination function (PCF), both of them will be described later [3].

There are two basic wireless network design structures defined, ad hoc and infrastructure networks.

1.1.1.1 Ad hoc Network

An ad hoc networking is the deliberate grouping of wireless user stations into a single BSS for internetworking communications without the aid of an infrastructure network. An ad hoc WLAN has no infrastructure, and therefore no ability to communicate with external networks. An ad hoc WLAN is normally setup purely to permit multiple wireless stations to communicate with each other while requiring as little external hardware or management support as possible. The BSS of an ad hoc network is referred to as an Independent BSS (IBSS) (see Figure 1.1).

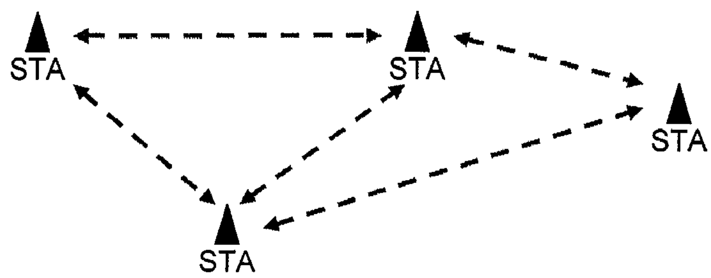


Figure 1.1 Ad hoc wireless LAN (IBBS)

1.1.1.2 Infrastructure

An infrastructure-based IEEE 802.11 wireless network (as shown in Figure 1.2) is composed of one or more BSSs that are interconnected through another network. This connecting infrastructure is called the Distribution System (DS). Within this infrastructure, each BSS must have exactly one wireless STA connected to the DS. This station provides the functionality to relay messages from the other STAs of the BSS to the DS. This STA is called the Access Point (AP) for its associated BSS. The AP supports range extensions by providing the integration points necessary for network connectivity between multiple BSSs, thus forming an extended service set (ESS) that are integrated using a common DS. The DS can be thought of as a backbone network that is responsible for transmission of the MAC layer service data units (MSDU). The DS, which is implemented independently, can be a wired network, such as IEEE 802.3 Ethernet LAN, IEEE 802.5 Token Ring LAN and FDDI LAN, or another IEEE 802.11 wireless LAN; and it is solely used as a transport backbone to transport packets between different BSSs in an ESS [20] [21].

An ESS can also provide a gateway to external networks, for example Internet, so that all STAs in the ESS can access the external networks via the gateway.

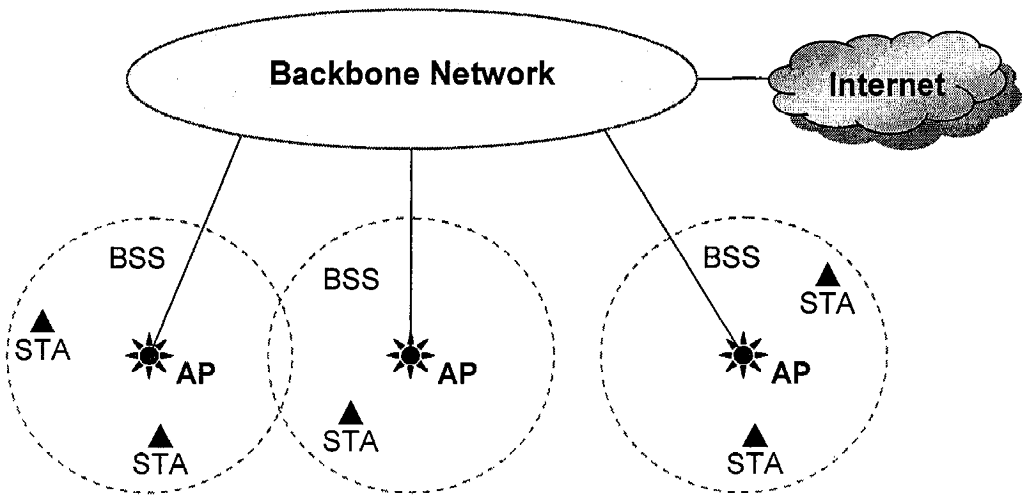


Figure 1.2 Infrastructure-based IEEE 802.11 wireless network

1.1.2 IEEE 802.11 Physical Layer

There are two specifications for radio system within the IEEE 802.11 physical layer definition: Direct Sequence Spread Spectrum (DSSS) and Frequency Hopping Spread Spectrum (FHSS). Both use spread spectrum techniques and employ radio transmission in the 2.4GHz ISM (industrial, scientific, and medical) band. Another specification Orthogonal Frequency Division Multiplexing (OFDM) is used in the IEEE 802.11a physical layer and employ radio transmission in 5GHz band [20] [21].

1.1.2.1 Direct Sequence Spread Spectrum (DSSS)

DSSS operates in 2.4GHz ISM band. DSSS signal symbol is spread with a sequence in wide bandwidth so that it has less power density. Basic DSSS has data rates of 1 and 2 Mbps. The basic rate 1Mbps is encoded using Differential Binary Phase Shift Keying (DBPSK). The 2Mbps is done using Differential Quadrature Phase Shift Keying

(DQPSK). To realize spectrum spreading, the total bandwidth is divided into 11 sub-channels of each 11MHz wide and 11 chips Barker sequence is used to spread each data symbol. The overlapping and adjacent BSSs can be accommodated by ensuring that the central frequencies of BSSs are separated by at least 30 MHz. This rigid requirement will enable only two overlapping or adjacent BSSs to operate without interference. The IEEE 802.11b uses a CCK scheme that is based on the same chip rate of DSSS channelization scheme to extend IEEE 802.11 DSSS by giving a higher rates of 5.5 and 11 Mbps [33].

1.1.2.2 Frequency Hopping Spread Spectrum (FHSS)

IEEE 802.11 FHSS operates in 2.4GHz unlicensed ISM band with bandwidth of 1MHz. It uses 79 non-overlapping hopping frequency channels with 1MHz channel spacing. The first channel centers at 2.4GHz and other subsequent channels are 1MHz apart. Three different hopping sequence sets are designed with 26 hopping sequences per set. The choosing of a certain channel is done through a pseudo random hopping pattern. Therefore, it enables the coexistence of multiple BSS, i.e. up to 26 BSS networks in the same geographical area. As a result, maximum throughput is achieved and congestion is reduced significantly. The minimum hop rate is 2.5 hops/s. The basic rate is at least 1Mbps using 2 levels Gaussian Frequency Shift Keying (GFSK), where a logical 1 is encoded as frequency F_c+f and 0 is F_c-f. The enhanced 2Mbps access rate uses 4 level GFSK, 4 frequencies are used to encode 2 bit at a time. FHSS is also a kind of resistant to multipath fading through the inherent frequency diversity mechanism [33] [34].

1.1.2.3 Orthogonal Frequency Division Multiplexing (OFDM)

IEEE 802.11a standard has been developed to extend the IEEE 802.11 standard in the 5 GHz band. In the 5-GHz band, OFDM modulation schemes are used to reach data rates ranging from 6Mbps up to 54Mbps. The main idea of OFDM is to spread a high rate data stream over a number of low rate streams and those low rate streams are transmitted through a number of subcarriers. OFDM uses inverse fast Fourier transform (IFFT) to generate the sum of a large number of subcarriers and make sure to maintain the orthogonality between the different subcarriers at the receiver side. BPSK, QPSK, 16QAM, 64QAM are available for OFDM's modulation and data rate can go up to 54Mbps. The guard interval is a key parameter to choose to reduce the intersymbol and intercarrier interference. With variable coding rates and different error corrections together, it makes the modulation robust enough to be used in any indoor and outdoor environment where directional antennas may be used. IEEE 802.11a physical layer still uses the existing IEEE 802.11 MAC protocol.

1.1.3 IEEE 802.11 MAC Layer

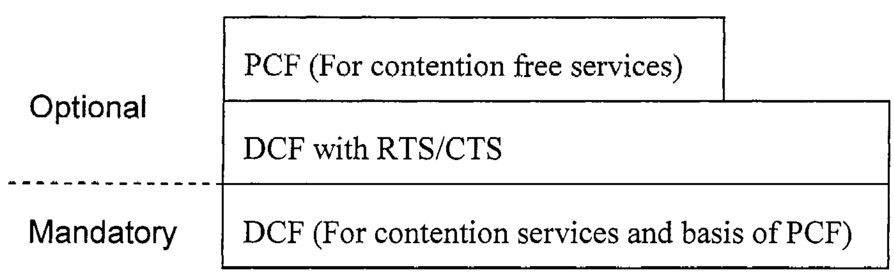


Figure 1.3 IEEE 802.11 MAC layer architecture

The basic medium access functionality of IEEE 802.11 MAC layer protocol [22] [26] provides asynchronous, time-bounded, and contention or contention free access control on a variety of physical layers through the use of the Carrier Sense Multiple Access/Collide Avoidance (CSMA/CA) protocol a fundamental method of DCF with acknowledgment (ACK) and a random back-off time procedure following a busy medium condition. In addition to DCF, IEEE 802.11 standard also employs an optional PCF as an alternative access method as described earlier. The IEEE 802.11 MAC architecture is depicted in Figure 1.3.

Frame control	Duration ID	Address 1	Address 2	Address 3	Sequence control	Address 4	Data frame (MSDU)	FCS

Generic IEEE 802.11 MAC frame (MPDU)

Frame control	Duration ID	Receiver address	FCS

IEEE 802.11 ACK frame

Figure 1.4 IEEE 802.11 MAC layer frame formats

The frame formats of IEEE 802.11MAC layer are illustrated in Figure 1.4. Besides data frame body (MSDU) and 4 octets frame check sequence (FCS) for error checking, 6 octets MAC address are used to identify source and destination stations. The two octets for duration ID field indicate the time (NAV value) the channel is reserved for successful transmission of an MAC layer protocol data unit (MPDU). The two octets frame control field indicates types of frame as for control (RTS, CTS, and ACK), data or management. The control and data frames work together to ensure reliable delivery of data. Management frames perform supervisory functions that are used to join and leave WLAN and move association from access point to access point. MAC has functions of channel allocation procedures, protocol data unit (PDU) addressing, frame formatting, error checking, and fragmentation & reassembly. It supports multiple physical layers, power management, security (including registration and authentication) and association. The purpose of IEEE 802.11 CSMA/CA is to solve the problem of collisions or reduce the probability of collision between multiple stations while they access the medium [9].

1.1.3.1 CSMA/CA MAC

The DCF is based on CSMA/CA [3] [22]. CSMA is a member of the ALOHA family of protocols. ALOHA is the first multiple/random access protocol to be applied to large-scale wireless networks. CSMA, which senses the status of a channel before transmitting packets, is the simplest way to improve ALOHA. Therefore, IEEE 802.3 workgroup chose persistent CSMA/CD as the MAC protocol for wired LAN. The success of CSMA/CD in Ethernet relies on the ease of sensing the carrier by measuring the current or voltage in the cable. This is the main reason why CSMA has been successfully applied in wired networks even though it was originally designed for radio networks. However, carrier sensing for radio networks is still a major problem due to the hidden terminal problem. Reliable carrier sensing is extremely difficult owing to severe channel fading in indoor environments and the use of directional antennas. 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 detects activity in the channel via relative signal strength from other sources. CSMA/CD is not used because a radio station is unable to listen to the channel for collisions while transmitting.

1.1.3.2 The Hidden Node Problem

In radio systems that depend on the physical sensing of the carrier, a problem called the hidden node problem arises [3] [9]. This problem comes from the fact that every wireless node has a limited radio transmitting range and that every wireless node can thus not be expected to be able to communicate with every other node in the network. The phenomenon is depicted in Figure 1.5.

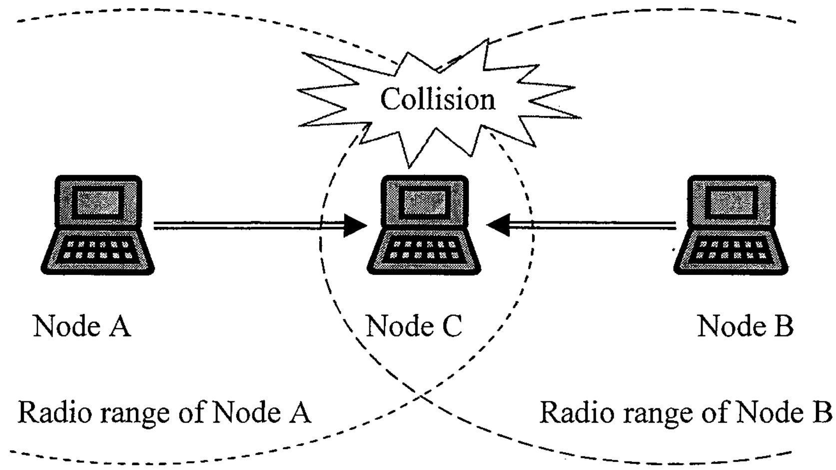


Figure 1.5 Hidden node problem

In this situation, a single receiving station can hear two different transmitters, but the two transmitters cannot hear the carrier signals of one another. This results in likelihood of collision while the two transmitters send frames. To alleviate the hidden node problem and to increase reliability, CSMA/CA is used. In general, it is associated with polling or handshaking, because multipath fading of indoor channels usually lasts for a time equal to a symbol.

1.1.3.3 Point Coordinator Function (PCF)

The PCF is an optional capability that is connection-oriented and provides contention-free frame transmission [22] [26]. The PCF uses a Point Coordinator (PC) to perform polling, enabling the polled stations to transmit without contending for channel access. The AP within each BSS that are capable of operating in the contention-free period (CFP) are known as CF-aware stations. The method by which the polling tables are maintained and the polling sequence is determined is left to the implementers. The main applications of the capability are time-bounded services (usually packetized voice or video) in which packet order is a main concern.

In IEEE 802.11 standard [21] [26], the PCF is required to coexist with the DCF and logically sits on top of the DCF (shown as Figure 1.6).

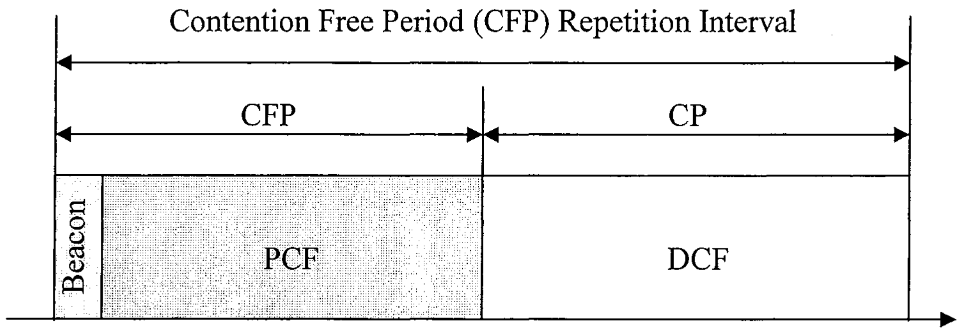
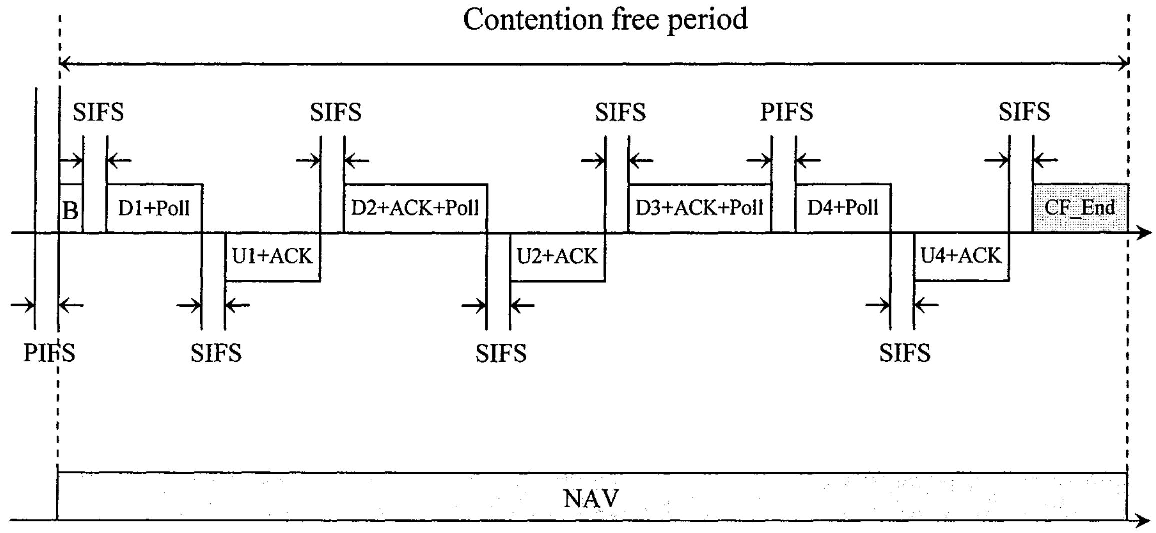


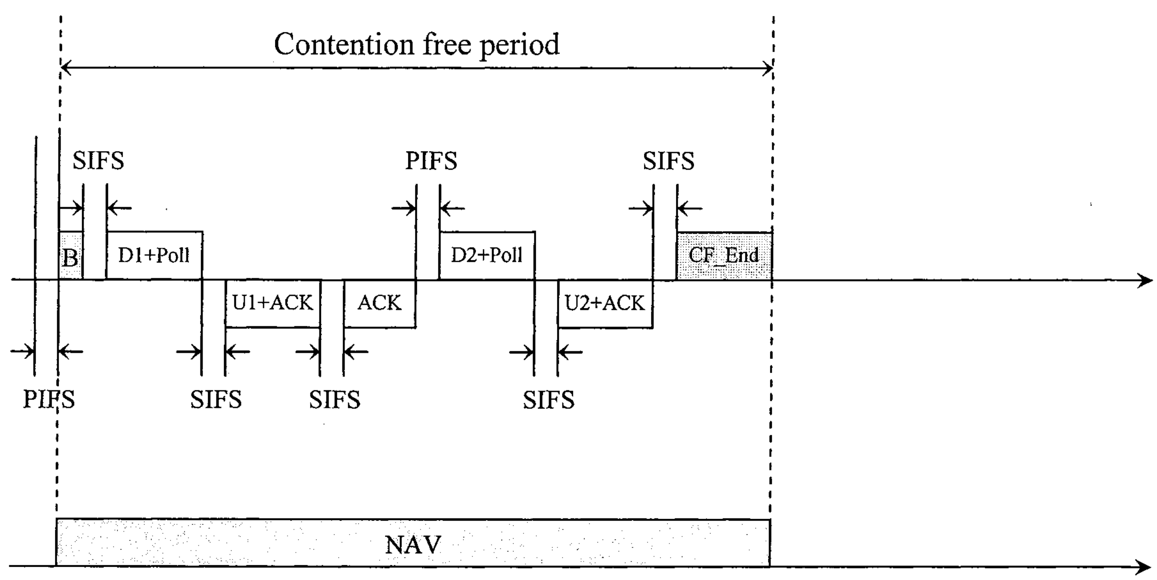
Figure 1.6 Coexistence of PCF and DCF

The CFP repetition interval (CFP_Rate) is repetition interval, a portion of the time is allotted to contention-free traffic, and the remainder is provided for contention-based traffic. A beacon frame initiates the CFP repetition interval, where the AP transmits the beacon frame. One of its primary functions is synchronization and timing. The duration of the CFP repetition interval is a manageable parameter that is always an integer number of beacon frames. Once the CFP_Rate is established, the duration of the CFP is determined. The maximum size of the CFP is determined by the manageable parameter CFP_Max_Duration. It varies between a minimum (time required to transmit two maximum-sized MPDUs, including overhead, the initial beacon frame, and a CF_End frame) and a maximum (CFP repetition interval minus the time required to successfully transmit a maximum-sized MPDU during the CP, including time for RTS/CTS handshaking and ACK). Therefore, time must be allotted for at least one MPDU to be transmitted during the CP. It is up to the AP to determine how long to operate the CFP during any given repetition interval. If traffic is very light, the AP may shorten the CFP and provide the remainder of the repetition interval for the DCF. The CFP may also be shortened if DCF traffic from the previous repetition interval carries over into the current interval. The timing diagram of PCF frame transmission is shown in Figure 1.7.

The PCF transmission procedure is presented as follows. At the nominal start of the CFP, the point coordinator senses the medium. If the medium remains idle for a PIFS interval, the point coordinator transmits a beacon frame to initiate the CFP. The point coordinator starts contention-free (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 point coordinator can immediately terminate the CFP by transmitting a CF_End frame, which is common if the network is lightly loaded and the point coordinator has no traffic buffered. If a CF-aware station receives a CF_Poll (no data) frame from the point coordinator, the station can respond to the point coordinator after a SIFS idle period, with a CF-ACK (no data) or a Data+CF_ACK frame. If the point coordinator receives a Data+CF_ACK frame from a station, the point coordinator can send a Data+CF_ACK+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 acknowledgement frames with data frames, transmitted between stations and the point coordinator, was designed to improve efficiency.



a) Timing diagram between the point coordinator and stations



b) Timing diagram between station-to-station

Figure 1.7 The timing diagram of the frame transmission in PCF

If the point coordinator 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 point coordinator. If the point coordinator fails to receive an ACK for a transmitted data frame, the point coordinator waits a PIFS interval and continues transmitting to the next station in the polling list.

After receiving the poll from the point coordinator, 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 point coordinator waits a PIFS interval following the ACK frame before transmitting and additional frames. The point coordinator may also choose to transmit a frame to a non-CF-aware station. Upon successful receipt of the frame, the station would wait a SIFS interval and reply to the point coordinator with a standard contention-period ACK frame.

1.1.3.4 Distributed Coordinator Function (DCF)

Distributed coordination function (DCF) is the fundamental access method used to support asynchronous data transfer on a best effort [3] [35]. DCF is based on the use of the CSMA/CA. The use of the CSMA/CA protocol is intended to enable access to the wireless medium in a manner that minimizes the risk of collisions. The mechanism involves sensing the medium for existing traffic. CSMA/CA first checks if the transmission medium is clear before transmission. Carrier sensing medium access can be applied on both physical and MAC layers. On the physical layer, carrier sensing means that the stations detect signal activity in the shared channel; whereas, on the MAC layer, virtual carrier sensing means network allocation vector (NAV) setting procedure that is used to indicate the amount of the time that must elapse before the current transmission session is completed and the channel can be sampled again for idle status. The channel is marked busy if either the physical or the virtual carrier sensing mechanism indicates that the channel is busy. The NAV value on the duration field (see Figure 1.7) of frames updates other stations' NAV to the latest reserved length of ongoing transmission.

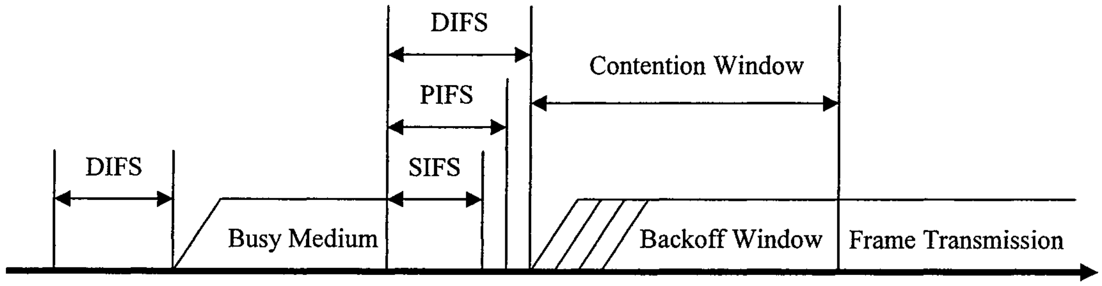


Figure 1.8 Inter-frame Space Relationship

In order to separate the different types of packets, different levels of access priority are implemented. Priority access to the wireless medium is controlled through the use of the inter-frame space (IFS), a time interval between the transmissions of frames. Three mandatory IFS intervals are specified in the standard: short IFS (SIFS), PCF IFS (PIFS), and DCF IFS (DIFS). The SIFS is the smallest IFS, followed by PIFS and DIFS, respectively; so, SIFS has the highest priority to access to the communication medium. The IFS relationship is shown in Figure 1.8.

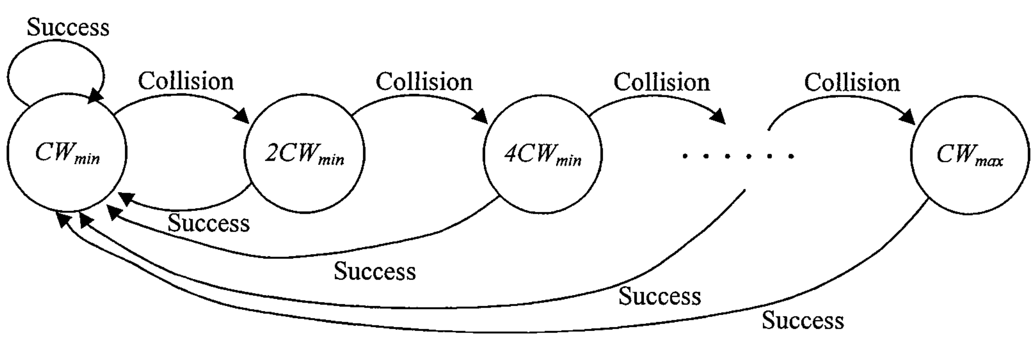


Figure 1.9 Binary slotted exponential backoff algorithm

If the medium is determined to be idle for DIFS period, a STA is permitted to transmit immediately. If the medium is not idle when initially sensed, a Binary Slotted Exponential Backoff algorithm shown in Figure 1.9 is invoked to control access when the medium first becomes idle.

If the medium is busy, the station defers until after a DIFS is detected and then generates a random backoff period for an additional defer interval before transmitting. Referred as the contention window (CW), this minimizes collisions during contention between multiple stations. The backoff period is the unit of measurement used by the backoff timer. The backoff timer is decreased only when the medium is idle; it is frozen when the medium is busy. After a busy period the decreasing of the backoff timer resumes only after the medium has been free longer than the DIFS. The backoff interval is chosen following

$$INT[2^{2+i} * rand()] * Slot\_Time$$

Where i is the number of consecutive times a station attempts to send an MPDU, rand() is a uniform random variable in (0,1), Slot_Time is different for every physical implementation, and INT[x] represents the largest integer less than or equal to x. If no other station started transmitting before the backoff interval is reached (i.e., another station that selected an earlier slot), it initiates its own transmission. After each unsuccessful transmission, the contention window takes the next value in the series until it reaches CW_{max}; and will remain at for the remaining retries. After a successful transmission, the contention window is reset to CW_{min}. Collisions can now only occur in the case that two stations selected the same slot. If another station selected an earlier slot, the station freezes its backoff timer, waits for the end of this transmission and now only waits for the slots remaining from the previous competition. Since the contention window is doubled for every collision, the probability that the two colliding stations will choose the same backoff interval decreases.

The IEEE 802.11 MAC protocol includes [22] [26 [35], as an option, a well-known mechanism for solving the hidden node problem. Two control frames are used: a RTS (Request To Send) frame that a potential transmitter sends to a receiver, and a CTS (Clear To Send) frame that a receiver sends in response to a transmitter's RTS frame (see Figure 1.10).

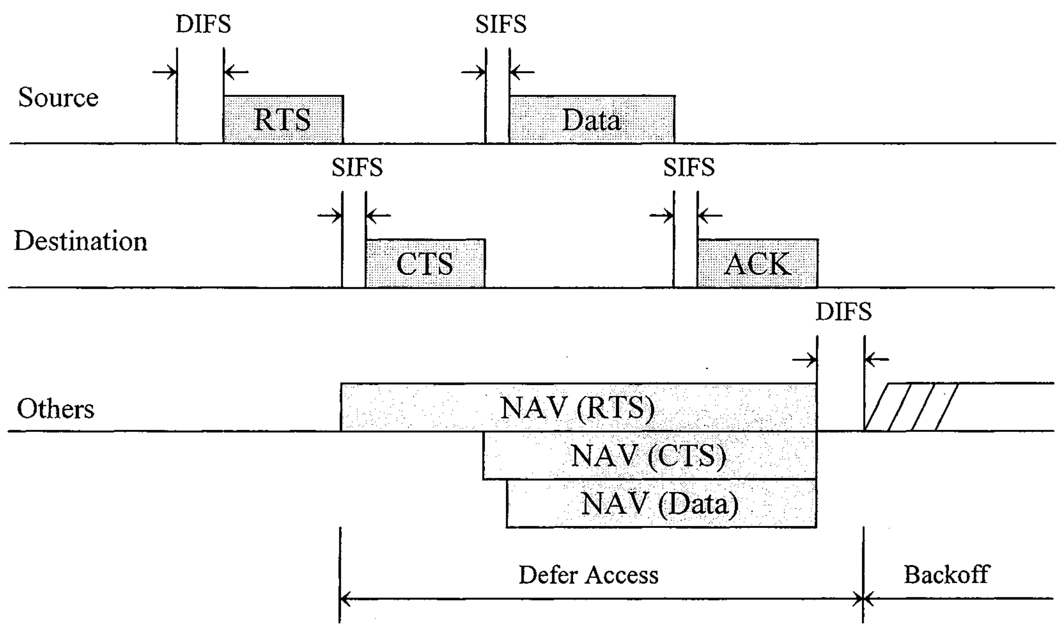


Figure 1.10 Transmission of MPDU with RTS/CTS

In the RTS-CTS negotiation mechanism, the station that wants to send data sends a RTS frame to all stations within radio range not to initiate any transmission for a given NAV time. The destination station responds with a CTS frame if it is ready to receive and the medium is idle. Upon reception of the CTS frame, the transmitting station transmits its data frame; and if the CTS is not received, it tries again after a random backoff time.

The RTS and CTS frames contain the duration of the coming data transmission, which allows stations overhearing the RTS/CTS exchange to refrain from transmitting during this time. To further explain this, consider the scenario in Figure 1.11. Here, any transmission between Node A and Node C could be interrupted by a sudden transmission from Node B to Node C. Using the RTS-CTS scheme here would eliminate such interruptions, since Node B will be able to hear the CTS response from Node C and thus to know how long Node C will be busy in a transmission with A.

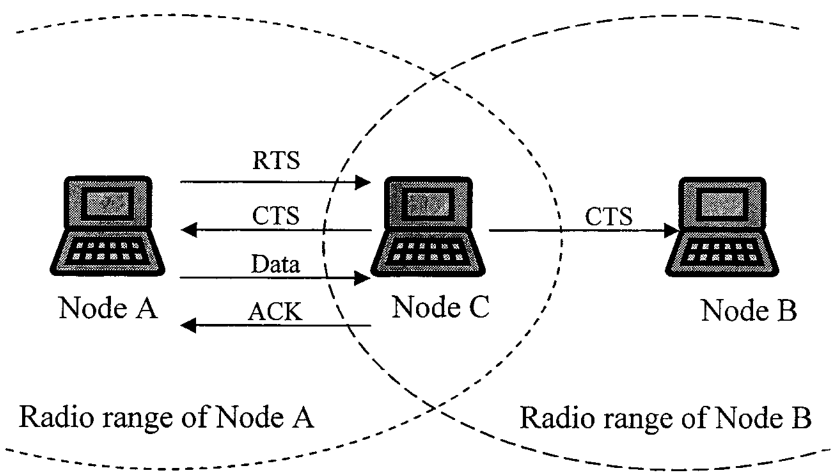


Figure 1.11 The RTS-CTS mechanism

RTS/CTS pair exchanges minimize the collision time because the lengths of RTS/CTS frames are much shorter than the length of a packet of MPDU usually. If a collision happens, only a period of RTS/CTS is wasted instead of the whole MPDU. It saves bandwidth and minimizes the waste a lot especially when MPDU (maximum data frame size up to 2346 octets) is relatively much larger than RTS/CTS control frames. RTS and CTS are 20 and 14 octets respectively. However, for a lightly loaded medium, additional delay is imposed by the overhead of the RTS/CTS frames. Therefore, it is helpful to use RTS/CTS when MPDU exceeds the value of RTS threshold in highly loaded environment with overlapping networks.

1.2 Introduction to HIPERLAN

The HIPERLAN Standards [12] [13] [14] [19] were developed by the European Telecommunications Standards Institute (ETSI). There are two primary versions of HIPERLAN named HIPERLAN/1 and HIPERLAN/2. These two wireless standards are very dissimilar in nature from each other; in this thesis, we only focused on the HIPERLAN/1.

The HIPERLAN is a high performance radio local area network in which all nodes communicate using a single shared communication channel. It has the following properties:

- It provides a service that is compatible with the ISO MAC service definition;
- Its operations are compatible with the ISO MAC bridges specification for interconnection with other LANs;
- It may be deployed in a pre-arranged or an ad-hoc fashion;
- It supports node mobility;
- It may have a coverage beyond the radio range limitation of a single node;
- It supports both asynchronous and time-bounded communication by means of a Channel Access Mechanism (CAM) with priorities providing hierarchical independence of performance;
- Its nodes may attempt to conserve power in communication by arranging when

they need to be active for reception.

1.2.1 HIPERLAN Architecture

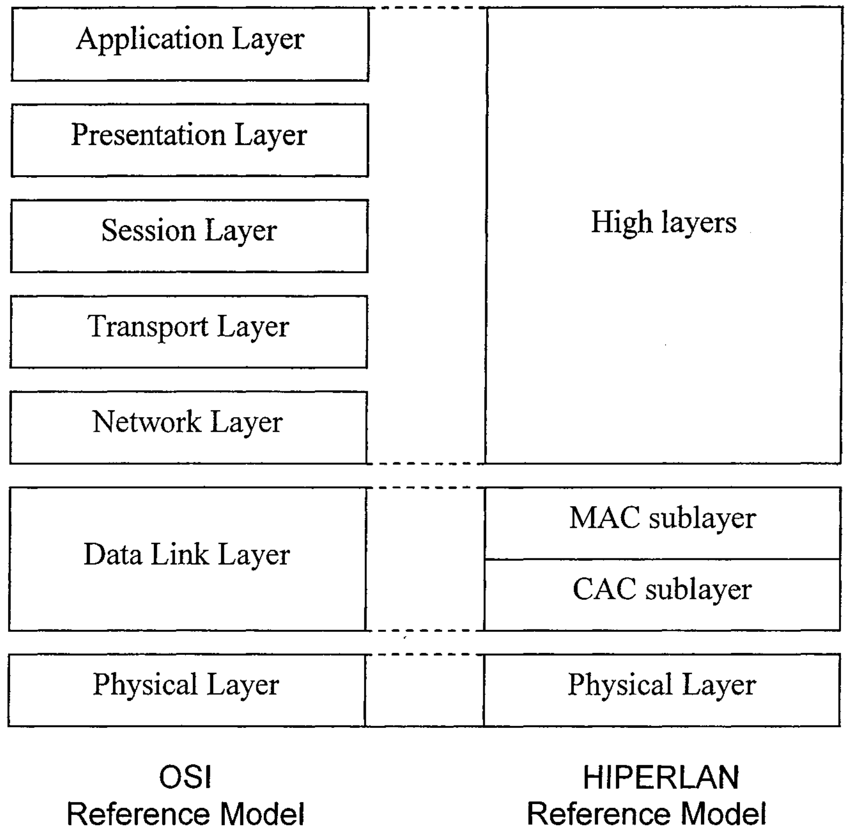


Figure 1.12 OSI and HIPERLAN reference models

The HIPERLAN reference model [3] [25] is composed of a Medium Access Control (MAC) sublayer, a Channel Access Control (CAC) sublayer and a physical layer [12]. The mapping of the HIPERLAN reference model to the OSI reference model is shown in Figure 1.12. In this layering model, HIPERLAN applications are outlined as higher layer protocols above the HIPERLAN layers.

The HIPERLAN communication model [3] [25] is shown in Figure 1.13.

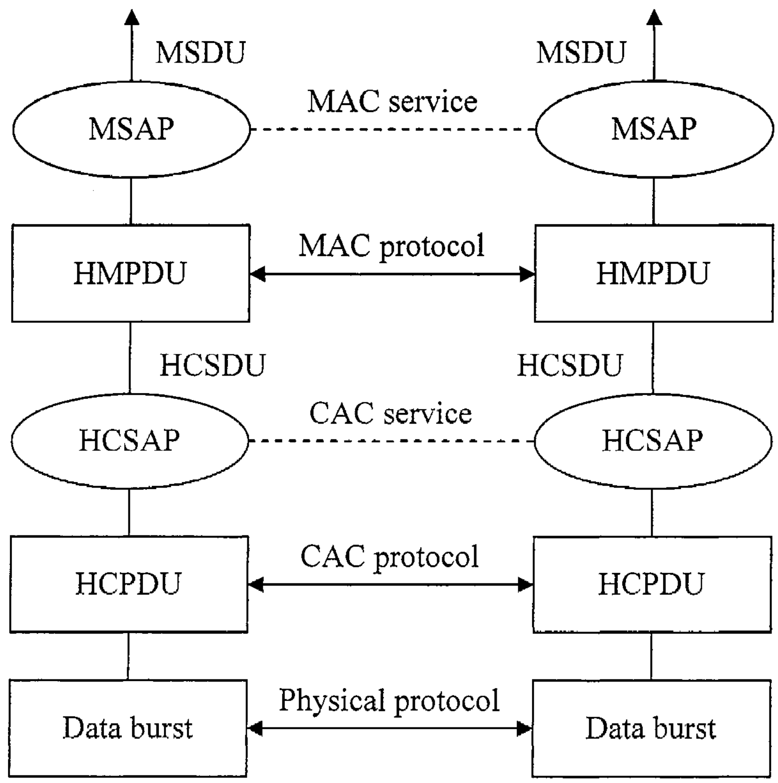


Figure 1.13 HIPERLAN communication model

When data transmission is needed, applications (users) access HIPERLAN at the MAC service AP (MSAP) and demand the transmission of their MSDUs. The AMC service is based on HIPERLAN MAC PDUs (HMPDUs). Some HMPDUs carry user data, from the MSDUs. Others are used to exchange information between the MAC entities of different terminals. MAC entities have to maintain a number of databases with information needed for the routing functions. This information is dynamically updated based on received data from other MAC entities.

HMPDUs are delivered to the HIPERLAN CAC service AP (HCSAP) for the CAC service. This service is based on the HIPERLAN CAC PDUs (HCPDUs). Some of them include data processing from the high layers (user data or MAC data) and others are generated by the CAC entity as part of the implementation of the channel access protocol.

HCPDUs are delivered to the physical layer, which codes then as bursts and them transfers them to other HIPERLAN nodes via radio. All users within a single HIPERLAN use the same radio channel, with sharing based on time division multiple access. The distribution of time between the users is part of the CAC service.

On the receiving side, the reverse process takes place. The physical layer receives the radio bursts and decodes them, and if their destination is in the same terminal, he HCPDU delivers them to the CAC. If the HCPDU contains data for the higher layers, the HMPDU delivers them to the MAC layer, and, eventually, the MSDU is delivered to the user.

1.2.2 HIPERLAN Physical Layer

The main functions of the HIPERLAN physical layer [12] is describer as follows:

- To establish a physical link to deliver data from a transmitter to one or several receivers using the modulation formats described for the low bit rate (LBR) and high bit rate (HBR) transmission and the techniques fir error correction as described in the standard.
- To assist the multiple access scheme by measuring the channel status according to predefined rules and maintaining an adaptive threshold to determine whether the channel is busy.

The physical layer allows a HIPERLAN to select one of five independent channels within the allocated bandwidth from 5,150MHz to 5,300MHz. Table 1.1 lists the carrier frequencies' nominal values [3]. While channel 1 to 3 are license-free in any country, channel 4 and 5 are not globally available. The channel transmits data at two different data bit rates – a low bit rate (LBR) (1.4706 Mb/s), that is used to transmit acknowledgment packets and the packet header, and a high bit rate (23.5294 Mb/s) to transmit the data packet itself.

Channel carrier	Frequency (MHz)
1	5,176.4680
2	5,199.9974
3	5,223.5268
4	5,247.0562
5	5,270.5856

Table 1.1 HIPERLAN channel carrier frequencies

The physical layer adds to the MPDU the LBR header, 450 HBR bits for synchronization and training sequence, *n*_496 high-rate bits payload coded with BCH(31,26) and a variable number of bits for padding. The selection of a new channel and the changing of the carrier must not take more than 1 ms. The Rx/Tx turnaround time is limited to < 5 μsec.

LBR modulation is two levels of frequency shift keying (FSK). The signaling frequencies are equal to the center frequency plus (for a one) or minus (for a zero) 368KHz. The nominal bit rate is 1.470588 Mbps. No filtering is performed on the baseband signal.

HBR modulation is Gaussian minimum shift keying (GSMK). This is a variation of FSK in which the frequency deviation from the carrier is set at 1/4 T_h, T_h being the bit period for the HBR. The baseband data are filtered with a Gaussian filter. The Gaussian filter bandwidth is usually related to the bit period.

1.2.3 HIPERLAN MAC Layer

The HIPERLAN MAC layer [12] performs several functions that are part of the DLL in the OSI model and that are described as follows.

- Provides the transmission of HMPDUs using the CAC services, to one or several destinations within the same HIPERLAN, with or without data encryption. This can be carried out with regard to the time available for traffic.
- Implements the feature of HIPERLAN regarding the network range larger than the radio range of individual nodes, with the aid of forwarding nodes acting as relaying stations that deliver to other nodes the data that are addressed to them.
- Maintains dynamic information regarding other stations within the same network. This information includes HIPERLAN identification and name, nodes and distance to reach them with indications on the best route, etc. All this information is updated dynamically.
- Allows the implementation of power-saving functions that are intended to reduce the stations' power consumption, particularly those that are battery-operated.

There are seven types of HMPDUs:

- Type 1: DT-HMPDU --- data HMPDU
- Type 2: LR-HMPDU --- look-up request HMPDU
- Type 3: LC-HMPDU --- look-up confirm HMPDU
- Type 4: IP-HMPDU --- individual-attention pattern HMPDU
- Type 5: GP-HMPDU --- group-attendance pattern HMPDU
- Type 6: TC-HMPDU --- topology control HMPDU
- Type 7: HO-HMPDU --- hello HMPDU

Each HMPDU awaiting transmission has an associated holding time, representing its residual HMPDU lifetime, upon expiry of which it is no longer subject to transmission and is removed.

1.2.4 HIPERLAN CAC Layer

The CAC sublayer [3] [12] governs the establishment of the communications on a shared radio channel with the following objectives.

- Allowing the user to establish priorities for the traffic;
- Preventing low-priority traffic from disturbing that of the highest priority;
- Freeing the user from the need to know the peculiarities of the communication channel;
- Reducing the probability of collisions to an acceptable minimum, even in the presence of hidden nodes;
- Allowing a reasonable use of the channel.

In CAC sublayer, the HPERLAN CAC protocol data units (HCPDUs) are used to exchanged information between HC-entities. There are two types of HCPDUs:

- LBR-HCPDU is made up of a sequence of bits that is transferred at the LBR;
- LBR-HBR-HCPDU is made up of and LBR-part and an HBR-part. The LBR-part consists of a sequence of bits that is transferred at the LBR. The HBR-part consists of a sequence of octets that is transferred at the HBR.

HIPERLAN CAC is responsible for the following:

- Selecting of the radio channel;
- Implementation of the multiple access protocol, known as the EY-NPMA;

- Formatting the HCPUDs in transmission, including, if necessary, the data in the HMPDUs and transferring them to the PHY once the channel access is guaranteed;
- Receiving from the PHY the HCPUDs and decoding the corresponding data to be transferred to the MAC layer in the form of HMPDUs.

1.2.4.1 Channel Access Control

The Channel Access Control (CAC) layer of HIPERLAN/1 [12] invokes a form priority in granting access to the wireless medium. The access scheme invoked is call Elimination-Yield, Non-preemptive Priority Multiple Access (EY-NMPA). It operates similar to CSMA. Basically, when the channel is assessed to be clear, packets ready for transmission are submitted to a prioritization phase during which the packets with the highest priority are selected. Channel access is attempted by the CAC entity according to the condition of the channel. The three possible channel conditions are described as follows:

- Free channel: When the channel is detected as idle for a sufficiently large interval of time, access can be readily attempted.
- Synchronized channel: When the channel is busy at times, access can be made only at certain times according to the EY-NPMA protocol.
- Hidden node elimination: As the HIPERLAN range is larger then the radio range of individual nodes, a station may arise where two transmitters that are out of range of both simultaneously, with the consequent collision. This situation is same as in IEEE 802.11 hidden nodes.

The CAC protocol relies on three factors to avoid collisions:

- Priority: High priority traffic has more immediate channel access. The protocol eliminates lower priority traffic from contention.
- Random yielding: The probability of equal priority traffic collisions is reduced by a yield phase that gives them equal access rights on average but that assigns random priorities to a particular transmission cycle.
- Detection of hidden transmitting nodes and transmission deferral for a short while to avoid collisions.

However, some collisions are still likely. To save the integrity of the data, and acknowledgement is sent when a unicast transmission has been correctly received; but this is not sent with a multicast transmission.

1.2.4.2 Channel Access in Free Channel Condition

Transmission can be attempted after the channel is sensed as idle for a long enough interval. This is called the free channel interval [3], and its length is not equal for all nodes to avoid the possibility of collisions when several nodes detect that the channel is free at the same time. The duration of the channel-free interval is:

$$i_{MF} + n \times i_{FS}$$

where i_{MF} is the minimum duration of the channel free interval. This is equal to $2000T_h$, T_h being the HBR bit period ($\approx$42.5nsec). Thus, $i_{MF} \approx 85\mu$sec.

i_{FS} is the duration of a slot of dynamic extension in the channel-free interval and is equal to $212T_h$ ($\approx 9\mu$sec).

n is a random number, with uniform probability between 1 and m_{FS}. In the implementation, $m_{FS} = 12$.

Thus, the channel-free interval length is between 85 and 193µsec. Different nodes that have been detecting that the channel is idle during the first 85µsec would attempt to transmit at different times depending on the value of n that each node is using. The node with the smaller n would transmit first. The rest would detect that the channel is busy and wait for a new whole channel-free interval or attempt the transmission when the channel condition is synchronized. There is still a small likelihood that two or more nodes have the same random number n and transmit at the same time, with the consequent collision and loss of traffic. On average, 1/12 of the expecting nodes would have the same random number n [3] [12].

1.2.4.3 Channel Access in Synchronized Channel Condition

The synchronized channel condition [3] [12] occurs when the channel is idle in the channel synchronization interval, which starts immediately after the end of the previous channel access cycle determined at the local antenna. In the synchronized channel condition, a channel access attempt shall take place using the synchronized channel access cycle, which follows immediately the channel synchronization interval. In the synchronized channel condition, Elimination-Yield Non-preemptive Priority Multiple Access (EY-NPMA) operates in the synchronized channel access cycle, comprising in series the prioritization, contention and transmission phases [12] [19].

- The prioritization phase: this phase is intended to eliminate all the traffic except that with higher priority from contending the access to the channel within this cycle.
- The contention phase: in this phase, equal priority traffic contends for

transmission within the cycle.

- The transmission phase: nodes that have survived the elimination processes of the previous cycles attempt the transmission. When unicast transmission is made, this cycle ends with the reception (or absence) of an ACK.

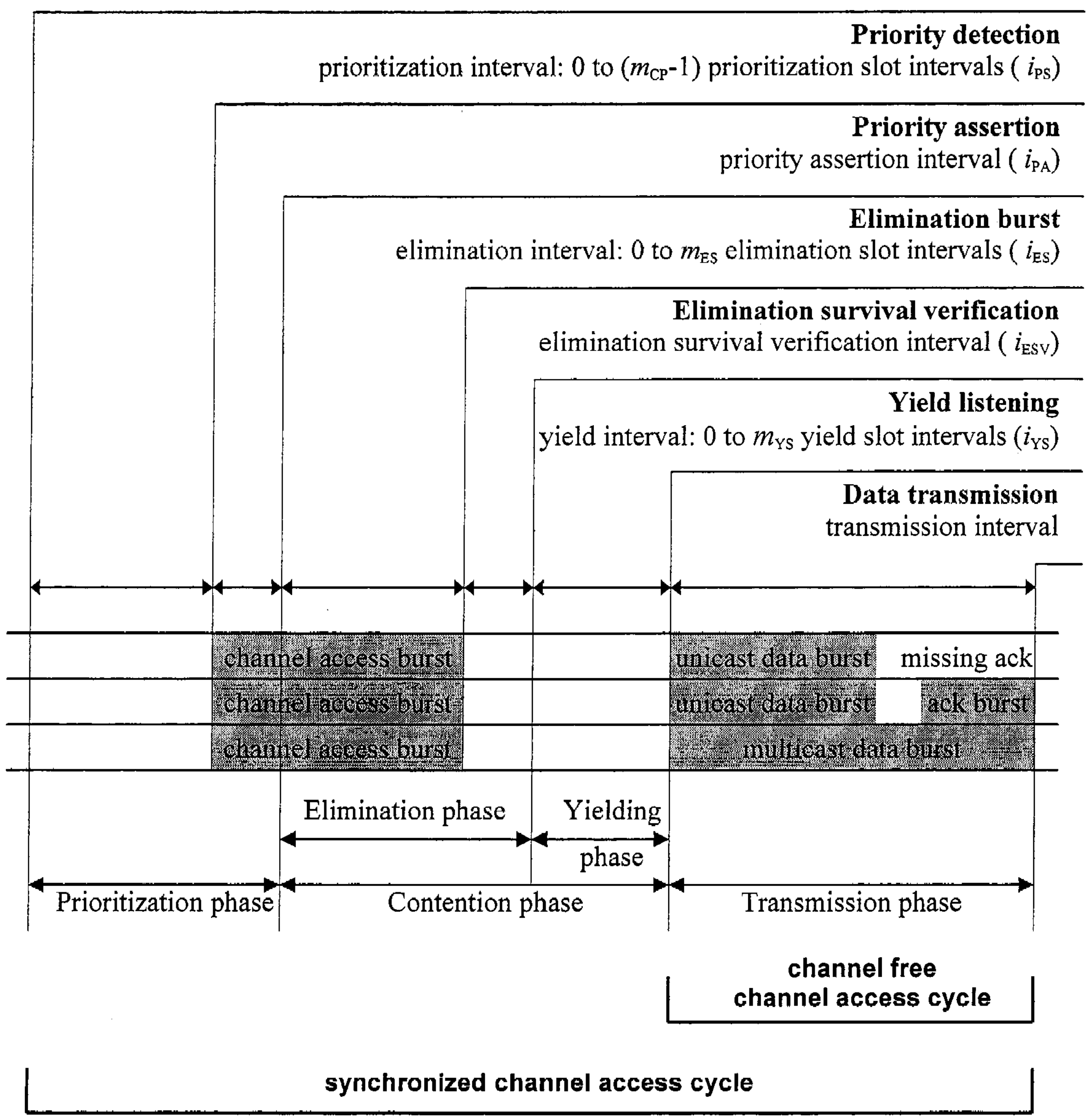


Figure 1.14 EY-NPMA activities

At least one node survives the transmission phase. On some occasions, more than one node can survive the transmission phase, and thus collisions may occur. The EY-NPMA activity procedure is shown in Figure 1.14 [12].

Prioritization phase

The priority resolution scheme provides hierarchical independence of performance between loads at different channel access priorities. The operation of the priority resolution scheme is outlined below:

- There are a total of m_{CP} channel access priorities, which are numbered from 0 to (m_{CP} -1), with 0 denoting the highest channel access priority;
- Prioritization slots are used for prioritization of different channel access priorities. The duration of the prioritization slot interval is denoted by i_{pj};
- Priority resolution takes place by means of priority detection and priority assertion. A contending node, whose data transmission attempt has a channel access priority n, shall listen for n prioritization slot intervals. If the channel is sensed idle in the n prioritization slot intervals, the node asserts the channel access priority by transmitting immediately a channel access burst for the duration of the priority assertion interval, denoted by i_{PA}. Otherwise, the node stops its data transmission attempt in the current channel access cycle;
- If the prioritization phase ends with a priority assertion for channel access priority n, the duration of the prioritization interval is n prioritization slot intervals;
- At least one contending node will survive the prioritization phase.

Elimination phase

The operation of the elimination scheme is outlined below:

- Elimination slots are used for elimination burst. The duration of the elimination slot interval is denoted by i_{ES};
- The length of an individual elimination burst is 0 to m_{ES} elimination slot intervals long, with the probability of burst in an elimination slot being p_E. Accordingly, the probability for an individual elimination burst to be n elimination slot intervals long, denoted by $P_E\,(n)$, is given by:

$$P_E(n) = p_E{}^n(1-p_E) \qquad \text{for } 0 \le n < m_{\mathrm{ES}}$$

$$P_E(n) = p_E{}^{m_{ES}} \qquad \text{for } n = m_{\mathrm{ES}}$$

  The accuracy of $P_E\,(n)$ demonstrated by an implementation is subject to a tolerance of ± 5 %;
- The elimination scheme resolves contention by means of elimination burst and elimination survival verification. A contending node transmits a channel access burst to eliminate other contending nodes and then listens to the channel in the elimination survival verification interval, denoted by i_{ESV}, to verify if it is eliminated by other contending nodes. A contending node survives the elimination phase if and only if the channel is sensed idle in its elimination survival verification interval; otherwise, the node is eliminated and withdraws from the competition for the right of transmission in the current channel access cycle;
- The duration of the elimination interval is the longest elimination burst among the contending nodes;

- At least one contending node will survive the elimination phase.

Yield phase

The operation of the yield scheme is outlined below:

- Yield slots are used for yield listening. The duration of the yield slot interval is denoted by i_{YS};
- The length of an individual yield listening is 0 to m_{YS} yield slot intervals long, with equal likelihood. Accordingly, the probability for an individual yield listening to be n yield slot intervals long, denoted by $P_Y(n)$, is given by:

  $$P_Y(n) = 1 / (m_{YS} + 1) \qquad \text{for } 0 \leq n \leq m_{YS}$$

  The accuracy of $P_Y(n)$ demonstrated by an implementation is subject to a tolerance of ± 5 %;
- The yield scheme resolves contention by means of yield listening. A contending node survives the yield phase if and only if the channel is sensed idle in its yield listening; otherwise, the node yields to the other contending nodes and withdraws from the competition for the right of transmission in the current channel access cycle;
- The duration of the yield interval is the shortest yield listening among the contending nodes;
- At least one contending node will survive the yield phase.

Transmission phase

During the transmission phase, the data transmission concerns either:

- a multicast LBR-HBR HCPDU transmission; or
- a unicast LBR-HBR HCPDU transmission, which is expected to be followed by a corresponding AK-HCPDU transmission from the destination HC-entity.

When a HC-entity transmits a AK-HCPDU to acknowledge a received unicast LBR-HBR HCPDU transmission, the acknowledgement time requirement, measured at its antenna, is imposed such that the start of its AK-HCPDU burst shall occur i_{AK}, with an accuracy of $\pm d_{AK}$, after the end of the received LBR-HBR HCPDU burst.

As far as the transmitter of an LBR-HBR HCPDU is concerned, its multicast LBR-HBR HCPDU transmission is always successful, whereas its unicast LBR-HBR HCPDU transmission is considered successful only if a corresponding AK-HCPDU is received.

1.2.5 HIPERLAN/2 Introduction

HIPERLAN/2 MAC protocol [13] [14] [15] is based on reservation TDMA. The MAC protocol has a centralized architecture, despite whether the topological architecture of the network is centralized or direct. The basic structure of the TDMA frame is shown in Figure 1.15. The frame consists of broadcast, downlink, direct link, uplink, and random access phases. The duration of the whole access cycle is fixed to 2 ms, but the boundaries between the separate phases are flexible and can be adapted according to the traffic load.

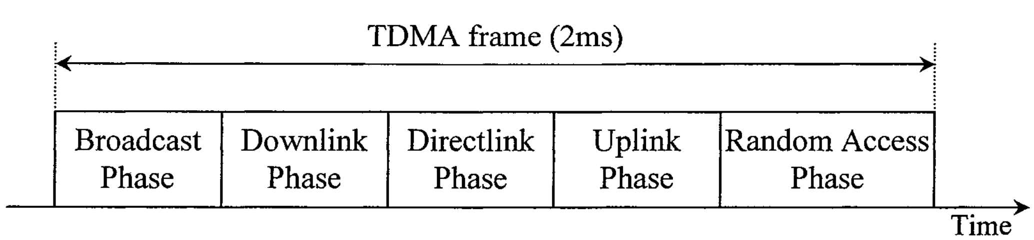


Figure 1.15 HIPERLAN/2 MAC TDMA frame

The broadcast phase carries network information, such as power levels and AP identifier, and defines the structure of the TDMA frame and the reservation status of time slots on different link phases. During the downlink phase, MPDUs are transmitted from AP to terminals and the uplink phase contains the reverse direction. The direct link phase is optional, and used for direct terminal-to-terminal communications.

The random access phase contains uplink slots that are used for terminals to send messages to AP when they do not have reserved uplink slots. The phase is divided into time slots that are utilized using the slotted ALOHA protocol with binary exponential backoff after collisions.

The basic unit of transfer is an ATM cell payload of 48 bytes that is transferred in 54bytes long MPDUs. For increasing efficiency, several MPDUs from a single terminal can be multiplexed into a single MPDU train [10]. Selective repeat ARQ with CRC calculation over MPDUs are utilized by the EC. Also, the physical layer performs convolutional FEC coding.

The HIPERLAN/2 MAC protocol is connection oriented. The QoS for each connection is a result of resource requests sent by mobile terminals and an AP scheduling function that allocates time slots for the DLC connections of a terminal. For scheduling, the AP concerns the amount of queued traffic at each terminal and the QoS definitions for a connection.

As HIPERLAN/2 has been targeted at the support of the QoS classes defined for ATM networks for virtual connections [15]. Each HIPERLAN/2 connection can thus be associated with separate parameters for throughput, delay, delay variation, and bit error rate. For compatibility with IEEE 802 based LANs, an Ethernet service specific convergence sub layer has been specified for HIPERLAN/2 [14]. The support of the

IEEE 802 based QoS is optional in the convergence layer. If supported, the different IEEE 802.11 traffic classes are mapped into separate HIPERLAN/2 DLC connections.

Chapter 2

Simulation Model

In this thesis, in order to compare the performances of different wireless LANs, simulation models are established. All simulation models are presented and are implemented by the computer software that is designed using C++ language based on object-oriented programming technology [17]. The software is consisted of 3 parts: PCF, DCF and HIPERLAN. It simulates different network conditions depending on input simulation parameters and outputs simulation results when simulation completes. In this chapter, the simulation models are established first and the different input parameters are listed; then, the metrics used to evaluate the performance are presented.

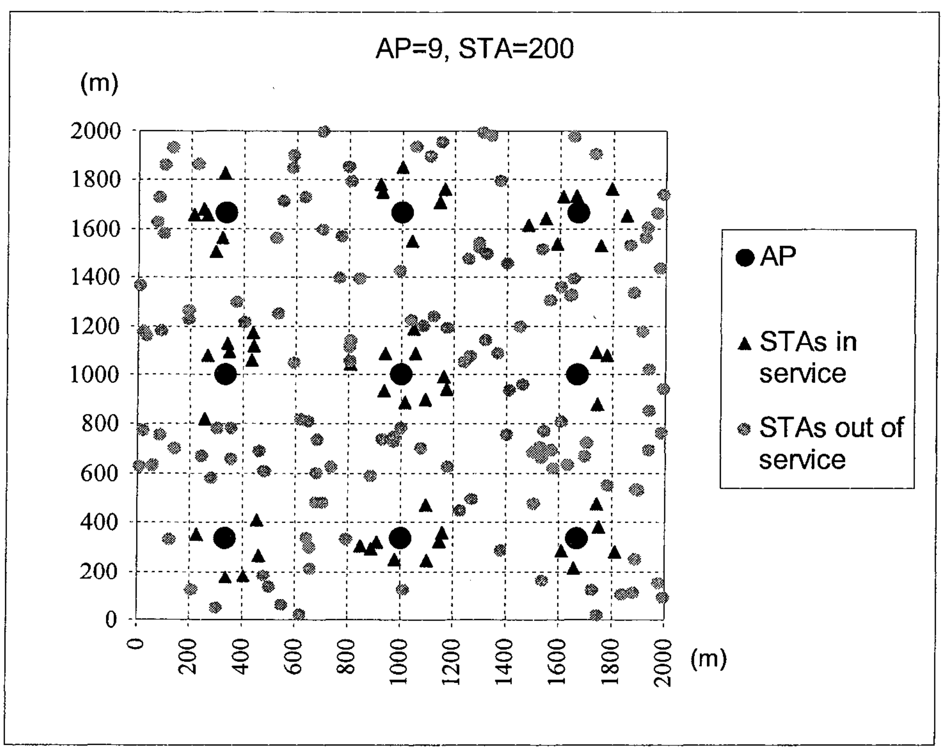
2.1 IEEE 802.11 PCF Simulation

For IEEE 802.11 WLAN, most of networks are infrastructure-based architecture network. The two infrastructure wireless LAN simulation models are established for PCF and DCF, respectively.

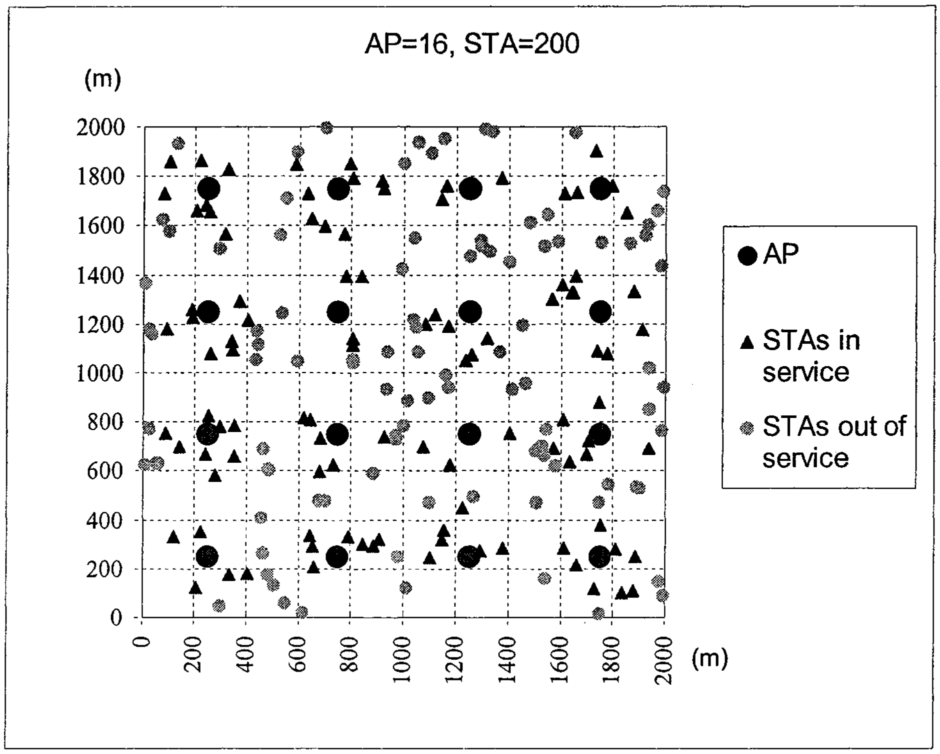
2.1.1 PCF simulation model

This model simulates an infrastructure-based wireless network within an ideal square area with 2000-meter length and 2000-meter width. In this area, 200 wireless stations are active and distribute randomly; they move only inside the area randomly. To setup an infrastructure network, access points (APs) must be deployed. All stations and APs have the same radio transmission coverage range of radius 200 meters. Different number of

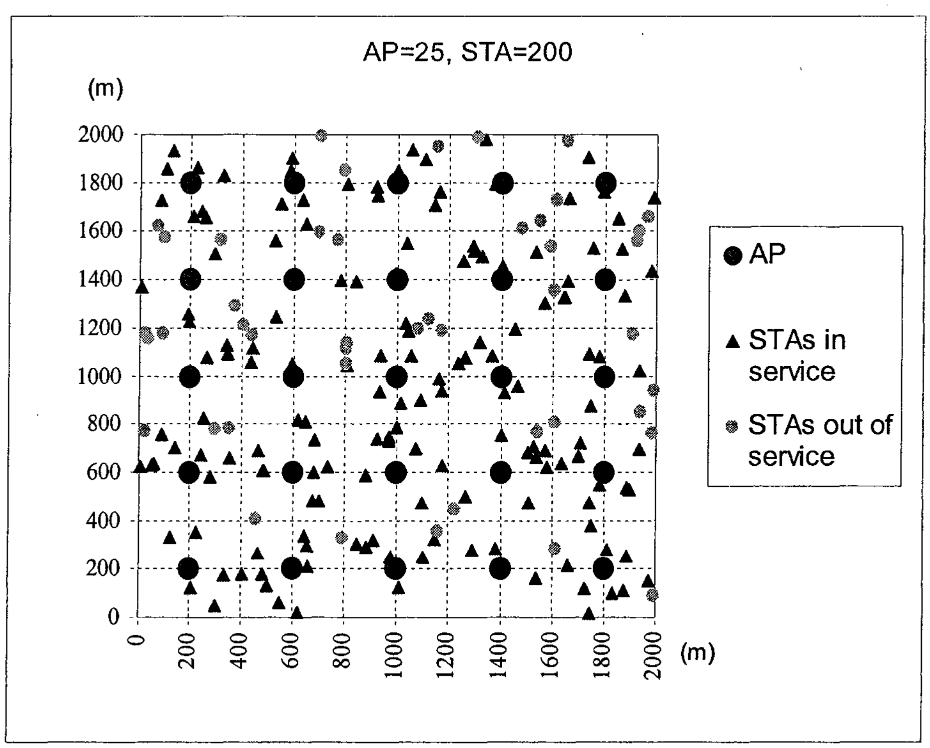
APs covers different number of STAs in every BSS. All BSSs radio service coverage illustrates as following charts. All APs are distributed uniformly; and in this square area, 9, 14, 25, 36 and 49 APs are deployed depending on the scenarios (see Figure 2.1). Based on the assumed transmission range, we can see that at least 49 APs can completely cover this area; that means each station is within one AP's radio transmission range. Assuming that all stations move only inside the area, but they can move from one AP's radio service area to another AP's radio service area or to out of service, but they will go back into the area when they reach the area border. We also assume all APs are connected by wired backbone network and packet transmission delay in the backbone network is neglected. The basic channel bit rate of 1Mb/s is used in the simulation. In every BSS, the medium control method is Point Coordination Function (PCF). The models of access points and stations distribution are shown in Figure 2.1.



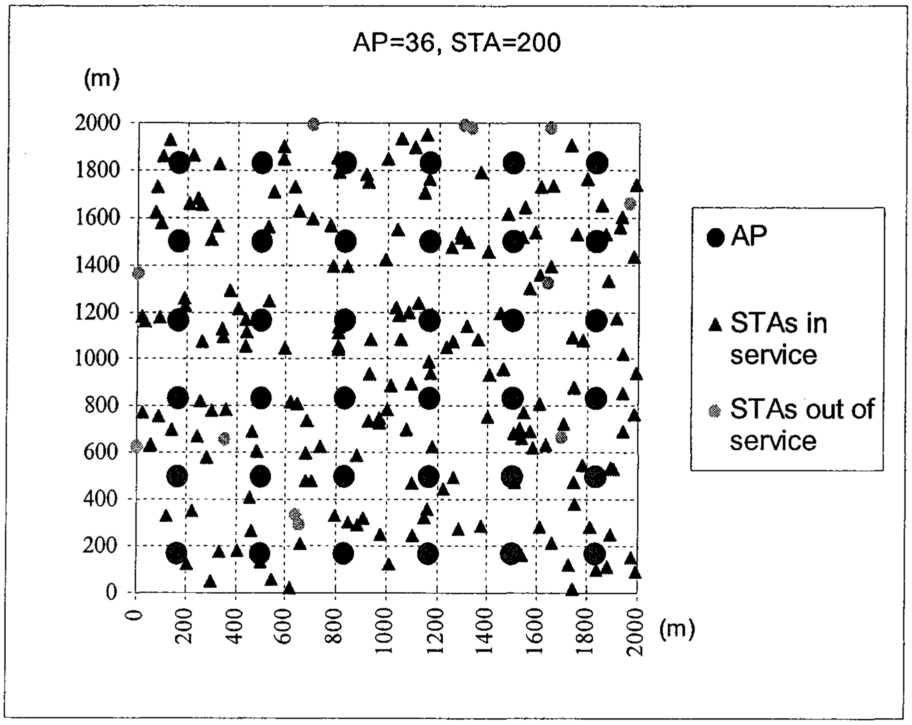
(a) AP=9


AP=16, STA=200
(m)
2000
1800
1600
1400
1200
1000
800
600
400
200
0
0
200
400
600
800
1000
1200
1400
1600
1800
2000
(m)
AP
STAs in service
STAs out of service

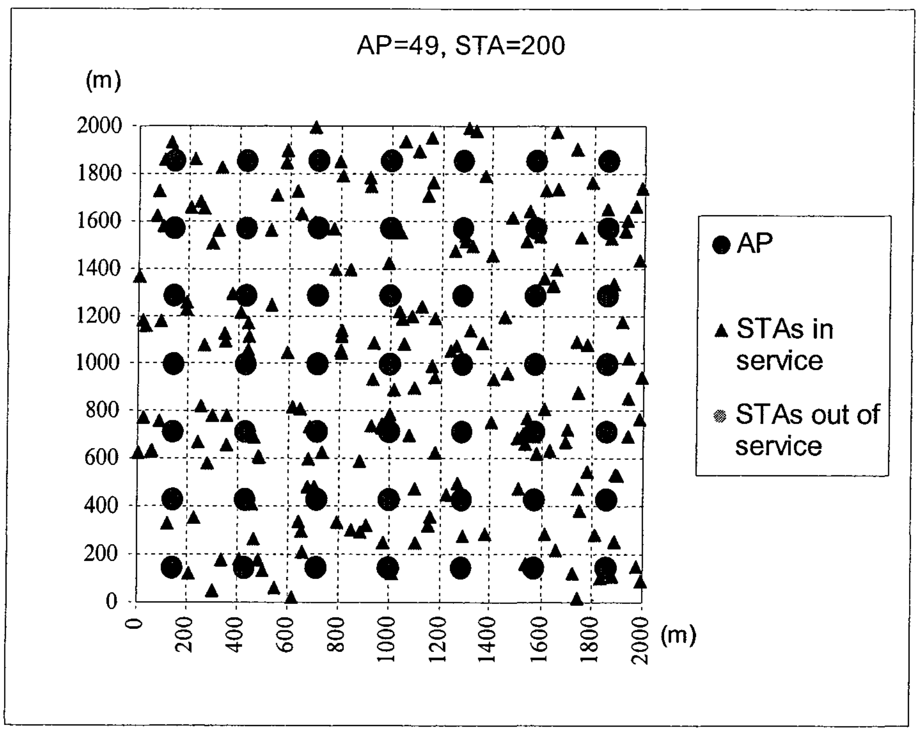
(b) AP=16


AP=25, STA=200
(m)
2000
1800
1600
1400
1200
1000
800
600
400
200
0
0
200
400
600
800
1000
1200
1400
1600
1800
2000
(m)
AP
STAs in service
STAs out of service

(c) AP=25



(d) AP=36



(e) AP=49

Figure 2.1 APs and STAs distribution models in different deployment scenarios

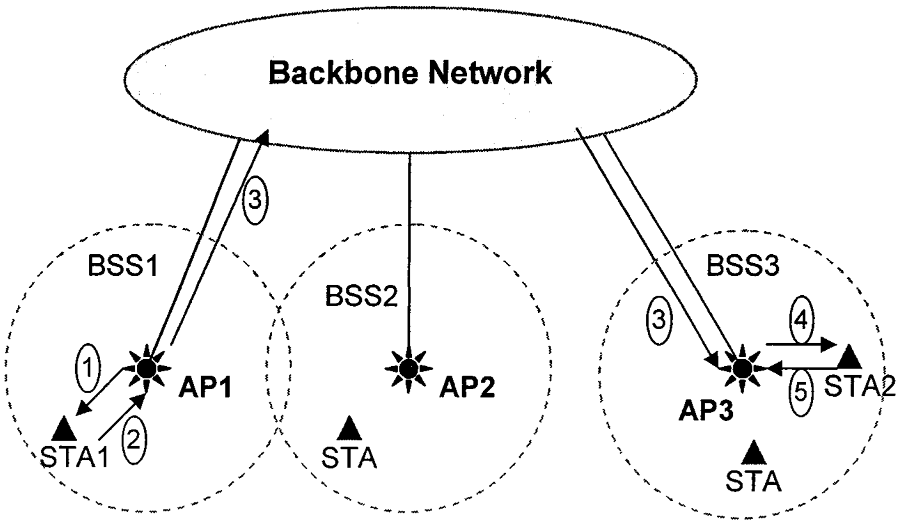


Figure 2.2 Infrastructure PCF transmission procedure

The procedure of infrastructure-based PCF is illustrated in Figure 2.2. In this example, three BSS are connected to a backbone network and all stations communicate through the access point of its BSS. Assuming STA1 in BSS1 has a packet to transmit to the STA2 in BSS3, the data packet transmission procedure is:

① AP1 send CF_Poll to STA1 in BSS1;

② STA1 transmits data packet to AP1;

③ AP1 forwards the packet to AP3 via backbone network;

④ AP3 sends (CF_Poll + Packet) to STA2 in BSS3;

⑤ STA2 replies ACK to AP3.

There are 4 cases when point coordinator communicates with stations depending on if access point or station has data packets to transmit, as shown in Figure 2.3.

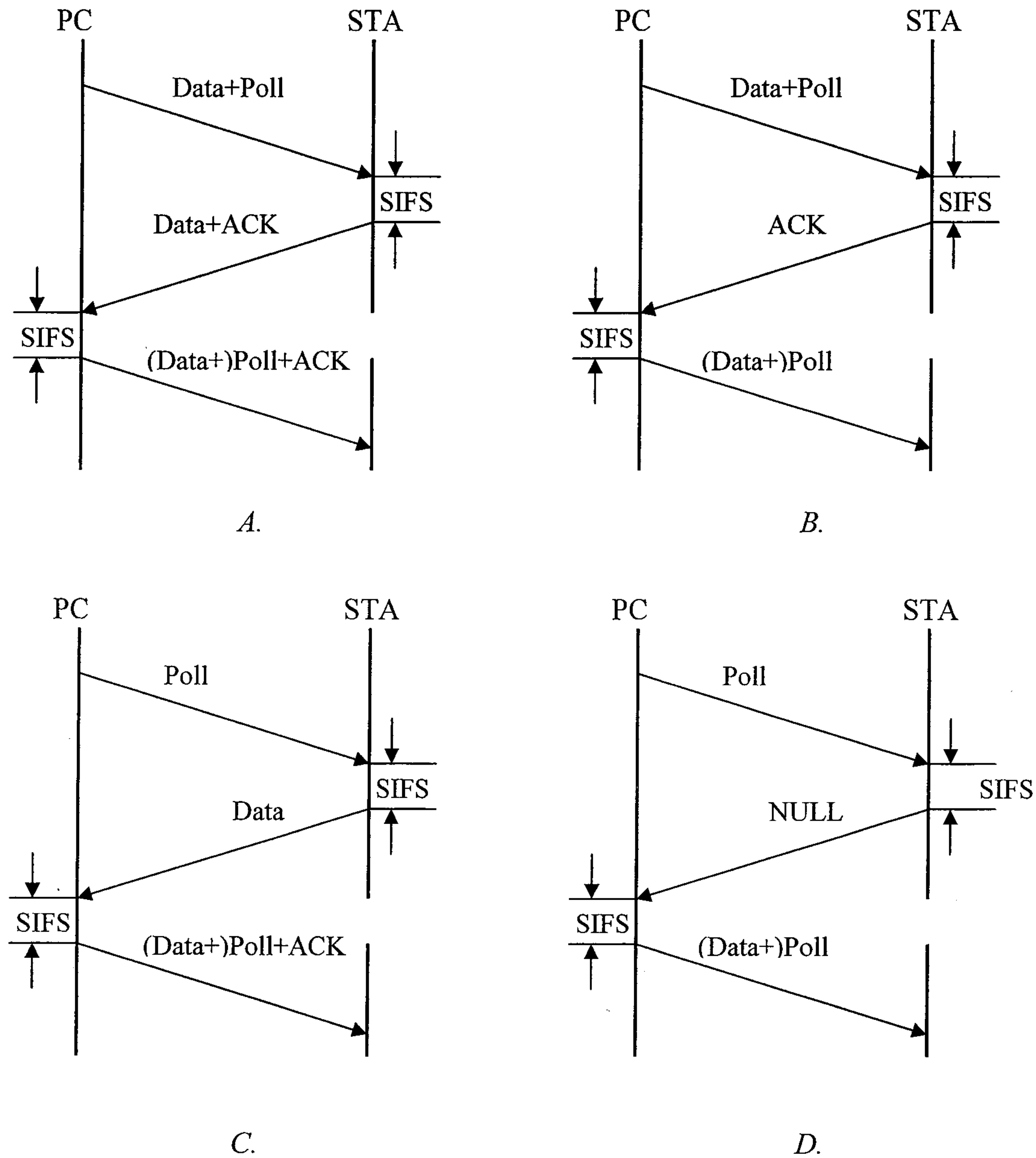


Figure 2.3 PC-to-Station transmission in PCF

2.1.2 Assumptions for Infrastructure PCF simulation

In the simulation of infrastructure-based PCF WLAN, all assuming values are given in Table 2.1.

Attribute	Value
Area	2000m × 2000m
Number of AP	9, 16, 25, 36, 49
Number of stations	200
Channel bit rate	1 Mb/s
Data packet size	8000 bits (8000 μs)
ACK	104 bits (104 μs)
Beacon	100 bits (100 μs)
CF_Poll	100 bits (100 μs)
Small Iteration	10 μs
Large Iteration	8000μs (8ms)
SIFS	10 μs
PIFS	30 μs
CFP_Repetition_Interval	0.40 s
CFP_Max_Duration	0.39 s
Station movement velocity	30m/s (maximum)
Station location update period	10 Large Iteration (80ms)
TTL	250ms, 1000ms

Table 2.1 Simulation attribute values (input parameters) for PCF

Assuming the position of station i in the simulation area is (X_i, Y_i) and the position of a AP_p is (X_p, Y_p), the new location (X'_i, Y'_i) of the station after each location update is

$$\begin{cases} X_i' = X_i + \Delta T \cdot v_i \cdot \cos\theta_i \\ Y_i' = Y_i + \Delta T \cdot v_i \cdot \sin\theta_i \end{cases}$$

where ΔT is the station location update period (=80ms), v_i is the movement speed of station i, and θ_i is the movement angle of the station i. If

$$\sqrt{(X_i' - X_p)^2 + (Y_i' - Y_p)^2} \leq 200m$$

then this station is connected with the AP_p (X_p, Y_p).

In this simulation, assume the probability that a station generates a packet is P_g. The packet generation process from each station is assumed to follow a Bernoulli trial. A uniformly distributed random generation is called every packet time for a specific station. If the returned value $x \leq P_g$, a packet is generated; when $x > P_g$, no packet generated.

P_g value is set to 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, and 1.0 respectively, to yield different simulation cases. As soon as a packet is generated, the packet is delivered to the buffer memory of the station. The size of buffer memory represents how many packets can be stored simultaneously, and the buffer is FIFO (First In, First Out) buffer. When a packet is transmitted successfully, the packet is removed out of the buffer. In this simulation, the buffer size of all APs and stations is set to 2, 16, and 128 respectively.

Assume the lifetime period of every packet, TTL (Time To Live), is 250ms and 1000ms respectively depending on the scenarios. A packet TTL is counted from the packet generation time. If a packet life period is counted up to its TTL, the packet will be discarded.

Another assumption is all simulations are operated in ideal channel condition without considering the interference and fading effects in details to simplify the channel status; in other words, the channel bit error rate is set to zero and correct packet probability P_c=1. Because the system performance is apparently worse when the channel bit error rate becomes higher.

2.1.3 Procedure for Infrastructure PCF simulation

Using object-oriented computer software design tools, this wireless LAN simulator simulates various operating details of wireless LAN environment. All simulation parameters can be set in a configuration file or input at beginning of the simulation. When all parameters for simulation are entered, the simulation starts to run. Firstly, based on the entered parameters, the simulation system initializes all access points and stations configuration, sets the number of the total simulation cycles, and randomly generates all stations location coordinates, movement speeds, and direction. Next, the PCF contention-free period starts and the detailed procedure is presented in section 1.1.3.3. After the preset simulation cycles ends, the system begins to analysis and to estimate all simulation data and outputs simulation results into a data file. The simulation procedure flowchart is illustrated as Figure 2.4, 2.5 and 2.6.

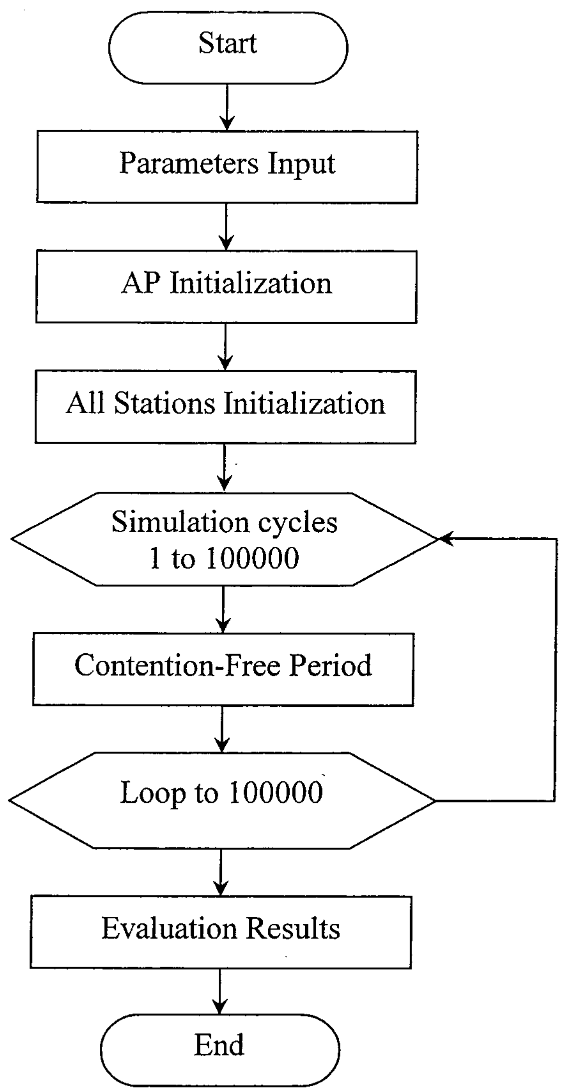


Figure 2.4 Flowchart of infrastructure PCF simulation

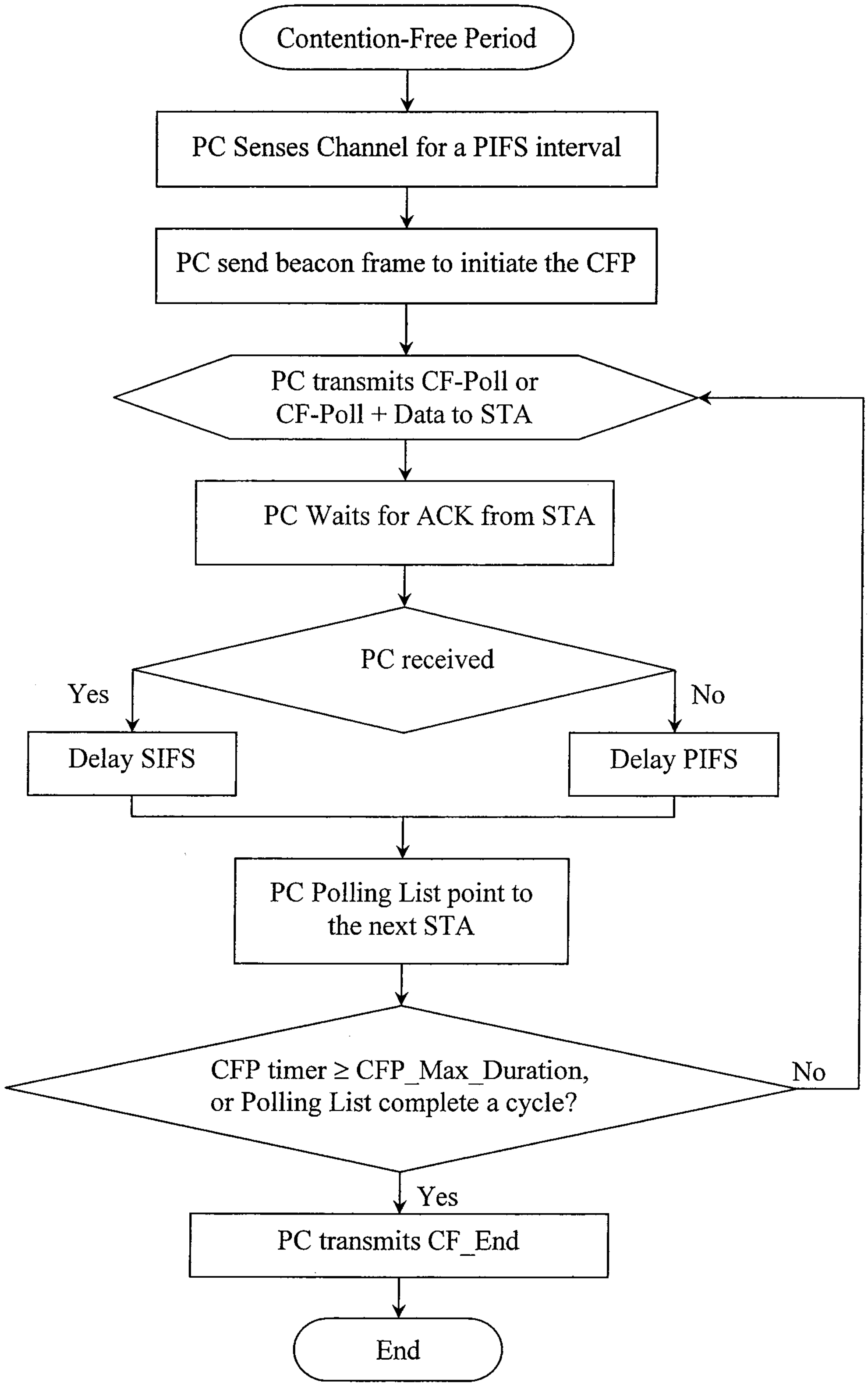


Figure 2.5 Point coordinator contention free period flowchart

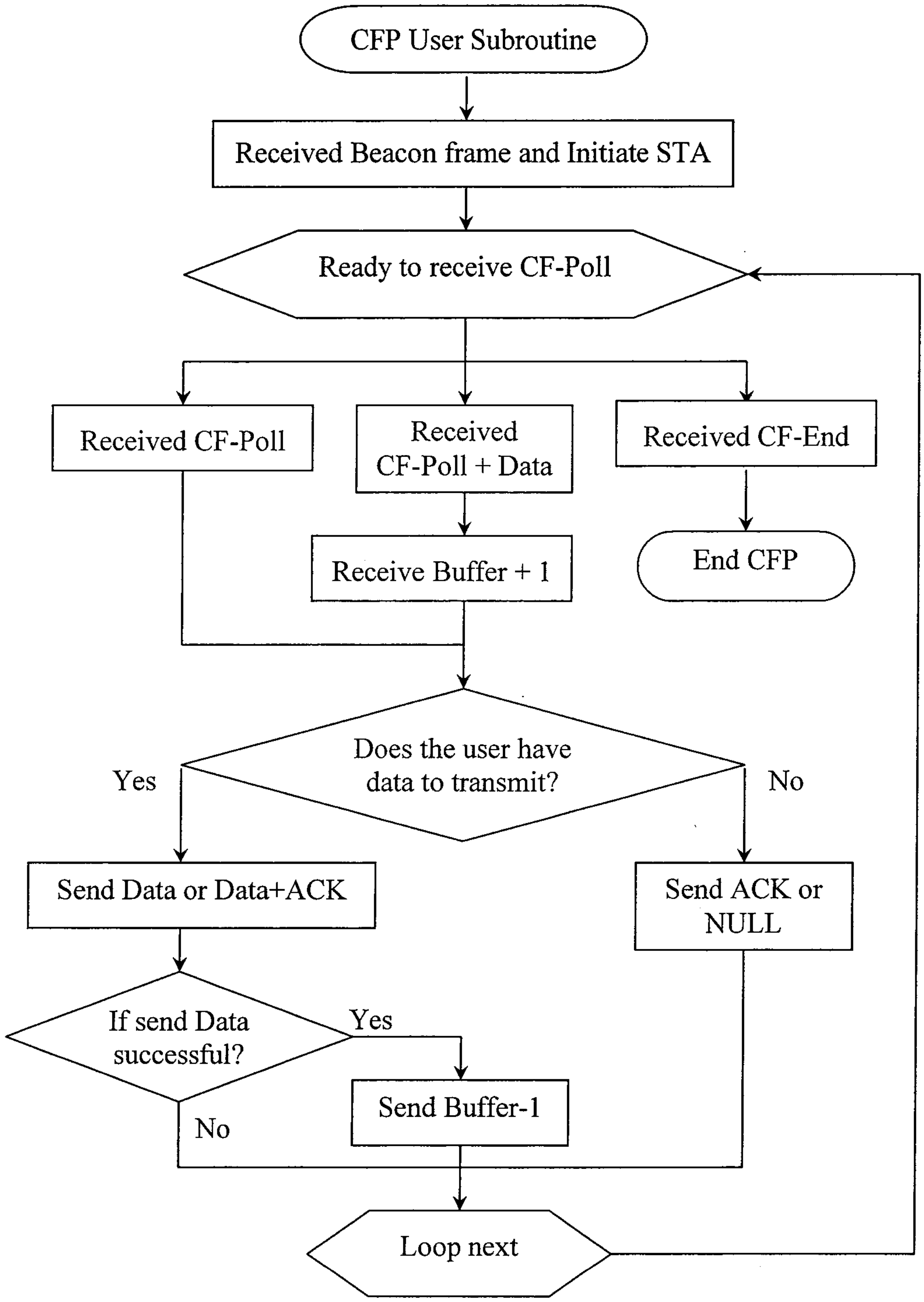


Figure 2.6 Station contention free period flowchart in PCF

2.1.4 Performance Criteria

Choosing the correct metrics to use in the evaluation of the WLAN is vital to the result and the validity of the evaluation. The metrics we have used are normalized average traffic throughput, average end-to-end delay, average packet loss rate, and average buffer overflow rate [17] [23].

1) Average Traffic Throughput

Throughput is definitely one of the most critical considerations in the design of a MAC protocol because the spectrum is a scarce resource. A station traffic normalized throughput η is defined as the percentage of the packets that a station transmits successfully from source to destination during the simulation period over the total packets that the station generated during the simulation period. A network average throughput $E[\eta]$ is the value that the sum of all station throughputs divided by the number of stations.

$$\eta_i = \frac{Ni_{success}}{Ni_{total}} \qquad E[\eta] = \frac{\sum_{i=1}^{N_s} \eta_i}{N_s}$$

η_i is the normalized throughput of station i.

$Ni_{success}$ is the number of data packets that a source station i transmits to the destination successfully.

Ni_{total} is the number of total data packets that station i generates.

N_s is the number of stations.

2) Throughput variance

Throughput variance is defined as the sum of the squares of difference between station i's average throughput η_i and all stations' average throughput $E[\eta]$. Throughput variance can be obtained by

$$\delta_\eta^2 = \frac{\sum_{i=1}^{N}(\eta_i - E[\eta])^2}{N_s}$$

N_s is the number of stations.

3) Average End-to-End Delay

The packet end-to-end delay is defined as a period from the time when the packet is generated at the source to the time when the packet is delivered to the destination and successfully acknowledged. It consists of the queuing delay, transmission delay and propagation delay. The queuing delay includes queuing delays in transmitter station, APs, and the intended receiver. For the range of wireless LAN, the propagation delay is so small and will be ignored. The queuing delay is the time the packet waits in buffer memory to be transmitted. The end-to-end transmission delay starts from the moment a data packet starts to transmit and ends when the transmitting station receives ACK successfully. Average end-to-end traffic delay (*E[D]*) can be calculated as the result of the sum of transmission times of total successful end-to-end transmitted data packets divided by the number of successful transmitted data packets.

$$D_i = \frac{\sum_{j=1}^{N_{ip}} d_{ij}}{N_{ip}} \qquad E[D] = \frac{\sum_{i=1}^{N} D_i}{N}$$

D_i is the average end-to-end delay of station i.

d_{ij} is the end-to-end delay of the jth packet of station i.

N_{ip} is the number of total successful transmitted packets of station i.

N is the number of total stations.

4) Average Buffer Overflow Rate

Generally, when a station generates a packet, the packet is delivered to the buffer memory of the station. However, the buffer size is limited. If the buffer is full, the new generated packet will be discarded, that means buffer overflow. The buffer overflow rate (*RBO*) is defined as the percentage of the packets that a station buffer discards over the total packets that the station generated during the simulation period. Average buffer overflow rate *E[RBO]* is the sum of all station buffer overflow rates divided by the number of stations.

$$RBO_i = \frac{Ni_{overflow}}{Ni_{total}} \qquad E[RBO] = \frac{\sum_{i=1}^{N_s} RBO_i}{N_s}$$

RBO_i is the buffer overflow rate of station i.

$Ni_{overflow}$ is the number of total discarded data packets of source station i.

Ni_{total} is the number of total data packets that station i generates.

N_s is the number of stations.

5) Average Packet Loss Rate

In the simulation, packet loss occurs in two conditions:

- When a packet lifetime period is counted up to its TTL and it is still not delivered, it will be discarded.

- When a packet is transmitted to its destination cell AP that covers the packet's destination station, but the AP cannot find the destination station in its BSS area, which means the destination station has moved out of the BSS; then the packet will be discarded.

The packet loss rate (*RPL*) is defined as the percentage of the total lost packets when a station transmits packets to the destination over the total packets that the station generated during the simulation period. Average packet loss rate *E[RPL]* is the value that the sum of all station packet loss rates divided by the number of stations.

$$RPL_i = \frac{Ni_{loss}}{Ni_{total}} \qquad E[RPL] = \frac{\sum_{i=1}^{N_s} RPL_i}{N_s}$$

RPL_i is the packet loss rate of station i.

Ni_{loss} is the number of lost data packets of the source station i.

Ni_{total} is the number of total data packets that station i generates.

N_s is the number of stations.

2.2 Infrastructure-based DCF simulation

2.2.1 DCF simulation model

This model simulates an infrastructure-based wireless network within an ideal square area of 2000-meter length and 2000-meter width based on distributed coordination function. In the simulation area, 200 wireless stations are active and distributed uniformly; they move only inside the area randomly. To setup an infrastructure network, access points (APs) must be deployed. All stations and APs have same radio transmission coverage range of radius 200 meters. Different number of APs covers different number of STAs in every BSS. All BSS radio service coverage charts are the same as those in infrastructure-based PCF model (refer Figure 2.1). Assuming all APs are distributed uniformly; hence, in this square area, we deploy 9, 14, 25, 36 and 49 APs depending on the case, respectively. From these charts, we can see that at least 49 APs can completely cover this area; that means, all stations are covered by AP's radio transmission range. Assuming that all stations move only inside the area, whereas they can move from one BSS to another BSS or to out of service, they will go back into the area when they reach the area border. We also assume all APs are connected by wired backbone network and packet transmission delay in the backbone network is neglected. The basic channel bit rate of 1Mb/s is used in the simulation. In every BSS, the medium control method is Distributed Coordination Function (DCF) based on CSMA/CA protocol.

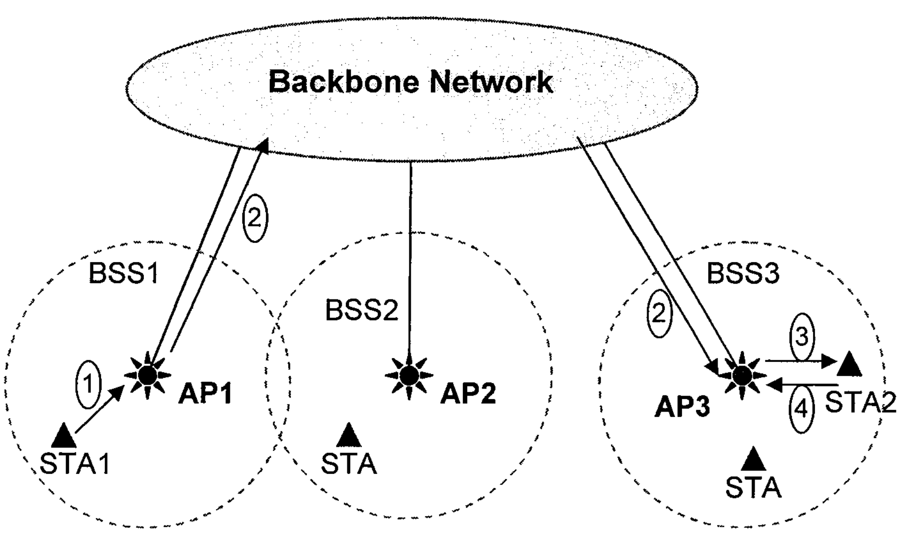


Figure 2.7 Infrastructure DCF transmission procedure

The procedure of infrastructure-based DCF is illustrated in Figure 2.7. In this example, three BSS are connected together via the backbone network. Assuming STA1 in BSS1 has a packet to transmit to the STA2 in BSS3, the data packet transmission procedure is:

① STA1 contends and captures the channel in BSS1 and transmits packet to AP1;

② AP1 forwards the packet to AP3 via backbone network;

③ AP3 captures the medium in BSS3 and transmits packet to destination STA2;

④ STA2 replies ACK to AP3.

2.2.2 Station State Transition in DCF simulation

IEEE 802.11 DCF is based on the use of the CSMA/CA. In the simulation, every station moves from state to state depending on different conditions. The station state transition diagram is shown in Figure 2.8.

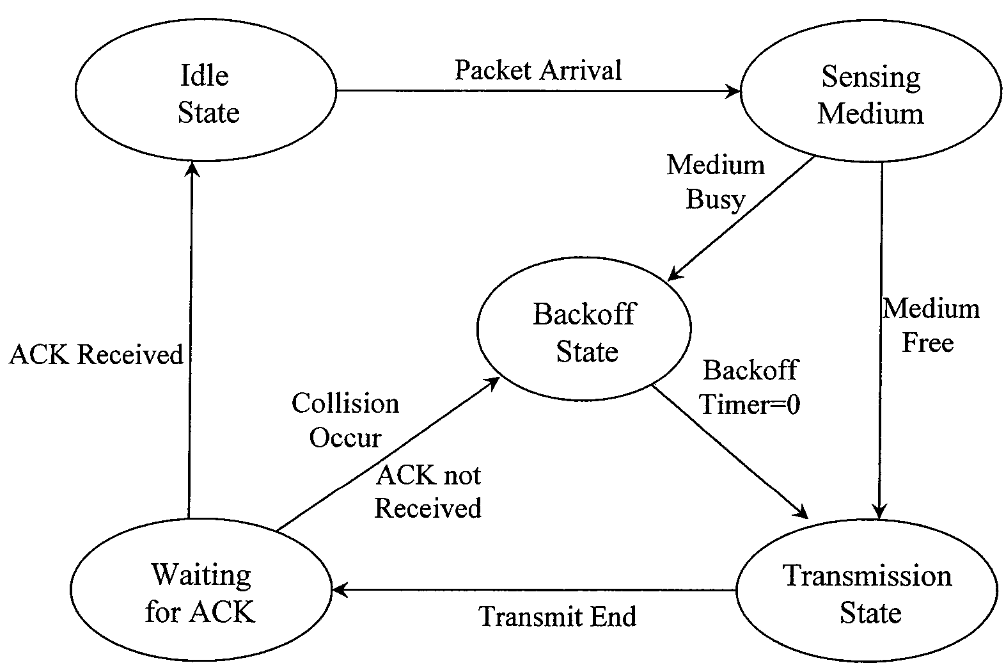


Figure 2.8 Station state transition of DCF

Initially, each station is in the Idle state. When a new packet arrives, it is stored in transmission buffer and the station moves to Sensing Medium state. Depending on free or busy result of sensing the medium, the station state changes either to Transmission state or to Backoff state. In the Backoff state, the station continues to sense the channel and decrement the backoff timer only when the channel is free. When the backoff timer decreases to zero and the channel is still free, the station state changes from Backoff state to Transmission state and the station starts to transmit a packet. On the other hand, when a station is in the Transmission state, if the transmission is completed, the station state transits from Backoff state to Waiting for ACK state. When a station is Waiting for ACK, if the ACK is received, it means the packet is transmitted successfully and then the station state moves to Idle state; if the ACK is not received, it means collision occurred and the transmission failed, then the station state moves to Backoff state after doubling the contention window.

2.2.3 Assumptions for Infrastructure DCF simulation

In the simulation of infrastructure-based DCF WLAN, all input parameters are given in Table 2.2.

Assuming the position of station i in the simulation area is (X_i, Y_i) and the position of a AP_p is (X_p, Y_p), the new location (X'_i, Y'_i) of the station after each location update is

$$\begin{cases} X_i' = X_i + \Delta T \cdot v_i \cdot \cos\theta_i \\ Y_i' = Y_i + \Delta T \cdot v_i \cdot \sin\theta_i \end{cases}$$

where ΔT is the station location update period (=80ms), v_i is the movement velocity of the station i, and θ_i is the movement angle of the station i. If

$$\sqrt{(X_i' - X_p)^2 + (Y_i' - Y_p)^2} \leq 200m$$

then this station is connected with the AP_p (X_p, Y_p).

Attribute	Value
Area	2000m × 2000m
Number of AP	9, 16, 25, 36, 49
Number of stations	200
Channel bit rate	1 Mb/s
Data packet size	8000 bits (8000 μs)
ACK	104 bits (104 μs)
Small Iteration	10 μs
Large Iteration	8000μs (8ms)
SIFS	10 μs
DIFS	50 μs
Station movement velocity	30m/s (maximum)
Station location update period	10 Large Iteration (80ms)
Minimum contention window CW_{min}	8
Maximum contention window CW_{max}	256
TTL	250ms, 1000ms
TxTimes	5, 10

Table 2.2 Default simulation attribute values for DCF

In this simulation, assume the probability that a station generates a packet is P_g. The packet generation process from each station is assumed to follow a Bernoulli trial. A uniformly distributed random generation is called every packet time for a specific station. If the returned value

$x \leq P_g$, the station generates a packet;

$x > P_g$, no packet generate;

P_g value is set to 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, and 1.0 respectively, to control the simulation system data traffic. As soon as a packet is generated, it is delivered to the buffer memory of the station. The size of buffer memory reflects how many packets can be stored simultaneously, and the buffer is FIFO (First In, First Out) buffer. When a packet is transmitted successfully, the packet is removed out of the buffer. In this simulation, the buffer size of all access points and stations is set to 2, 16, and 128 depending on the simulation scenarios.

For the random backoff algorithm, assume the minimum contention window (initial contention window) CW_{min} is 8, and the maximum contention window CW_{max} is 256.

We assume the life period of every packet, TTL (Time To Live), is 250ms and 1000ms, respectively. A packet TTL is counted from its generation time. If a packet life period is counted up to its TTL, it will be discarded.

Another assumption is: all simulations are operated in ideal channel condition without considering the interference and fading effects so as to simplify the simulation; in other words, the channel bit error rate is set to zero and correct packet probability P_c=1.

We assume the maximum number of packets attempting transmission (retransmission), TxTimes, is 5 and 10, respectively. Packets retransmission is due to collision only.

2.2.4 Procedure for the Infrastructure DCF simulation

Similar to the simulation of PCF, firstly, based on the entered parameters, the simulation system initializes all access points and stations configuration, sets the number of the total simulation cycles, and randomly generates all stations location coordinates, movement velocity, and movement direction. Next, the DCF contention period starts and the detailed procedure is presented in section 1.1.3.4. After the preset simulation cycles ends, the system begins to compute all simulation criteria and outputs simulation results into a data file. The simulation procedure flowchart is illustrated as Figure 2.9 and 2.10.

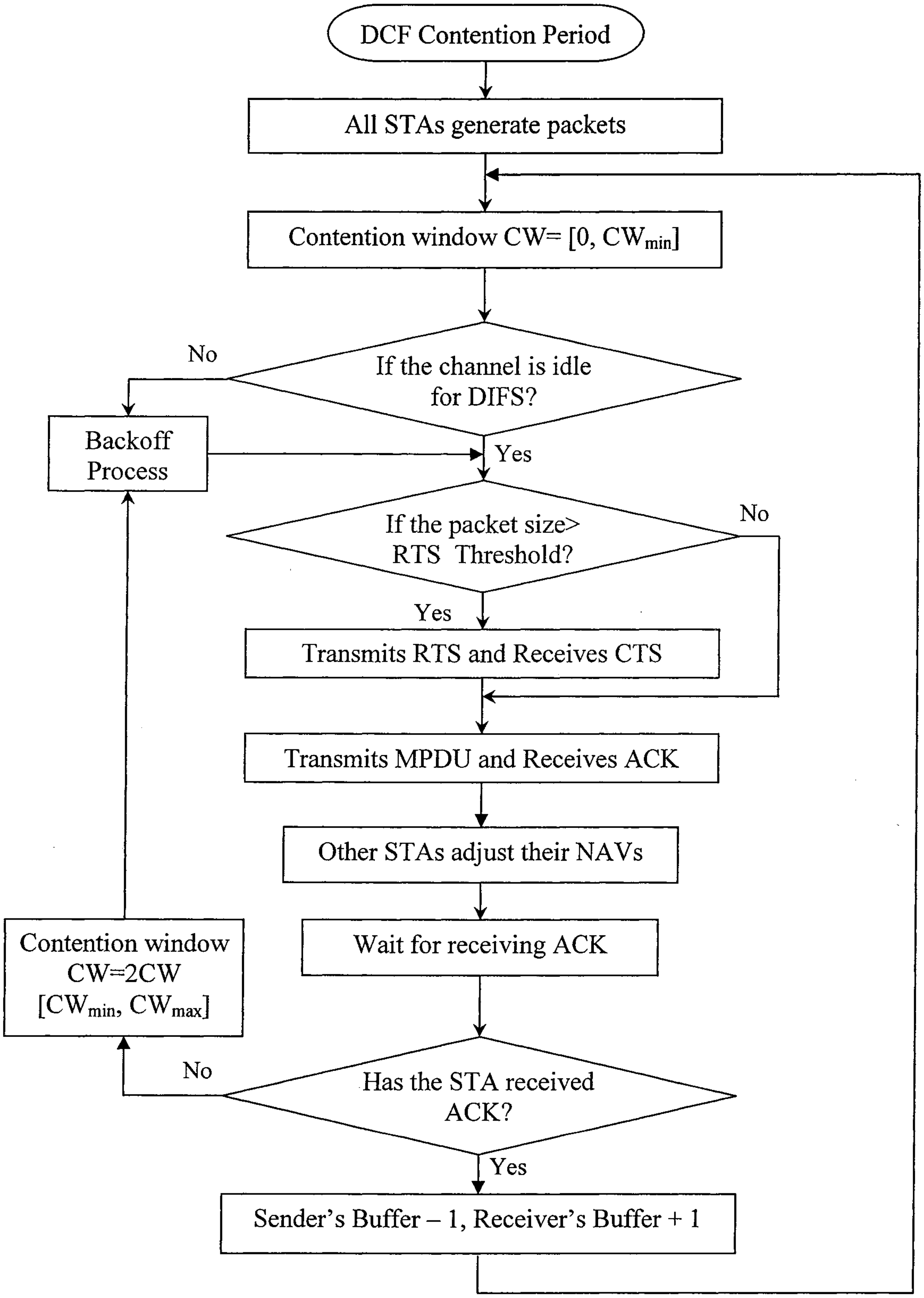


Figure 2.9 Distributed coordinator contention period flowchart

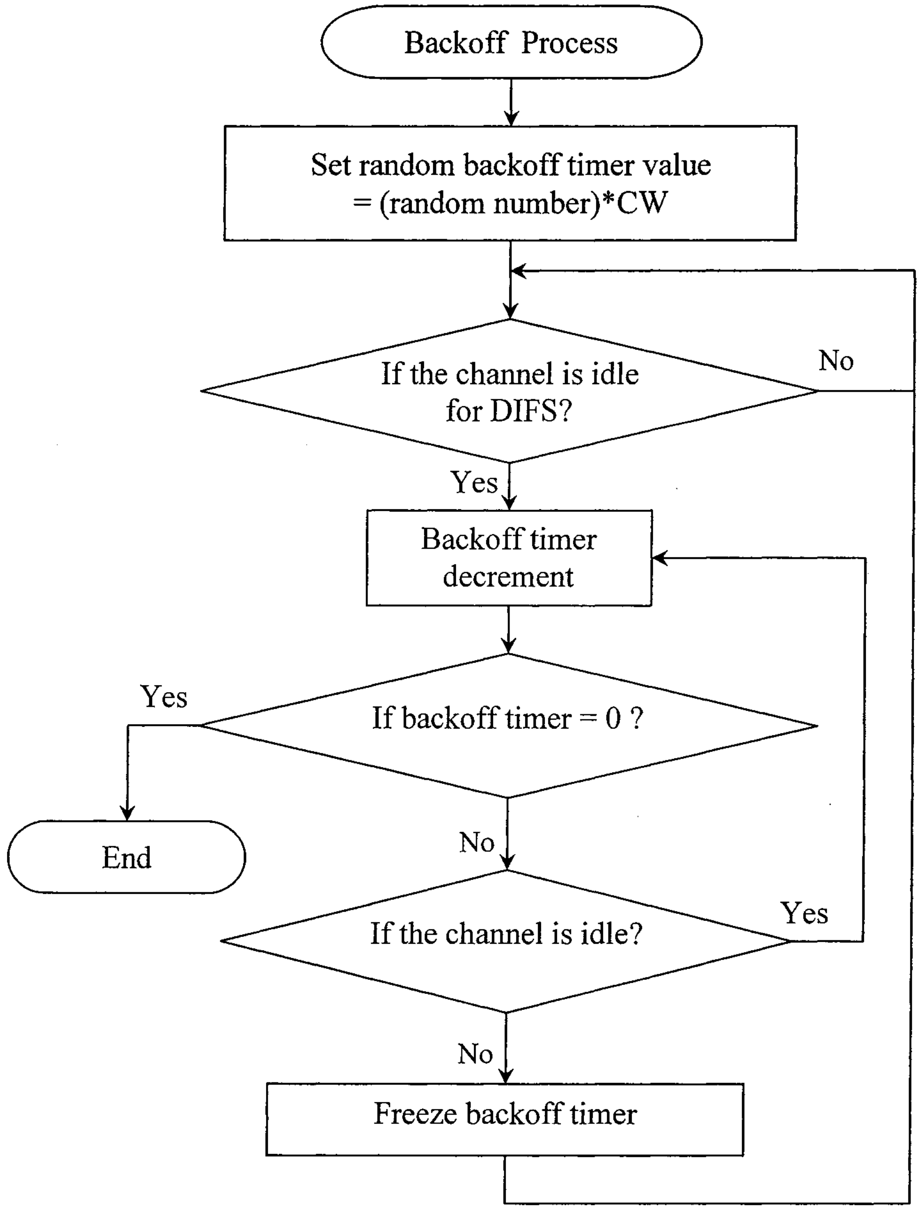


Figure 2.10 Contention backoff flowchart in DCF

2.2.5 Performance Evaluation Criteria

The metrics we use in infrastructure-based DCF simulation are same as those in infrastructure-based PCF, which include normalized average traffic throughput, average end-to-end delay, average packet loss rate, and average buffer overflow rate. In addition, they have same definitions and calculation equations as in section 2.1.1.4.

2.3 HPERLAN Simulation

HIPERLAN standard is intended to establish two type of WLAN: predefined LANs that can be made to form part of, or supplement, fixed LANs, and Ad hoc HIPERLAN that can be rapidly configured for Ad hoc functionality. The simulation goal is evaluating the performance of the EY-NPMA protocol, which is applied in HIPERLAN medium access control.

2.3.1 HIPERLAN simulation model

This model simulates a HIPERLAN wireless network within an ideal round area with 200-meter radius as shown in Figure 2.11. In the simulation area, 100 wireless stations are active and distributed randomly. Every station has a 200-meter-radius radio coverage range and can directly communicates with other stations located in the same area. Assuming all stations move only inside the area randomly, they will go back into the area when they reach the area border. A channel bit rate of 23.5294Mb/s is used in the simulation. For HIPERLAN, channel access is attempted by the CAC entity according to three possible channel conditions: free channel condition, synchronized channel condition, and hidden terminal condition. Here, we only simulate the synchronized channel condition, for which the medium control method is EY-NPMA protocol.

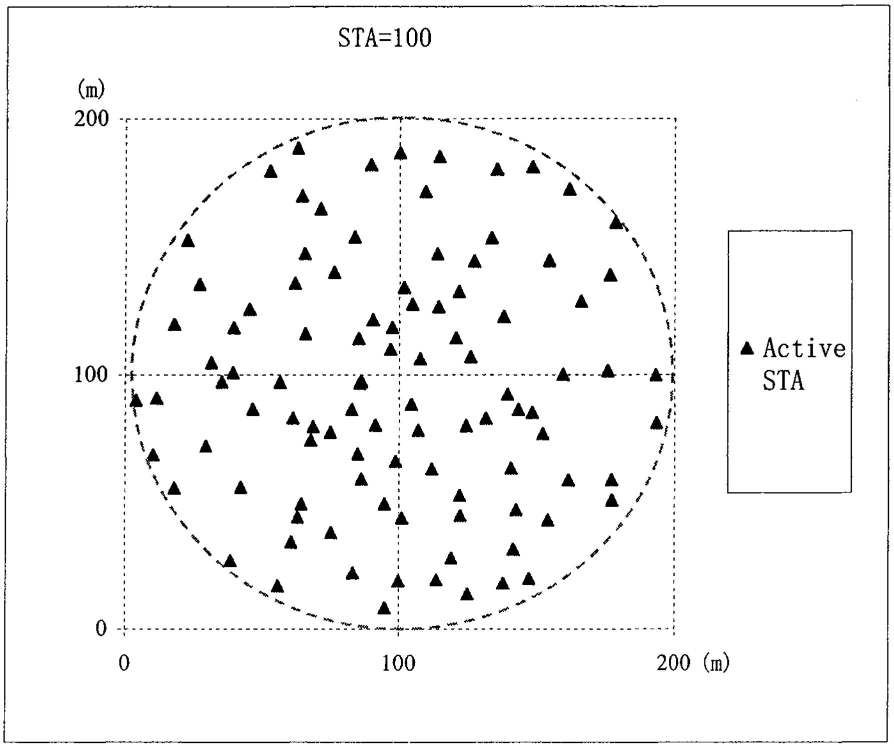


Figure 2.11 HIPERLAN simulation area and stations distribution

2.3.2 Assumptions for HIPERLAN simulation

In the simulation of HIPERLAN WLAN, all assuming values are given in Table 2.3.

Attribute	Value
Area	Radius=200m
Number of stations	100
Channel bit rate	23.5294 Mb/s
Data packet size	8000 bits (340 μs)
ACK	104 bits (4.4 μs)
Small Iteration	10 μs
Large Iteration	340μs
Station movement velocity	30m/s (maximum)
Station location update period	10 Large Iteration (3.4ms)
TTL	250ms, 1000ms
TxTimes	5, 10
Channel packet correct probability P_c	1.0

Table 2.3 Default simulation parameter values for HIPERLAN

Assuming the position of station i in the simulation area is (X_i, Y_i) and the center point coordinate of the simulation area is (X_c, Y_c), the new location (X'_i, Y'_i) of the station after each location update is

$$\begin{cases} X_i' = X_i + \Delta T \cdot v_i \cdot \cos\theta_i \\ Y_i' = Y_i + \Delta T \cdot v_i \cdot \sin\theta_i \end{cases}$$

where ΔT is the station location update period (=3.4ms), v_i is the movement velocity of the station i, and θ_i is the movement angle of the station i. If

$$\sqrt{(X_i' - X_c)^2 + (Y_i' - Y_c)^2} > 100m,$$

this station moves out of the area. Then, let the station movement direction angle

$$\theta_i = \theta_i + \pi/2$$

and recalculate the new location coordinate (X'_i, Y'_i). If the new location is still out of the area, the simulation program will apply the method repeatedly until the new location coordinate is inside the area.

In this simulation, assume the probability that a station generates a packet is P_g. The packet generation process from each station is assumed to follow a Bernoulli trial. A uniformly distributed random generation is called every packet time for a specific station. If the returned value

$x \le P_g$, the station generates a packet;

$x > P_g$, no packet generate;

P_g value is set to 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, and 1.0 respectively, to control the simulation system data traffic. As soon as a packet is generated, the packet is delivered to the buffer memory of the station. The size of buffer memory reflects how many packets can be stored simultaneously, and the buffer is FIFO (First In, First Out) buffer. When a packet is transmitted successfully, the packet is removed from the buffer. In this simulation, the buffer size of all stations is set to 2, 16, and 128 depending on the simulation scenarios.

Assume the life period of every packet, TTL (Time To Live), is 250ms and 1000ms, respectively. A packet TTL is counted from its generation time. If a packet life period is counted up to its TTL, it will be discarded.

Assume the maximum number of packets attempting transmission (retransmission), TxTimes, is 5 and 10, respectively. Packets retransmission is due to collision only.

2.3.3 Procedure for HIPERLAN simulation

Similar to the simulation of IEEE802.11 PCF, the HIPERLAN simulation system first initializes all stations configuration, sets the number of the total simulation cycles, and randomly generates all stations location coordinates, movement velocity, movement direction angle, and priority level based on the entered parameters. After the initialization, the channel access cycle simulation of synchronized channel condition starts. There are three phases in every channel access cycle including the prioritization phase, the contention phase, and the transmission phase. The contention phase is made up of two sub-phases: elimination phase and yielding phase [3] [12] [25].

Prioritization phase

After the start of the access cycle, the station must detect the channel for a number of prioritization slot intervals equal to the priority. Thus, stations that have to transmit traffic with priority 0 may immediately access the channel with a channel access burst. Stations with priority 1 must sense the channel for one slot, stations with priority 2 must wait for 2 slot, etc. the stations send a channel access burst only if they detect that the channel is idle for all the previous slots. Thus, a station with traffic with priority 1 would only access the channel if it did not detect any transmission with priority 0. Five priority levels are defined in HIPERLAN, from 0 to 4. Apparently, the priority 0 is the highest level, and the priority 4 is the lowest. The duration of the prioritization slot is $168T_h \approx 7\mu s$ (T_h is the HBR bit period). The channel access bursts are sent only if in the previous slots the channel is idle, or if there are no stations with higher priority demanding transmission. The prioritization phase procedure is illustrated in Figure 2.12.

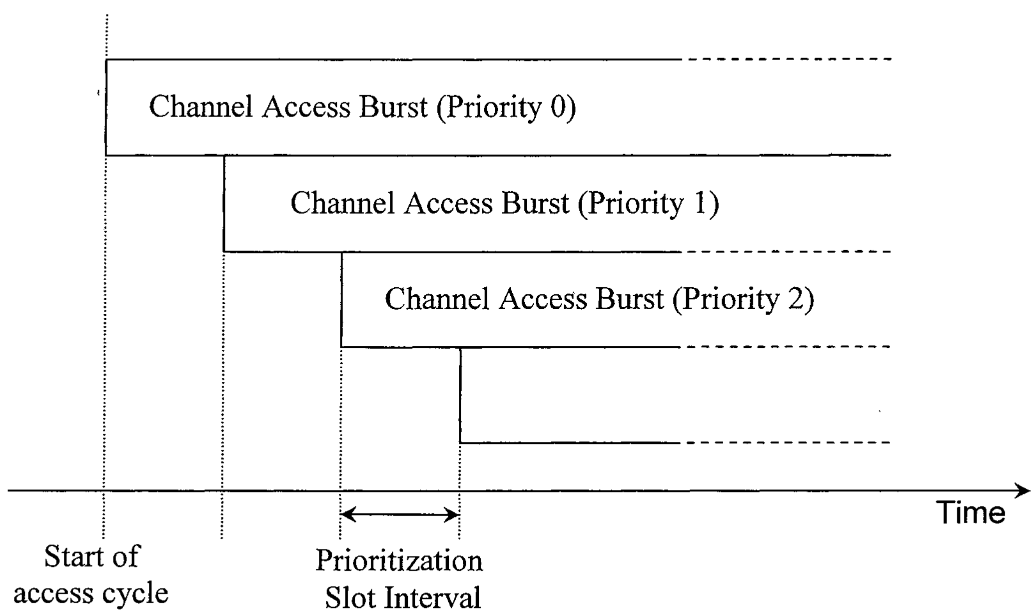


Figure 2.12 HIPERLAN Prioritization phase procedure

Elimination phase

After the prioritization phase, several (or at least one) stations will survive the competition. In face, all stations with the highest priority among the first contenders will survive. To reduce the number of stations, a random procedure is performed in this elimination phase.

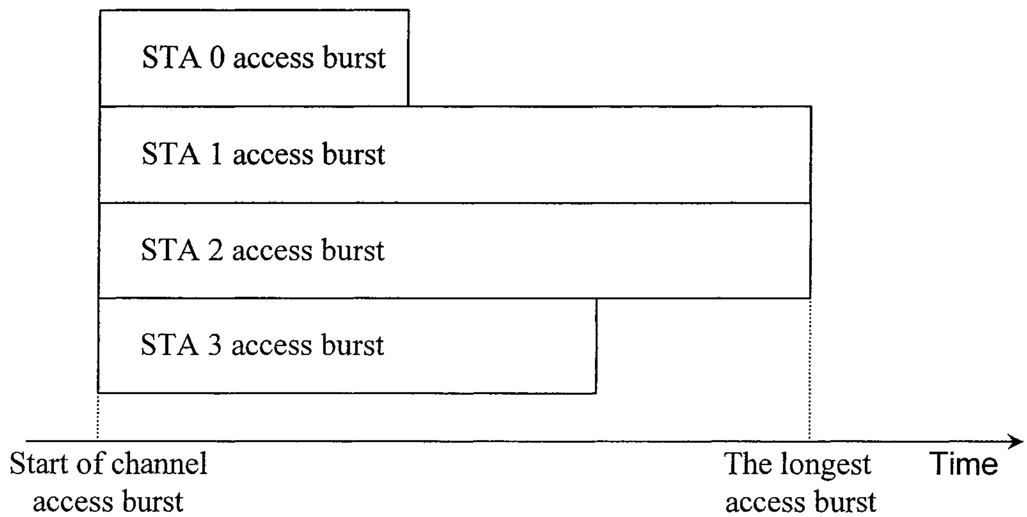


Figure 2.13 HIPERLAN Elimination phase procedure

The channel access burst minimum duration is $168T_h$ equal to the duration of the prioritization slot. However, this is incremented by a random number of elimination slot intervals of $212T_h$ (≈9μs). The range of this random number is between 0 and 12. Thus, every station has a different channel access burst duration. However, on average, 1/12 of the all stations would have the same random number and the same channel access burst duration. After a station completes transmitting the channel access burst, it must check the channel status. If the channel is busy, the station is eliminated by other stations; if the channel is idle, which means this station has the longest channel access burst duration; it survives and goes to the yielding phase. Again, one or more contenders possibly survive this phase. The survivors must detect whether the channel will be idle for the following $256T_h$ after terminating the channel access burst and before proceeding to the next phase. The elimination phase is illustrated in Figure 2.13. STA 0, 1, 2, and 3 have the same priority levels and transmit channel access bursts at the start of the elimination phase. Only STA 1 and STA 2 have the same longest access burst duration, so the two stations survive. STA 0 and STA 3 are eliminated.

Yielding phase

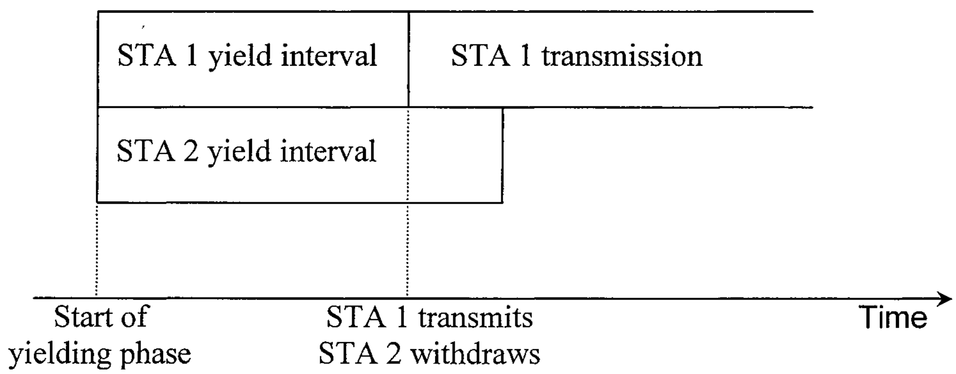


Figure 2.14 HIPERLAN Yielding phase procedure

Once the elimination phase is completed, the surviving stations are asked to yield for a random time interval before the start of transmission. Yielding slot interval is $168T_h$ long. A station must yield for a random number of slot intervals. The random number n is integer variable with uniform distribution between 0 and 9. Thus, the stations with n=0 would be the first to transmit just at the end of elimination phase. If there are no stations with n=0, then the stations with n=1 would transmit, and so on. During the yield interval, if a station detects that the channel is busy, it withdraws. Only the stations with the smallest random n would transmit. Even through random n, 1/10 of the all stations in yielding phase would have the same random number and transmit at the same time; thus, the collision is still possible to occur. Figure 2.14 illustrates the yielding phase. the STA 1 has shorter yielding interval than STA 2; hence, the STA 1 survives to transmit and the STA 2 withdraws.

2.3.4 Performance Evaluation Criteria

The metrics we use in HIPERLAN simulation are same as those in infrastructure-based PCF, which include normalized average traffic throughput, average end-to-end delay, average packet loss rate, and average buffer overflow rate. In addition, they have the same definitions and calculation equations as in section 2.1.1.4.

Chapter 3

WLAN Simulation Results

In this chapter, all simulation results based on the above simulation models of IEEE 802.11 infrastructure-based PCF and DCF wireless LAN and ETSI HIPERLAN are presented. These include average traffic throughput, average end-to-end delay, average buffer overflow rate, and average packet loss rate as function of the offered traffic load P_g, station buffer size, packet lifetime TTL and the maximum packet transmission times TxTimes. These simulation results are presented and then analyzed. Comparison of different WLAN and medium access control algorithms then follows.

3.1 IEEE 802.11 PCF Simulation Results

For the infrastructure-based PCF, we simulate the performance metrics of the polling medium access control method. The following simulation performance results are obtained under certain conditions and simulation environment specified earlier. All simulation results including average traffic throughput, average end-to-end delay, average buffer overflow rate and average packet loss rate are shown versus the offered traffic load, which is represented by the probability of packet generation P_g, the value is set in the range 0.1 to 1.0. The buffer memory size of stations and access points is set to small size of 2, middle size of 16 and large size of 128 to evaluate the effect for performance metrics in different buffer sizes. The packet lifetime period parameter TTL is set to 250ms and 1000ms to evaluate the effect for performance metrics in different packet lifetime period values. The performance metrics could be affected simultaneously by several parameters. The details are shown below.

3.1.1 Average Traffic Throughput

From Figure 3.1 and Figure 3.2, we can see that the average traffic throughput decreases with increasing the probability of offered traffic load. The reason is that more packets are generated during the simulation as the probability of packet generation increases. Heavier traffic load certainly causes longer queuing delay in the buffer; if a packet queuing delay is larger than the packet TTL, the packet will be discarded. Moreover, more offered traffic load causes more buffer overflow and results of more packets dropped. Therefore, the throughput decreases with increasing the probability of offered traffic load.

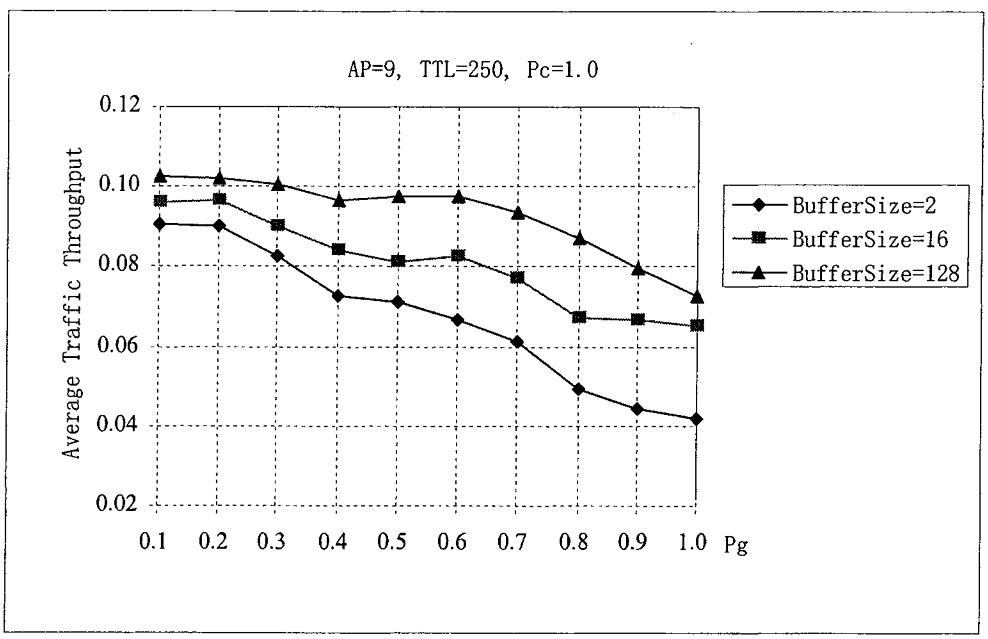


Figure 3.1 Average Traffic Throughput vs. Buffer size (1) (PCF)

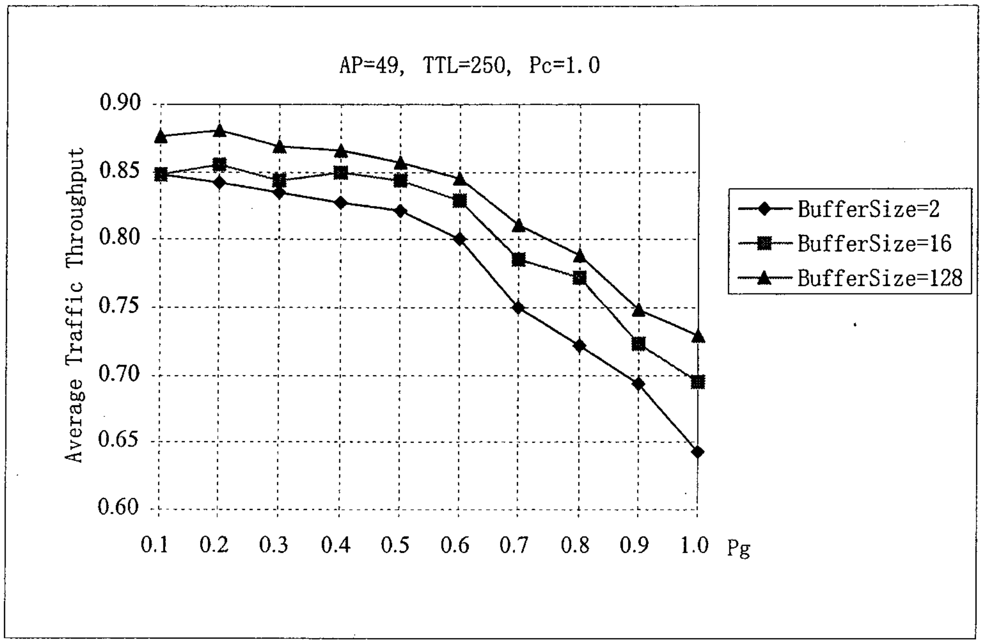


Figure 3.2 Average Traffic Throughput vs. Buffer size (2) (PCF)

On the other hand, the larger buffer size stations have, the larger average throughput than can be obtained. The reason is that if the buffer size is small, only a small number of packets can be put into the buffer. More generated packets cannot be put into the buffer (buffer overflow) and the packet will be discarded. Thus, the larger buffer size, the less generated packets are dropped, and the larger throughput obtained.

The difference between Figure 3.1 and Figure 3.2 is the number of access points. The traffic throughput difference in these two cases is very large. The reason will be explained later.

Figure 3.3 illustrates the difference of average traffic throughputs when the TTL value is different. The larger value TTL results in a little bit more average traffic throughput. The reason is that if a packet has longer lifetime, it can have longer queuing delay in buffer and is not easier to get dropped by TTL timer. On the other hand, the longer lifetime possibly results in more packets discarded due to buffer overflow.

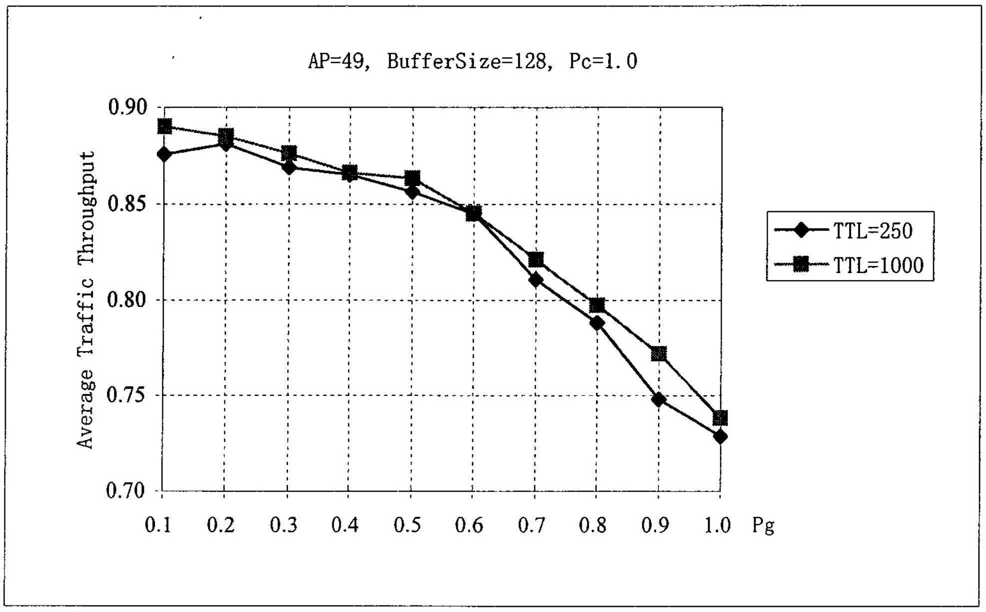


Figure 3.3 Average Traffic Throughput vs. TTL (PCF)

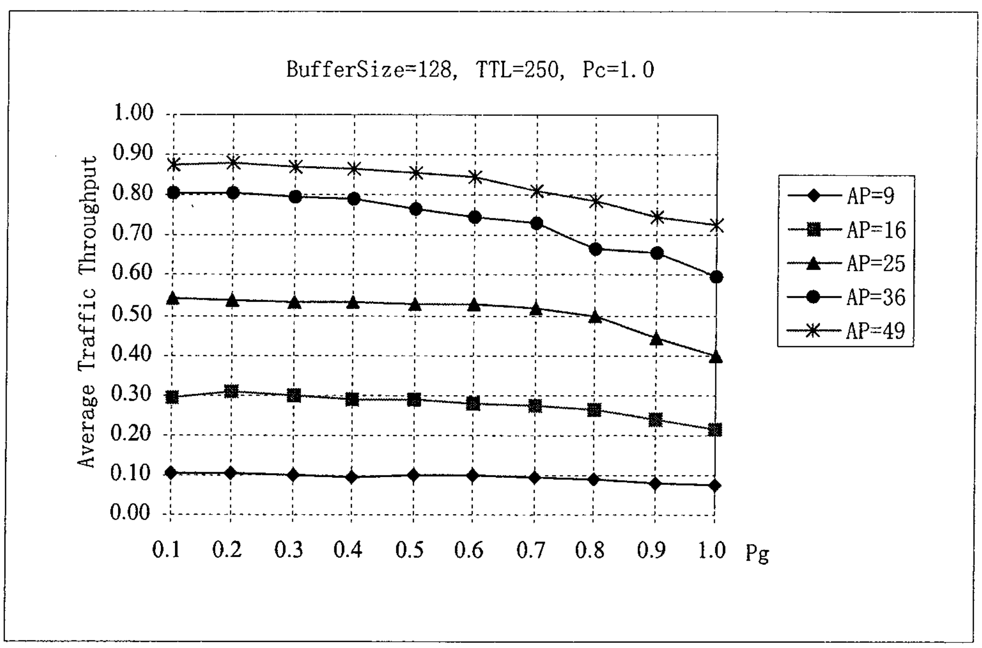


Figure 3.4 Average Traffic Throughput vs. APs (PCF)

Figure 3.4 illustrates the difference of average traffic throughputs when the simulation area is covered by different number of access points. From figure 2.8, we can see that the less is number of AP employed, the less stations are covered by AP radio service. When AP=9, only approximately 28 percent of all stations are covered by 9 access points. Therefore, 72 percent stations are out of radio service; and then, when these stations generate new packets, they will be dropped while their lifetime comes to an end or buffer memory overflow. Hence, the simulation will obtain very low average traffic throughput. On the other hand, when AP=49, 100 percent stations are covered by access points radio service. Most of the packets can be transmitted to their destination; thus, the system maintains the highest average traffic throughput.

3.1.2 Average End-to-end Delay

For the average end-to-end delay, the input parameters are buffer memory size, packet lifetime period TTL, and the number of APs.

From Figure 3.5 and Figure 3.6, we can see that the average end-to-end delay increases with increasing the probability of offered traffic load. The reason is that more packets are generated during the simulation as the probability of packet generation increases. Heavier traffic load certainly causes packets larger queuing delay in the buffer. Because the packet transmission delay and propagation delay do not change with the number of packet in the buffer, the end-to-end delay increases only due to queuing delay.

On the other hand, we can see, the average end-to-end delay increases quickly with larger buffer sizes. The reason is that if the buffer size is large, the more generated packets can be put into the FIFO buffer for queuing. Thus, the queued packets can be transmitted only when its predecessor packets have been transmitted. Therefore, the packet will have more queuing delay in the larger buffer.

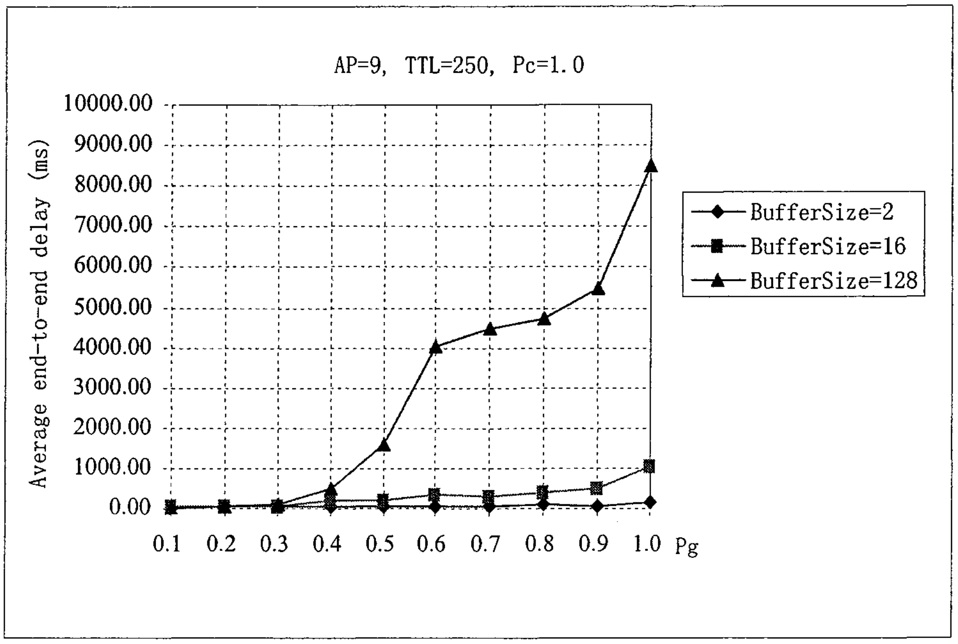


Figure 3.5 Average End-to-end Delay vs. Buffer Size (1) (PCF)

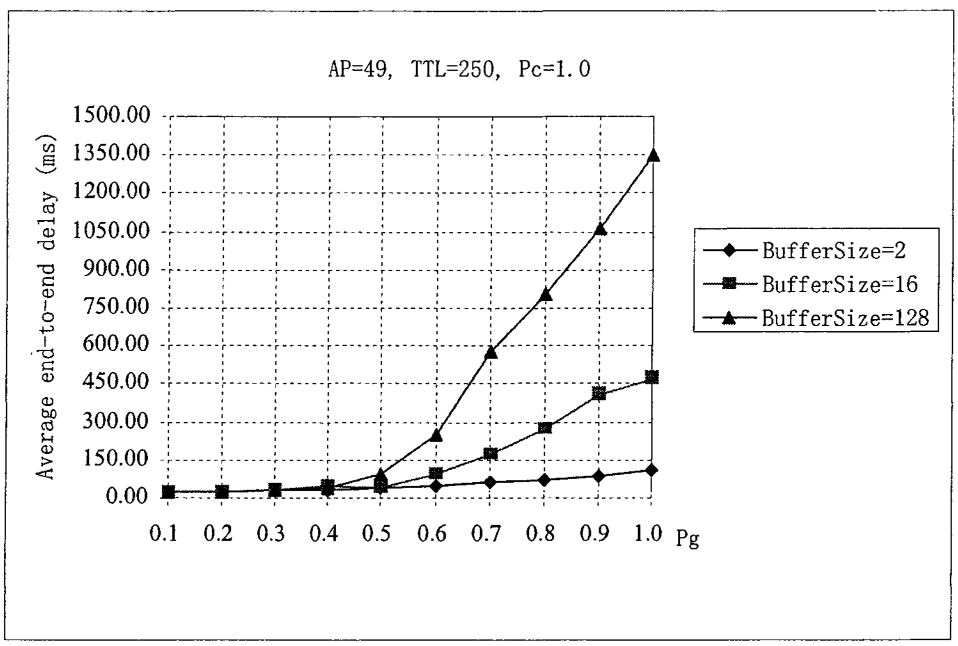


Figure 3.6 Average End-to-end Delay vs. Buffer Size (2) (PCF)

The difference between Figure 3.5 and Figure 3.6 is the number of access points. The average end-to-end delay difference in these two cases is very large. The reason will be explained later.

Figure 3.7 illustrates the difference of average end-to-end delays when the TTL value changes. The larger value TTL results in a little bit more average end-to-end delay. The reason is that if a packet has longer lifetime, it can have longer queuing delay to stay in the buffer and is not easier to be dropped by the TTL timer. On the other hand, the longer lifetime possibly results in more packets discarded due to buffer overflow.

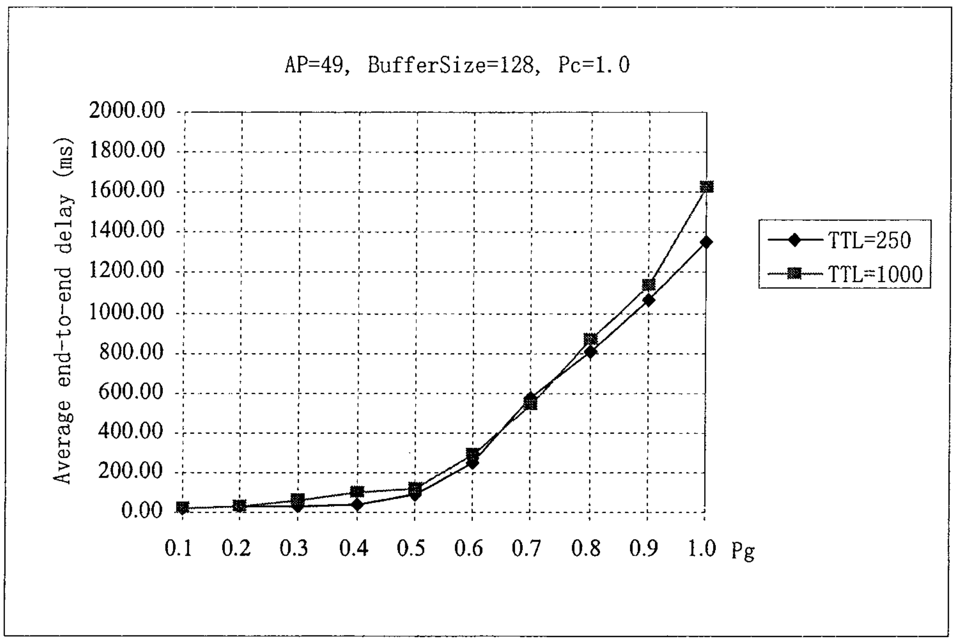


Figure 3.7 Average End-to-end Delay vs. TTL (PCF)

Figure 3.8 illustrates the difference of average end-to-end delays when the simulation area is covered by different number of access points. From Figure 2.8, we can see that the less is number of AP employed, the less stations are covered by AP radio service. For example, when AP=9, only approximately 28 percent of all stations are covered by 9 access points; that means, 72 percent stations are out of radio service. The destination of packets is uniformly distributed among all stations in the area. Thus, we can say that average 72 percent packets that the active stations transmit to APs would be discarded because the AP cannot transmit the packet to the appropriate destination. In this case, it means whole system costs the most of time to transmit useless packets so that the successful packets have to stay in the buffer longer time to wait for service. Hence, the average end-to-end delay becomes very large. On the other hand, when AP=49, 100 percent stations are covered by access points radio service. The packets that the active stations transmit to APs can be transmitted to appropriate destination except the destination station has moved out of the cell. Therefore, the average end-to-end delay becomes very small.

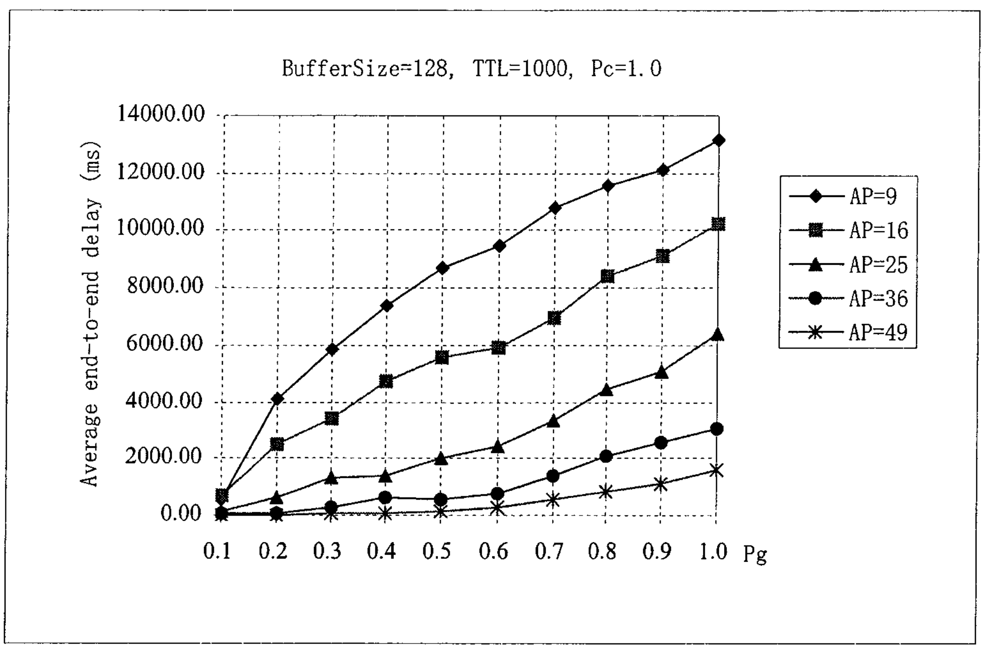


Figure 3.8 Average End-to-end Delay vs. the number of APs (PCF)

3.1.3 Average Buffer Overflow Rate

For the average buffer overflow rate, the simulation input parameters are buffer memory size, packet lifetime period TTL, and the number of APs.

Figure 3.9 depicts the average buffer overflow rate versus buffer size with respect to offered traffic load. We can see that the average buffer overflow rate increases with increasing the probability of offered traffic load. The reason is that more packets are generated during the simulation as the probability of packet generation increases, so more generated packets are put into the FIFO buffer. Since the buffer size is fixed, the buffer is filled by packets faster with increasing offered load; thus, the buffer overflow increases.

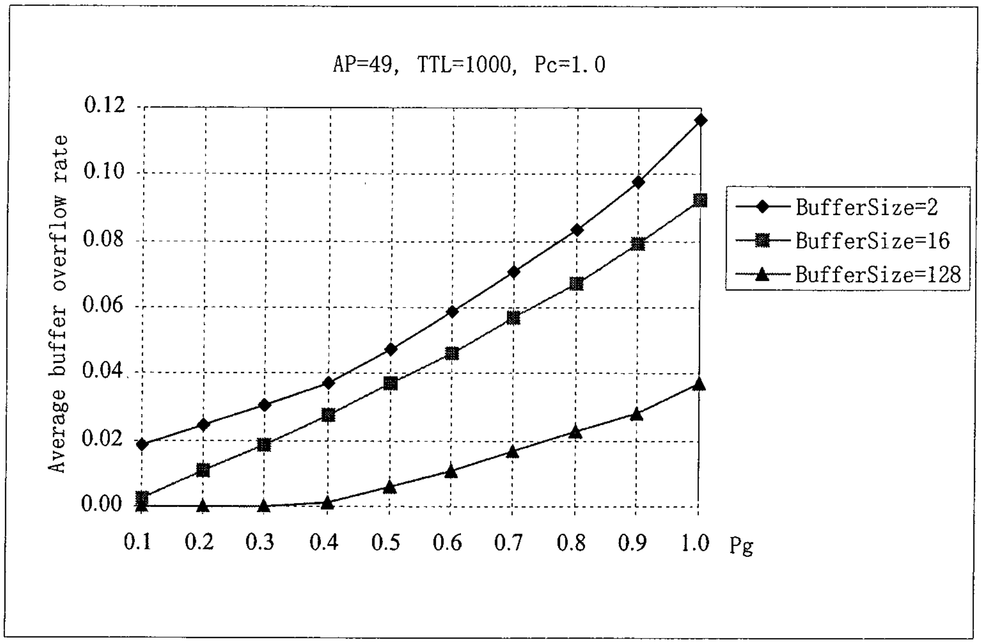


Figure 3.9 Average Buffer Overflow Rate vs. Buffer Size (PCF)

In addition, we can see from the chart, the average buffer overflow rate with smaller buffer size is always larger than that with larger buffer size. When the buffer size is 2, approximately 2 percent packets are dropped due to buffer overflow if offered load rate is 0.1; whereas no packets are dropped when the buffer size is 128 if offered load rate is 0.1 and approximately 2 percent packets are dropped due to buffer overflow until the offered load rate is up to 0.75. Therefore, the larger buffer size gives system performance more promotion.

Figure 3.10 shows the average buffer overflow rates versus packet lifetime TTL with respect to the offered traffic load. The average buffer overflow rate increases with increasing the probability of offered traffic load. Reasons remain the same as explained above. Another point is the larger TTL results in higher buffer overflow rate. The reason is that if a packet has longer lifetime, it has longer queuing delay to stay in the buffer and is not easier to be dropped by TTL timer. Then, more packets will stay in the buffer waiting for transmission so that the buffer is filled quickly and then overflows. With smaller TTL, packets may be dropped by TTL timer when the packets stay enough time equal to TTL in the buffer.

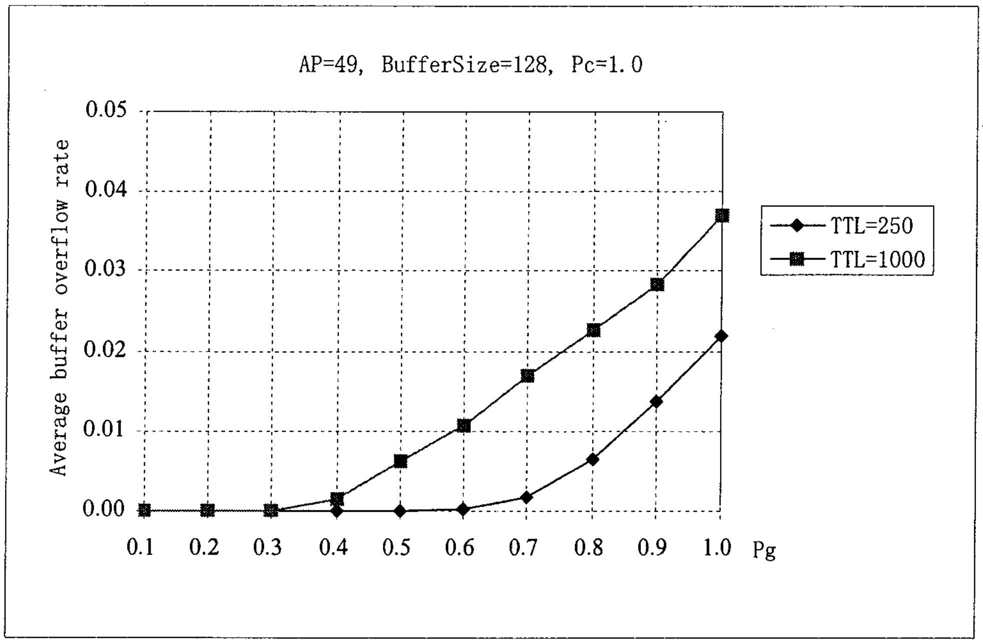


Figure 3.10 Average Buffer Overflow Rate vs. TTL (PCF)

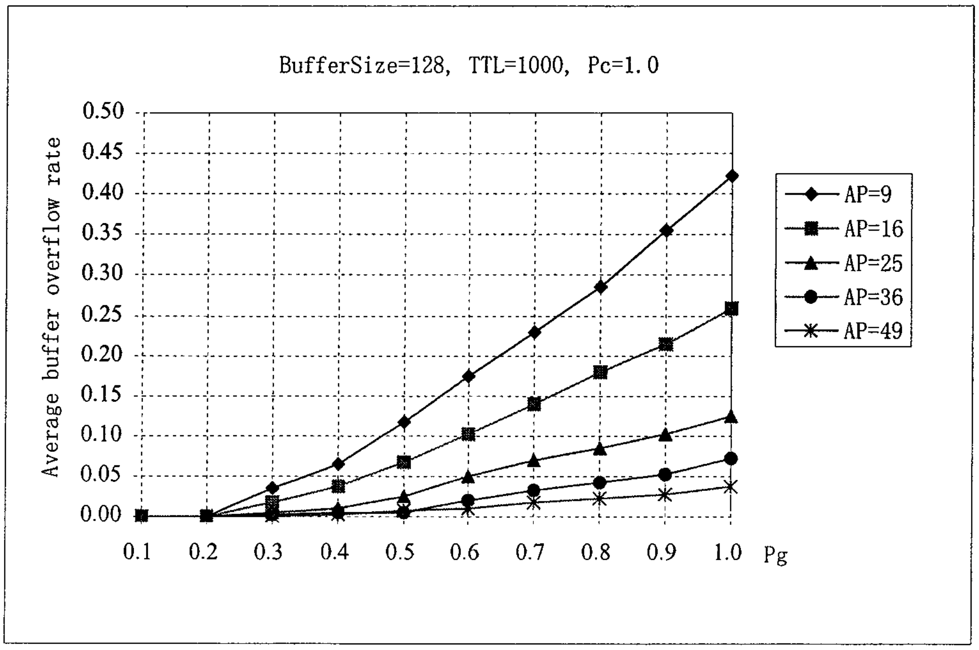


Figure 3.11 Average Buffer Overflow Rate vs. the number of APs (PCF)

Figure 3.11 shows the average buffer overflow rates versus the number of APs as a function of the offered traffic load. From Figure 2.8, we can see that the less the number of AP employ, the less stations are covered by AP radio service and lower average buffer overflow rate. For example, when AP=9, only approximately 28 percent of all stations are covered by 9 access points; that means, 72 percent stations are out of radio service. The destination of packets is uniformly distributed among all stations in the area. Thus, we can say that an average 72 percent of the packets that the active stations transmit to APs would be discarded because the AP cannot transmit the packet to the appropriate destination. This means most of time is spent to transmit useless packets so that the packets have to stay in the buffer longer time to wait for transmitting. Hence, more generated packets would be dropped due to buffer overflow; so, the average buffer overflow rates becomes higher. On the other hand, when AP=49, 100 percent stations are covered by access points radio service. The packets that the active stations transmit to

APs can be transmitted to appropriate destination except if destination station has moved out of the cell. Therefore, the average buffer overflow rate becomes lower.

3.1.4 Average Packet Loss Rate

For the average packet loss rate, the simulation input parameters are the buffer memory size, packet lifetime TT, and the number of APs. Packet loss is attributed to two cases: a) a packet lifetime period is counted up to its TTL and it is still not transmitted, it will be discarded; and b) when AP cannot find the destination station in its cell, which means the destination station has moved out of the BSS, then the packet will be discarded.

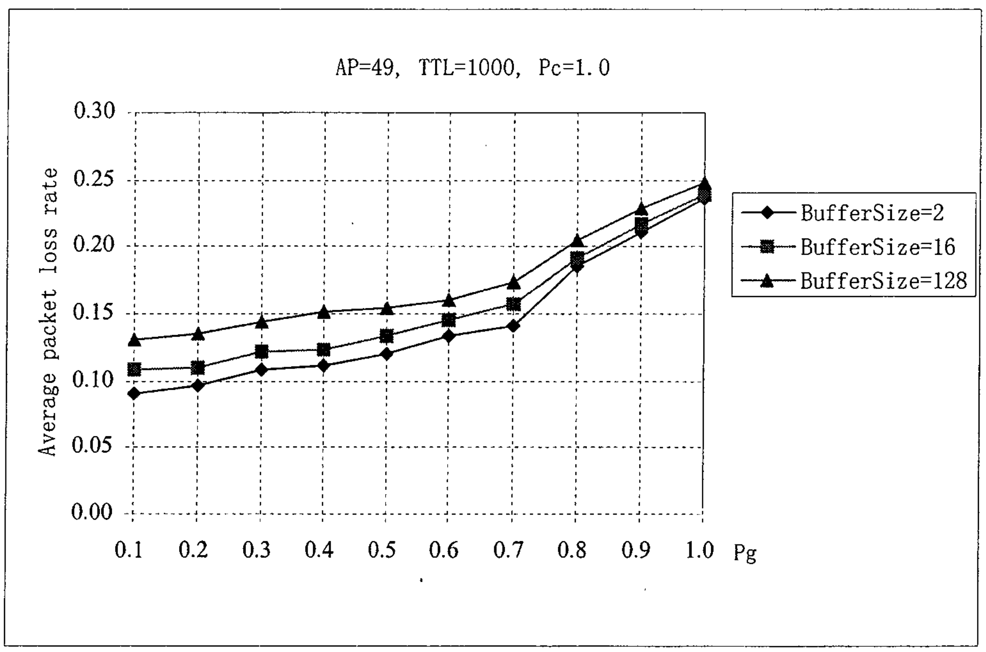


Figure 3.12 Average Packet Loss Rate vs. Buffer Size (PCF)

Figure 3.12 shows the average packet loss rate versus buffer size with respect to offered traffic load. From that, we can see that the average packet loss rate smoothly increases with increasing the probability of offered traffic load. The reason is that more packets are generated during the simulation as the probability of packet generation increases and then causes packets a longer queuing delay in the buffer. When the queuing delay of a packet is counted up to its TTL, the packet will be dropped and the packet is lost. A larger buffer size will also cause longer queuing delay, and then causes higher packet loss rate.

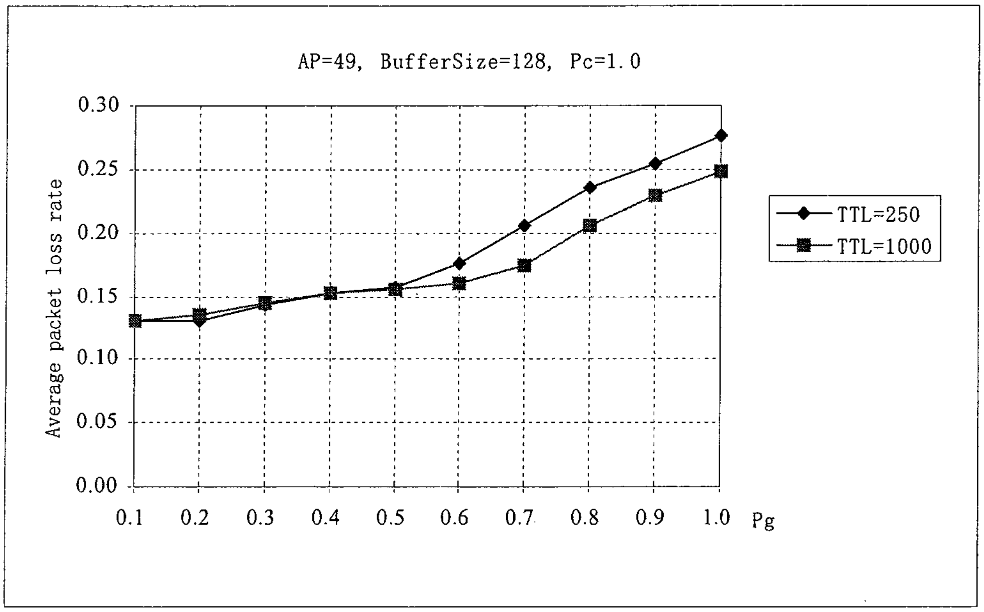


Figure 3.13 Average Packet Loss Rate vs. TTL (PCF)

Figure 3.13 shows the average packet loss rate versus TTL with respect to the offered traffic load. We can see that the average packet loss rate with smaller TTL becomes higher than that with larger TTL as the offered load increases. The reason is that the larger TTL results in fewer packets being dropped due to TTL timer; and then the packet loss rate becomes lower.

Figure 3.14 shows the average packet loss rate versus the number of APs with respect to offered traffic load. From this figure, we can see that the higher number of APs causes a lower average packet loss rate. The reason is that the more APs cover more stations and forward more packets to the destination successfully; so the packet loss rate is lower. By contrary, a lower number of APs cover less stations and most of packets are lost, so the packet loss rate is higher.

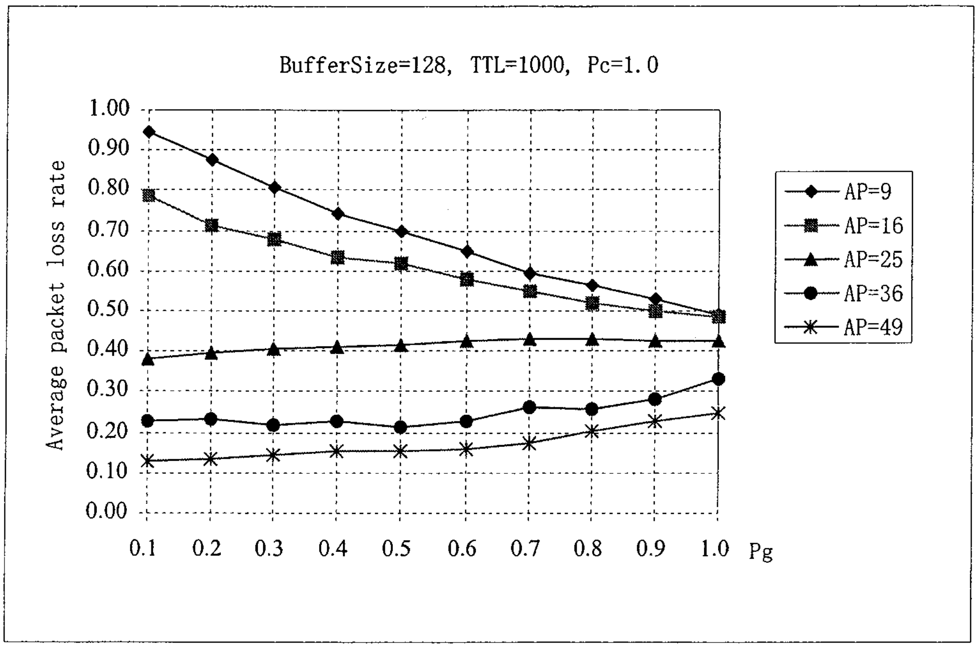


Figure 3.14 Average Packet Loss Rate vs. the number of APs (PCF)

We can also see that different varying tendencies with different number of APs as the probability of offered traffic load increase. With larger number (25, 36 and 49) of APs, the average packet loss rate gently increases with increasing the probability of offered traffic load. On the other hand, with smaller number (9 and 16) of APs, the average packet loss rate gently decreases with increasing the probability of offered traffic load. This can be explained by Figure 3.15, which combines Figure 3.11 and Figure 3.14.

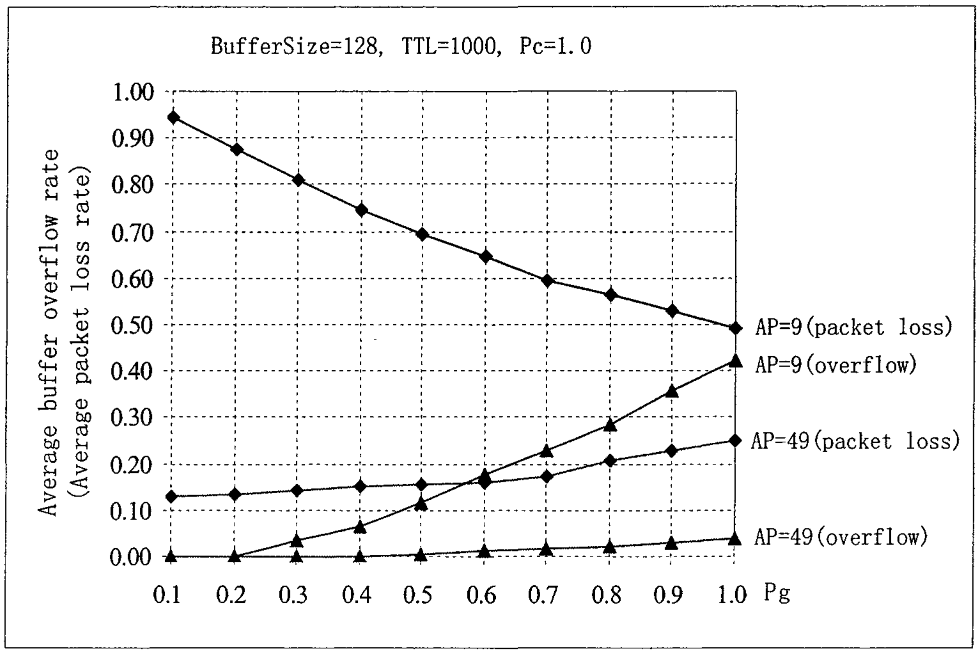


Figure 3.15 Average Buffer Overflow Rate &Packet Loss Rate
vs. the number of APs (PCF)

From Figure 3.15, we can see that when AP=9, the average buffer overflow rate increases from R_{bo}=0 at P_g =0.1 to R_{bo}=0.42 at P_g =1.0. As explained before, when AP=9, average 72 percent packets would be discarded because the AP cannot transmit the packet to the appropriate destination. Assuming the total number of generated packets is N_{total}, the packet loss rate $RPL_{0.1}$ at P_g =0.1 is

$$RPL_{0.1} = \frac{N_{total} \times (1 - R_{bo}) \times 72\%}{N_{total}} = \frac{N_{total} \times (1 - 0) \times 72\%}{N_{total}} = 0.72 \qquad (3\text{-}1)$$

and the packet loss rate $RPL_{1.0}$ at P_g =1.0 is

$$RPL_{1.0} = \frac{N_{total} \times (1 - R_{bo}) \times 72\%}{N_{total}} = \frac{N_{total} \times (1 - 0.42) \times 72\%}{N_{total}} \approx 0.41 \qquad (3\text{-}2)$$

From the figure, we can see the value $RPL_{0.1}$ is 0.94 (> 0.72), since this value includes not only dropped packets due a destination AP cannot find the destination station in its cell (0.72) but also dropped packets by TTL timer. Similarly, the value $RPL_{1.0}$ from the figure is 0.49 (> 0.41) for the same reason. Therefore, the average packet loss rate decreases while the probability of offered traffic load increases is reasonable.

When AP=49, we can see from the figure, the average packet loss rate increases from $RPL_{0.1}$ =0.13 at P_g =0.1 to $RPL_{1.0}$ =0.25 at P_g =1.0. As explained before, when AP=49, no packets would be discarded except if the destination station has moved out of the cell (with a very small probability). In this scenario, the packet loss is mainly attributed to the packets dropped by TTL timer, and its average packet loss rate gently increases with increasing the probability of offered traffic load.

3.2 IEEE 802.11 DCF Simulation Results

In the simulation of the infrastructure-based DCF, we simulate the performance metrics of the CSMA/CA MAC channel access control protocol. The following simulation performance results are obtained under certain conditions and simulation environment specified earlier in the simulation model. We assume the channel packet correct probability P_c=1.0, which means there are no interference, no fading in the channel to cause transmission failure. All simulation results including average traffic throughput, average end-to-end delay, average buffer overflow rate and average packet loss rate are evaluated as functions of the offered traffic load, which is represented by the probability of packet generation P_g, the value is set in the range 0.1 to 1.0. The buffer memory size of stations and access points is set to small size of 2, middle size of 16 and large size of 128 to evaluate the effect on performance metrics of different buffer sizes. The packet lifetime period parameter TTL is set to 250ms and 1000ms to evaluate the effect on performance metrics for different packet lifetime period values. The maximum attempting transmission times parameter TxTimes is set to 5 and 10 to evaluate the effect for performance metrics in different maximum transmission times. The performance metrics could be affected simultaneously by several parameters. The detail will be discussed as following.

3.2.1 Average Traffic Throughput

For the average traffic throughput, the simulation parameters are the buffer memory size, packet lifetime period TTL, the maximum attempting transmission times TxTimes, and the number of APs.

From Figure 3.16, we can see that the average traffic throughput decreases with increasing the probability of offered traffic load. The reason is that more packets are generated during the simulation as the probability of packet generation increases. Heavier traffic load certainly causes longer queuing delay in the buffer; if a packet queuing delay is larger than the packet TTL, the packet will be dropped. In addition, more offered traffic load causes more buffer overflow and results in more packets dropping. Therefore, the throughput decreases with increasing the probability of offered traffic load.

Moreover, the larger the buffer size stations have, the larger the average throughput that can be obtained. The reason is that if the buffer size is small, only a small number of packets can be put into the buffer. More generated packets will be discarded because the buffer overflows. Thus, the larger buffer size the stations have, the less generated packets are dropped, and the larger the throughput obtained.

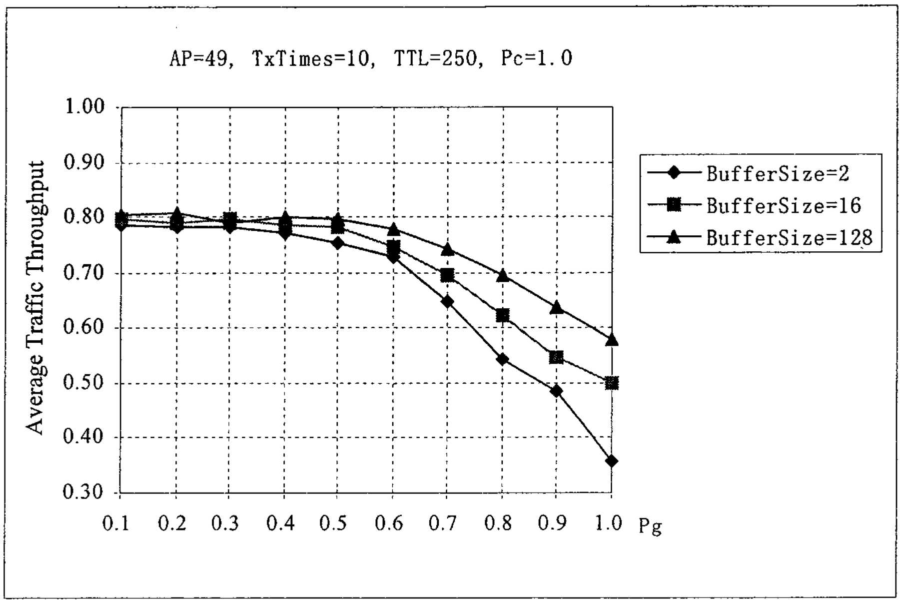


Figure 3.16 Average Traffic Throughput vs. Buffer Size (DCF)

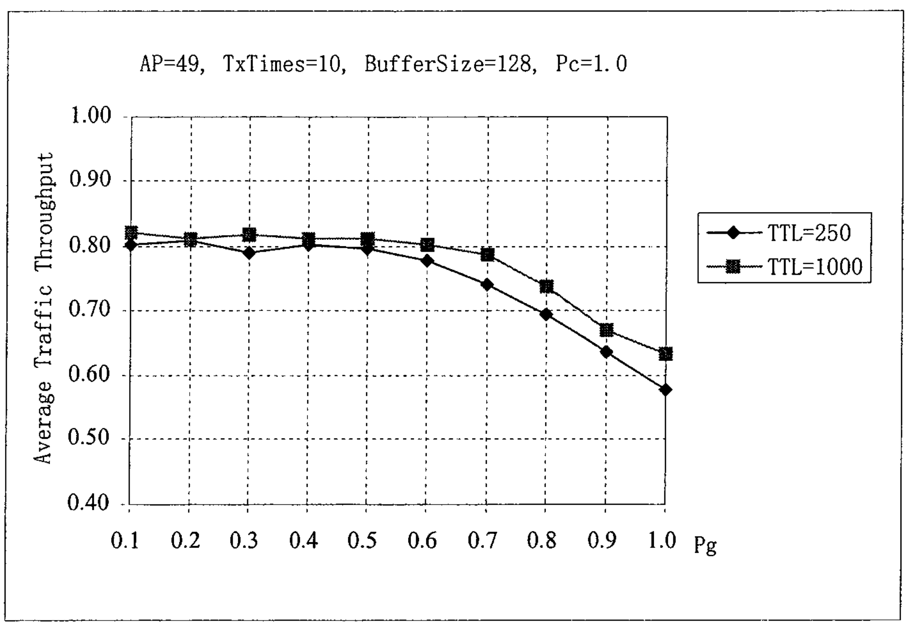


Figure 3.17 Average Traffic Throughput vs. TTL (DCF)

Figure 3.17 illustrates the difference in average traffic throughputs for different TTL values. The figure shows that larger value TTL results in a little bit more average traffic throughput. The reason is that if a packet has longer lifetime, it can have longer queuing delay in the buffer and is not easier to be dropped by TTL timer. On the other hand, the longer lifetime possibly results in more packets being discarded due to buffer overflow.

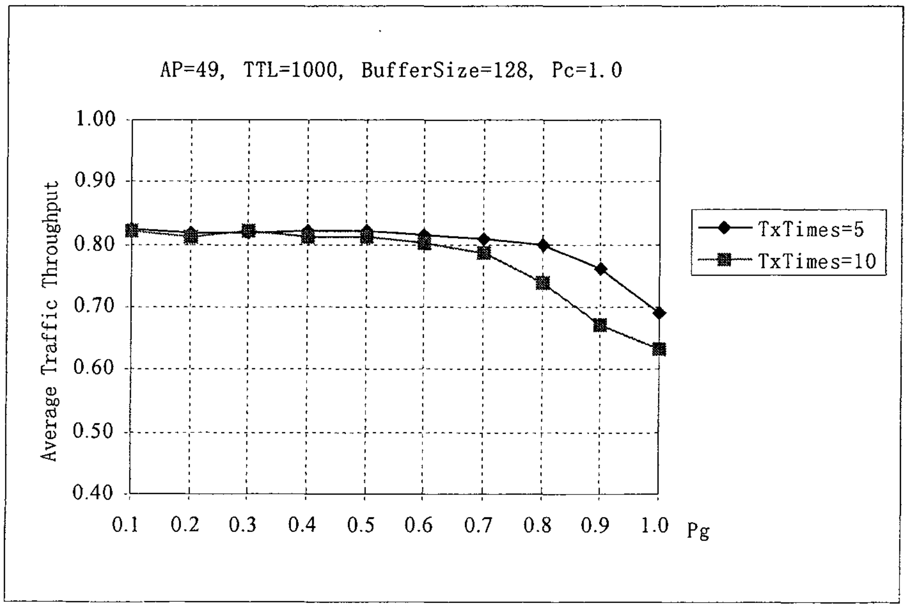


Figure 3.18 Average Traffic Throughput vs. TxTimes (DCF)

Figure 3.18 shows the difference of average traffic throughputs when the maximum transmission times (TxTimes) value is different. From the figure, we can see that the larger TxTimes value results in a little bit more average traffic throughput while the offered traffic load increases. The reason is that if a packet has larger TxTimes value, it has more chances to attempt transmissions; so, more packets would be transmitted. On the other hand, the larger TxTimes value definitely results in longer queuing delay in the buffer so that some of the packets would be discarded due to the TTL timer or the buffer overflow.

Figure 3.19 illustrates the difference of average traffic throughputs when the simulation area is covered by different number of access points. Despite the number of

APs, the average traffic throughput decreases with increasing the probability of offered traffic load, only the larger number of APs will obtain higher average throughput. From earlier discussion, we know that the less number of APs employed, the less stations are covered by AP radio service.

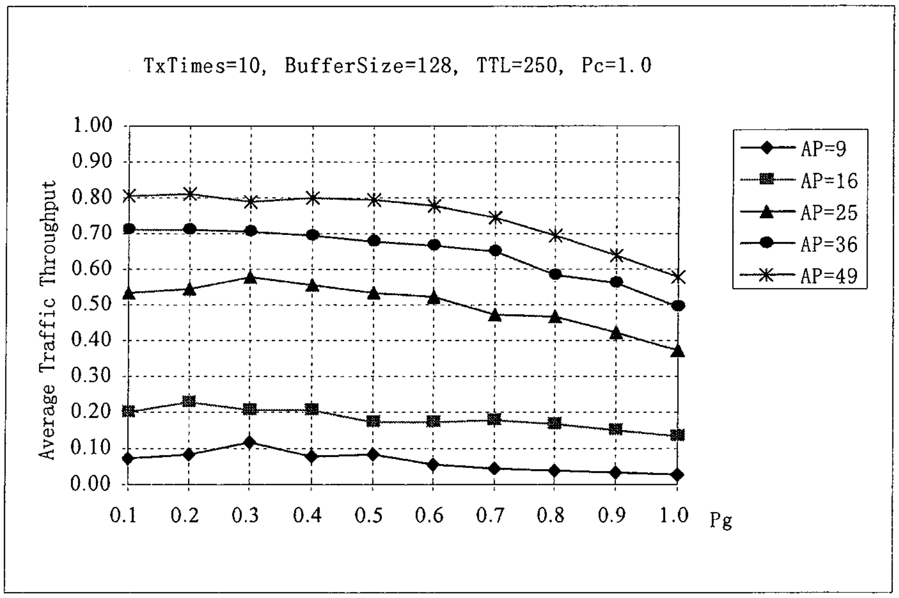


Figure 3.19 Average Traffic Throughput vs. the number of APs (DCF)

From the figure, we can see only approximately 28 percent of all stations are covered by 9 access points when the number of AP is 9. Therefore, 72 percent stations are out of radio service; and then, when these stations generate new packets, they will be dropped definitely while their lifetime comes to end or buffer memory overflows. Hence, the simulation shows very low average traffic throughput. On the other hand, when AP=49, 100 percent stations are covered by access points radio service. The most number of packets can be transmitted to their destination; thus, the system obtains the highest average traffic throughput.

3.2.2 Average End-to-end Delay

For the average end-to-end delay, the simulation input parameters are buffer memory size, packet lifetime period TTL, the maximum attempting transmission times TxTimes, and the number of APs.

Figure 3.20 shows that the average end-to-end delay increases with increasing the probability of offered traffic load. The reason is that more packets are generated during the simulation as the probability of packet generation increases. Heavier traffic load certainly causes packets larger queuing delay in the buffer. Because the packet transmission delay and propagation delay do not change with the number of packet in the buffer, the end-to-end delay increases only due to queuing delay.

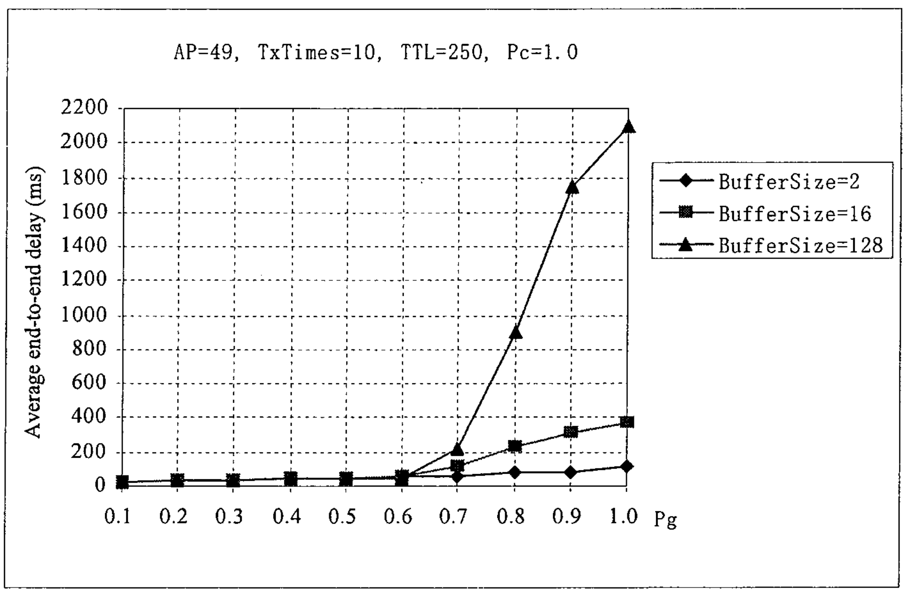


Figure 3.20 Average End-to-end Delay vs. Buffer Size (DCF)

On the other hand, we can see, the average end-to-end delay increases quickly with larger buffer size. The reason is that if the buffer size is large, the more generated packets can be put into the FIFO buffer for queuing. Thus, the queued packet can be transmitted only when its predecessor packets have been transmitted. Therefore, the packet will have more queuing delay in the larger buffer.

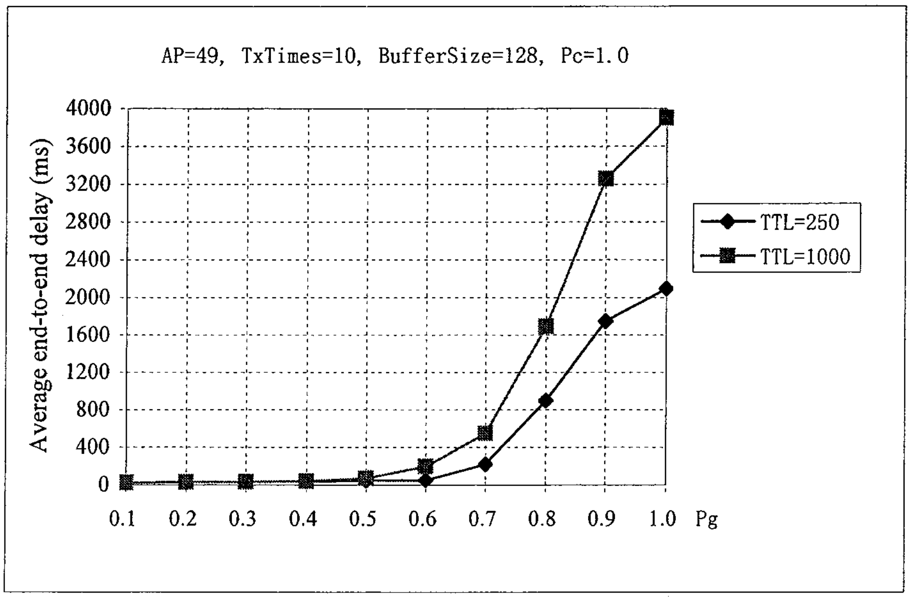


Figure 3.21 Average End-to-end Delay vs. TTL (DCF)

Figure 3.21 illustrates the difference of average end-to-end delays when the TTL value changes. The larger value TTL results in a little bit more average end-to-end delay. The reason is that if a packet has longer lifetime, it can have longer queuing delay to stay in the buffer and is not easier to be dropped by the TTL timer. On the other hand, the longer lifetime possibly results in more packets discarded due to buffer overflow.

Figure 3.22 shows the difference of average end-to-end delays when the maximum attempting transmission TxTimes value changes. The larger TxTimes value results in a little bit more average end-to-end delay when the probability of offered load is larger than 0.6. The reason is that if a packet has more chances to attempt transmitting, it will have longer queuing delay to stay in the buffer. When the traffic load is light, a packet would transmit quickly without retransmissions. However, when the traffic becomes heavier, a packet would need more and more retransmissions since the heavier traffic will result in more collisions.

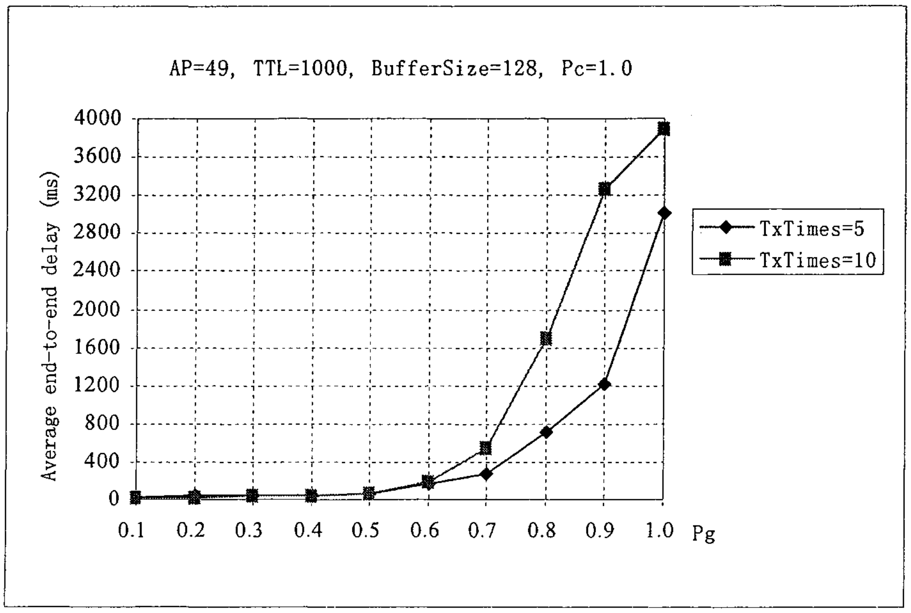


Figure 3.22 Average End-to-end Delay vs. TxTimes (DCF)

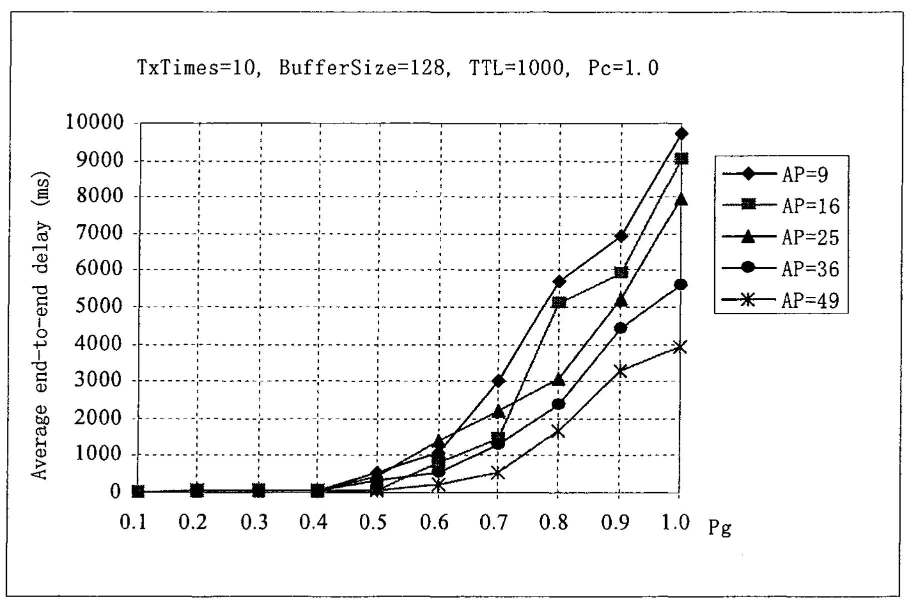


Figure 3.23 Average End-to-end Delay vs. the number of APs (DCF)

Figure 3.23 shows the difference of average end-to-end delays when the simulation area is covered by different number of access points. Despite the number of APs, the average end-to-end delay increases quickly with increasing the probability of offered traffic load; only the larger number of APs will obtain a little smaller average end-to-end delay. We have known that the less is the number of APs employed, the less stations are covered by AP radio service. In this case, the system has to cost the most of time to transmit useless packets so that the successful packets have to stay in the buffer longer time to wait for service. Hence, the average end-to-end delay becomes very large. On the other hand, when the number of APs equals to 49, 100 percent stations are covered by access points radio service. The packets that the active stations transmit to APs can be transmitted to appropriate destination except the destination station has moved out of the cell. Therefore, the average end-to-end delay becomes very small.

3.2.3 Average Buffer Overflow Rate

For the average buffer overflow rate, the simulation input parameters are buffer memory size, packet lifetime period TTL, the maximum attempting transmission times TxTimes, and the number of APs.

Figure 3.24 shows the average buffer overflow rate versus buffer size with respect to offered traffic load. We can see that the average buffer overflow rate increases with increasing the probability of offered traffic load. The reason is that more packets are generated during the simulation as the probability of packet generation increases; so, more generated packets are put into the FIFO buffer. Since the buffer size is fixed, the buffer is filled by packets faster with increasing offered load; thus, the buffer overflow increases.

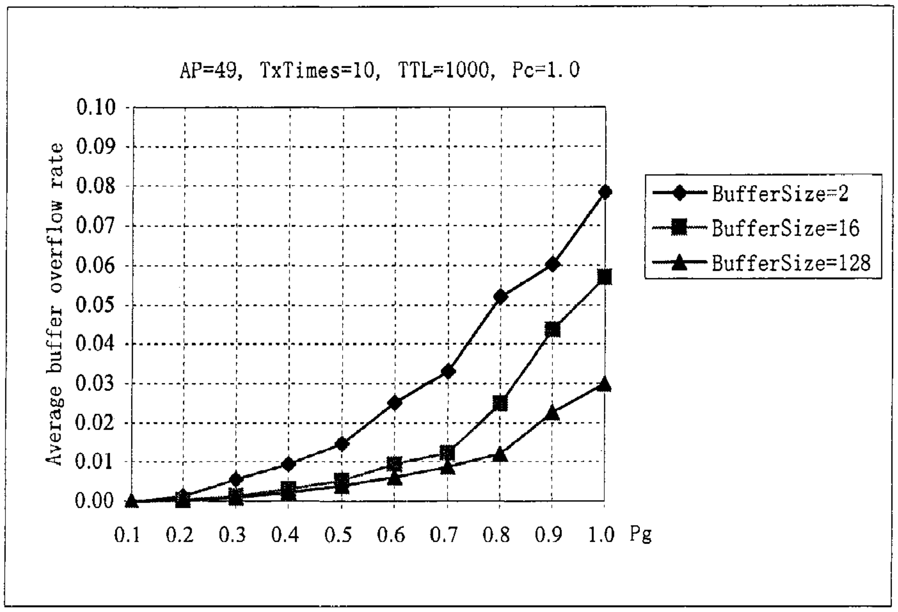


Figure 3.24 Average Buffer Overflow Rate vs. Buffer Size (DCF)

In addition, we can see from the chart, the average buffer overflow rate with smaller buffer size is always larger than that with larger buffer size. Apparently, the larger buffer size gives system performance more promotion.

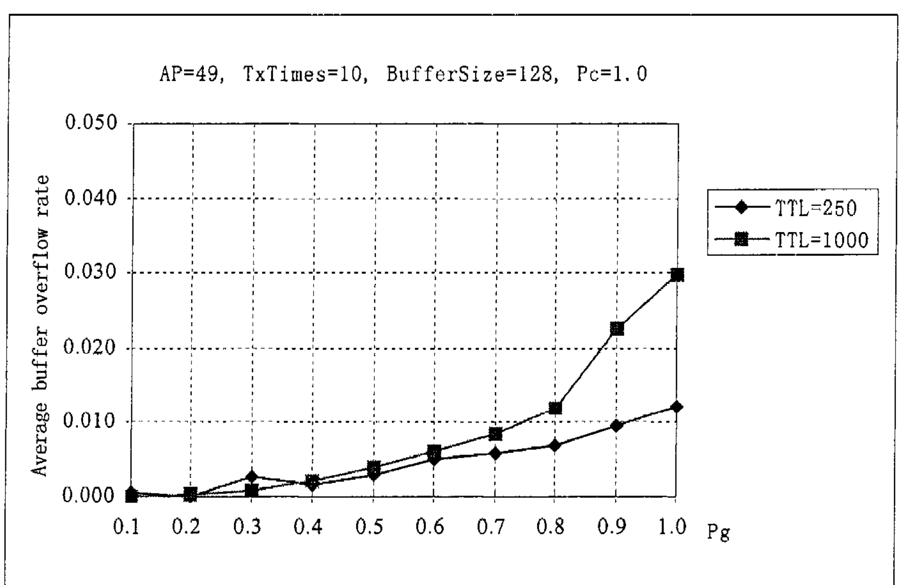


Figure 3.25 Average Buffer Overflow Rate vs. TTL (DCF)

Figure 3.25 shows the average buffer overflow rates versus packet lifetime TTL with respect to the offered traffic load. The average buffer overflow rate increases with increasing the probability of offered traffic load. The reason remains the same as explained above. Another point is the larger TTL results in higher buffer overflow rate when the offered load increases. The reason is that if a packet has longer lifetime, it has longer queuing delay to stay in the buffer and is not easier to be dropped by TTL timer. Then, more packets will stay in the buffer waiting for transmission so that the buffer is filled quickly and then overflows. With smaller TTL, packets may be dropped by TTL timer when the packets stay enough time equal to TTL in the buffer.

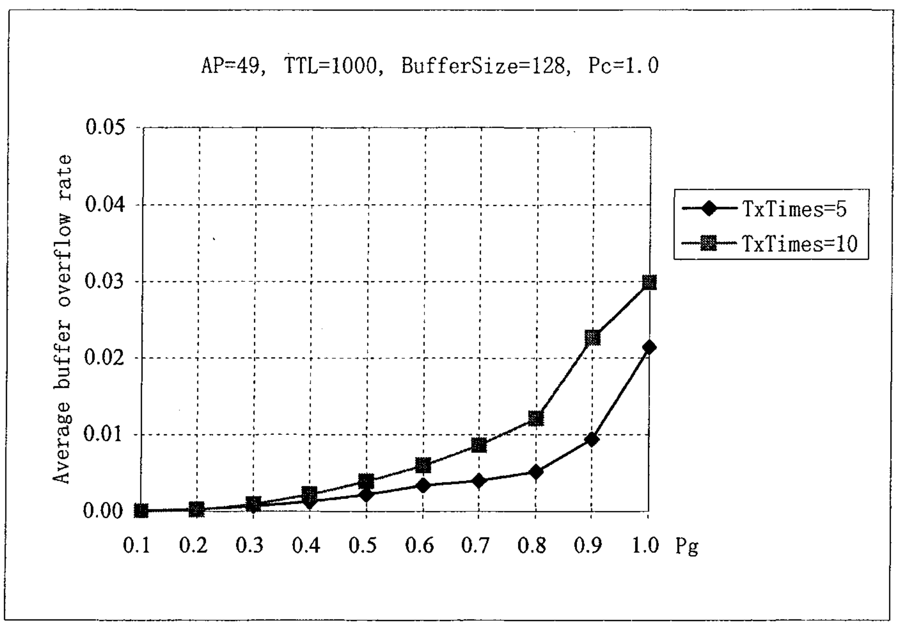


Figure 3.26 Average Buffer Overflow Rate vs. TxTimes (DCF)

Figure 3.26 shows the difference of average buffer overflow rates when the maximum attempting transmission TxTimes value is different. The larger TxTimes value results in a little bit more average buffer overflow rates when the probability of offered load increases. The reason is that if a packet has more chances to attempt transmitting, it will have longer queuing delay to stay in the buffer. When the traffic becomes heavier, more packets would enter the buffer and results in higher buffer overflow rate.

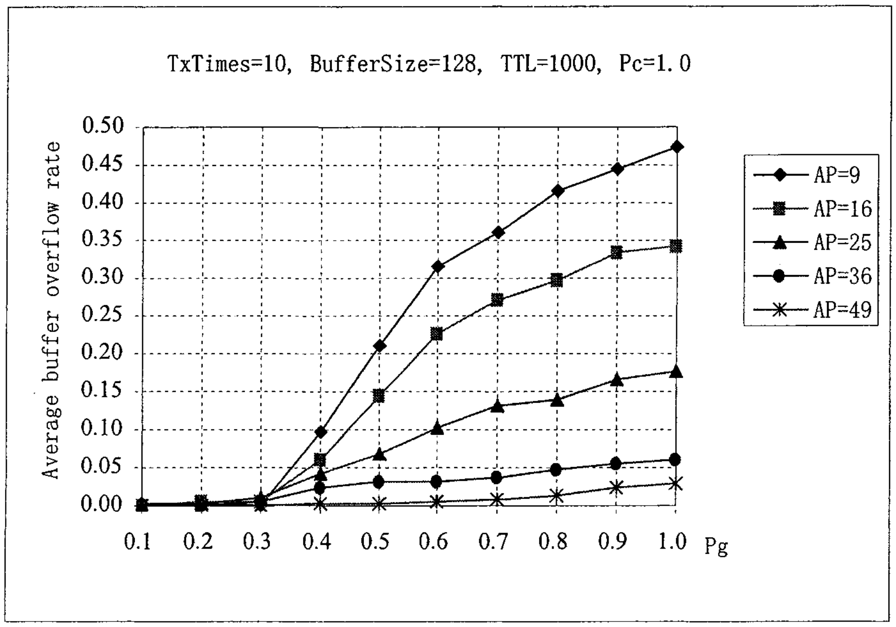


Figure 3.27 Average Buffer Overflow Rate vs. the number of APs (DCF)

Figure 3.27 shows the average buffer overflow rates versus the number of APs as a function of the offered traffic load. Despite the number of APs, the average buffer overflow rates always quickly increases with increasing the probability of offered traffic load, only the larger number of APs will obtain a little smaller average end-to-end delay. We have known that the less the number of AP employ, the less stations are covered by AP radio service and the lower average buffer overflow rates are obtained. In this scenario, the most of time is spent to transmit useless packets so that the packets have to stay in the buffer longer time to wait for service. Hence, more generated packets would be dropped due to buffer overflow; so, the average buffer overflow rates becomes higher. On the other hand, with the largest number of AP=49, 100 percent stations are covered by access points radio service. The packets that the active stations transmit to APs can be transmitted to appropriate destination except if the destination station has moved out of the cell. Therefore, the average buffer overflow rate becomes lower.

3.2.4 Average Packet Loss Rate

For the average packet loss rate, the simulation input parameters are the buffer memory size, packet lifetime period TTL, the maximum attempting transmission times TxTimes, and the number of APs. Packet loss attributed to two cases: a) a packet lifetime period is counted up to its TTL and it is still not transmitted, it will be discarded; and b) when AP cannot find the destination station in its cell, which means the destination station has moved out of the BSS, then the packet will be discarded.

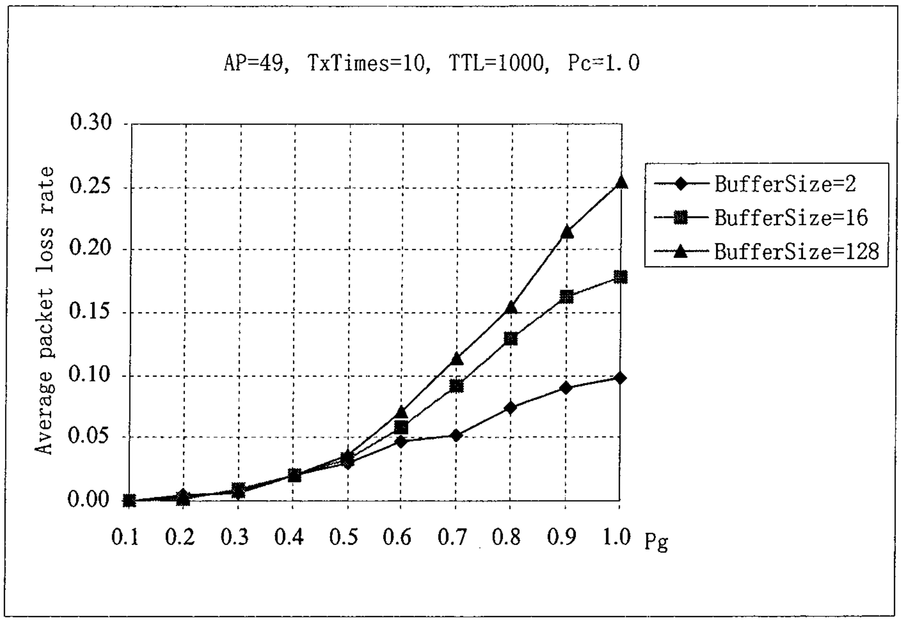


Figure 3.28 Average Packet Loss Rate vs. Buffer Size (DCF)

Figure 3.28 shows the average packet loss rate versus buffer size with respect to offered traffic load. From that, we can see that the average packet loss rate increases with increasing the probability of offered traffic load. The reason is that more packets are generated during the simulation as the probability of packet generation increases and then more generated packets are put into the FIFO buffer. Since the buffer size is fixed, the queuing delay of the packets in the buffer would increases. When the queuing delay is larger than the TTL, the packet will be dropped. Thus, packet loss rate will increase.

In addition, we can see from the chart, the average packet loss rate with smaller buffer size is always lower than that with larger buffer size. Apparently, the larger buffer size results in longer queuing delay; so the packet loss rate becomes higher.

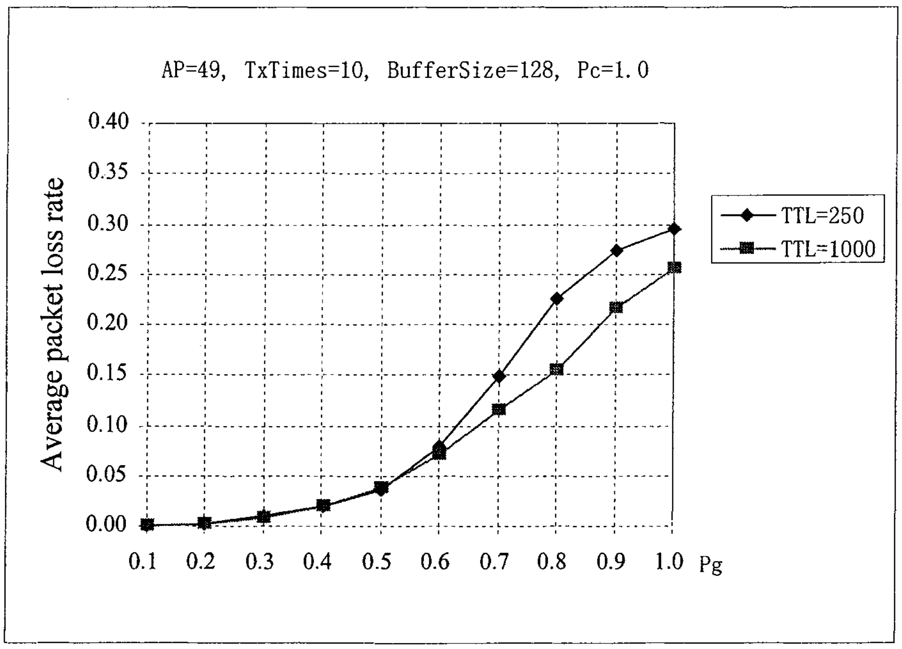


Figure 3.29 Average Packet Loss Rate vs. TTL (DCF)

Figure 3.29 shows the difference of the average packet loss rate with different TTL values. We can see that the average packet loss rate with smaller TTL becomes higher than that with larger TTL as the offered load increases. Apparently, the larger TTL results in fewer packets dropped due to the TTL timer; so average packet loss rate becomes lower.

Figure 3.30 shows the difference of the average packet loss rate with different TxTimes values. The average packet loss rate with larger TxTimes becomes higher than that with smaller TxTimes as the offered load increases. The more retransmission chances would cause longer queuing delay, and the more packets are dropped due to the TTL timer; so the packet loss rate becomes higher.

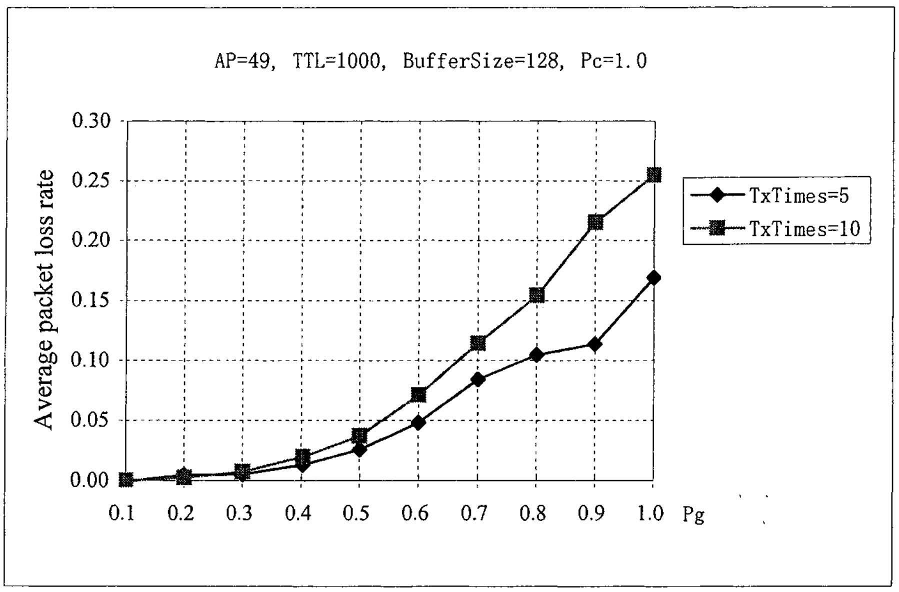


Figure 3.30 Average Packet Loss Rate vs. TxTimes (DCF)

Figure 3.31 shows the average packet loss rate versus the number of APs as a function of the offered traffic load. We can see that the more number of APs cause the lower average packet loss rate. The reason is that the more APs cover the more stations and forward more packets to the destination successfully; so the packet loss rate is lower. By contrary, the less APs cover the less stations and most of packets are lost, so the packet loss rate is higher.

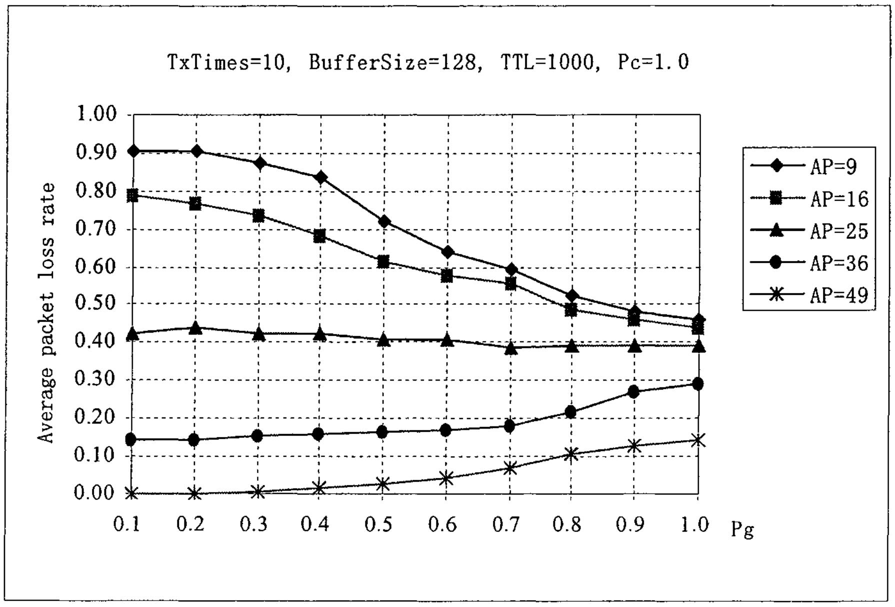


Figure 3.31 Average Packet Loss Rate vs. the number of APs (DCF)

We can also see that different varying tendency with different number of APs as the probability of offered traffic load increases. With larger number (25, 36 and 49) of APs, the average packet loss rate smoothly increases with increasing of the offered traffic load. On the other hand, with smaller number (9 and 16) of APs, the average packet loss rate smoothly decreases with increasing of the offered traffic load. The similar explanation has been figured out in section 3.1.1.4.

3.3 HIPERLAN Simulation Results

In the HIPERLAN simulation, the channel access control protocol utilizes the Elimination-Yield Non-Pre-emptive priority Multiple Access (EY-NPMA) method. The designed simulating program completely follows the protocol and configures simulation parameters according to the model that we have illustrated before. Similar to IEEE 802.11 simulation, we assume the channel quality is in ideal quality condition with channel packet correct probability P_c=1.0. The following simulation performance results are obtained under certain conditions and simulation environment specified earlier. All simulation results including average traffic throughput, average end-to-end delay, average buffer overflow rate and average packet loss rate are computed as a function of the probability of offered traffic load, which is represented by the probability of packet generation P_g and the value is set in the range 0.1 to 1.0. The buffer memory size of stations and access points is set to small size of 2, middle size of 16 and large size of 128 to evaluate the effect on performance metrics in different buffer sizes. The packet lifetime period parameter TTL is set to 250ms and 1000ms to evaluate TTL effect on performance metrics in different packet lifetime period values. The maximum attempting transmission parameter TxTimes is set to 5 and 10 to evaluate the effect on performance metrics in different maximum transmission times. The performance metrics could be affected simultaneously by several parameters. The details are shown below.

3.3.1 Average Traffic Throughput

For the average traffic throughput, the simulation results are computed as functions of the buffer memory size, packet lifetime period TTL, the maximum attempting transmission times TxTimes, and the number of APs, whereas the channel packet correct probability P_c is 1.0.

Figure 3.32 shows that the average traffic throughput decreases with increasing the probability of offered traffic load at different channel quality. The reason is that more packets are generated during the simulation as the probability of packet generation increases. Heavier traffic load certainly causes longer queuing delays in the buffer; if a packet queuing delay is larger than the packet TTL, the packet will be dropped. In addition, more offered traffic load causes more buffer overflow and results of more packets loss. Therefore, the throughput decreases with increasing the probability of offered traffic load.

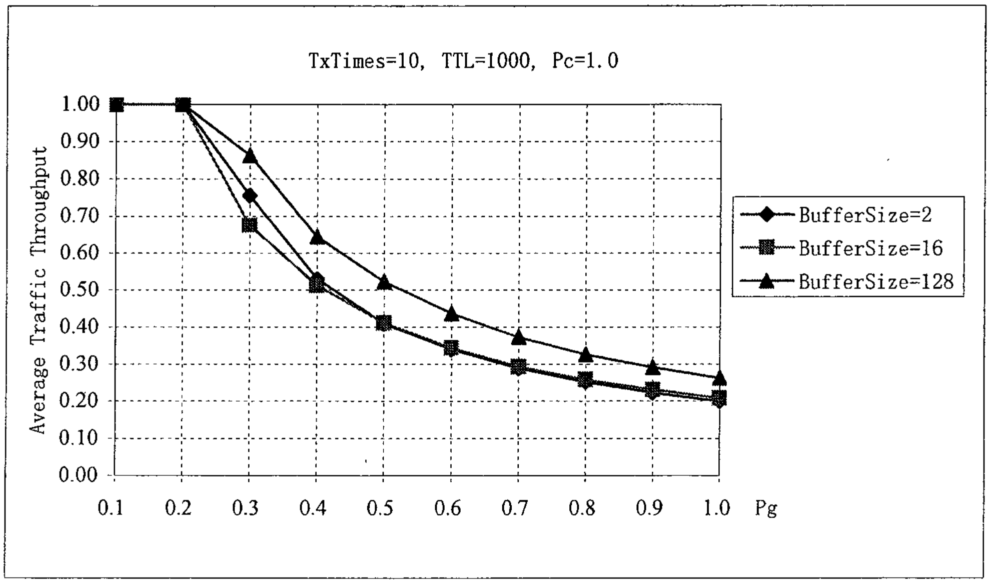


Figure 3.32 Average Traffic Throughput vs. Buffer Size (Hiperlan)

Moreover, the larger buffer size a station has, the higher average throughput can be obtained. The reason is that if the buffer size is small, only a small number of packets can be put into the buffer. More generated packets will be discarded because the buffer overflows. Thus, the larger buffer size, the less generated packets are dropped, and then the larger throughput obtained.

Figure 3.33 illustrates the difference of average traffic throughputs with different the TTL value. The figure shows that the different TTL values hardly affect the average traffic throughput.

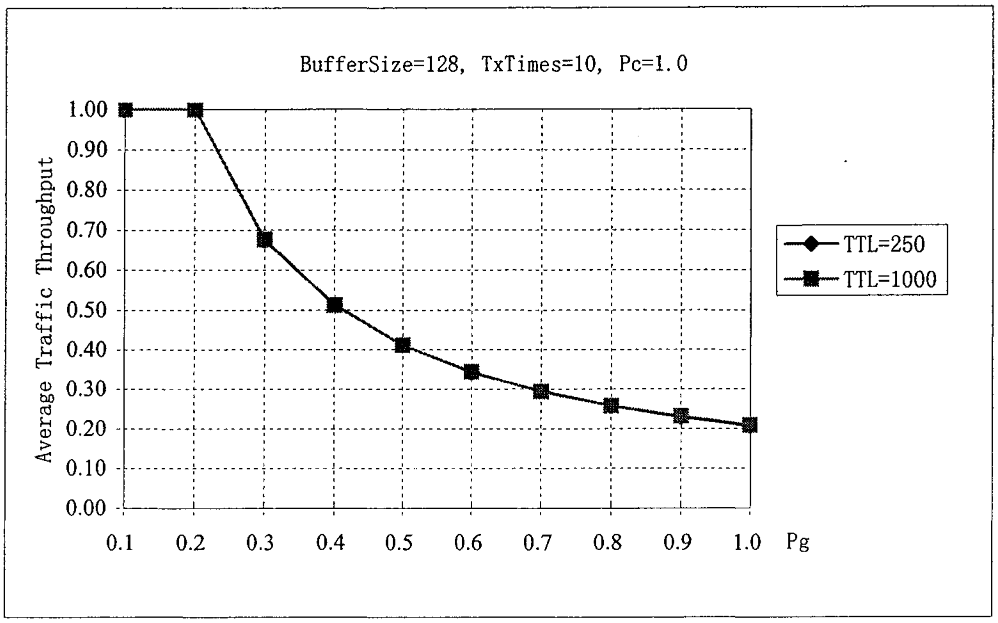


Figure 3.33 Average Traffic Throughput vs. TTL (Hiperlan)

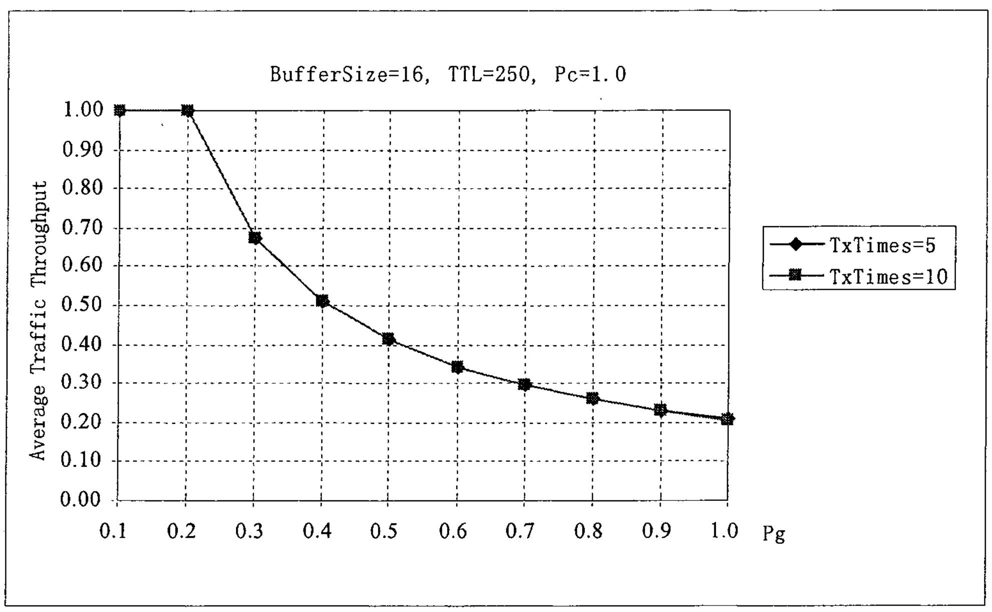


Figure 3.34 Average Traffic Throughput vs. TxTimes (Hiperlan)

Figure 3.34 illustrates the difference of average traffic throughputs with different the TxTimes value. The figure shows that the system obtains the almost identical results when it has different TxTimes values; that means the TxTimes value hardly affects the average traffic throughput.

3.3.2 Average End-to-end Delay

For the average end-to-end delay, the simulation results are computed as functions of the buffer memory size, packet lifetime period TTL, and the maximum attempting transmission times TxTimes. The value of average end-to-end delay is equal to the value of queuing delay in this simulation.

Figure 3.35 shows that the average end-to-end delay increases with increasing the probability of offered traffic load. The reason is that more packets are generated during the simulation as the probability of packet generation increases. Heavier traffic load certainly causes packets longer queuing delay in the buffer memory. Because the packet transmission delay and propagation delay do not change with the number of packet in the buffer, the end-to-end delay increases only due to queuing delay.

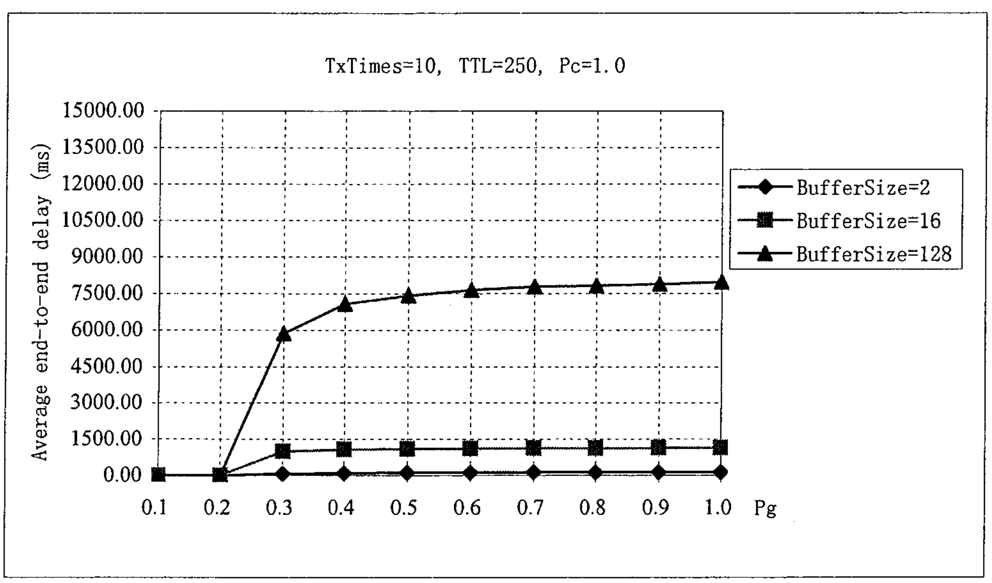


Figure 3.35 Average End-to-end Delay vs. Buffer Size (Hiperlan)

On the other hand, we can see that the average end-to-end delay increases largely with larger buffer sizes. The reason is that if the buffer size is large, more generated packets can be put into the FIFO buffer for queuing. Thus, the queued packets can be transmitted only when its predecessor packets have been transmitted. Therefore, the packet will have longer queuing delay in the larger buffer.

Figure 3.36 shows the difference of average end-to-end delay with different TTL values. The figure shows that the simulation obtains almost same results when it sets to different TTL values; that means the TTL value hardly affects the average traffic throughput.

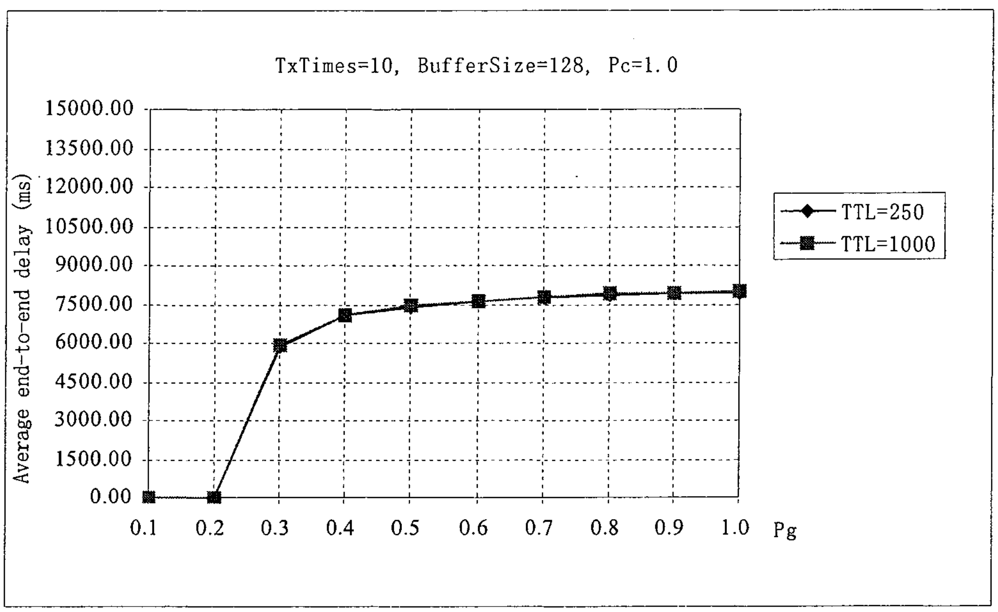


Figure 3.36 Average End-to-end Delay vs. TTL (Hiperlan)

Figure 3.37 shows the difference of average end-to-end delays when the maximum attempting transmission TxTimes value changes. The larger TxTimes value results in a little bit more average end-to-end delay when the probability of offered load increases.

The reason is that if a packet has more chances to attempt transmitting, it will have longer queuing delay to stay in the buffer. When the traffic load is light, a packet would transmit quickly without retransmissions. When the traffic becomes heavier, a packet would attempt more and more retransmissions since the heavier traffic will result in more collisions.

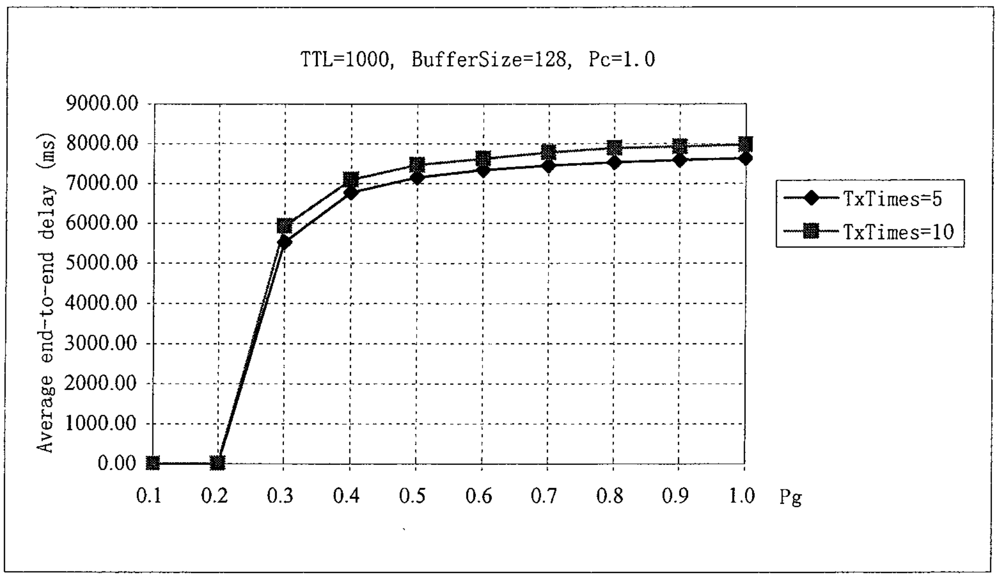


Figure 3.37 Average End-to-end Delay vs. TxTimes (Hiperlan)

3.3.3 Average Buffer Overflow Rate

For the average buffer overflow rate, the simulation results are computed as functions of the buffer memory size, packet lifetime period TTL, and the maximum attempting transmission times TxTimes. The value of buffer overflow rate mainly attributes the buffer size and offered traffic load.

Figure 3.38 shows the average buffer overflow rate versus buffer size with respect to offered traffic load. We can see that the average buffer overflow rate increases with increasing the probability of offered traffic load. The reason is that more packets are generated during the simulation as the probability of packet generation increases, so more generated packets are put into the FIFO buffer. Since the buffer size is fixed, the buffer is filled by packets quickly with increasing offered load; thus, the buffer overflow increases.

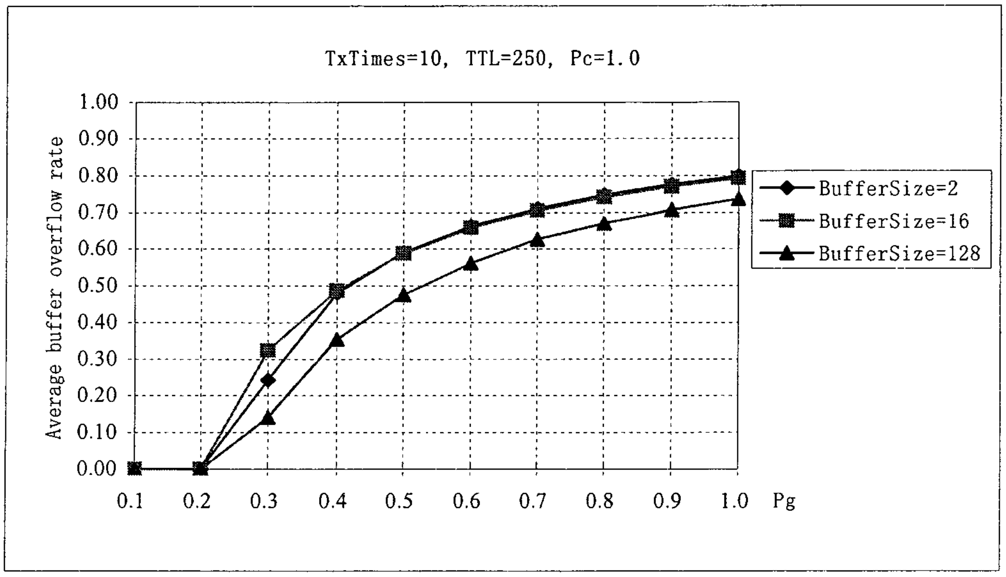


Figure 3.38 Average Buffer Overflow Rate vs. Buffer Size (Hiperlan)

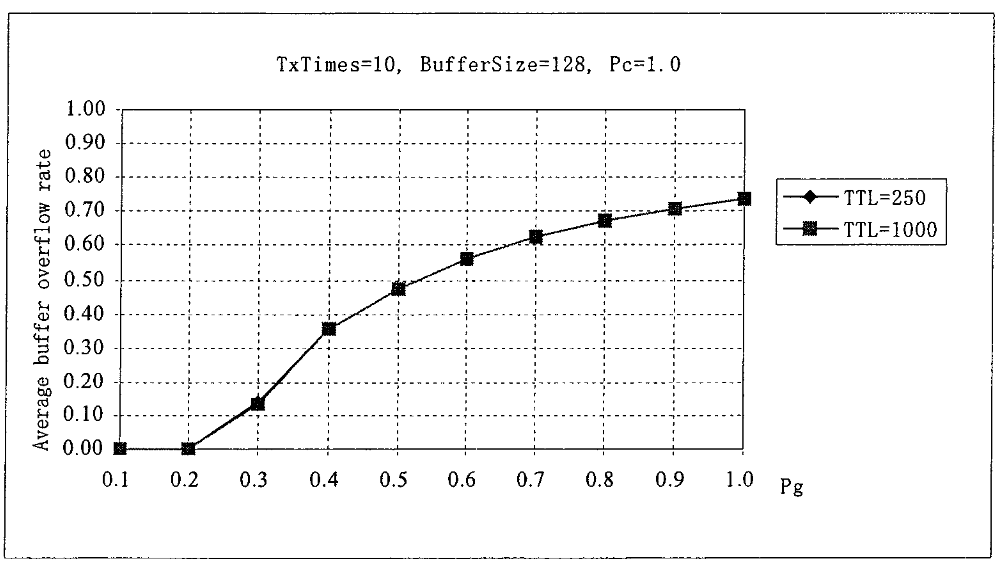


Figure 3.39 Average Buffer Overflow Rate vs. TTL (Hiperlan)

Figure 3.39 shows the difference of average buffer overflow rates with different TTL values. The figures show that the different TTL values hardly affect the average traffic throughput.

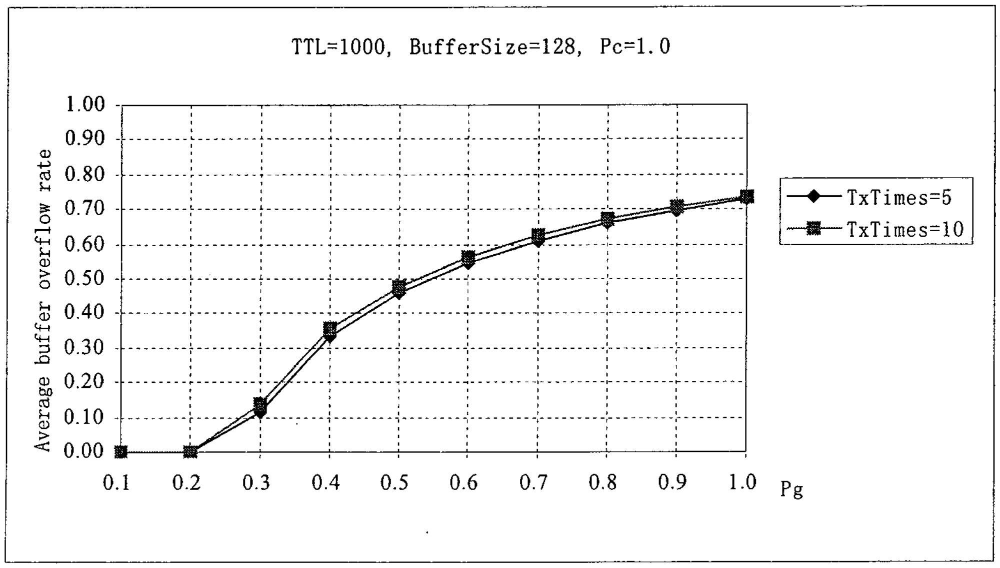


Figure 3.40 Average Buffer Overflow Rate vs. TxTimes (Hiperlan)

Figure 3.40 shows the difference of average buffer overflow rates when the maximum attempting transmission TxTimes value changes. The larger TxTimes value results in a little bit high average buffer overflow rates when the probability of offered load increases. The reason is that if a packet has more chances to attempt transmitting, it will have longer queuing delay to stay in the buffer for retransmission. When the traffic becomes heavier, more packets would be put into the buffer and result in higher buffer overflow rates.

3.3.4 Average Packet Loss Rate

For the average packet loss rate, the simulation respects to buffer memory size, packet lifetime TTL, and the maximum attempting transmission times TxTimes. Packet loss attributes to two causes:

a) a packet lifetime period is counted up to its TTL and it is still not be transmitted, it will be discarded; and

b) when a station attempts to transmit a packet, if collision occurs, the packet transmission fails; and then the packet will be retransmitted later. If the number of retransmission times of a packet increases up to TxTimes, the packet will be dropped.

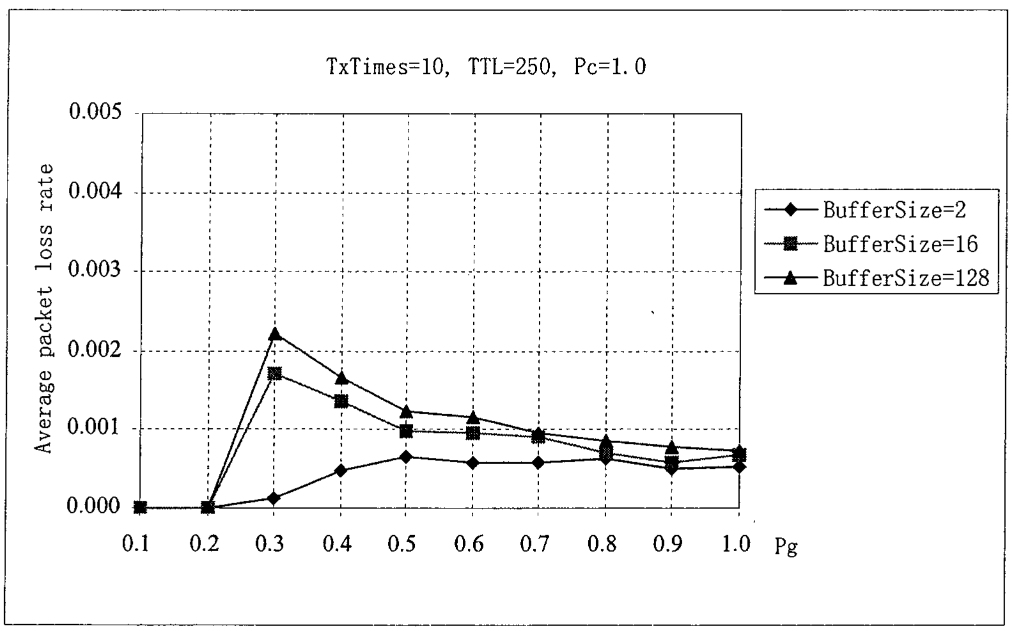


Figure 3.41 Average Packet Loss Rate vs. Buffer Size (Hiperlan)

Figure 3.41 shows the average packet loss rate versus buffer size as a function of the offered traffic load in ideal channel quality condition (P_c=1.0). When the probability of offered load P_g is less than and equal to 0.2 (very light load), no packets lose. When P_g is equal and larger than 0.3 (light, middle and heavy load), the packet loss rates gradually decreases from the highest values. When the traffic load is very light, all packets would be transmitted very quickly since there are hardly collisions and very less retransmissions is needed; hence, the packet loss rate is zero. When the traffic load increases, collision possibly occurs and more retransmission is needed, and more packets will be dropped due to TTL or TxTimes; however, the number of generated packets is much larger than the number of lost packets, the packet loss rate decreases.

We noticed that the varying of average packet loss rate with offered load is very small. For example, referring the line of Buffersize=128 in Figure 3.41, when P_g =0.2, the total traffic load is 20,000 packets, which is very light traffic load, so no packet loss. When P_g =0.3 (the load becomes to 30,000 packets), the packet loss rate is 0.22%=0.0022, which means 66 packets are dropped due to TTL or TxTimes. When P_g=0.4 (the load becomes to 40,000 packets), the packet loss rate is 0.17%=0.0017, which means 68 packets are dropped. Until P_g =1.0 (the load becomes to 100,000 packets), the packet loss rate is 0.08%=0.0008, which means 80 packets are dropped. From the above example, we can see that the number of packet loss increases with the increasing offered traffic load, only the increment is slight. Because the increment of total traffic load is much larger than the increment of the number of dropped packets, the average packet loss rate (the number of packet loss divided by total traffic load) smoothly decreases with the increasing offered load.

In addition, we can see from the chart, the average packet loss rate with smaller buffer size is always lower than that with the larger buffer size. Apparently, the larger buffer size results in longer queuing delay; more packets would be dropped by TTL and then the packet loss rate becomes higher.

Figure 3.42 shows the difference of average packet loss rates with different TTL values. The figure shows that the smaller TTL values results in a little higher packet loss rate. The varying tendency of average packet loss rate has the same explanation as for average packet loss rates versus buffer sizes.

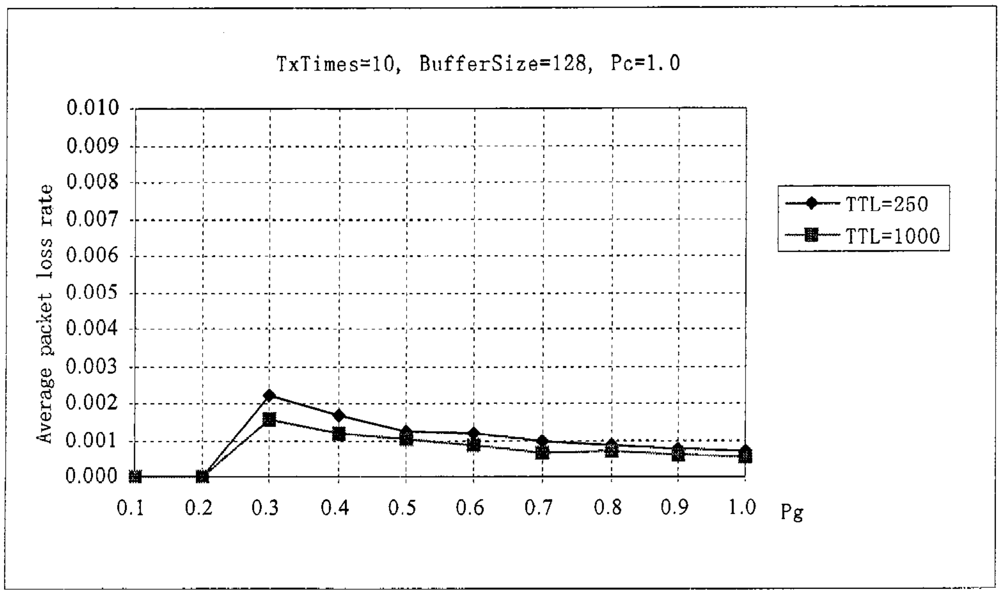


Figure 3.42 Average Packet Loss Rate vs. TTL (Hiperlan)

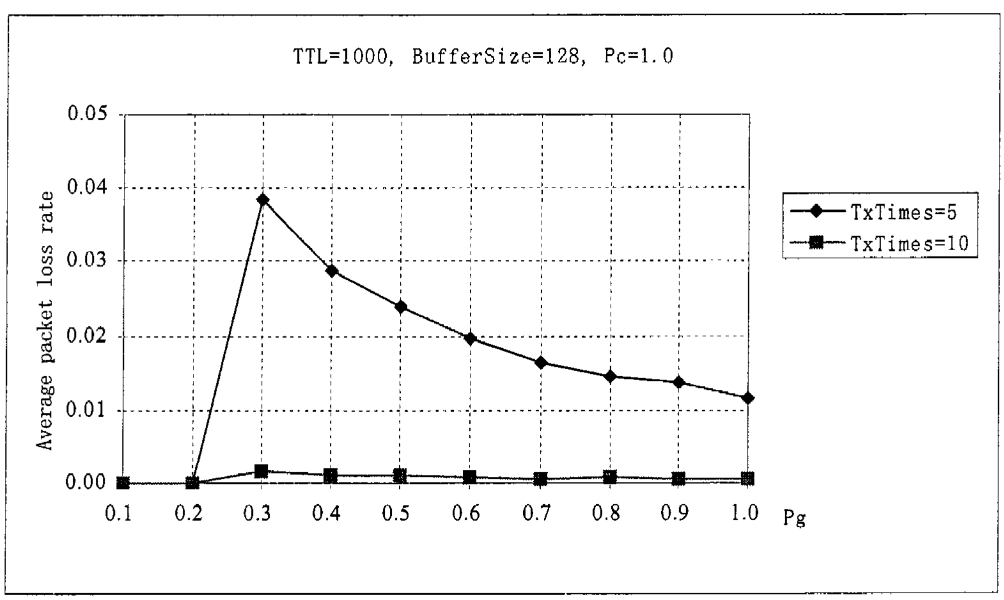


Figure 3.43 Average Packet Loss Rate vs. TxTimes (Hiperlan)

Figure 3.43 shows the difference of average packet loss rates with different TxTimes values. The figure shows that the smaller TxTimes value results in the higher packet loss rate. The varying tendency of average packet loss rate has the same explanation as for average packet loss rates versus buffer sizes.

3.4 Comparison of Simulation Results

The simulation results and their analysis have been shown above. Here, we compare the performance results of different WLAN MAC medium access protocols. We only compare the two important performance metrics: average traffic throughput and average end-to-end delay.

3.4.1 Comparison of IEEE 802.11 PCF and DCF

In the simulation, IEEE 802.11 PCF and DCF have the same network architecture model, but different medium access control protocols. We compare the average traffic throughput and average end-to-end delay results under the same input parameters.

From Figure 3.44, we can see that the PCF has higher average traffic throughput than DCF. When the offered traffic load is light, the difference is small; but, when the offered traffic load becomes heavier, their difference increases. The reason is that the PCF performs the polling medium access control method to assign the channel access for every station, and no contending collisions occur between stations. Because only polling packet and other channel control packets are overhead and their sizes are very small compared to the data packet size, the PCF obtains higher average traffic throughput. On the other hand, the DCF uses CSMA/CA medium access control protocol; and all stations contend for the channel for transmission. Inevitably, contending collision occurs in the shared channel; and the number of collision definitely increases as the offered traffic load increases.

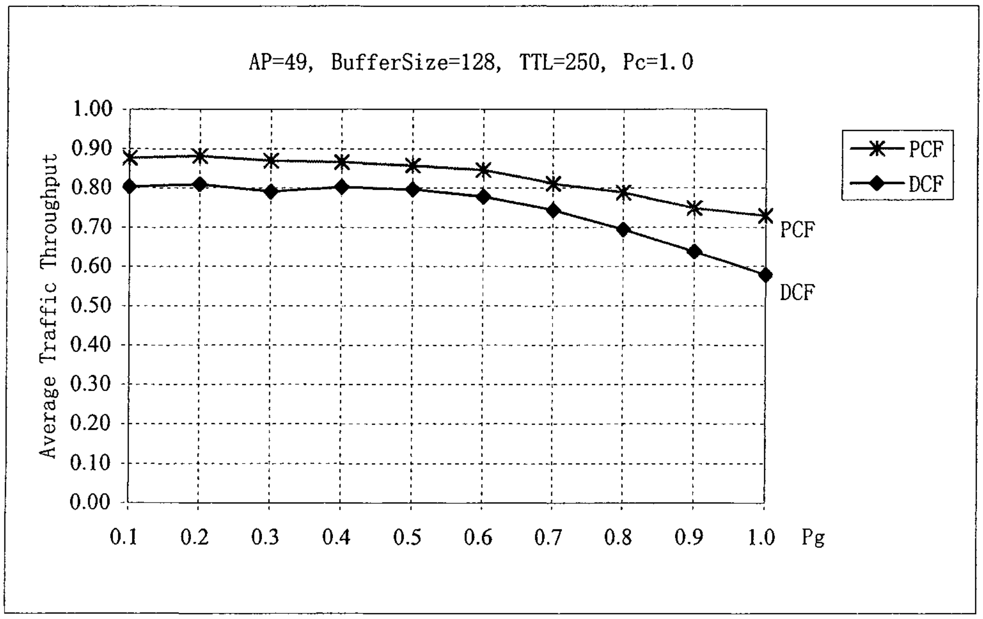


Figure 3.44 Comparison of Average Traffic Throughput (PCF/DCF)

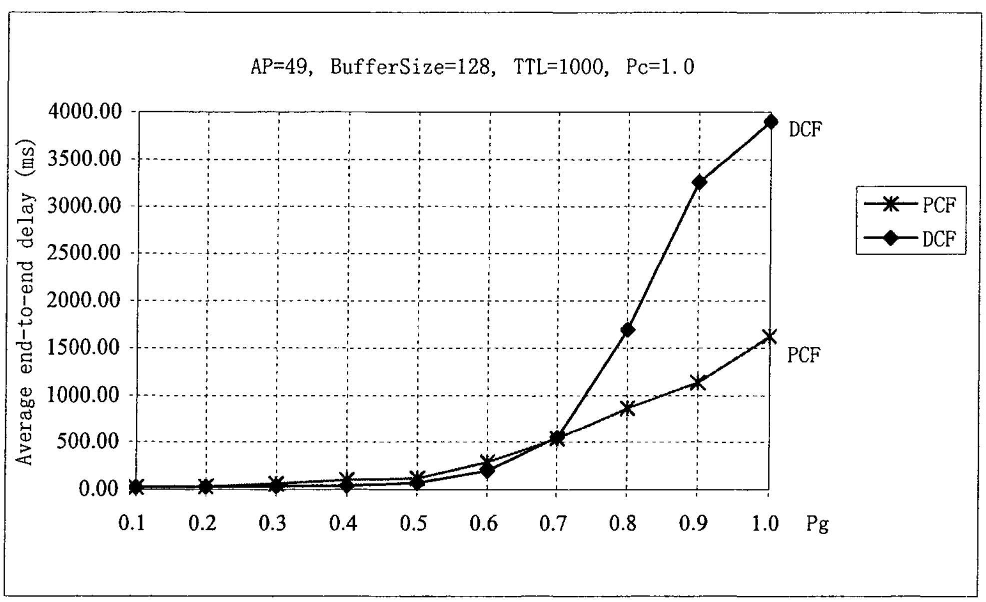


Figure 3.45 Comparison of Average End-to-end Delay (PCF/DCF)

From Figure 3.45, we can see the difference of average end-to-end delay between PCF and DCF. When the offered traffic load is light or middle, the two average end-to-end delays are almost identical, even the DCF has a little bit lower delay than that PCF does; but, when the offered traffic load becomes heavier, their difference increases markedly because the average end-to-end delay of DCF increases very quickly. The reason is that the PCF performs the polling method to assign the channel access for every station, and no contending collisions occur between stations. Heavier traffic load only increases the transmission time. On the other hand, when the offered traffic load is light or middle, a station using DCF protocol is possibly faster to obtain the control of medium to transmit packets since the collisions is less. When the traffic load is heavier, much more collisions would occur and then the packets queuing delay evidently increases.

Based on the above analysis, we can say the PCF medium access control protocol is appropriate for time-bounded data services, such as packetized voice or video. The DCF is appropriate for random data burst services and it is applied widely since it is a simple protocol and is similar to popular Ethernet, which uses CSMA/CD MAC protocol.

3.4.2 Comparison of IEEE 802.11 and HIPERLAN

In IEEE 802.11 DCF, the station contends for medium using CSM/CA protocol, whereas the station in HIPERLAN contends for medium using EY-NPMA protocol. In the simulation, the channel bit rate of IEEE 802.11 is 1Mbit/s, while the bit rate of HIPERLAN is 23.5294Mbit/s. It is difficult to compare all performances completely.

From the Figure 3.46, we can see that the HIPERLAN has higher average traffic throughput than IEEE 802.11 DCF when the offered traffic load is light. When the offered traffic load increases, the average traffic throughput of HIPERLAN decreases more quickly than that of DCF. The IEEE 802.11 DCF employs CSMA/CA as MAC protocol and performs Exponential Backoff algorithm while collision occurs. On the other hand, the HIPERLAN utilizes EY-NPMA as channel access control protocol and performs prioritization, random elimination and random yielding algorithms to avoid collisions. When the offered traffic load increases, the efficiency of the complicated EY-NPMA algorithm decreases. Thus, its average traffic throughput decreases more quickly.

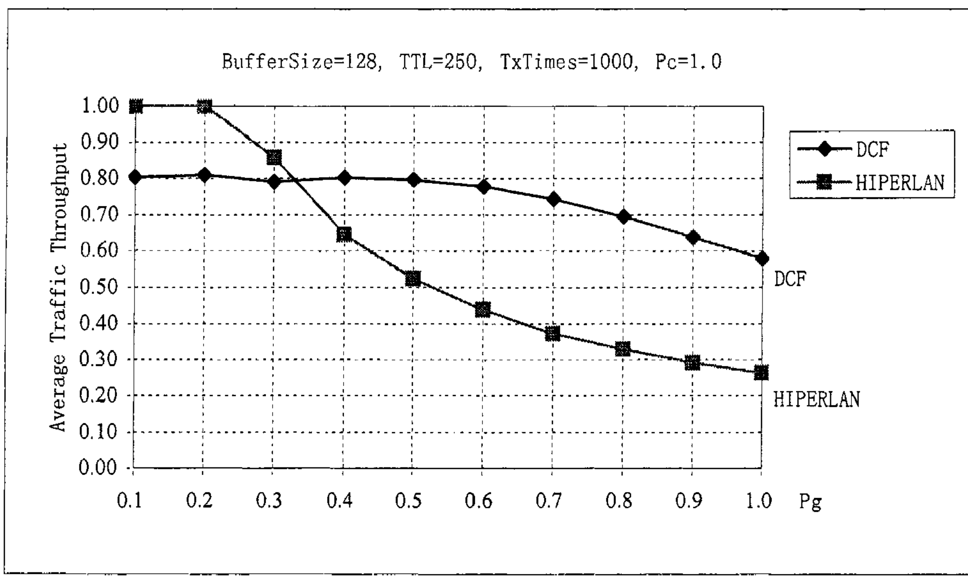


Figure 3.46 Comparison of Average Traffic Throughput (DCF/Hiperlan)

However, the HIPERLAN standard supports a QoS for packet delivery, which can provide higher performance for real-time data services. The QoS value can be assigned by the application using HIPERLAN services and priority of a packet. In the prioritization phase of EY-NPMA, channel access priority can fall into one of five levels. A packet with the highest priority level can capture the channel firstly. Nevertheless, IEEE 802.11 standard has no QoS mechanism like this.

Chapter 4

Conclusions and Future Consideration

4.1 Conclusion

Wireless LAN medium access control protocols have been studied extensively since the 1970s. Today, WLAN technology and application have matured and become very popular not only for data transmission services but also for voice and video multimedia stream services. Since the wireless medium is a kind of scarce resource, the medium access control protocols play a crucial role in efficient and fair sharing of this scarce wireless bandwidth. In this thesis, we have introduced the two main WLAN standards currently in development and application – IEEE 802.11 and ETSI HIPERLAN, and presented the simulations results in various network conditions. We have been concentrating on the performance of the different medium access control protocols, simulating general application scenarios and looking at special issues relevant for WLAN. We have evaluated the performance for different numbers of wireless terminal stations, access points, station buffer memory size, the packet lifetime period, and the maximum transmission times under different offered traffic load conditions.

According to the simulation results, both protocols perform satisfactorily in general configurations. The IEEE 802.11 standard provides two kinds of medium access method: PCF and DCF. PCF performs the polling medium access control method to assign the channel access for every station, and has no contention and collision between stations; hence, PCF can obtain higher average traffic throughput than that of DCF. The average end-to-end delay of DCF is little smaller than that of PCF in light traffic; but it increases rapidly as the traffic load increase such that it becomes much larger than PCF. The average memory buffer overflow rate of DCF is little bit smaller than that of PCF. The average packet loss rate of DCF is also little bit smaller than that of PCF.

However, PCF needs more additional hardware and software overhead to manage operation tables and other functionality in practical implementation. DCF performs CSMA/CA medium access control protocol and all stations contend the channel for transmission. Inevitably, contending collision occurs in the shared channel; and the number of collision definitely increases as the offered traffic load increases. The PCF's polling mechanism can provide time-bounded data services, whereas DCF provides better performance for burst data services.

HIPERLAN standard can obtain higher average traffic throughput than that of IEEE 802.11 as the traffic load is light; but it degrades so rapidly compared to the IEEE 802.11 DCF as the traffic load increase that it becomes much lower than the later. In addition, HPERLAN EY-NPMA mechanism is more complicated then IEEE 802.11 CSMA/CA mechanism. Consequently, IEEE 802.11 DCF has higher efficiency then HIPERLAN. However, HIPERLAN standard can provide QoS data delivery using the prioritization algorithm and can support time-bounded services in ad-hoc networks whereas IEEE 802.11 requires PCF mode.

4.2 Future Consideration

Since the main concentration of this thesis is on the performance of different medium access control protocols, some detailed factors were not considered. In future, it is worth to consider to the following issues:

1) Research the performance of WLANs using varying packet size in different

MAC protocols;

2) Compare the performance of time-bounded read-time data services in IEEE 802.11 PCF and HIPERLAN;

3) Study multicasting transmission performance for IEEE 802.11 and HIPERLAN.

Bibliography

[1] Alberto Leon-Garcia, Indra Widjaja, "Communication Networks-Fundamental Concepts and Key Architectures", McGraw-Hill, 2000.

[2] Ajay Chandra V.Gummalla and John O.Limb, "Wireless Medium Access Control Protocols", IEEE Communications Surveys, Second Quarter 2000.

[3] Asuncion Santamaria, Francisco J.Lopez-Hernandez, "Wireless LAN Standards and Applications", Artech House, 2001.

[4] Aura Ganz, Kitti Wongthavarawat, "IEEE 802.11 wireless LAN association procedure for multimedia applications", MILCOM 1999 - IEEE Military Communications Conference, October 1999, 1287-1291

[5] Benny Bing, "Wireless Local Area Networks", John Wiley & Sons, 2002.

[6] Bhagwat, P. and Screenan, C.J., Eds., "Future Wireless Applications", IEEE Wireless Communications, 9(1), 6-59, 2002.

[7] Borko Furht and Mohammad Ilyas, "Wireless Internet Handbook – Technologies, Standards and Applications", CRC Press, 2003.

[8] Brahim Bensaou, Yu Wang, and Chi Chung Ko, "Fair medium access in 802.11 based wireless ad-hoc networks", In IEEE/ACM The First Annual Workshop on Mobile Ad Hoc Networking & Computing (MobiHoc), Boston, USA, August 2000.

[9] Crow, Brian P., Indra Kim Widjaja, Geun Jeong, and Prescott T. Sakai., "IEEE 802.11 Wireless local Area Networks" IEEE Communications Magazine, September 1997, vol. 35, No.9: 116-126.

[10] Doufexi A., Armour S., Butler M., Nix A., Bull D., McGeehan J., Karlsson P., "A Comparison of the HIPERLAN/2 and IEEE 802.11a Wireless LAN Standards", IEEE Communications Magazine, Volume 40 Issue 5, 172-180, May 2002.

[11] Ellen Kayata Wesel, "Wireless Multimedia Communications: Networking Video, Voice, and Data", Addison-Wesley, 1998.

[12] EN 300 652 V1.2.1 (1998-07), Broadband Radio Access Networks (BRAN); High Performance Radio Local Area Network (HIPERLAN) Type 1; Function specification, ETSI, Valbonne, France, 1998.

[13] ETSI TR 101 683 V1.1.1, Technical Report, Broadband Radio Access Networks (BRAN); HIPERLAN Type 2; System Overview, 19 pages, 2002.

[14] ETSI TS 101 493-1 V1.1.1 (2000-04), Broadband Radio Access Networks (BRAN); HIPERLAN Type 2; Packet based Convergence Layer; Part 1: Common Part, ETSI, Valbonne, France, 2000.

[15] ETSI TS 101 761-1 V1.3.1, Technical Specification, Broadband Radio Access Networks (BRAN); HIPERLAN Type 2; Data Link Control (DLC) Layer; Part 1: Basic Data Transport Functions, December 2001.

[16] F. Cali, M. Conti and E. Gregori, "IEEE 802.11 wireless LAN: Capacity analysis and protocol enhancement", in Proceedings of IEEE INFOCOM'98, vol. 1, Mar. 1998, 142-149.

[17] Hirishi Harada, Ramjee Prasad, "Simulation and Software Radio for Mobile Communication", Artech House, 2002

[18] Hiperlan/2 Global Forum, http://www.hiperlan2.com

[19] http://www.hiperlan.com

[20] http://www.ieee802.org/11/

[21] IEEE 802.11 Standard, New Jersey: IEEE Press, 1999

[22] IEEE 802.11 WG, Draft Supplement to LAN/MAN Specific Requirements - Part 11: Wireless Medium Access Control (MAC) and physical layer (PHY) specifications: Medium Access Control (MAC) Enhancements for Quality of Service (QoS), IEEE 802.11e/D2.0, Nov. 2001.

[23] Jost Weinmiller, Morten Schl¨ager, 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 (1997) 55–67.

[24] J. L. Sobrinho and A. S. Krishnakumar, "Quality-of-Service in ad hoc Carrier Sense Multiple Access Wireless Networks", IEEE JSAC, Vol.17, No.8, Aug. 1999, 1353–1414.

[25] K. FU, Y.J. GUO and S.K. BARTON, "Performance of the EY-NPMA Protocol", Wireless Personal Communications 4: 41–50, 1996.

[26] LAN MAN Standards Committee of the IEEE Computer Society, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications", June 1997. IEEE 802.11.

[27] Martin Johnsson, "HiperLAN/2 – The Broadband Radio Transmission Technology Operating in the 5 GHz Frequency Band", Ver.1.0, HiperLAN/2 Global Forum, 1999.

[28] M. Joa-Ng and I-Tai Lu, "Spread spectrum medium access protocol with collision avoidance in mobile ad-hoc wireless network," In Proceedings of IEEE INFOCOM '99, vol. 2, 776 – 783, March 1999.

[29] Neeli Prasad, Anand Prasad, "WLAN Systems and Wireless IP for Next Generation Communications", Artech House, 2002.

[30] Andren, C., IEEE 802.11 Wireless LAN: can we use it for multimedia? IEEE MultiMedia Magazine, 5(2), April-June, 1998, 84-89.

[31] Richard Van Nee, Geert Awater, Masahiro Morikura, Hitoshi Takanashi, Mark Webster and Karen W.Halford, "New High-Rate Wireless LAN Standards", IEEE Communications Magazine, December 1999.

[32] R. Van Nee, "A new OFDM standard for high rate wireless LAN in the 5 GHz band", Vehicular Technology Conference 1999, VTC 1999-Fall, IEEE VTS 50th, Vol.1, 258-262, 1999.

[33] Simon, Marvin Kenneth, et al. "Spread Spectrum Communications Handbook", New York: McGraw-Hill, 1994.

[34] Simon Haykin and Michael Moher, "Modern Wireless Communications", Upper Saddle River, New Jersey, Pearson Prentice Hall, 2005

[35] The Institute of Electrical and Electronics Engineers, Inc. "IEEE Std 802.11 – Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications", IEEE Press, 1999 edition.

[36] T. Nandagopal, T. Kim, X. Gao and V. Bharghavan, "Achieving MAC layer fairness in wireless packet networks", In Proceedings of IEEE ACM Mobicom2000, August 2000, 87-98.

[37] Wilkinson, T.,T.G.C.Phipps . "IEEE Standard 802.11 –Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications", IEEE Press, 1999.