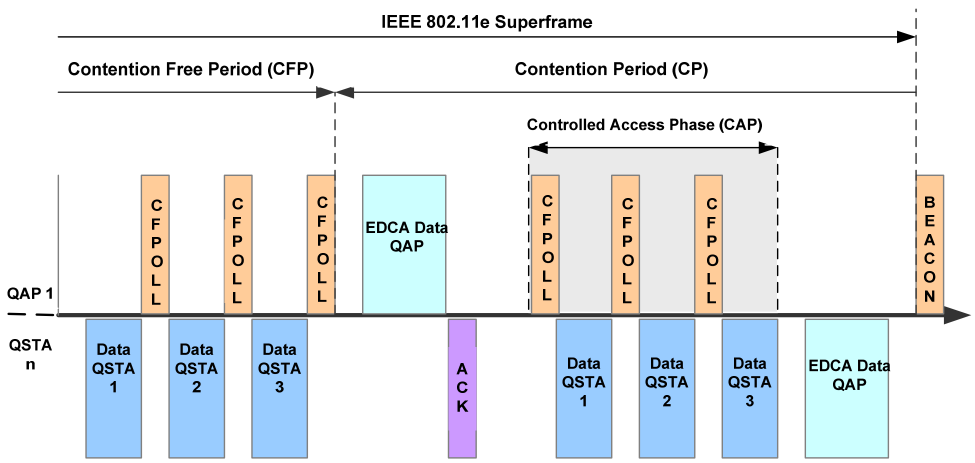
Department of Electrical and Information Engineering

Institute Industrial IT, Lemgo/Germany

Performance Evaluation of IEEE802.11e for Industrial Wireless Networks

Mohsin Hameed



August 2007

Fachhochschule Lippe und Höxter
University of Applied Sciences
Department of Electrical and Information Engineering

Master Thesis

written by

Mohsin Hameed
Registration-Nr.: 15181037

according to the MPO-IT of April 20, 2004
for the Master of Science in Information Technology at the
Fachhochschule Lippe und Höxter in Lemgo

Title: Performance Evaluation of IEEE802.11e for Industrial Wireless Networks

Supervisor: Prof. Dr.-Ing. Jürgen Jasperneite

Affirmation

I state, that the subsequent master thesis was written on my own and without any assistance from third parties. I used only the indicated resources and sources.

Lemgo, date ______________________________

(Mohsin Hameed)

Abstract

The advantages of using IEEE 802.11-based Wireless Local Area Networks (WLAN) in industrial automation applications are substantial and include: mobility, ease and speed of installation, flexibility and costs. But wireless applications for industrial automation applications have rigorous requirements on quality of service (QoS) for the transmission of real-time critical process data. IEEE 802.11-based WLANs, which were initially designed only for best effort traffic, did not provide any QoS support for this kind of traffic. Therefore the IEEE 802.11e standard amendment was introduced and ratified in 2005. It defines the concept of a Hybrid Co-ordination Function (HCF) at the MAC layer for medium access control. HCF is a combination of HCF Controlled Channel Access (HCCA) with parameterized quality of service (QoS) and Enhanced Channel Access (EDCA) with prioritized QoS.

The contemporary work deals with the performance evaluation of HCCA for industrial wireless network. A HCCA simulation model has been implemented using OPNET modeler. The simulation results are compared with EDCA in terms of delays for various scenarios.

Acknowledgments

First of all, I am thankful to almighty for all His help during my entire Msc, with that I am very grateful to Prof. Dr.-Ing. Jürgen Jasperneite for giving me an opportunity to do my thesis in inIT and for sharing his ideas and providing a brilliant guidance throughout the whole Master Thesis project.

I also would like to thank Dipl.-Ing. Henning Trsek (Master) for cooperating me during my initial research period and in many other intermediate tasks that I wouldn't have finished without his support.

I would like to thank all my family; in particular my parents and fiancé, for their prayers and frequent communication that helped me survive abroad.

Table of Contents

List of Figures

List of Tables

1 Introduction

As a result of wireless technology, the traditional ways of wired networks have become inadequate in meeting the challenges of present arena posed by our collective lifestyles. Wireless technology provides us with cheap and flexible wireless access. It is also easy to install on campuses, airports, in stock markets, offices, hospitals, and other places. Mobility, ease and speed of installation, flexibility and cost are the core characteristics that place wireless solutions on great demand in today's commercial market. Nowadays wireless technology has become a part of our life, all present applications are being upgraded to support wireless technology and all upcoming applications are being manufactured with integrated wireless support. Wireless technology has become a basic requirement. All the wireless applications need over-the-air accessibility of up to 100m of area that can be provided by IEEE 802.11 [1] which supports over-the-air interface between the wireless client and a base station or between 2 wireless clients. The advantages of using IEEE 802.11-based Wireless Local Area Networks (WLAN) in industrial automation applications are also substantial. The annual industry revenues have already exceeded US$1 billion, and are expected to pass US $4 billion by 2007 [9].But In contrast to mainstream office products, wireless applications for the industrial automation applications require high quality of service (QoS) for the transmission of real-time data.

The IEEE 802.11 is a standard that defines the specifications of both Physical (PHY) and Media Access Control (MAC) layers of WLAN. According to that standard mandatory distributed coordination function (DCF) and an optional point coordination function (PCF) are the two medium access coordination functions at MAC layer. The IEEE 802.11 based WLAN is initially designed only for the best effort data traffic i.e. it does not provide any support for real-time traffic. While if we want to use wireless components in industrial automation, where reliable and time conscious communication is a key factor, the real-time behavior must be considered because whether it is Ethernet or wireless, industrial automation system has rigorous requirements on quality of service (QoS) such as jitter and delay. As described in [16] jitter value lower then 1 ms and delay lower then 10 ms is required in process control application.

The IEEE 802.11 [1] standard was introduced in 1999, After that IEEE established different task groups that are working to provide the enhanced features in WLAN, among these task groups is one task group by the name of 802.11i, focusing on enhanced security and authentication mechanisms of WLAN, similarly 802.11n, which tries to support a maximum (throughput) of at least 100 Mb/s. 802.11f, proposing an inter-AP protocol to allow stations to roam between multi-vendor access points. 802.11e is a task group that is working to provide the QoS support, in WLAN, for the transmission of real-time data. IEEE802.11e introduces the new Hybrid Coordination Function (HCF) for the medium access, which consists of the contention based Enhanced Distributed Channel Access (EDCA) for prioritized QoS along with contention free HCF Controlled Channel Access (HCCA). These new functions are built on DCF defined in the 802.11 legacy model as shown in Figure 1.1

.

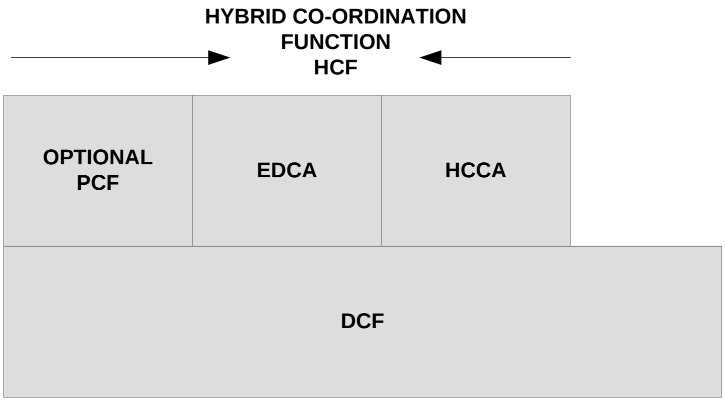


Figure 1.1: Mac Architecture in IEEE802.11e [4]

In this project we mainly focused on the HCCA with respect to the industrial requirements. The OPNET [2] simulation tool has been used to analyze the existing 802.11 PCF model as well as to develop the required 802.11e HCCA model.

In the first chapter, a brief introduction to WLAN and HCCA model has been given, followed by a brief motivation, related and future work and detailed definition of the problems to be addressed throughout the project. Chapter 2 tries to summarize the most important aspects of the state of the art system such as MAC layer of 802.11 and 802.11e.

Chapter 3 explains the OPNET simulation tool used for this project. Chapter 4 addresses the HCCA OPNET Enhancement while Chapter 5 validates HCCA Enhanced model.

In Chapter 6, a statistical analysis and an evaluation of the obtained results are presented and system suitability is discussed. The last Chapter encloses everything up in order to draw sufficient conclusions and serve as a basis to inspire discussions for future research activities.

The conventions used in this document try to follow the common standard. The document is arranged in chapters, sections and subsections. All the references and citations appear in between square brackets ([ ]). A list of figures and tables is provided.

1.1 Motivation

The main motivation of this project has emerged with a deliberate consideration of the amount of work still remaining to provide the real-time data transmission support, with required QoS, for the industrial automation system. This section presents some real world industrial scenarios in which contemporaneous work in the field of wireless technology has true importance.

The advantages of using wireless technology in industrial automation applications are substantial. With the sophisticated advancements in industries, plant manager can use wireless technology to view every detail of production i.e. he can view every machine's output, maintenance schedule, stress and strain level, and based on results he can shift the load of one machine to another machine. For all this work he simply needs to make a few strokes on his keyboard. Similarly sophisticated advancements in manufacturing and assembling systems in industries, lead to minimal or no human interaction required in transportation of machine components from one robot to another to work on it. Off the shelf robots are currently available, especially in the automobile industry, to perform sophisticated operations like welding, painting, fastening etc. Wireless technology has a premium flexibility in automated plants where all the clients are controlled by a central controller like in conveying and transport systems which instead needs a cumbersome and inflexible wiring. Wired networks in industries are not only inflexible and cumbersome but are also very expensive i.e. approximately $10 per foot to install and more to modify when plant configurations changes [17] whereas wireless technology provides the benefits of mobility, ease and speed of installation, flexibility and cost.

To provide the QoS support in WLAN, the IEEE has approved a new standard, IEEE 802.11e [4], however, the standard only specifies the features required for the new service provisioning and leaves out the design of specific scheduling disciplines that utilize these features to the developers and equipment vendors [3]. Several researchers have already measured the performance of HCCA in [7, 8, 9] and discussed some solutions but all of their findings are not to provide any QoS support in industrial automation but mainly based to provide QoS support in home or office applications such as Voice over IP (VOIP) and video streaming.

1.2 Related Work

The research carried out in the field of industrial wireless networks with provisioning QoS is very confined. In [8], Grilo et al and in [9] Ramos et al measured the performance of the HCCA but their findings are based on the home or office applications such as Voice over IP (VOIP) and video streaming. Similarly [5, 6, 7] also measured the performance of HCCA and suggested some solutions but were also based on multimedia applications.

In [12], Krommenacker and Lecuire presented results on wireless automation system in which 802.11g with focus on the PCF was considered. According to [13], the HCCA mechanism was found to be inefficient for the real-time applications, such as voice over IP and video streaming, even though this kind of traffic does not allow an assessment of the same mechanism in an environment with different traffic characteristics like industrial communication systems [15]. It has been shown in [14] that the characteristics of voice and video traffic differ significantly from those of industrial real-time communication,

1.3 Problem Definition

The IEEE 802.11 [1] is a standard that defines the specifications of both Physical (PHY) and Media Access Control (MAC) layers of WLAN. According to that standard mandatory distributed coordination function (DCF) and an optional point coordination function (PCF) are the two medium access coordination functions at MAC layer. But IEEE 802.11 standard is initially designed only for the best effort data traffic so it does not provide any support for real-time traffic. Whereas if we want to use WLAN in industrial automation, the real-time behavior must be considered because industrial automation system has rigorous requirements on quality of service (QoS) such as jitters and delays.

To solve this problem, IEEE established a task group with the name 802.11e which is working to provide the QoS support, in WLAN, for the transmission of real-time data. IEEE 802.11e introduces the new Hybrid Coordination Function (HCF) for the medium access, which consists of the contention based Enhanced Distributed Channel Access (EDCA) for prioritized QoS along with contention free HCF Controlled Channel Access (HCCA) so that a single access point can facilitate both contention based traffic (EDCA based traffic) as well as contention free traffic (HCCA based traffic) at the same time as shown in Figure 1.2. But currently there is no HCCA model present. So the main objective of this project is to add HCCA features so that AP can facilitate both EDCA and HCCA with respect to the industrial requirements.

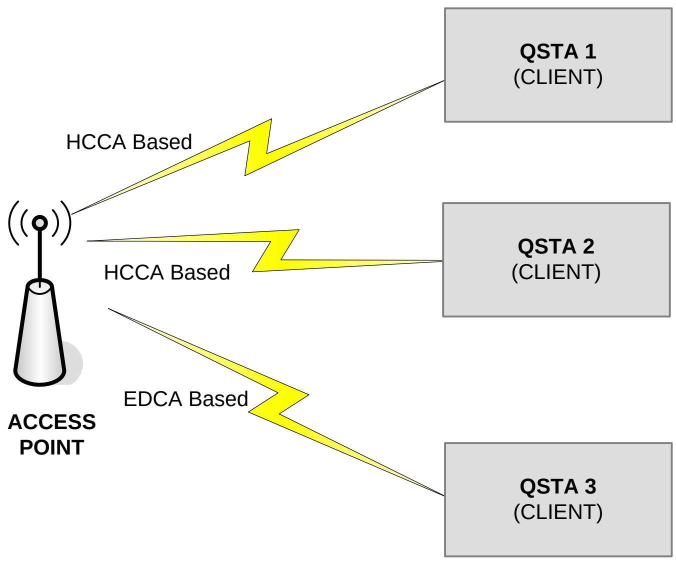


Figure 1.2: Wireless LAN in an Industrial Environment

1.4 Objectives

In order to start data transmissions in the HCCA model, QSTA sends a QoS request frame containing a TSPEC requirement to QAP. On receiving these QoS request frames; QAP allocates HCCA-TXOP to each sender QSTA with respect to its TSPEC requirements. Figure 1.3 presents the sequence diagram of HCCA TXOP allocation to each registered station.

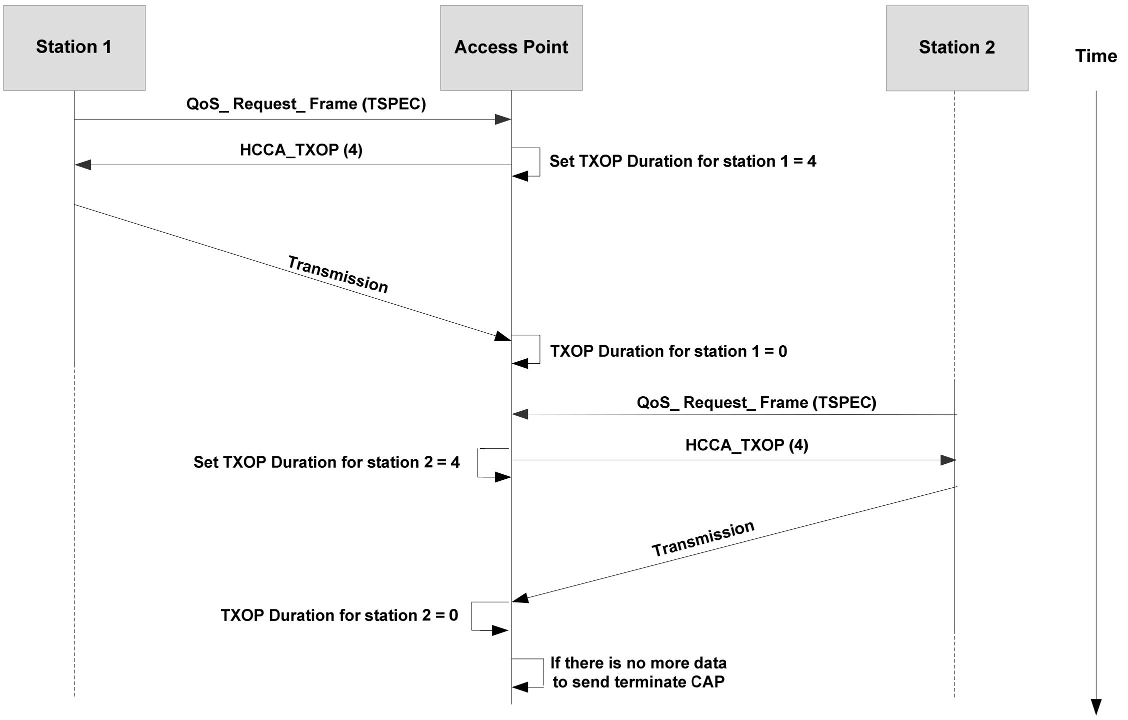


Figure 1.3: TXOP assignment to each QSTA

The main objective of this project is the extension of the standard PCF model in accordance to provide the HCCA features in OPNET by means of an HCCA admission control as shown in Figure 1.3. The extended model will comprise capabilities of HCCA admission control mechanism, scheduling mechanism, generation of controlled access phase (CAP) interval, and allocation of TXOP to each registered station and finally HCCA enabled stations can only transmit in CFP not during CP. In this project we are mainly interested in adding the mentioned features in the current PCF model using OPNET modeler and investigating their behavior in distributed real-time systems.

2 State of the Art

The development of proposed system needs a core research in the areas of real-time systems, IEEE802.11 and IEEE802.11e amendments. Starting with real-time systems, whose fundamental concepts are described in this chapter, we further describes the elementary concepts of IEEE802.11 and IEEE802.11e amendments.

2.1 Real-Time Systems

As already defined that if we want to use wireless component in industrial automation, where reliable communication is key factor, the real-time behavior must be considered because industrial automation system has rigorous requirements on quality of service (QoS) such as jitter and delay. A system is said to be **real-time** if the total correctness of an operation depends not only upon its logical correctness, but also upon the time in which it is performed [25]. Figure 2.1 shows an example of real-time scenario where start time =10ms and deadline =30ms.

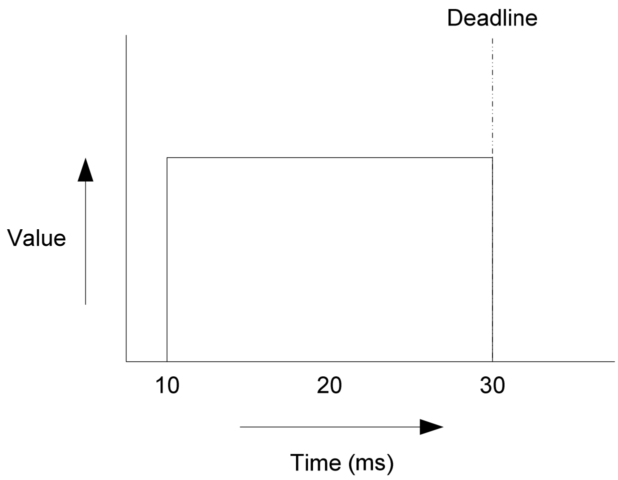


Figure 2.1: Example scenario of Real-Time System

According to time requirements, the real-time system is further divided into two categories:

- Hard Real-Time System
- Soft Real-Time System

2.1.1 Hard Real-Time System

The hard or immediate real-time system is such a system in which completion of an operation after deadline is considered useless because ultimately this may lead to a complete failure of the system. Example of hard real-time system include industrial process controller where a delayed operation may cause complete system failure. Figure 2.2 shows an example of hard real-time system where any operation after the deadline is considered useless or damage.

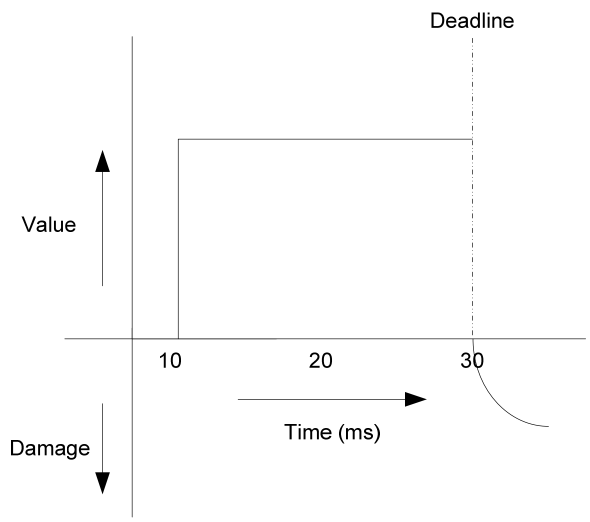


Figure 2.2: Example scenario of Hard Real-Time System

2.1.2 Soft Real-Time System

A soft real-time system is such a system in which completion of an operation will tolerate some delays, and may respond with decreased service quality for example dropping frames while playing audio. Figure 2.3 shows an example of soft real-time system where any operation after the deadline may decrease service quality (within some tolerable delay) or may considered useless (after tolerable delay).

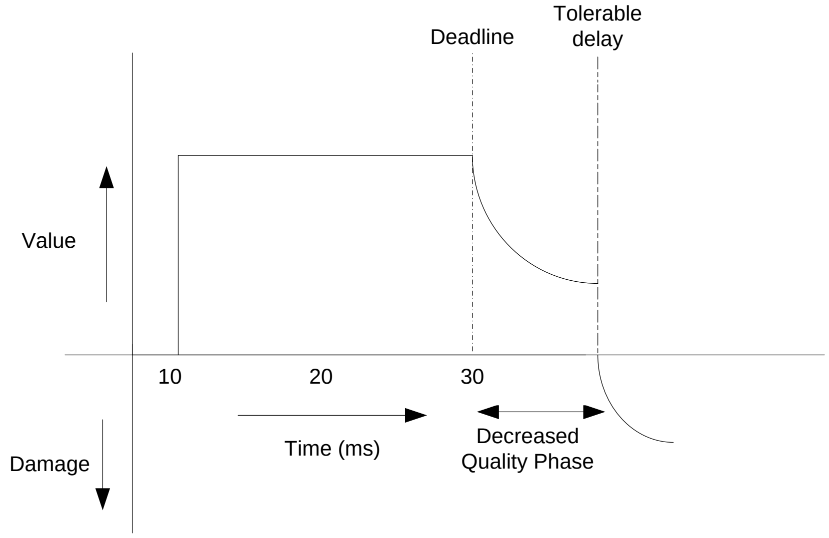


Figure 2.3: Example scenario of Soft Real-Time System

2.2 IEEE 802.11- AN OVERVIEW

IEEE 802.11 is a branch of 802 family of standards which defines the series of specification for local area network (LAN). The comprised standards fall within the scope of layer one PHY or physical layer and layer two data link layer of the Open Systems Interconnection (OSI) reference model and specify the data link layer in two sub layers, the Logical Link Control (LLC) and Medium Access Control (MAC). The MAC is a set of rules to determine how to access the medium and send data, but the details of transmission and reception are left to the PHY. They have some standards within the data link layer in common, such as the IEEE 802.2 for LLC and 802.1 standard for management, but they differ significantly when it comes down to the physical layer (PHY) and some other parts of the MAC sub layer as shown in Figure 2.4. IEEE 802.3 (Ethernet) defines the standard for a wired local area network, whereas the IEEE 802.11 and its supplements are dealing with wireless connectivity of fixed, portable and moving stations within such a network.

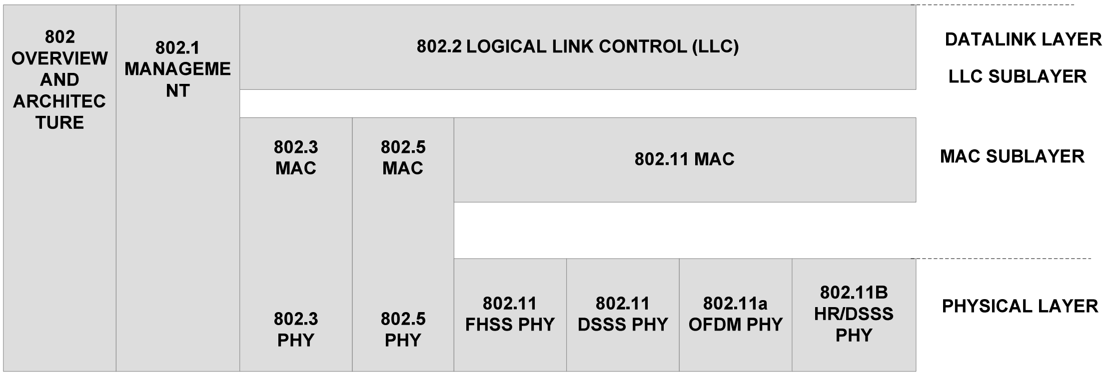


Figure 2.4: The IEEE 802 family and its relation to the OSI model [18]

Targeted environments for WLAN include offices, airports, stations and industrial environment as an indoor use while campuses and buildings complexes as an outdoor use.

2.2.1 IEEE 802.11 Standard

The IEEE 802.11 standard is basically divided into logical architecture of two main parts; the specification of the PHY layer for a wireless medium, which is necessary to define transmission functions and specification of MAC sub layer for medium access. 802.11 split the PHY into two generic components: The Physical Layer Convergence Procedure (PLCP) and a Physical Medium Dependent (PMD). The IEEE 802.11 standard defines the service access point, like PHY service access point (PHY_SAP) between MAC and PLCP, to provide the interface for the communication with other sub layer. The protocol reference model for 802.11 is shown in Figure 2.5.

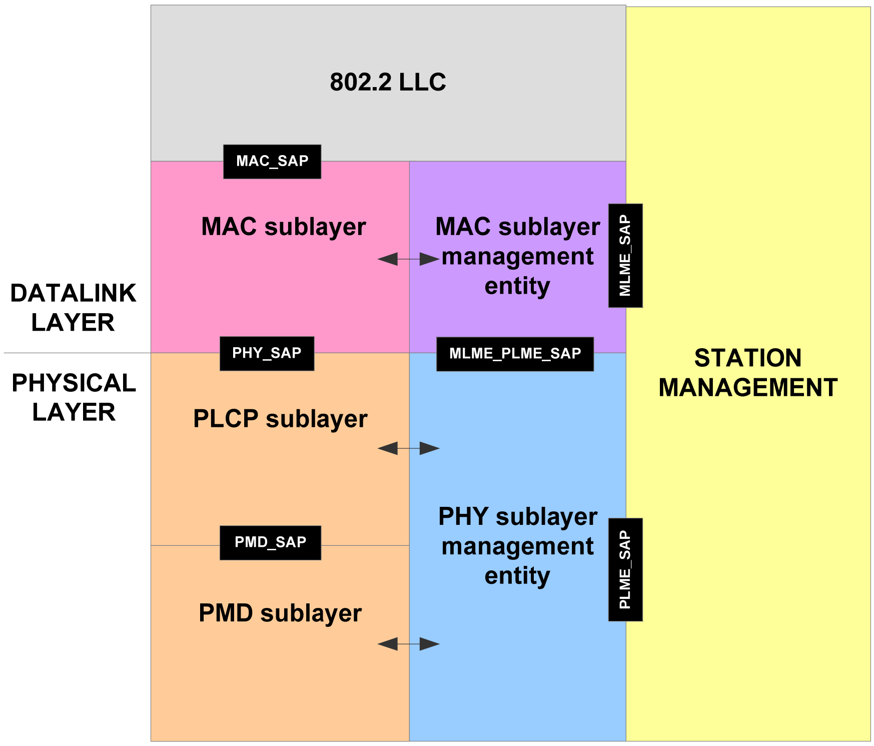


Figure 2.5: 802.11 Protocol Reference Model [27]

2.2.2 802.11 Components

802.11 networks consist of four major physical components which are summarized in Figure 2.6. The components are:

- **Distribution System**

  The distribution system is a logical component that is used to connect several access points to form a large area. The main function of the distribution system is to get frame from source access point and forward it to destination access point.

- **Access Point**

  A wireless access point (AP) is a device that connects wireless communication devices together to form a wireless network. The AP usually connects to a wired network, and can relay data between wireless devices and wired devices.

- **Wireless Medium**

  To move frames from one station to another station, the standard uses a wireless medium. For this task, IEEE defined several physical layers; the architecture allows multiple physical layers to be developed to support MAC layer. Initially, IEEE 802.11 standardized two radio frequencies (RF) and one infrared physical layer [1].

- **Stations**

  Networks are built to transfer data between stations. Stations are computing devices with wireless network interface.

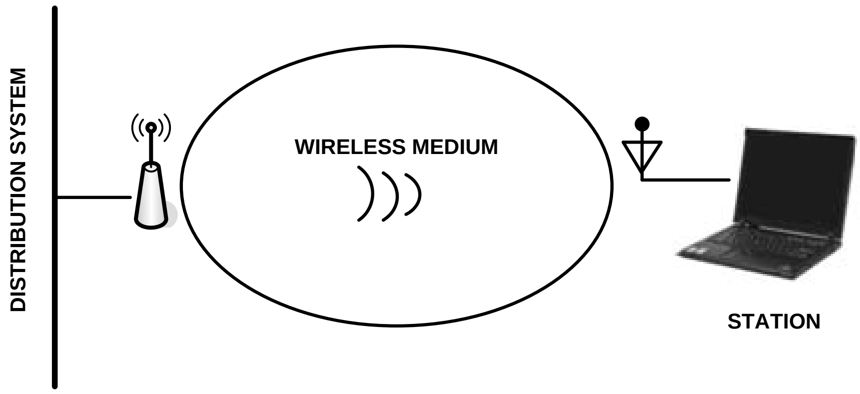


Figure 2.6: Components of 802.11 LAN's [18]

2.2.3 Types of Networks

The basic building block of an 802.11 network is the basic service set (BSS), which is a group of stations that communicate in the basic service area with each other. BSS can be two types; the Independent Network (IBSS) and Infrastructure Network or Extended Service Set.

- **Independent Network (IBSS)**

  In IBSS, stations communicate directly with each other and thus have to keep direct communication range. The smallest possible 802.11 network is an IBSS with two stations. Figure 2.7 shows an example IBSS network.

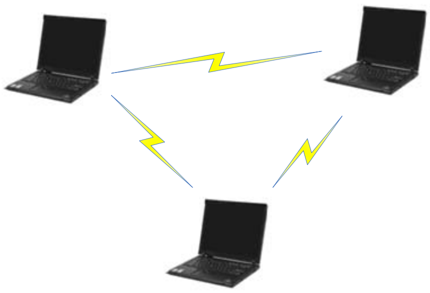


Figure 2.7: Independent Network [18]

- **Infrastructure Network**

  All the communication in the infrastructure network is done using access point, including communication between mobile nodes in the same service area, as shown in Figure 2.8. If one mobile station in an infrastructure BSS needs to communicate with another mobile station, who is a part of the same network, the communication will take two hops i.e. first from the source station to the access point and then from the access point to the destination station.

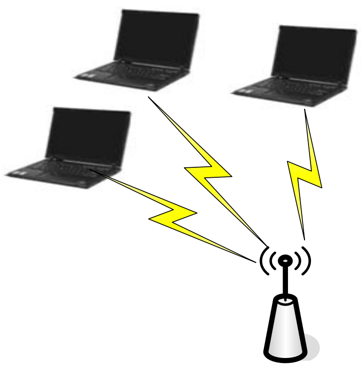


Figure 2.8: Infrastructure Network [18]

- **Extended Service Set**

  Extended Service Set is a combination of infrastructure network to provide the large coverage area. Stations within same ESS and different BSS may communicate with each other using the distribution system, even though these stations may be in different basic service areas or may be moving between basic service areas.

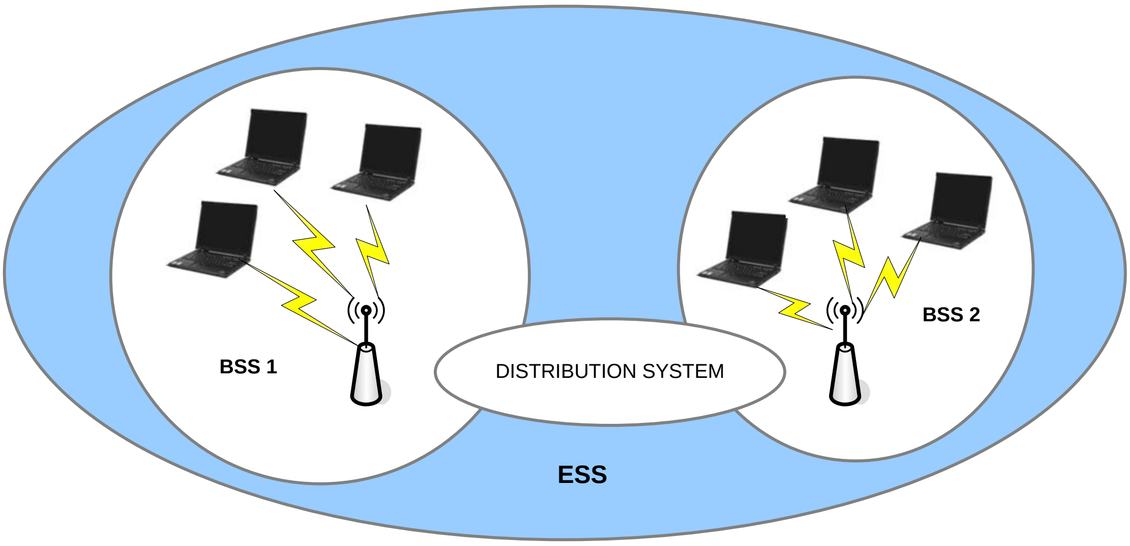


Figure 2.9: Enhanced Service Area [27]

2.2.4 Physical Layer

In 1997 the IEEE released the first version of the 802.11 WLAN standards with three kinds of PHY layer options: an infrared (IR) base band PHY, a frequency hopping spread spectrum (FHSS) radio, and a direct sequence spread spectrum (DSSS) radio. All three options only support 1–2 Mb/s data rates. In 1999 two high-rate PHY extensions were released by the IEEE:

- 802.11b, based on the DSSS technology, with data rates up to 11 Mb/s in the 2.4 GHz band
- 802.11a, based on orthogonal frequency-division multiplexing (OFDM) technology, with data rates up to 54 Mb/s in the 5 GHz band

In 2003 another version of the standard, 802.11g that extends the 802.11b PHY layer to support data rates of up to 54 Mb/s in the 2.4 GHz band was finalized.

With direct sequence spread spectrum (DSSS), each bit in the original signal is represented by multiple bits in the transmitted signal, using a spreading code. The spreading code, also known as pseudorandom noise (PN) code, spreads the original signal across a wider frequency band for the transmission. While at the receiver side, the receiver performs the correlation process using the same PN codes to get the original signal.

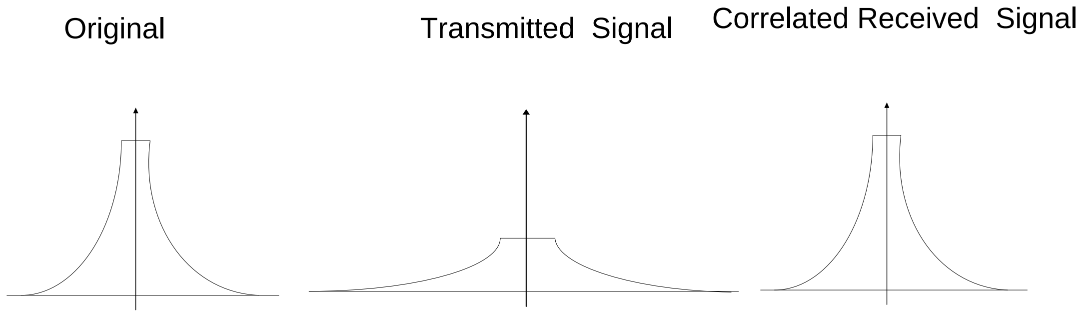


Figure 2.10: Basic DSSS Technique [24]

For the PN code, 802.11 use 11 bit barker code {10110111000}. Barker code is applied to the each bit in the data sequence by a modulo-2 adder as shown in the example Figure 2.11. When a 1 is encoded, all the bits in the spreading code change, while for 0 they remain the same. 802.11 uses two modulation schemes differential binary shift keying (DPSK) for 1 Mbps and differential quadrature shift keying (DQSK) for 2 Mbps, both with the same barker code. But to achieve the high data rate, the IEEE 802.11 working group turned to an alternate encoding method named Complementary Code Keying (CCK). CCK divides the chip stream into a series of 8 bit code symbols, hence providing the series of 1.375 million code symbols per second. CCK, which is based on a sophisticated mathematical transforms, allows the use of 8 bit sequence to encode 4 or 8 bits per code word thus providing the throughput of 5.5 Mbps or 11 Mbps.

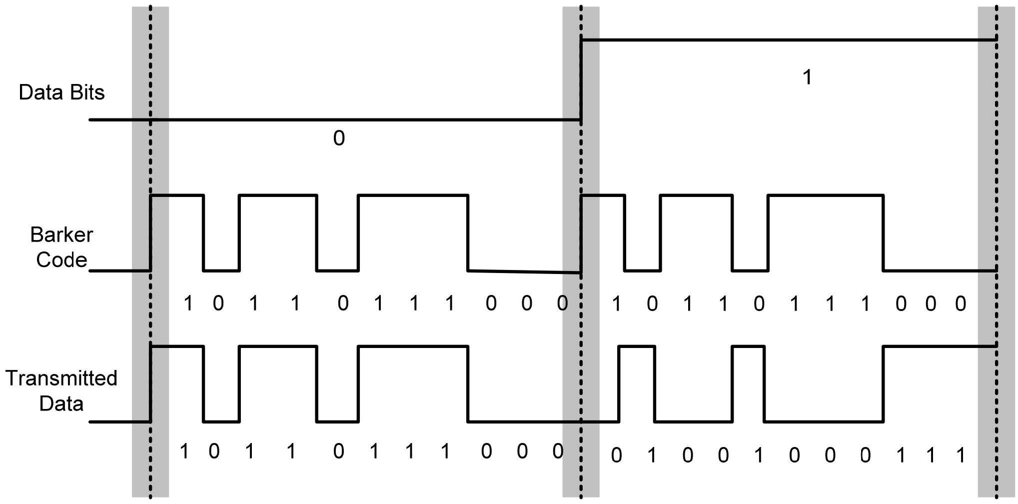


Figure 2.11: Encoding with the Barker Code [18]

802.11 split the PHY into two generic components: Physical Medium dependent (PMD) and Physical Layer Convergence Procedure (PLCP). PMD defines the characteristics of and method of transmitting and receiving data through, a wireless medium (WM) between two or more STA's [1]. While PLCP defines a method of mapping the MAC sub layer protocol data units (MPDU) into a framing format suitable for sending and receiving user data and management information between two or more STA's using the associated PMD system. For the completion of this task PLCP added its own preamble and header with PLCP service data unit (PSDU) which is based on the MPDU as shown in Figure 2.12.

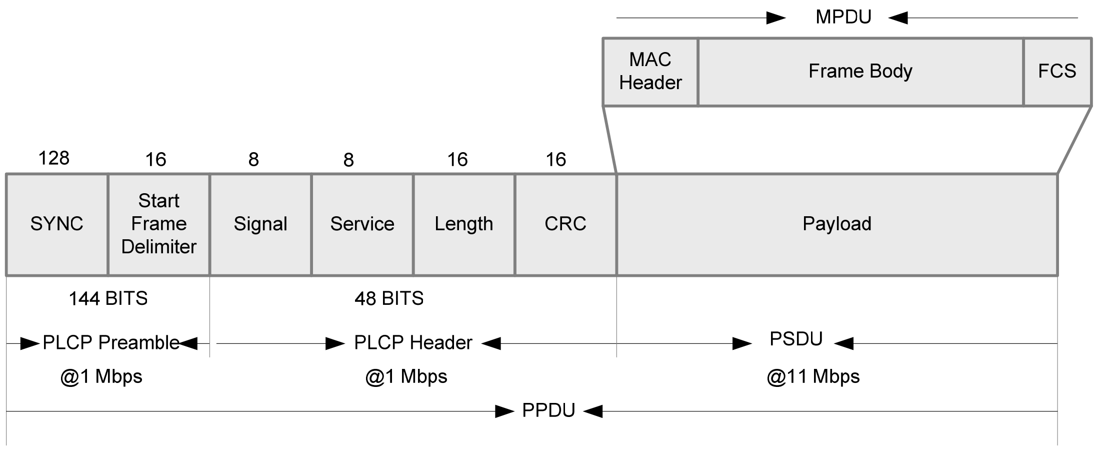


Figure 2.12: PLCP Frame of IEEE 802.11b [27]

- PLCP Preamble

  The preamble is used to synchronize the transmitter and receiver in terms of deriving common timing relationship. The different fields in the preamble are:

  - PLCP Synchronization (SYNC) field:

    This is a 128 bits field which is consists of sequence of 1`s (pulse) to provide the synchronization between sender and receiver.

  - PLCP Start Frame Delimiter (SFD) field:

    The SFD allows the receiver to find the start of the frame, even if some of the sync bits were lost in transit.

- PLCP Header

  The PLCP header follows the preamble. The header has PHY-specific parameters used by the PLCP. The different fields in header are:

  - PLCP Signal field:

    This is a 8 bit field which is used by receiver to identify the transmission rate of the encapsulated MPDU.

  - PLCP Service field:

    This 8 bit field is reserved for future use.

  - PLCP Length field:

    This 16 bit field consists of unsigned integers which show the amount of microseconds required to transmit the frame.

  - PLCP Cyclic Redundancy Check (CRC) Field:

    This 16 bit field is used to protect the header to be corrupt on the radio link. To assure accuracy, the sender calculates 16 bit CRC over the contents of the header and send it to receiver. When frame will arrive at the receiver, receiver will recalculate that CRC value over the received header contents and match it with the original CRC value, the header will be considered bug free if the CRC value is same otherwise corrupted.

- PSDU field:

  PSDU is actually the MPDU sent by the MAC sub layer and can have a size from a minimum of 0 to a maximum of 2500.

2.2.5 Medium Access Control

The MAC sub layer, as the name denotes, defines the controlling mechanism in the BSS for all the stations to access the medium. The most important functionality of MAC sub layer is frame formatting while common functionalities include power management, multiple access and security. The general MAC frame format is shown in Figure 2.13 in which frame control, destination ID, address1 and FCS are the compulsory fields while Address2, Address3, sequence control, address4 and frame body can be omitted, depending on the frame type.

Each frame starts with a two byte Frame Control sub field which defines the control information, such as type and sub type etc. The Duration ID field is used to set the value of network allocation vector (NAV) which shows the number of microseconds that the medium is expected to remain busy for the transmission currently in progress. An 802.11 frame may contain up to four address fields. The address field are numbered because different fields are used for different purposes depending on the frame type. 802.11 frames use two byte field for both de-fragmentation and discarding duplicate frames. The frame body, also known as data field, moves the data of the higher layer from station to station. The frame check sequence (FCS) allows stations to check integrity of received frames.

The 802.11 standard defines two access coordination functions; contention based period and contention free period. In contention based period, all stations try to access medium using Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) while in contention free period, all station can transmit their data at the given time slot. The contention based period is provided by distributed coordination function (DCF) while contention free period is provided by point coordination function (PCF).

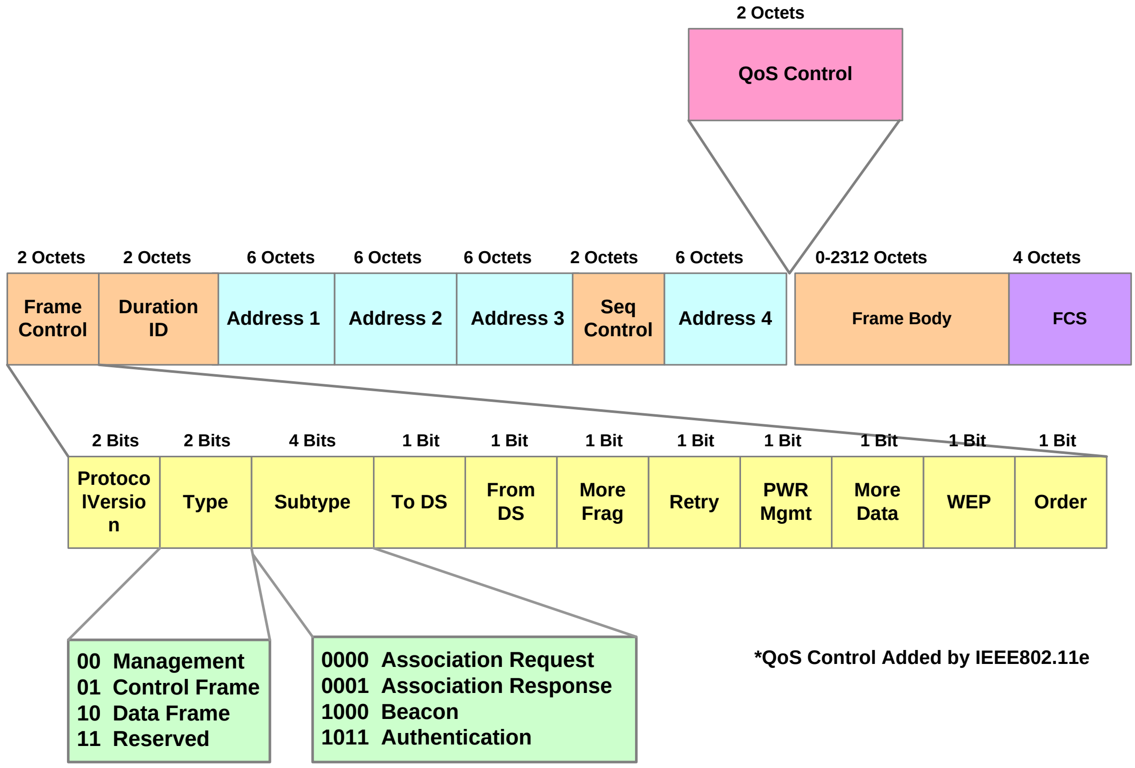


Figure 2.13: Mac Frame Format [4]

2.2.5.1 Distributed Coordination Function

The DCF allows multiple independent stations to interact without central control, and thus may be used in either IBSS networks or in infrastructure network. DCF operates on listen-before-talk scheme known as CSMA/CA. In DCF mode, each station is required to sense the medium before initiating transmission and perform a binary exponential backoff. If the medium has been sensed idle for a time interval called DCF interframe space (DIFS), the station enters a backoff procedure. A slotted backoff time is generated randomly from a contention window (CW). The backoff time is computed with the consideration of the equation 2.1

$$\text{backoff time} = \text{rand}\,[0; \text{CW}]\ \text{x slot time} \qquad (2.1)\ [7]$$

While if the medium is busy; the station must wait for the channel to become idle for DIFS and then start exponential backoff procedure again. At the first transmission attempt, the CW is set to minimum value, CW_{min}. It is doubled after each unsuccessful transmission until reach a maximum value, CW_{max}. When the contention window reaches its maximum size, it remains there until it can be reset to CW_{min} after the successful transmission. The contention window example is as shown in Figure 2.14 where CW_{max} = 1,023 slots.

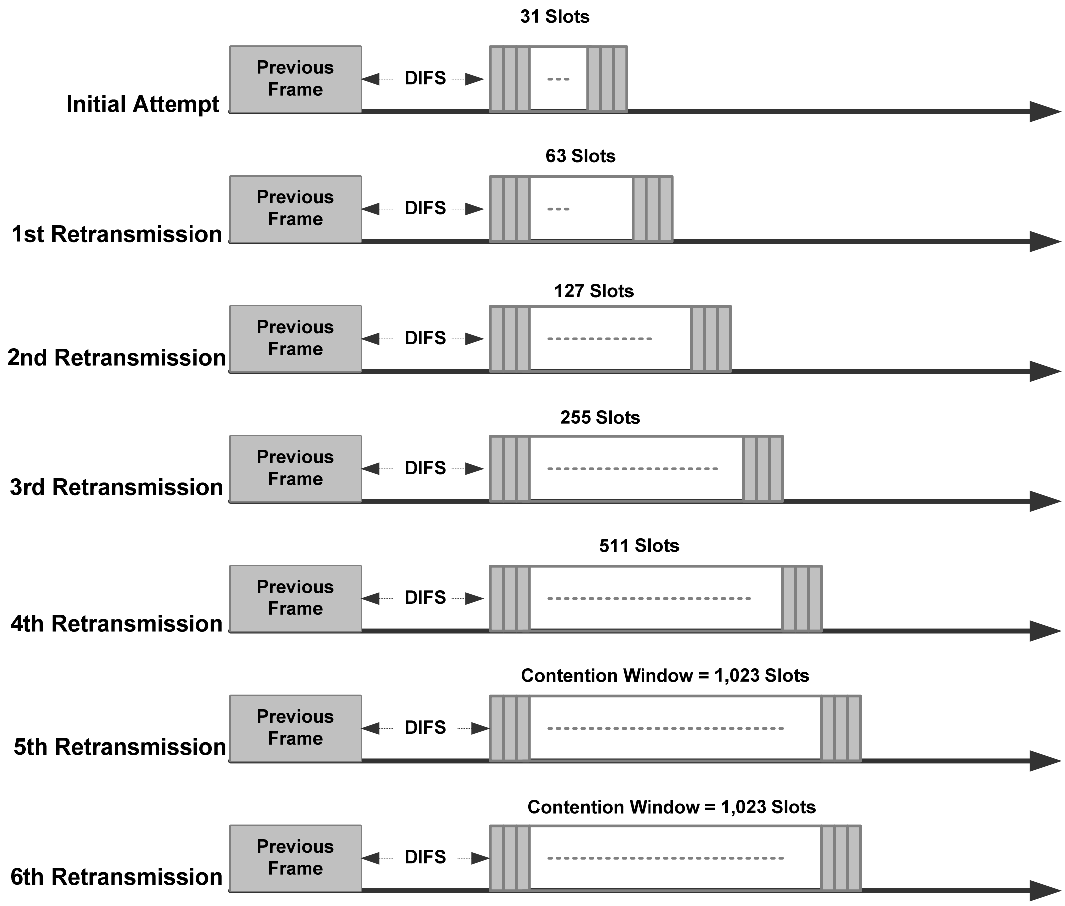


Figure 2.14: Example of Contention Window size [18]

The purpose of the backoff procedure is to reduce the possibility of collision by selecting a different random backoff time for different stations. DCF, which was designed to support best effort data traffic, provides no services to differentiate the frames in terms of different priorities. Thus, no support for real-time traffic is provided in DCF.

2.2.5.2 Point Coordination Function

To support applications that require near real-time services, the 802.11 standard includes a second coordination function named point coordination function (PCF) which is built on the top of DCF. PCF can only be used if a WLAN operated in an infrastructure mode. 802.11 standard defines PCF as an optional MAC function because the hardware implementation of PCF was thought to be too complicated [7]. PCF is a polling-based contention free scheme in which access point (AP) act as a point coordinator. When a WLAN system is set up with PCF enabled, the channel access time is divided into periodic interval called beacon intervals. A beacon interval is composed of a contention free period (CFP) and a contention based period (CP) as shown in Figure 2.15.

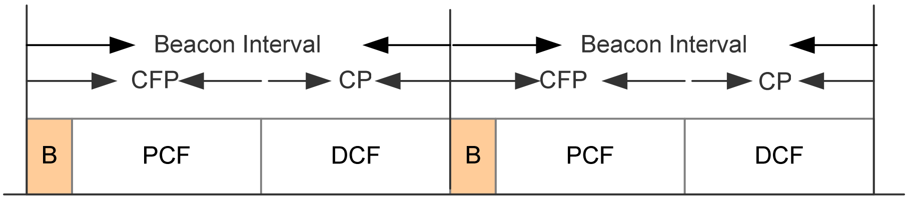


Figure 2.15: 802.11 Beacon Interval

During a CFP, the AP maintains a list of registered PCF enabled stations and polls them according to list. Only after a station is polled it can start its data transmission. While PCF was designed to support real-time applications, this mode has three major problems that lead to poor QoS performance [7, 19, 20, and 21].

- PCF defines single class round robin algorithm that cannot handle various QoS requirements.
- With PCF, stations are allowed to transmit even if the frame transmission cannot finish before the next beacon.
- PCF allows station to send variable length frames (0, 2304).

A common problem with both DCF and PCF is that no admission control mechanism is specified in the standard 802.11 legacy MAC [1]. When traffic load is very high, the performance of both functions can be degraded. 802.11 MAC coordination function implementation is shown in Figure 2.16.

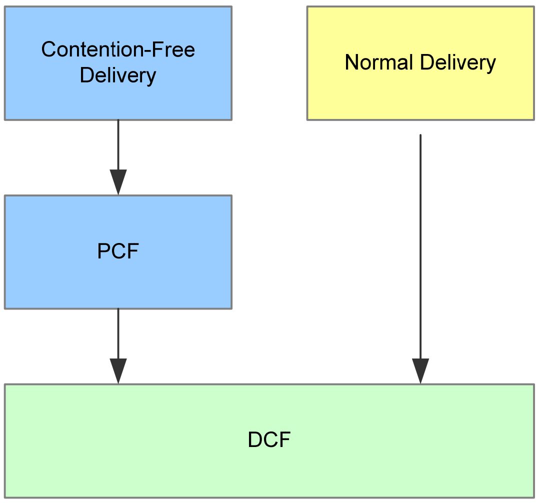


Figure 2.16: MAC Coordination Function [18]

2.3 IEEE 802.11e for Quality of Service

After the establishment of IEEE 802.11 in 1999, IEEE established different task groups that are working to provide the enhanced features in WLAN. Figure 2.17 shows the 802.11 supplements.

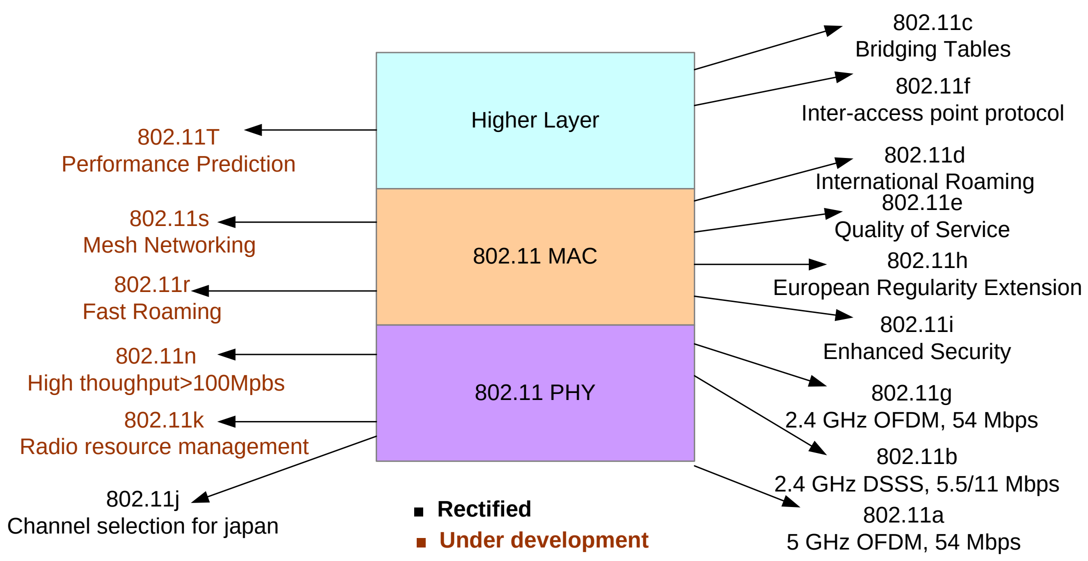


Figure 2.17: IEEE 802.11 Supplement [22]

Among these task groups, 802.11e is a task group that is working to provide the QoS support, in WLAN, for the transmission of real-time data. IEEE802.11e introduces the new Hybrid Coordination Function (HCF) for the medium access, which consists of the contention based Enhanced Distributed Channel Access (EDCA) for prioritized QoS along with contention free HCF Controlled Channel Access (HCCA). The HCF combines the methods of PCF and DCF, which is the reason it is called hybrid. These new functions are built on DCF defined in the 802.11 legacy model as shown in Figure 2.18. The station that operates as the central coordinator for all other stations within the QoS supporting BSS (QBSS) is called the Hybrid Coordinator (HC). According to 802.11e, QoS enabled station is known as QSTA while QoS enabled access point is known as QAP. IEEE802.11e provides some basic improvement in legacy 802.11 MAC which includes; an 802.11e station that obtains medium access must not utilize radio resource for duration longer than a specified limit. This important new feature is known as transmission opportunity (TXOP) which refers to the time duration in which a registered station has the right to deliver MSDU. A TXOP is defined by its starting time and duration. TXOP obtained via contention based medium access are referred to as EDCA-TXOP while TXOP obtained by the HC via controlled medium access is referred as HCCA-TXOP or polled TXOP.

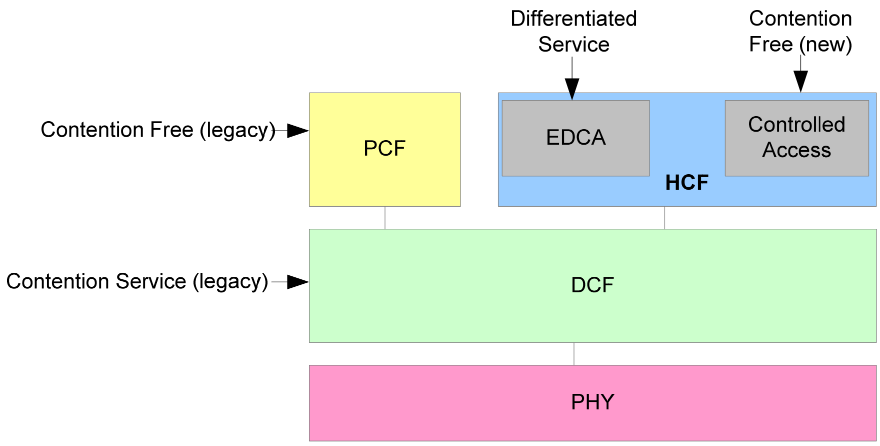


Figure 2.18: Mac Architecture in IEEE802.11e [4]

Another enhancement is that; station is only allowed to initiate its frame exchange if it can be completed before the start of the next beacon interval.

2.3.1 HCF Contention-Based Channel Access (EDCA)

Enhanced distribution coordination function (EDCF) is an extension of the legacy DCF which introduce the concept of Access Category (AC) and multiple independent backoff entities to provide QoS support. Before entering the MAC layer every data packet received from the upper layers is assigned a specific user priority value which depends on the user application. While at the MAC layer these data packets, after reading user specific priority, will be mapped into their corresponding AC. Every EDCA enabled station should implement four access categories namely Background (BK), Best-effort (BE), Video (VI) and Voice (VO) where BK traffic has the lowest priority whereas VO traffic has the highest priority. Figure 2.19 illustrate the different AC's and parallel backoff entities. In EDCA, every AC behaves as a single station. EDCA introduces a new type of interframe space (IFS) called arbitration IFS (AIFS), in place of DIFS in DCF. The minimum value of AIFS is equal to DIFS. Each AIFS is an IFS interval with an arbitrary length as shown in 2.2:

$$AIFS[AC] = SIFS + AIFSN[AC] * slot\text{-}time \qquad (2.2)$$

where *AIFSN*[*AC*] is called the arbitration IFS number. After sensing the medium idle for a time interval of *AIFS*[*AC*], each AC calculates its own random backoff time (*CWmin*[*AC*] <= backoff time <= *CWmax*[*AC*]). The purpose of using different contention parameters for different queues is to give a low priority class a longer waiting time than a high-priority class, so the high-priority class is likely to access the medium earlier than the low-priority class. Note that the backoff times of different AC's in one QSTA are randomly generated and may reach zero simultaneously. This can cause an internal collision. In such a case, a virtual scheduler inside every QSTA allows only the highest-priority AC to transmit frames as shown in Figure 2.19

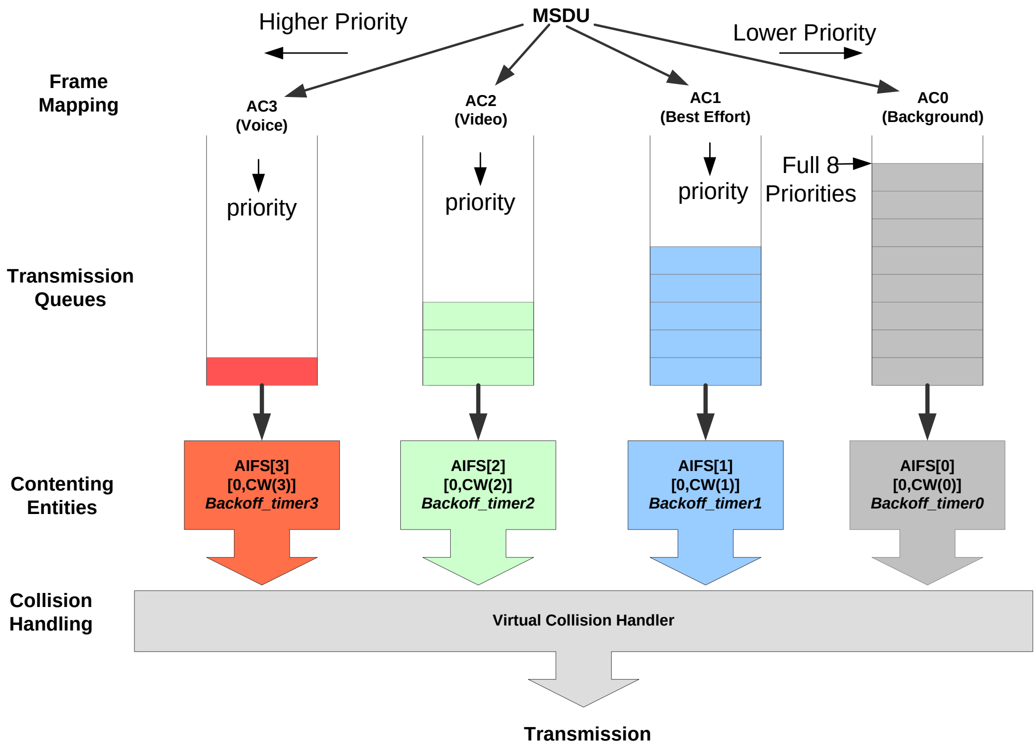


Figure 2.19: Access Category in EDCA [27]

A big difference between legacy DCF and 802.11e EDCA in terms of the backoff and countdown rule is as follows [23]:

- The first backoff countdown occurs at the end of the *AIFSN[AC]* interval.
- A frame transmission is initiated after a slot from the moment when the backoff counter becomes zero.

The purpose of using AC-specific contention parameter is to give low priority class a longer waiting time than a high priority class, so the high priority class is likely to access the medium earlier. Even the CW_{max} of highest priority AC is smaller then the CW_{min} of the lowest priority AC so that highest priority AC can access the medium before lowest priority AC. After any unsuccessful transmission the value for CW is recalculated using 2.4:

$$\text{newCW[AC]} = ((\text{oldCW} + 1) * 2) - 1 \qquad (2.4)$$

In order to start transmission using EDCA, each AC in a registered station transmits a QoS request to the QAP containing a traffic specification (TSPEC) of its application (e.g., mean/peak data rate, mean/maximum frame size). If the QAP accept that request, it calculates the amount of time per second for admitted traffic to access the medium which is called *medium_time* and send to station. After receiving *medium_time*, the admitted station updates its local *admitted_time* and uses another local variable, used _time, to record how long the station has accessed the medium.

The *used_time* is updated after every transmission attempt and if *used_time* is larger than *admitted_time*, the corresponding AC is not allowed to transmit any data frame until *used_time* is reset. If in case station needs more *admitted_time*, it has to send a new request to the QAP.

Figure 2.20 shows an example scenario with three QoS enabled stations contending for the medium with having different parameter for the channel access.

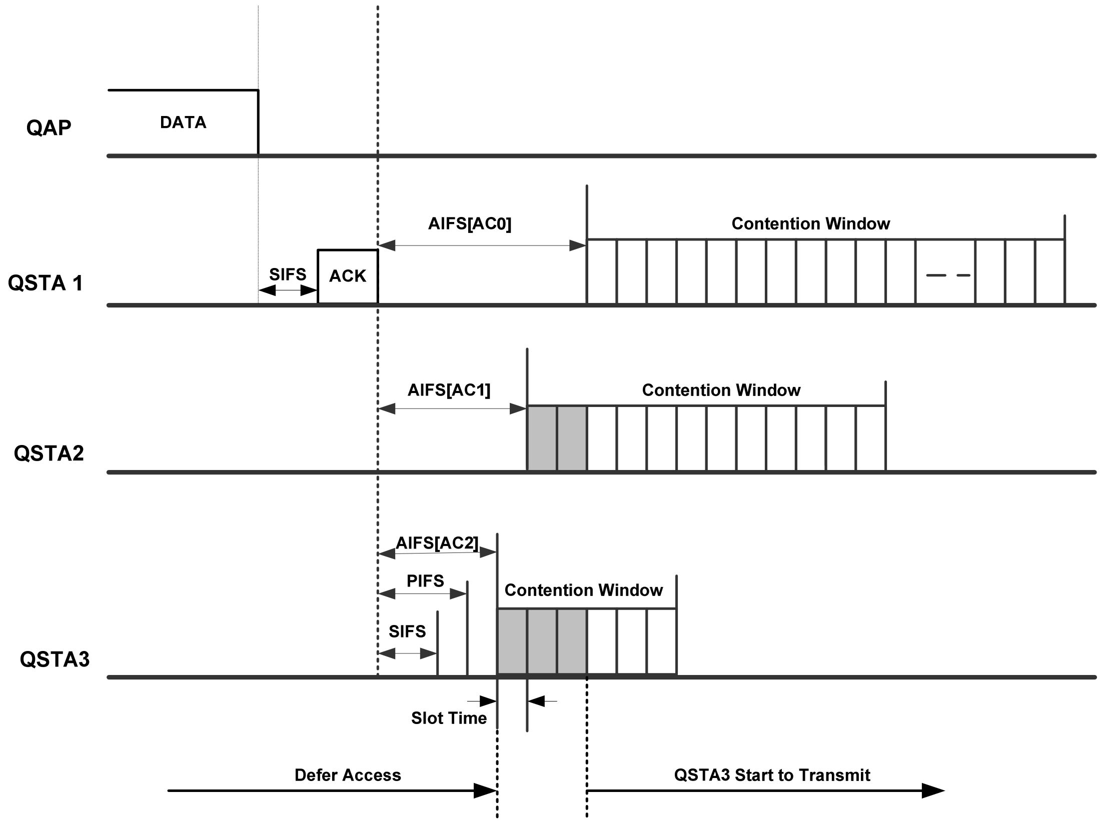


Figure 2.20: Channel Access with different parameters

2.3.2 HCF Controlled Channel Access (HCCA)

In 802.11e, HCCA mechanism provides the controlled medium access for contention free transfer. The time between two consecutive beacon frames, which is called superframe, is divided into contention free period (CFP) and contention based period (CP). To provide the controlled medium access, HCCA uses a centralized coordinator called a Hybrid Coordinator (HC). Similar to the Point Coordinator (PC) of the PCF, the HC resides within an 802.11e AP. The HC controls the access to the channel during CFP by polling its associated stations and allocating TXOPs either to itself or to its associated stations with their QoS requirements. However, the HC is also allowed to initiate a controlled access phase (CAP) during the CP after detecting that the channel is idle for a time interval longer than a PIFS and whenever there is a need to transfer real-time data. The controlled access period (CAP) is limited to a maximum duration, $CAP_{LIMIT,}$ in order to leave space for stations operating under EDCA. Since PIFS is shorter then DIFS and AIFS, HC is given a greater opportunity to start HCCA than EDCA. HCCA is more flexible than PCF because the latter is only allowed in a CFP while HC can initiate HCCA whenever it wishes during the whole beacon interval as shown in Figure 2.21.

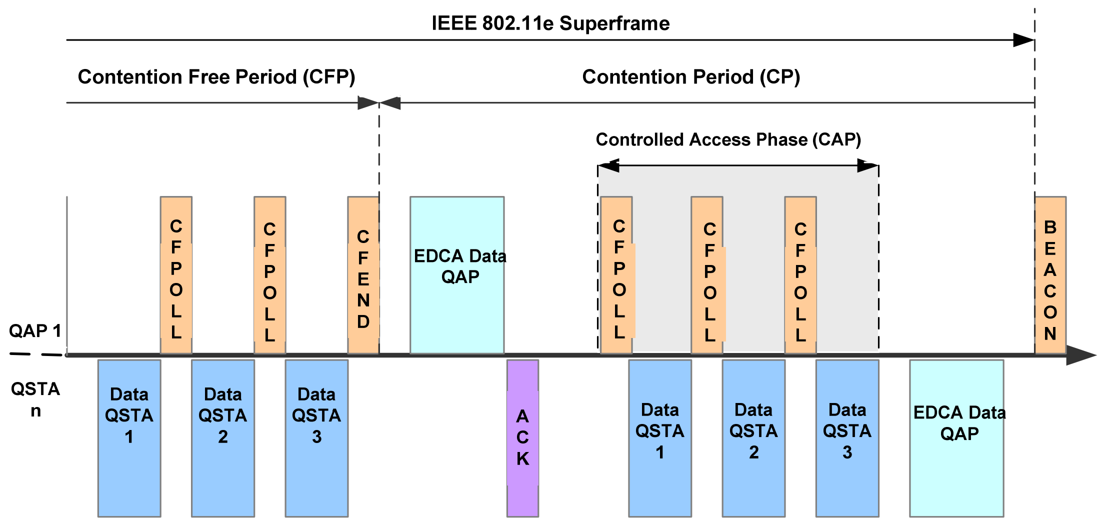


Figure 2.21: IEEE 802.11e Superframe [4]

HCCA solves the above three major problems of PCF as follows [7]:

- Different traffic classes called traffic streams (TSs) are introduced in HCCA. Manufacturers can design multiclass scheduling algorithms to support different types of applications. In addition, scheduling algorithms are treated as implementation-dependent in HCCA, can be enhanced by manufacturers.

- An 802.11e QSTA is not allowed to transmit a packet if the frame transmission cannot finish before the next beacon.
- A $TXOP_{LIMIT}$ is used to bind the transmission time of a polled QSTA.

Before any data transmission using HCCA mechanism, a traffic stream (TS) is first established. Each QSTA is allowed to have up to eight TSs with different priorities. TSs, in HCCA, and ACs, in EDCA, can use different MAC queues. In order to initiate TS connection, a QSTA sends a QoS request frame containing a traffic specification (TSPEC) to QAP. TSPEC describes the QoS requirements of TS and is shown in Figure 2.22 where maximum service interval (MSI) refers to the maximum duration between the start of successive TXOPs that can be tolerated by a requesting application. If both MSI and Delay Bound are specified by the non-AP QSTA in the TSPEC, the Scheduler uses the MSI for the calculation of the TXOP.

	Element ID (13)	Length (55)	TS Info	Nominal MSDU Size	Maximum MSDU Size	Minimum Service Interval	Maximum Service Interval	Inactivity Interval	Suspension Interval
Octets	1	1	3	2	2	4	4	4	4

	Service Start Time	Minimum Data Rate	Mean Data Rate	Peak Data Rate	Burst Size	Delay Bound	Minimum PHY Rate	Surplus Bandwidth Allowance	Medium Time
Octets	4	4	4	4	4	4	4	2	2

Figure 2.22: Traffic Specification Elements [4]

According to IEEE 802.11e when an AP receives these QoS parameters, it will schedule the admitted stream in two steps.

- First the scheduler calculates the minimum of all Maximum Service Intervals for all admitted streams. Second, the scheduler chooses a number lower than or equal minimum MSI that is a sub-multiple of the beacon interval. This value is the Scheduled Service Interval for all non-AP QSTAs with admitted streams.
- In the second step, the TXOP duration for a given SI is calculated for the stream. For the calculation of TXOP, scheduler first calculate the number of arriving frames (N) during one SI in (2.6) for every traffic stream $\mathbf{i}$ of a station.

$$N_i = \left\lceil \frac{SI \times p_i}{L_i} \right\rceil \quad i = 1 \ldots n \quad \textbf{(2.6)[4]}$$

Then the scheduler calculates the TXOP duration as the maximum of N_i frames at R_i and time to transmit one maximal MSDU at R_i plus the additional overhead (MAC header + PHY header) [5]

$$TXOP_j = \sum_{i=1}^{n} \max\left(\frac{N_i \times L_i}{R_i} + O, \frac{M_i}{R_i} + O \right) \qquad \textbf{(2.6)[4]}$$

An example is shown in Figure 2.24 where beacon interval is 20ms and the minimum MSI among 3 streams is 12 ms. The scheduler calculates a Scheduled Service Interval (SI) equal to 10ms using the steps explained above. The TXOP is calculated using equation (2.5) and (2.6). CAPs are constructed after concatenating TXOPs from all stations. The CAP is always repeated after a period equal to the SI value and a length of sum of all TXOPs

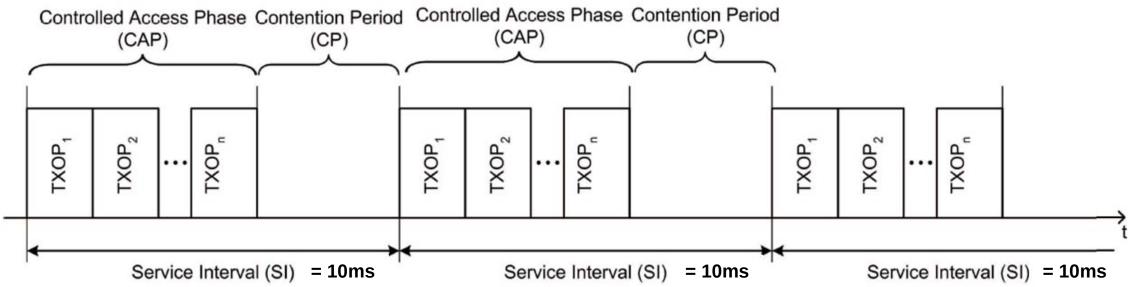


Figure 2.23: A sample TXOP allocation schedule

3 OPNET- An Overview

This chapter presents an overview of OPNET simulation tool which is a comprehensive integrated development environment for modeling and performance evaluation of communication networks and distributed system. Next section explains the importance and different aspects of simulation

3.1 Simulation

Simulation is a useful technique for the computer systems performance and analysis as quoted in [26]. Similarly [28] defines simulation as an important feature in engineering systems or any system that involves many processes, to measure its system performance. A simulation model provides an easy way to predict the performance when the system to be characterized is not available. Further, some times the simulation model may be preferred even if a system under consideration is available for measurement because the simulation model generally allows alternatives to be compared under a wider variety of workloads and environments. For every simulation study we have to consider several assumption and simplifications. According to [26], the level of detail in a simulation model is limited only by the time available for simulation development because a more detailed simulation also require a greater computer time to run. It is generally assumed that a more detailed model is a better model, since it makes fewer assumptions. Some analyst often tries to save their own time as well as computer's time by running simulation that is too short. The results in such cases are not true representative of the real system. In some cases even if a simulation program has no error, it may not represent the real system correctly because of incorrect assumptions about the system's behavior that's why it is essential that the models must be validated before implementing in the real system, to ensure that the conclusion reached would be the same as expected from the real system. Figure 3.1 presents a flow diagram of a simulation study which shows that if our results are not accurate or sufficiently detailed then we have to define our input and desired output again.

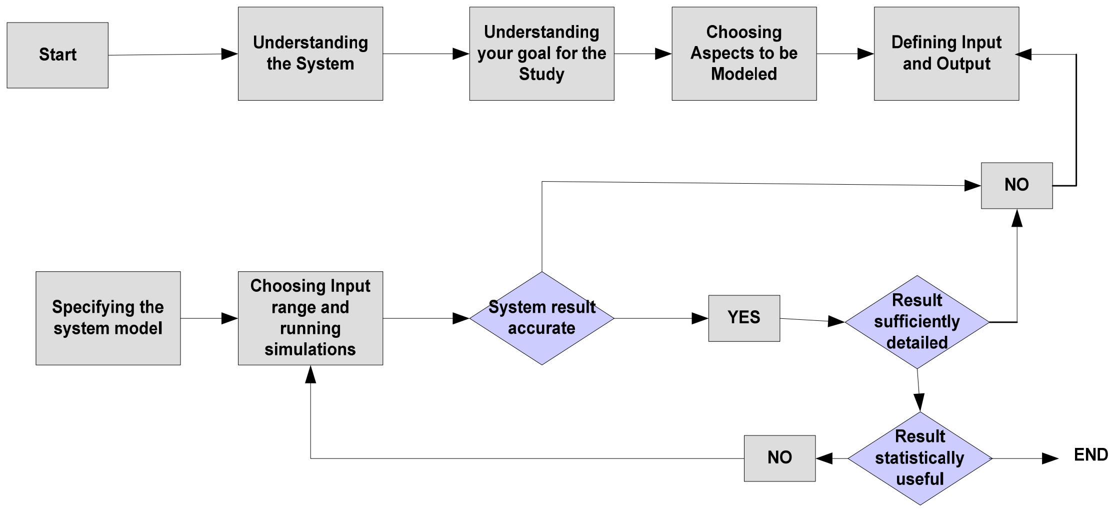


Figure 3.1: Flow Diagram of a Simulation Study [29]

3.2 OPNET Modeller

This section precisely deals with OPNET architecture and an explanation of representing a real-world system with an OPNET model. Figure 3.2 presents the OPNET model architecture and its corresponding OPNET editors. Here we only define first three OPNET editors.

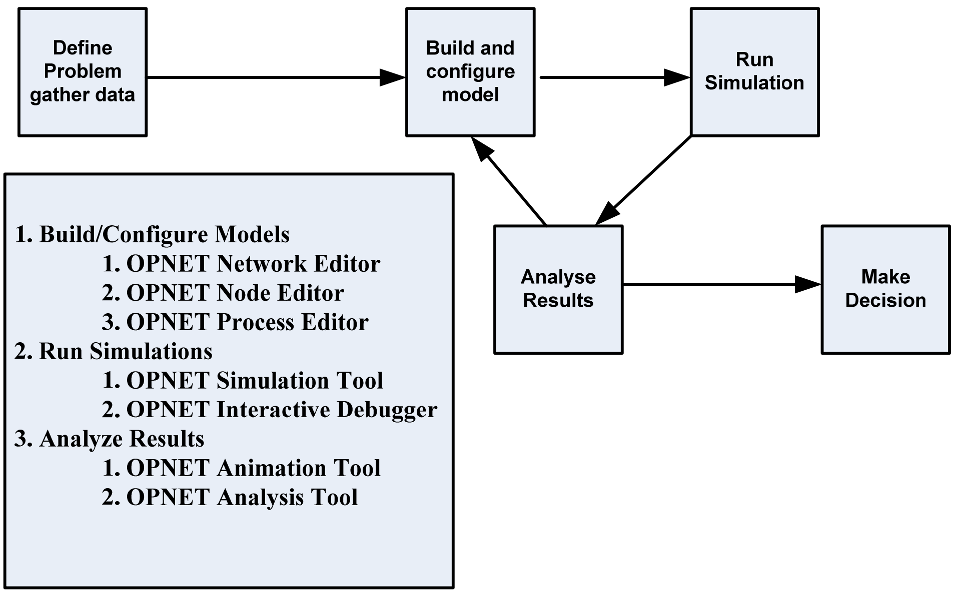


Figure 3.2: OPNET Model Architecture [29]

3.2.1 Project Editor

The function of Project editor

- Create a network model using models from the standard library
- Collect statistics
- Run the simulation and view the result
- Other editors are accessible from the project editor

Modeler uses a project-and-Scenario approach to modeling networks

- A Project is a collection of related network scenarios
- A Scenario is a single instance of a network. It presents a unique configuration for the network.
- Exp: WLAN Scenario: PCF_vs_DCF is shown in Figure 3.3 where WLAN is a name of the project and PCF_DCF is one of its scenarios.

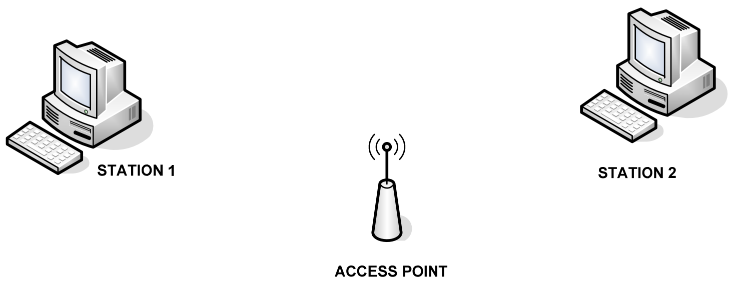


Figure 3.3: Project Editor

3.2.2 Node Editor

- Define the behavior of each network object.
- The behavior of the each network object is shown graphically which represents data flow within a communications device (source, transmitter, sink etc.)
- Depicts layering of protocols
- Multiple modules, packet stream, statistic wires
- Exp: wlan_station_adv shown in Figure 3.4

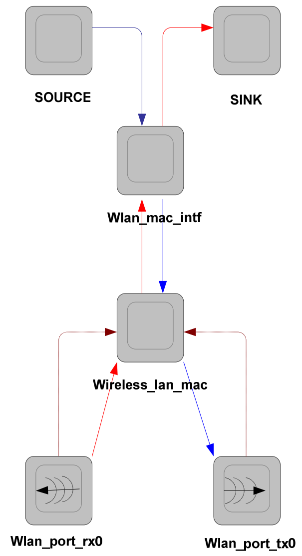


Figure 3.4: Node Editor

3.2.3 Process Editor

- Create process model which control the functionality of the node model
- These functionalities are defined in form of FSM(finite state machines)
- Finite state machine is a combination of States and Transitions.
- Operations performed in each state or for a transition
    - o Embedded C or C++ code blocks
- Exp: *bursty_source* shown in Figure 3.5

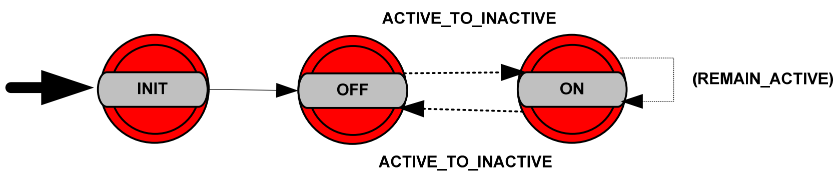


Figure 3.5: Process Editor

Figure 3.6 show the hierarchy of the OPNET editors which shows the Proto- C code of off state of *bursty_source* process model which is process model for source node of *wlan_station_adv* node model which is a node model for *PCF_vs_DCF* at network model.

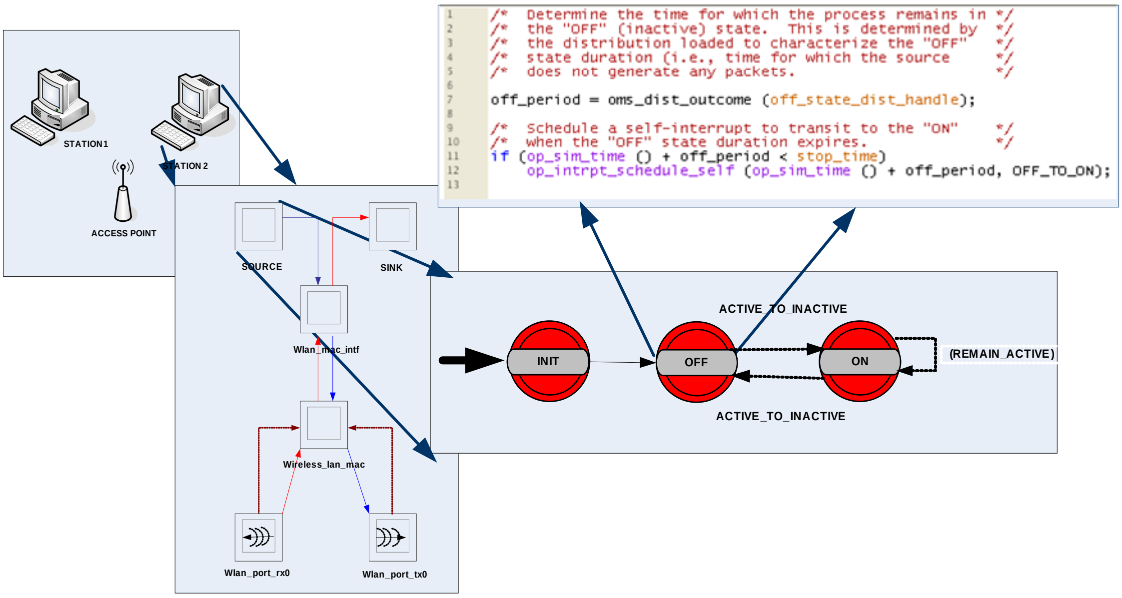


Figure 3.6: Hierarchy of OPNET Editors

3.2.4 OPNET–An Event Driven Simulation

- An event is a request for a particular activity to occur at a certain time.
- OPNET simulations are event-driven. Time, in the simulation, advances only when some event occurs.
- An OPNET simulation maintains a single global event list.
- All objects access a shared simulation time clock.
- Events are scheduled on the list in time order. The first event on the list is the head.
- An event has data associated with it.
- When an event completes it is removed from the list.
- An event becomes an interrupt when it reaches the head of the event list and is delivered by the Simulation Kernel to the designated module.
- Data associated with the event can be obtained by the module when the interrupt occurs.
- Certain modules, processes, and queues can be selected to place initial interrupts on the event list.
- An entity, the Simulation Kernel (SK), manages the event list.
- The SK delivers each event, in sequence, to the appropriate module.

- The SK receives requests from modules and inserts new events on the event list.

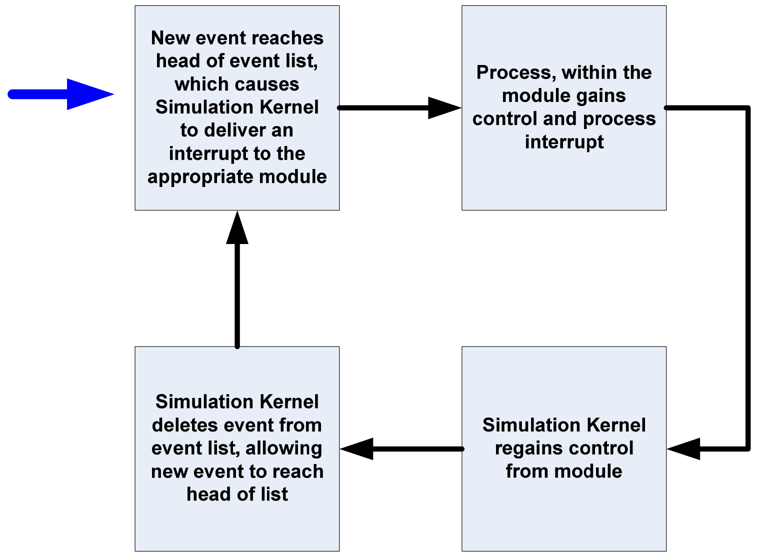


Figure 3.7: Working principle of Event List [29]

3.2.5 OPNET Example Project

Consider an example project that counts the number of packets sent from one node to another.

In the start of the simulation, the simulation kernel passes an interrupt to subnet (*subnet_0*), network level, as shown in Figure 3.8. As *subnet_0* is a subnet so the interrupt will be delivered to the module (SOURCE shown in Figure 3.9) running within subnet. When an interrupt is delivered to a module, control passes from the Simulation Kernel to the module.

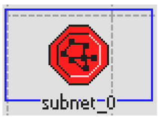


Figure 3.8: Subnet at Network Level

As the module (SOURCE) is a processor, the interrupt is delivered to the process, (*bursty source* shown in Figure 3.10) running within the module.

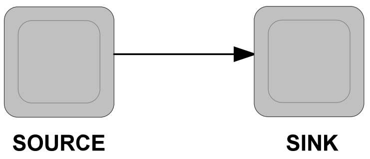


Figure 3.9: Packet_Count_node at Node Level

When an interrupt is delivered to a process model, control passes from the Simulation Kernel to the process model initialization state (INIT). In INIT, Enter Executive will initialize the variable pk_count == 0 and pass control to Exit Executive. Exit Executive will check the transition condition pk_count == 0; if the transition condition is true, transition will occur as shown with (3) in Figure 3.10. The control will remain in OFF state until the transition condition INACTIVE_TO_ACTIVE evaluates true. When the control transferred to ON state, ON state will create new packet and send it to next module (SINK). The control once again transferred to OFF state when the transition condition ACTIVE_TO_INACTIVE evaluates true.

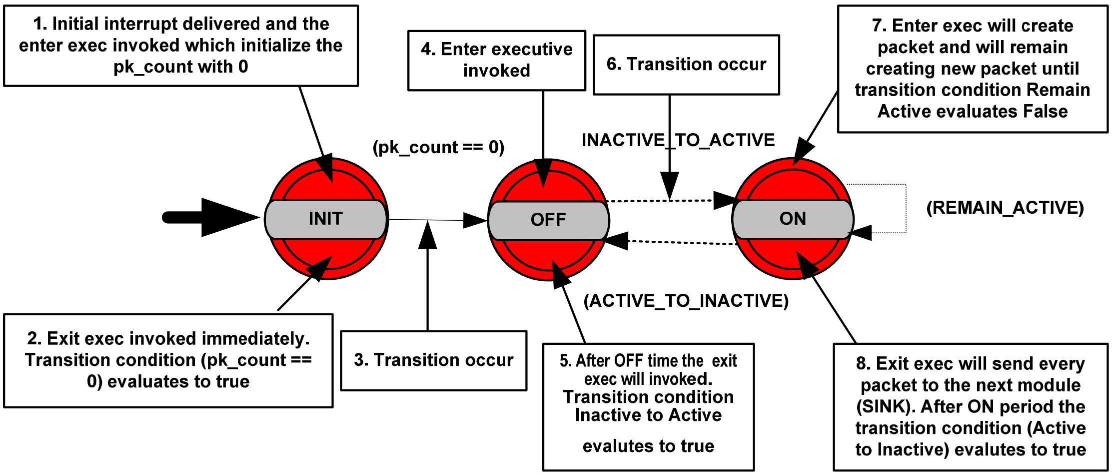


Figure 3.10: Busty Source Process Model

When an interrupt is delivered to a module, control passes from the Simulation Kernel to the module. As the module (SINK) is a processor, the interrupt is delivered to the process (SINK) running within the module. When an interrupt is delivered to a process model, control passes from the Simulation Kernel to the process model initialization state (INIT). The INIT state will increment the packet count variable and pass the control to Exit Executive.

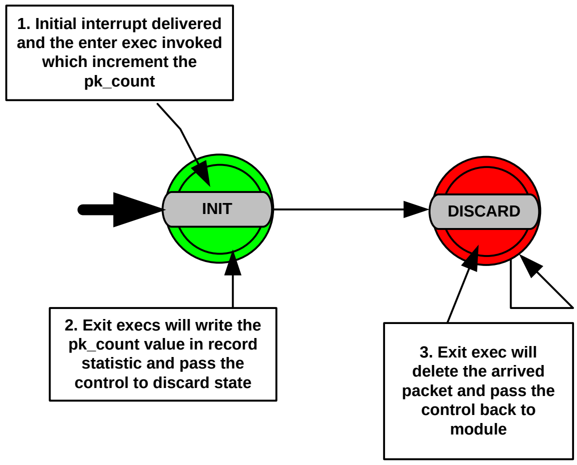


Figure 3.11: Sink Process Model

Exit Executive will write the value of the packet count in record statistics using the script given below

op_stat_write (pk_cnt_stathandle, pk_count);

and pass the control to DISCARD state. The DISCARD state will delete the packet and pass the control back to the SINK module which will pass the control back to simulation kernel.

4 HCCA OPNET Enhancement

As specified in the objective the primary objective of this project is to add HCCA enhancement in the current WLAN-MAC OPNET simulation model and do enhanced HCCA examination for different scenarios. The enhanced OPNET simulation model includes the functionalities of scheduler, CAP interval generation, HC assigning TXOPs to individual stations, disable contention based transmission during CAP duration and handling of non real-time traffic during contention based transmission. The designing of the OPNET model starts at the node level and ends up at process level implementation. From the network level, the station sends its values to node level and every node sends those values to its appropriate state machine (process level) in the form of interrupts to complete desired task. In the proposed system the contention free period of existing WLAN legacy MAC model has been modified to include the functionality of HCCA, because the existing PCF functionality of WLAN MAC model can easily be extended to obtain the required characteristics. Hence the first part of this chapter describes the working principle of WLAN MAC model.

4.1 WLAN MAC Model 802.11

Stations at the network level sends its values to node level (wlan_station_adv), which includes the packet generation distribution, type of traffic service and Wireless Lan characteristics, as shown in Figure 4.1.

4.1.1 Node Model

The node level modeling domain includes Physical layer, MAC layer, Transport layer characteristics exempting the higher layer functionalities as shown in Figure 4.2. The source and sink modules define the generation and reception process model of the packets.

4.1.1.1 SOURCE MODULE

The source model generates the packet according to the received packet generation distribution in the model attributes as well as all the initial packet generation statistics like number of bits/packets sent per second are registered and calculated here. The legacy model is using constant packet generation distribution. There are three types of destination addressing mechanism namely random, specific destination and deterministic. In random mechanism every generated packet is assigned a random destination address and specific destination is where all packets are sent to the same station. While in deterministic addressing mechanism in detail is where the packets are sequentially generated to each and every station in BSS. In the proposed system we use deterministic destination address mechanism.

4.1.1.2 SINK MODULE

The sink module on the other hand is typically used for the calculation of packet delays after which the packets are normally destroyed.

4.1.1.3 WLAN_MAC_INTERFACE MODULE

The wlan_mac_interface module provides an interface between application layer and MAC layer. The module handles the packet from the source forwarding them to wireless_lan_mac module and also handles the packet from the physical layer destined to this station forwarding them to the sink.

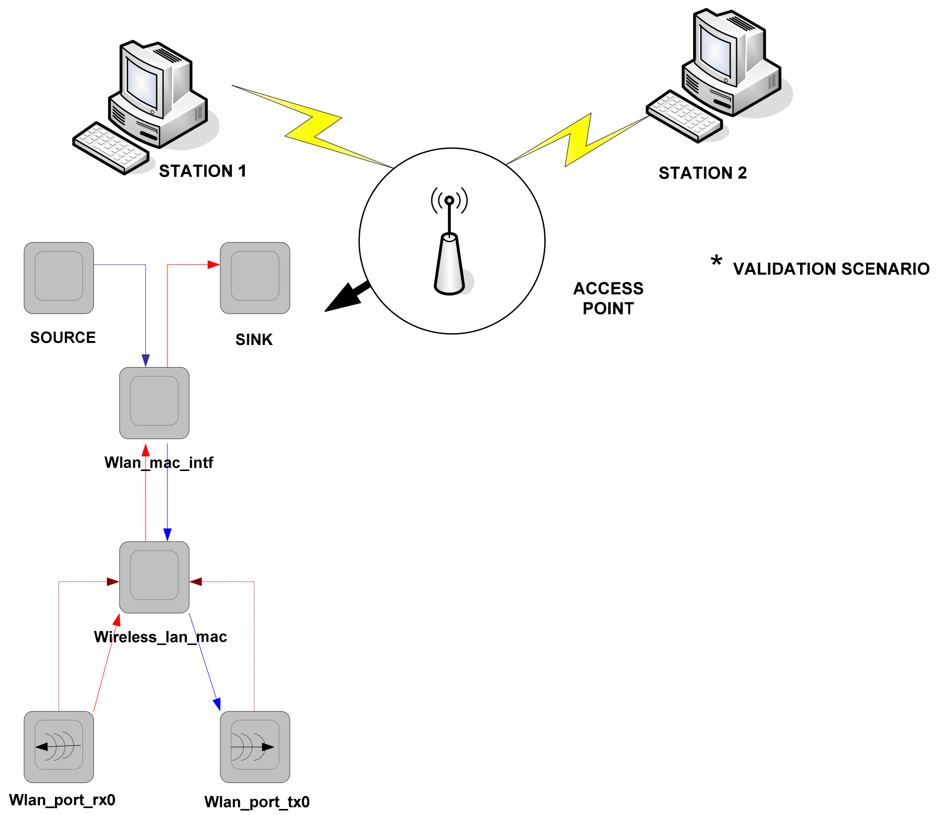


Figure 4.1: Wlan Stations in Validation Scenario

Figure 4.2: Wlan Node Model

4.1.1.4 WIRELESS_LAN_MAC MODULE

The wireless_lan_mac module contains wlan_mac model which implements the scheduling of generated packets as shown in Figure 4.3. All states in this model define stable condition while the simulation model is waiting for the next event to occur. In the propose system, Wireless_lan_mac module first check that whether the sending station is PCF enabled or not. If the station is PCF enabled then it will call wlan_hcca process model otherwise for EDCA traffic it will use wlan_mac_enhanced model.

All the important states and functions of wlan_mac model are defined in next section.

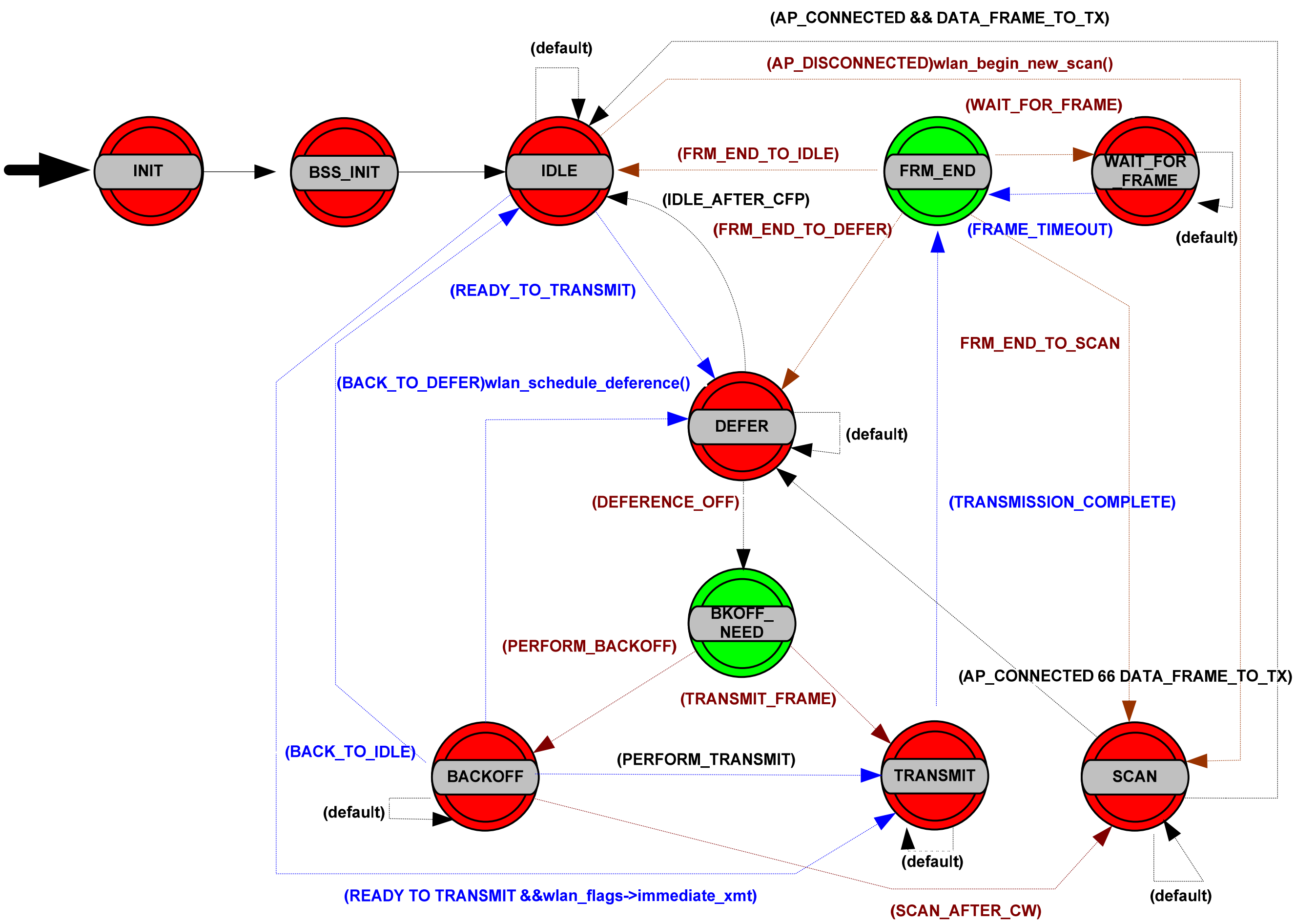


Figure 4.3: WLAN MAC 802.11 Process Model

4.1.2 Process Model

The basic description all states are described here:

- INIT

  All the variables required by the stations are initialized in this state for example initialize state variables, wireless LAN parameters, read model attribute values in variables, determine the final MAC address, create global lists and register statistics handlers etc.

- BSS_INIT:

  All the initializations at the Basic Service Set level are performed in this state. The polling-list for HCCA enabled station is initialized. Calculation of minimum MSI and CAP service interval has also done in this state.

- IDLE:

  The purpose of this state is to wait until the packet has arrived from the higher or lower layer. Empty transmission buffer, no ongoing transmission, no response to send and not presently in Contention Free Period are the typical idle conditions.

- DEFER:

  This state defer until the medium is available for transmission. The typical condition for defer state is: when the transmission buffer is not empty or when there is some response to send or when in CFP then the station has to wait for the medium to be idle if it is busy at present and after getting medium idle, station has to wait for IFS before to start actual transmission.

- BKOFF_NEED:

  Before initiating any frame transmission, a slotted *BackOff Time* is generated randomly from the contention window (CW) to prevent collision. *Backoff time: rand [0,CW] * slot time.* At the first transmission attempt, CW is set equal to a minimum value, CW_{min}. It is doubled after each unsuccessful transmission until reaching a maximum value of CW_{max}. Backoff is decremented by each slot when the medium is sensed idle for that slot. Only when the backoff time reaches zero is the station authorized to access the medium. This state determines whether a backoff is necessary for the frame we are trying to transmit. Backoff is needed when station preparing to transmit frame and discovers that the medium is busy or when the station infers collision. Backoff is not needed when the station is responding to the frame.

  If backoff needed then check whether the station has completed its backoff in the last attempt. If not then resume the backoff from the same point, otherwise generate a new random number for the number of backoff slots.

- TRANSMIT:

  This state mainly deals with packet transmission and collision detection during transmission.

- FRM_END:

  The purpose of this state is to determine the next unforced state after completing transmission.

- WAIT_FOR_FRAME:

  The purpose of this state is to wait for the response after transmission. If station does not receive any response before the waiting time expires, then the station will retransmit that frame.

- SCAN:

  Scan for AP-connectivity in roaming enabled stations

According to IEEE 802.11 standard [1], after the initialization of system parameters, a station would initially be in the idle state waiting for an interrupt to occur. This interrupt can be a stat interrupt i.e. interrupt from the receiver or from the transmitter or this can be a stream interrupt from the lower layer i.e. receiving a packet from another station or from the upper layer i.e. need to transmit a packet to another station or a Contention Free period initialized. For either of these cases the station would go to the defer state and wait until the medium is idle. A CFP initialization interrupt would block a PCF disabled station until the current CFP is finished. A non-AP PCF enabled QSTA would wait to be polled by the AP for transmission opportunity and a AP would initially send a beacon frame after PCF Inter Frame Space (PIFS) which is the minimum duration IFS providing the highest priority access to AP. Network Allocation Vector (NAV) is controlled by beacon interval which shows the duration in which network will be occupied by some station. In the CFP the NAV is equal to the CFP. A non - AP with PCF disabled station which needs to transmit a frame while currently waiting for the completion of the CFP would not transmit until the current CFP is elapsed. As during CFP the station receiving a poll can transmit a frame after a Short Interframe Space (SIFS) thus the next two states Backoff needed and backoff is no more important in PCF. During CFP the AP would transmit a poll frame and a data packet included if there is any data to be sent to the currently polling station as shown in Figure 4.4. After transmission of the poll frame the AP would naturally wait for the response (ACK) until the waiting time expires. This waiting for the response is accomplished in the WAIT-FOR state and the decision regarding whether to wait for a response or not is realized in FRM- END state. If the waiting time expires, then the AP would retransmit an unacknowledged frame after a period of PIFS. A non-AP station which is polled by the AP would be currently waiting in defer state and when receiving a poll frame would transmit a data frame if there is a data to transmit else a null-frame if there is no data to transmit, after a duration of SIFS. After receiving the response frame, the AP would check if the same station needs to be polled again or to poll the next station in the list. The acknowledgement for the previously received frame is piggybacked along with the next poll.

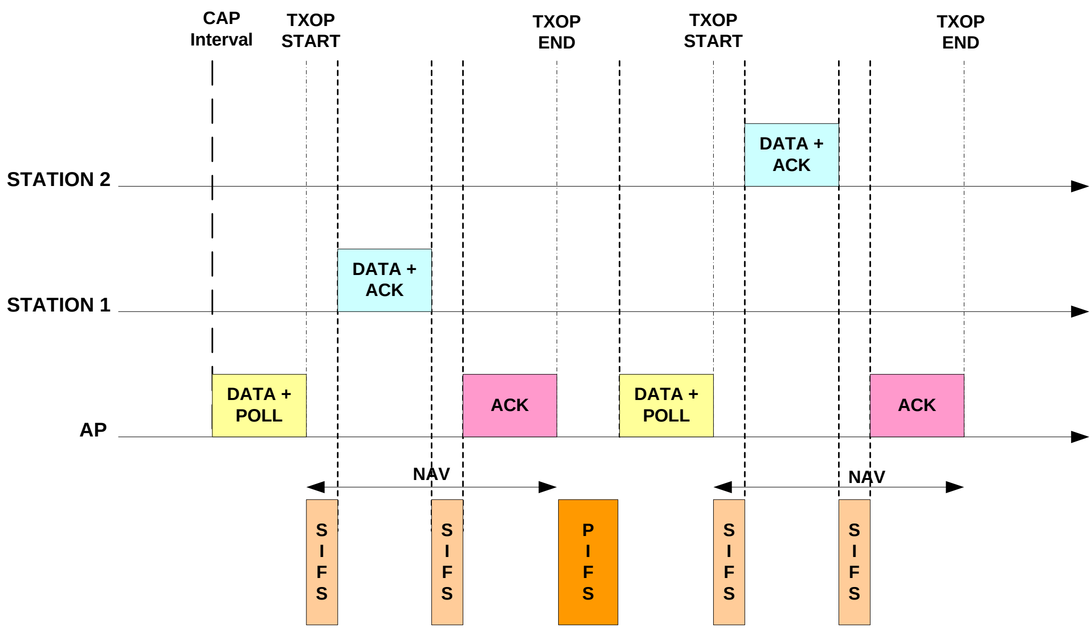


Figure 4.4: Frame Exchange Sequence [30]

The most important functions of the 802.11 MAC process model for the implementation of HCCA functionality is as follows:

- wlan_mac_sv_init (void);
- wlan_interrupts_process (void);
- wlan_higher_layer_data_arrival (void);
- wlan_physical_layer_data_arrival (void);
- wlan_schedule_deference (void);
- wlan_poll_list_member_find ();
- wlan_hlpk_enqueue (Packet* hld_pkptr, int dest_addr, int orig_src_addr,
  int protocol_type, OpT_Packet_Size data_size, Boolean polling);
- wlan_prepare_frame_to_send (WlanT_Mac_Frame_Type frame_type);
- wlan_frame_transmit (void);
- wlan_frame_discard (void);
- wlan_pcf_frame_discard (void);

The INIT state in the state machine call

- wlan_mac_sv_init (void);

This function to initialize all the system parameters as well as registering local and global statistics. After the initialization of all variable it will wait in idle state unitil the packet has arrived from the higher or lower layer. The interrupt handling of the state machine is performed with

- wlan_interrupts_process ();

by setting flags. There can be two types of interrupts stream interrupt which indicates that the packet has arrived either from lower layer or higher layer and stat interrupt which can be from transmitter or receiver.

When the interrupt handler indicates that a packet has arrived from the higher layer the following function

- wlan_higher_layer_data_arrival ();

is called for processing while if interrupt handler indicates that a packet has arrived from the lower layer the following function

- wlan_physical_layer_data_arrival ();

will be called for processing.

- wlan_schedule_deference ()

This routine schedules self interrupt for deference to avoid collision and also deference to observe interframe gap between the frame transmissions.

When a station receives a packet from the higher layer it will first check the type of the packet using function

- wlan_poll_list_member_find ();

that whether it is PCF based or DCF based. If it is PCF based then the packets are queued, using the function

- wlan_hlpk_enqueue (),

into the cfpd_list_ptr and transmitted only during the PCF period. Likewise packets for a DCF destination node will be inserted into the hld_list_ptr and transmitted only during DCF. In order to transmit, first the format of the packet to be sent is constructed using function

- wlan_prepare_frame_to_send()

After that the frame is transmitted using function

- wlan_frame_transmit ()

After the frame has been successfully received at the receiving end it is discarded by calling either of these functions

- wlan_frame_discard () or wlan_pcf_frame_discard ();

according to whether it's a PCF frame or not.

The flowcharts for these procedures are presented in appendix A.

4.2 HCCA Functionality Enhancement

As already mentioned in objectives HCCA functionality has been added by modifying the legacy MAC. The main procedures that need to be taken care of in order to add HCCA functionality enhancement are; admission control mechanism, scheduling mechanism, generation of controlled access phase (CAP) interval, and allocation of TXOP to each registered station and finally HCCA enabled stations can only transmit in CFP not during CP. The next section presents details regarding the appended functionality in more precise.

4.2.1 Admission Control Mechanism

802.11e standard [4] specifies an optional admission control to administer the available bandwidth resources. The 802.11e standard leaves the implementation of admission control on manufacturers. Admission control depends on the implementation of scheduler, available channel capacity, retransmission limits and the scheduling requirements of given stream. According to reference model, the hybrid controller which is present in QAP is used to administer admission control in the network. There are two distinct admission control mechanism one for EDCA or contention based traffic and other one is for HCCA or controlled access mechanism. As the objective of this project is to implement HCCA functionality, our further focus is on controlled access admission control. As a reference design a simple admission control is suggested in 802.11e [4]: Before any data transmission, a traffic stream (TS) is first established. In order to start TS connection, a QSTA send a QoS request frame containing traffic specification (TSPEC) to QAP. A TSPEC described the QoS requirements of TS, such as mean/peak data rate, mean/ maximum frame size, delay bound and maximum service interval. If the access point can fulfill these requirements then it will allow QSTA to start traffic stream and data transmission otherwise it will reject. In the contemporary model, the TSPEC parameters are defined as the model parameters and admission control is theoretically evaluated to determine the scenario. The number of admitted station for a given service interval are theoretically determined according to the service time of each station which is equal to TXOP of that particular station. The conditioning parameter for the number of stations that can be admitted for service by an access point is the maximum service interval (MSI).

4.2.2 Scheduling Mechanism

The QAP scheduler uses the TSPEC parameters for calculating TXOP for each station. On receiving these TSPEC, the QAP scheduler first determine the selected service interval, which should be the highest sub multiple value of beacon interval and also no larger than all MSI required by the different TS's from different QSTA. For the calculation service interval (SI), first QAP scheduler first finds the minimum of MSI for all admitted streams. Second, the scheduler chooses a number lower or equal to selected minimum MSI that is sub multiple of the beacon interval. In the second step, the TXOP duration for a given SI is calculated for admitted streams. For the calculation of TXOP, scheduler first calculate the number of arriving frames (N) during one SI in (4.1) for every traffic stream **i** of a station.

$$N_i = \left\lceil \frac{SI \times p_i}{L_i} \right\rceil \quad i = 1 \ldots n \qquad \textbf{(4.1)[4]}$$

Then the scheduler calculates the TXOP duration as the maximum of N_i frames at R_i and time to transmit one maximal MSDU at R_i plus the additional overhead (MAC header + PHY header) [5]

$$TXOP_j = \sum_{i=1}^{n} \max\left(\frac{N_i \times L_i}{R_i} + O, \frac{M_i}{R_i} + O \right) \qquad \textbf{(4.2)[4]}$$

A similar scheduling mechanism is implemented in the procedure wlan_cap_interval_get (int minimum_msi) which is called in BSS-INIT station and this procedure returns the Maximum Service Interval (MSI) value which dictates the CAP interval.

An example is shown in Figure 4.5 where minimum MSI is 10 ms. the scheduler calculates a Scheduled Service Interval (SI) equal to 10ms using the steps explained above. The TXOP is calculated using equation (4.1) and (4.2). CAPs are constructed after concatenating TXOPs from all stations. The CAP is always repeated after a period equal to the SI value and a length of sum of all TXOPs.

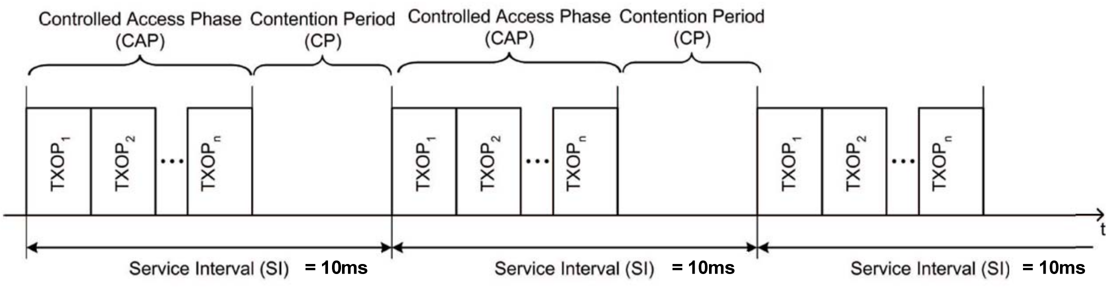


Figure 4.5: A sample TXOP allocation schedule

4.2.3 Controlled Access Phase

The Hybrid Controller periodically generates a controlled access phase in which all the data transmissions are controlled by the HC. The duration of this access phase is determined by the Maximum Service Interval. This controlled access phase is referred by an Interrupt in the model. As mentioned above, the Hybrid Controller can only generate Controlled Access Phase in contention free period. The non-AP QSTA can start its data transmission only in the controlled access phase (CAP). The interrupt for the next CAP interval is scheduled while processing the current interrupt. The interrupt processing sets a flag indicating a CAP interrupt at the AP. This leads to the initiation of the polling mechanism. During the CAP interval all the stations except the polled station sets its NAV for the CAP duration. If there is still time remaining in the CAP Interval after all the stations is polled then the AP would terminate the CAP Interval with a Poll frame providing an opportunity to contention based traffic to start transmission.

4.2.4 HCCA transmission regulation

According to 802.11e standard [4], an HCCA enabled station can only transmit during CAP interval, which is in our implement, only generated during CFP. Due to this fact HCCA enabled stations have to remain in idle state until the HC generates a new CAP interval. The first poll from the AP indicates that a CAP interval is generated and all HCCA enabled stations would wait for their chance in the WAIT state of the process model. Other PCF disabled stations can get data transmission opportunity after the end of CAP duration.

4.2.5 Transmission Opportunity (TXOP)

For contention free traffic, AP sends CF-poll frame to its registered stations as shown in Figure 4.6. These CF-poll frames assign TXOP to each station which defines during which time duration a station can transmit its data. The poll frame set the network allocation vector (NAV) for the station which looks after polled TXOP as shown in figure 4.6 and figure 4.7. Station is not allowed to initiate transmission if its transmission, ACK and expected response frame cannot finish within polled TXOP. The TXOP duration in our implementation is theoretically assigned, which is equal to the time required transmitting a packet of size 70 bytes (including header) and a SIFS time and the time required to transmit an ACK frame. The TXOP control in the present implementation is implemented in the FRM-END state of the process model as the next state after the successful reception is decided in this state.

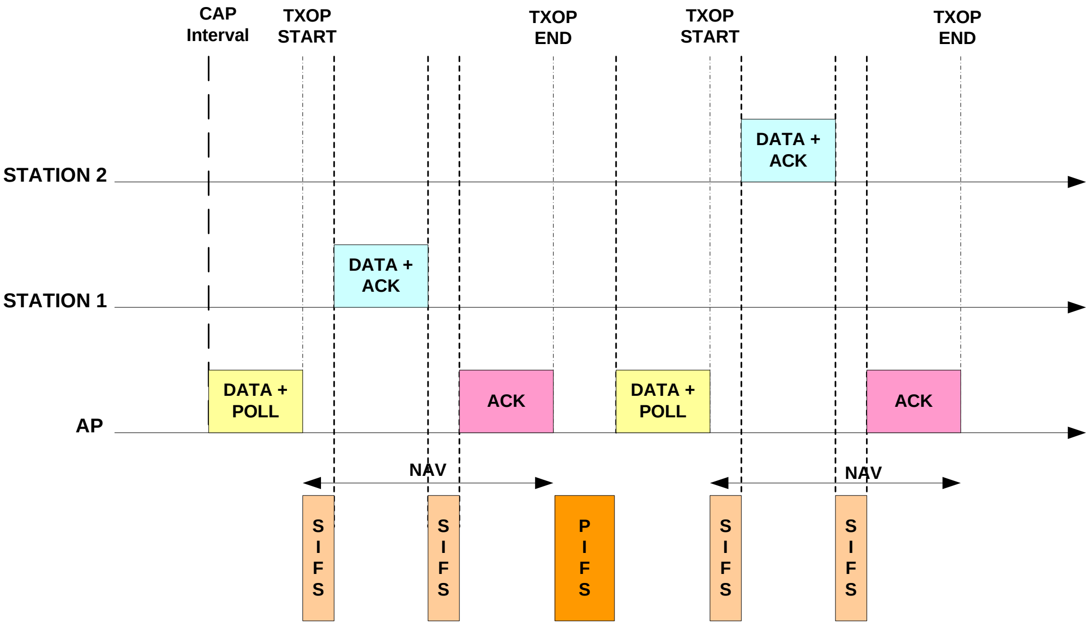


Figure 4.6: Frame Exchange Sequence [30]

Figure 4.7 presents a sequence diagram of TXOP allocation to each QSTA in the implemented model

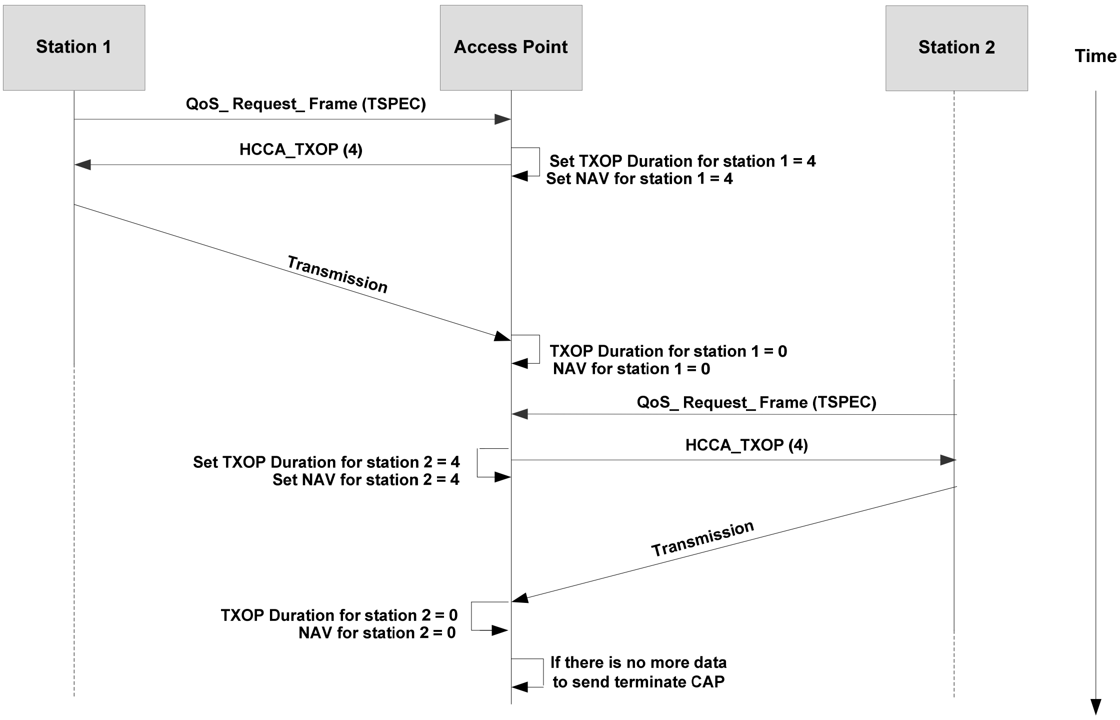


Figure 4.7: TXOP assignment to each QSTA

.

5 Validation

Validation refers to ensuring that the assumptions used in developing the model are reasonable in that, if correctly implemented, the model would produce results close to that observed in real system [26]. The validation of a simulation model is a very important part in simulation project.

There are three different methods to validate a model depending on the availability of the data

1. Expert Intuition
2. Real-Time System Measurement
3. Theoretical Results

In this project the implemented model is validated using Theoretical Results. As defined in [26], a simulation model can be validated using theoretical results obtained from an analytical model of the system under simplified assumptions and with predefined inputs. In such cases, theoretical results and simulation results are then compared to validate the simulation model.

HCCA model is validated with simple scenarios in order to eliminate the randomness for the theoretical results.

5.1 HCCA Model Validation

The basic algorithm of the previously defined HCCA model will be validated by comparing the theoretical results with the simulation results. For the HCCA simulation, the validation scenario consists of 2 QSTA’s and an AP all transmitting the same traffic class. To keep the validation simple, there is no randomness introduced in validation scenario and hence all the parameters are deterministic.

For non-QAP station, an inter arrival times of data frames are set to 10 ms. The QAP should be able to transmit at least one data packet within a single time interval to each station so the inter arrival time for AP is set to 5 ms. The start time parameter for the QAP simulation model is set to 1 ms and 8 ms for both non-QAP models. The simulation is run and the results are analyzed. The downstream and upstream delays for the packet are separately calculated.

The transmission sequence begins with the beginning of the CAP interval as shown in Figure 5.1. Initially the QAP send poll frame along with some data to first station assigning TXOP and allowing it to transmit the data packet and receive ACK within its TXOP. Similarly after the end of first TXOP, QAP would wait for PIFS time and then send a poll frame with some data to second station assigning TXOP and allowing it to transmit the data packet and receive ACK within its TXOP. If there is no station to poll and CAP interval is still is there the QAP would end the CAP interval with a poll to itself and CP follows as shown in Figure 5.2.

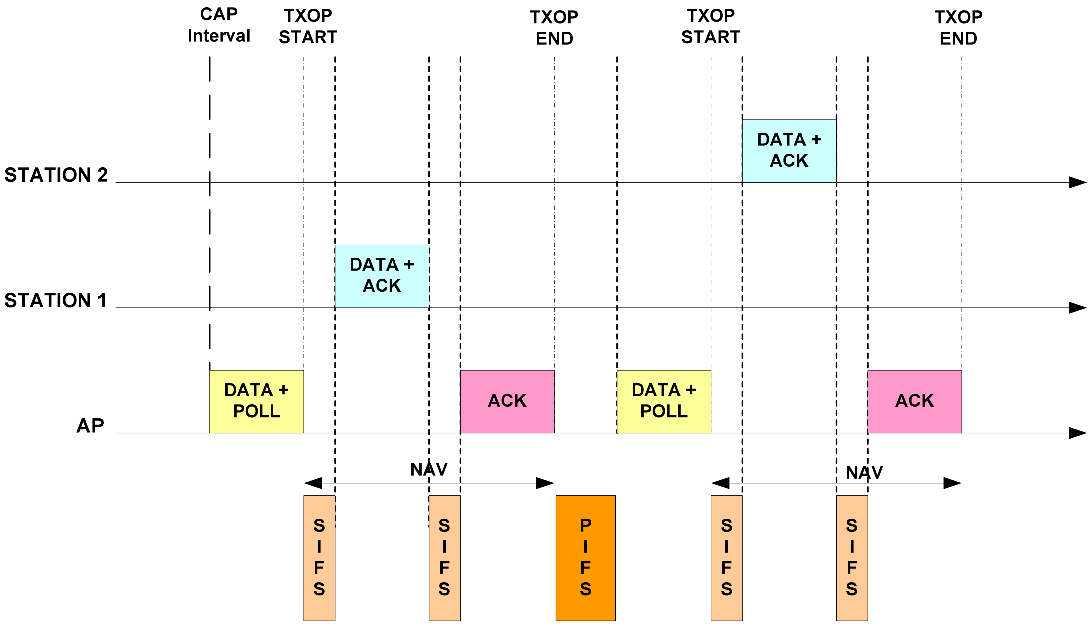


Figure 5.1: Frame Exchange Sequence [30]

$$TXOP_j = 2 \text{ x } SIFS + T_{data}\left(M, r_f\right) + T_{ack}(r) \tag{5.1}$$

Here TXOP is a function of the twice of SIFS, T_{data}, which is the data frame transmission time, and ACK time.

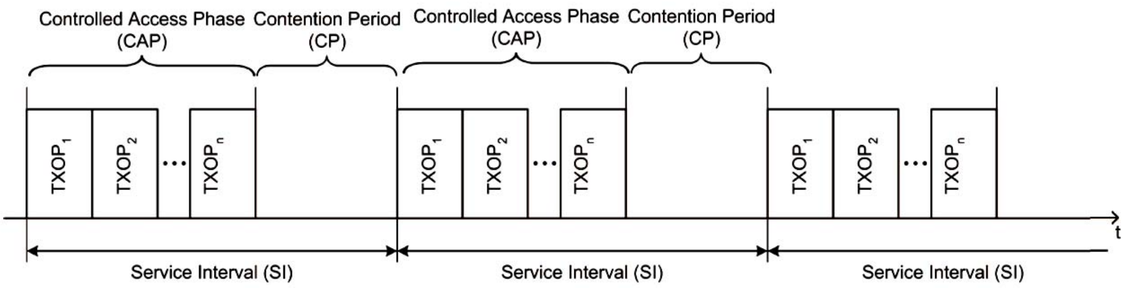


Figure 5.2: A sample TXOP allocation schedule

$$T_{data}\left(M, r_f\right) = \frac{1}{r_{plcp}} \text{ x } \left(PLCP_{preamble} + PLCP_{header}\right) + \frac{1}{r_f} \text{ x } \left(\frac{1}{r_f} \text{ x } 8\text{bit}\right) \tag{5.2}$$

The frame transmission duration is calculated using (5.2) where $PLCP_{preamble}$ and $PLCP_{header}$ is the overhead of the physical layer and r_f is its transmission rate.

The total delay for upstream and downstream traffic is calculated separately using equation 5.3 and equation 5.4.

$$\begin{aligned} d_{upstream}(M,r) = {} & WaitTime + (n-1)x \\ & PIFS + n \; x \; T_{poll}(M,r_f) \\ & + (n-1)TXOP \end{aligned} \tag{5.3}$$

$$\begin{aligned} d_{downstream}(M,r) = {} & WaitTime + (n)x \\ & PIFS + n \; x \; T_{poll}(M,r_f) + \\ & (n-1)TXOP \; (n)x \; SIFS \end{aligned} \tag{5.4}$$

Where WaitTime is equal to the time duration a packet has to wait for a consecutive CAP interval and n is the rank of the station in the polling sequence and SIFS = 10us, PIFS = 30us, SlotTime = 20us and T_{data} = T_{poll} = 241.45us and T_{ack} = 202.18us. In case of beacon frame transmission the beacon transmission delay will be added to the total delay. For these values and formulas (5.1, 5.2, 5.3, 5.4) we refer [30]. The values analytically calculated with equation 5.3 and 5.4 almost match the corresponding simulation results which are shown in Figure 5.3 and Figure 5.4. Therefore model is validated.

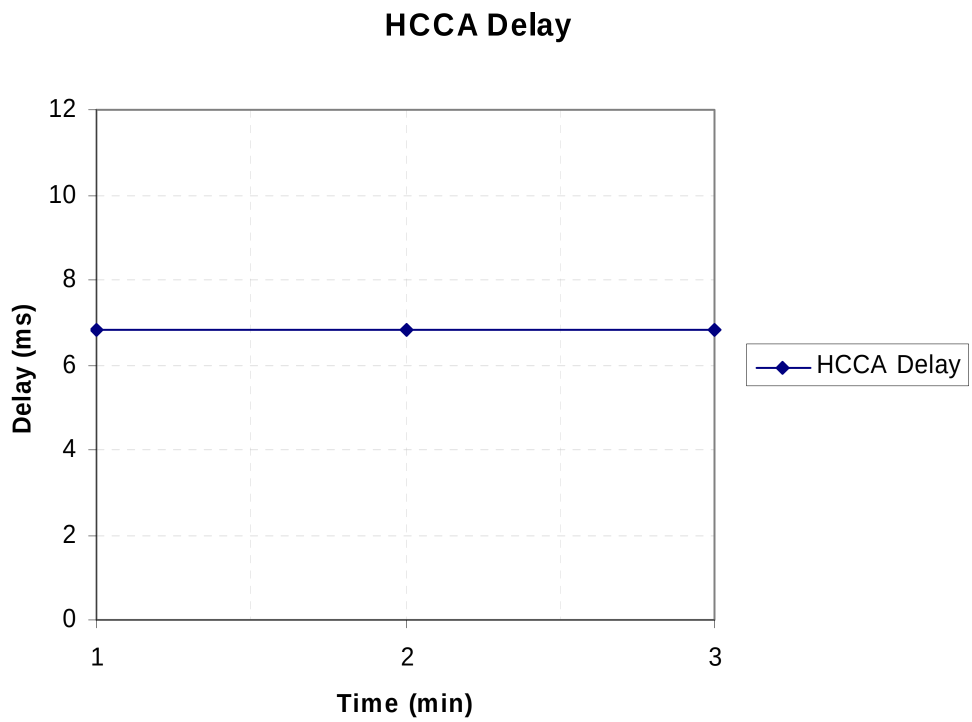


Figure 5.3: HCCA Upstream Delay

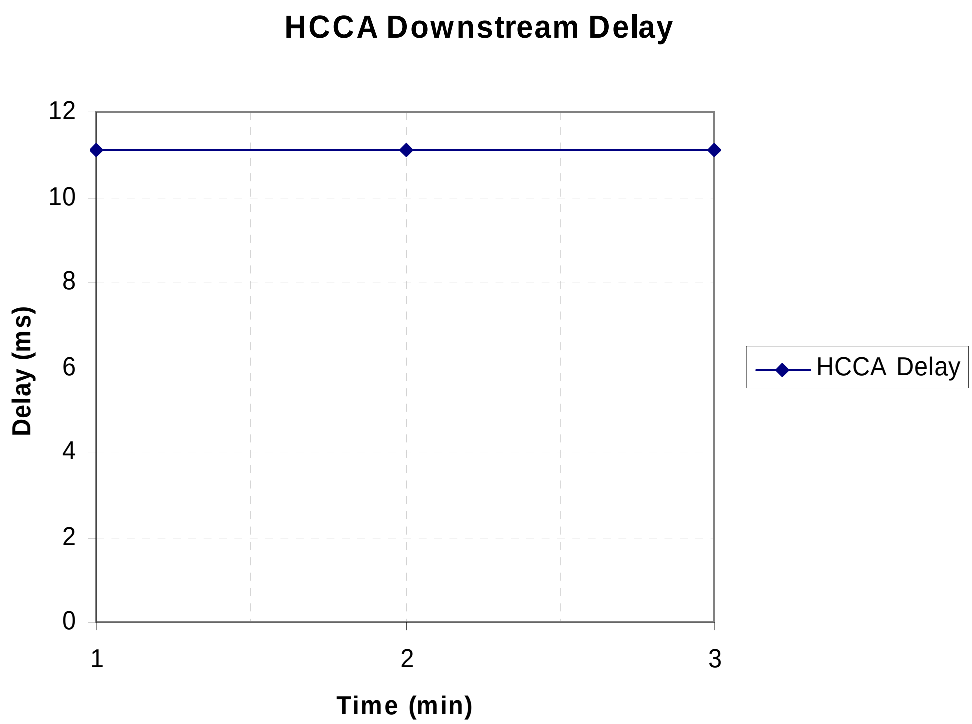


Figure 5.4: HCCA Downstream Delay

6 Scenarios and Results

In this simulation case study we mainly look up for the viability of different 802.11e access schemes for its deployment in deterministic industrial automation systems. This chapter presents an analysis and results of HCCA performance in an industrial environment and compares those results with other 802.11e access scheme EDCA to prove that HCCA outperformed the EDCA. For this purpose 2 different scenarios are considered and are explained in next section. Results of these scenarios are explained and presented with plots.

6.1 Simulation Scenarios

In an industrial automation environment programmable logic controller (PLC) and I/O nodes are typical communication elements exchanging process data via real-time Ethernet protocol at periodic interval of time, which is application dependent [12]. In this simulation case study we consider two scenarios; in the first scenario only real-time traffic is considered while in the second scenario both real-time and non real-time traffic are considered. Figure 6.1 shows the first scenario where a PLC is connected directly with QAP through real-time Ethernet connection and generates cyclic data in order to update the output of remote I/O devices. The I/O nodes send cyclic update data of their inputs via the WLAN client to the PLC. The frame sizes in both scenarios are constant and equal to 40 bytes. The traffic flow from node to PLC is defined as upstream and from PLC to node as downstream. Both of scenarios are set up with QoS-enabled wireless stations and are configured to work in infrastructure mode. Furthermore all the included stations remain stationary during the whole simulation. The physical layer consisted of the OPNET 802.11b PHY module with a maximum data rate of 11 Mbps and a normal line-of-sight link. The transmitted real-time frames are always of the highest priority and the corresponding EDCA access parameters (AIFS, CWmin, CWmax) were set to the default values suggested in [4].

In the first scenario, the number of clients is altered in each simulation between 2 to 80 for EDCA and 2 to 100 for HCCA. As we know that Maximum Service Interval (MSI) is the conditioning factor for the chosen service interval (SI) which is the main constrained in the maximum number of clients in HCCA, so the value of MSI for different number of stations has changed according to table 6.1

Maximum Number of Clients	8	25	50	100
Maximum Service Interval	10 ms	25 ms	50 ms	100 ms

Table 6.1: Number of Clients and Maximum Service Interval

The maximum number of stations that can be employed for service interval is taken from [30]. The statistics of interest in this scenario are the maximum delays as a function of the number of the stations.

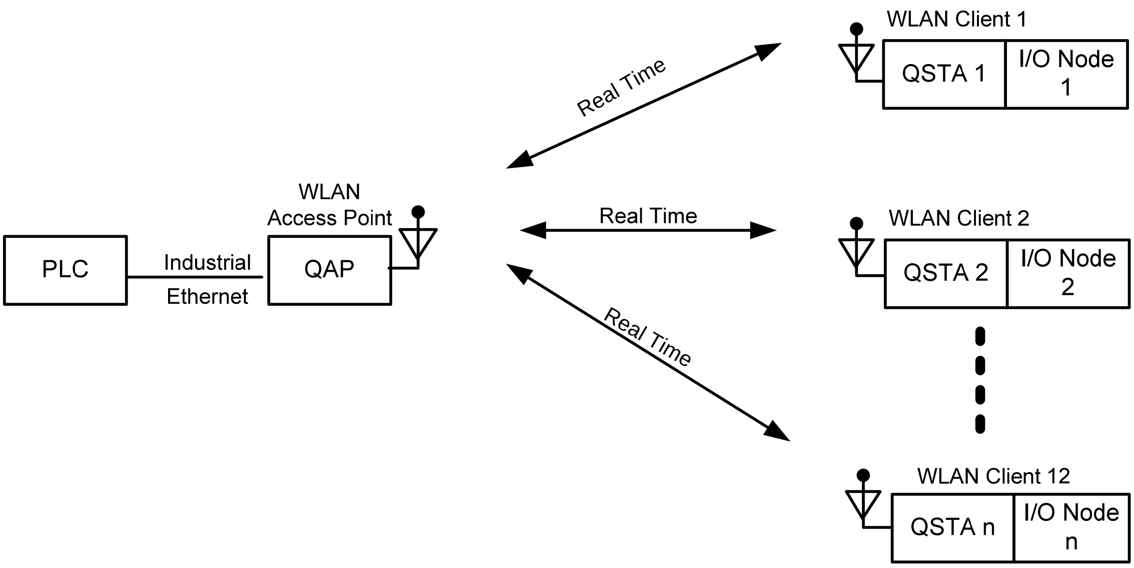


Figure 6.1: Scenario with a varying number of clients

Figure 6.2 shows the second scenario where a PLC is connected directly with QAP through real-time Ethernet connection and generates cyclic data in order to update the output of remote I/O devices. The I/O nodes send cyclic update data of their inputs via the WLAN client to the PLC. The numbers of real-time clients are fixed to 12 while two additional non real-time (Best Effort) nodes are also considered. These non real-time nodes exchange best effort (BE) data of different size with file server using AP. Therefore this best effort data is considered as source and these nodes as traffic generator for AP. The maximum service interval for real-time traffic remains constant as 20 ms. The obtained results show the isolation of real-time and best effort traffic, as well as best effort throughput, both under different network load conditions.

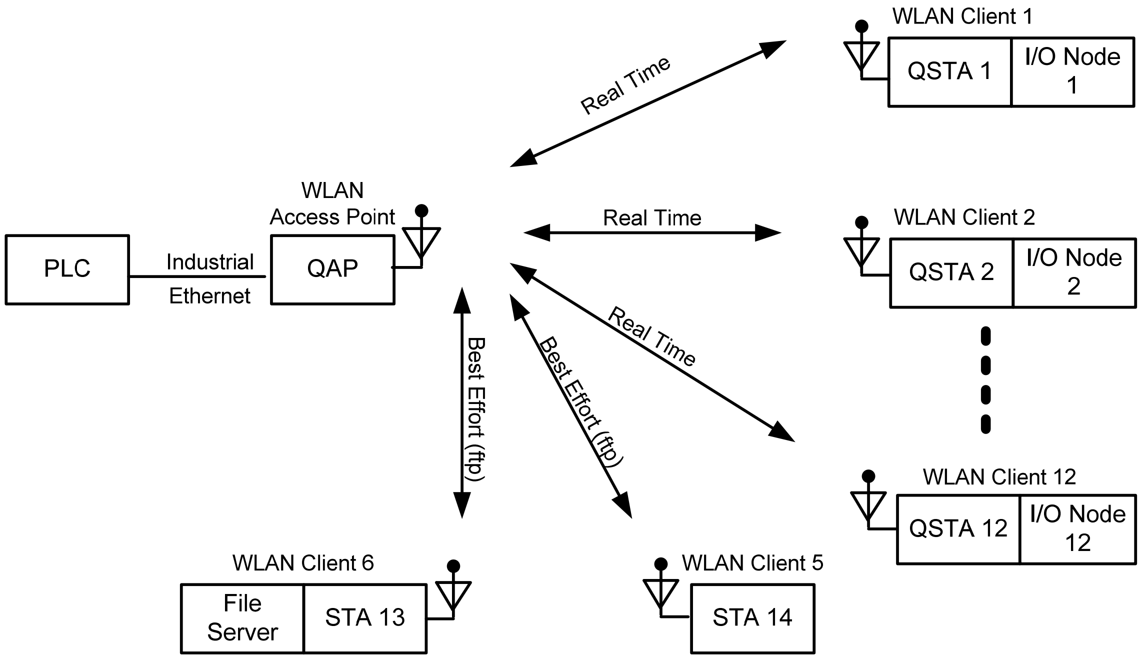


Figure 6.2: Scenario with additional best-effort traffic

6.2 Results

This section provides the comparison of results of the HCCA and EDCA mechanism with respect to their upstream (traffic flow from node to PLC) and downstream (traffic flow from PLC to node) delays. These upstream and downstream delays represents the end to end delay of all data packets that are successfully received by the WLAN MAC and forwarded to higher layer. As already defined in introduction that industrial communication has rigorous requirements on quality of service (QoS) such as delay, so in this evaluation maximum delay values are considered which includes queuing and medium access delays at the source MAC and reception of all fragments and relaying the frame via the AP .

Detailed results are appended in Appendix B

6.2.1 First Scenario

The results of first scenario are shown in Figure 6.3 and Figure 6.4. Figure 6.3 shows the maximum delay, in the upstream direction, depending on the number of stations that can be reliably employed for a particular service interval.

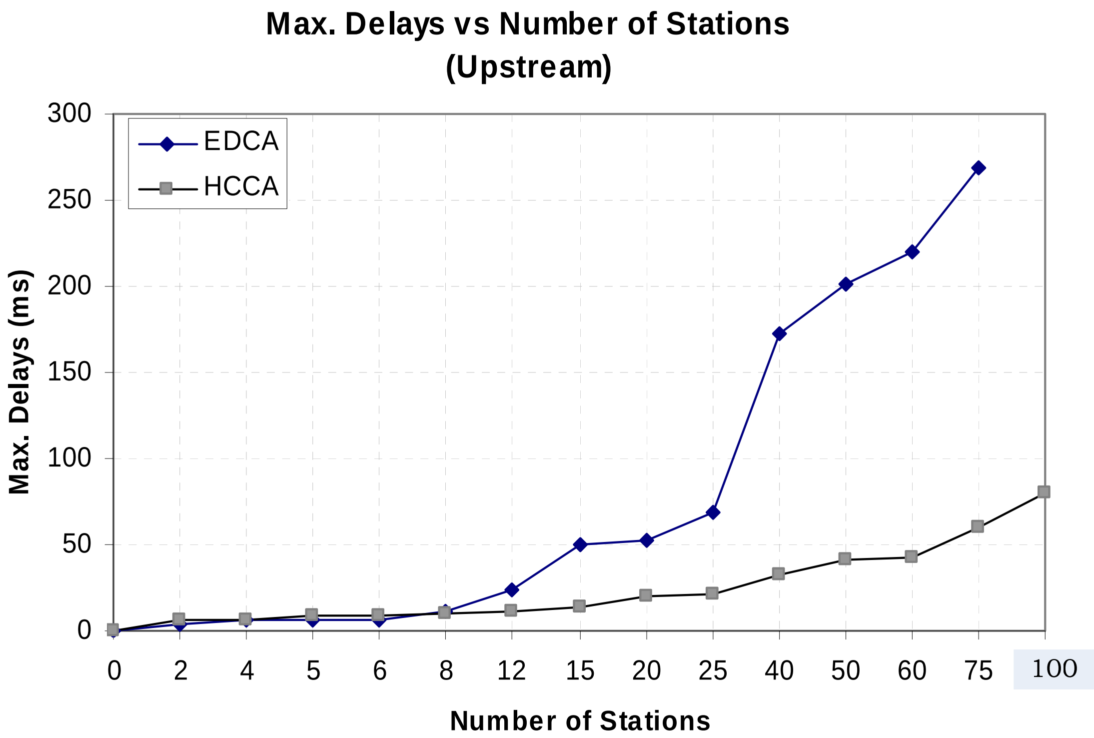


Figure 6.3: Maximum Delay vs. Number of Clients (Upstream)

It can be seen that the HCCA outperformed the EDCA. The HCCA maintained its parameterized bounds whereas the EDCA only fulfills the requirements in setups with a few number of clients. For instance, when 50 stations are active with a corresponding service interval of 50 the maximum HCCA delay is 40.64 ms while the maximum EDCA delay is 201ms. The number of stations that can be employed for a given delay bound is also larger with the HCCA mechanism, e.g. for a delay bound of 100ms you can have up 100 stations with HCCA and only approx. 30 with EDCA.

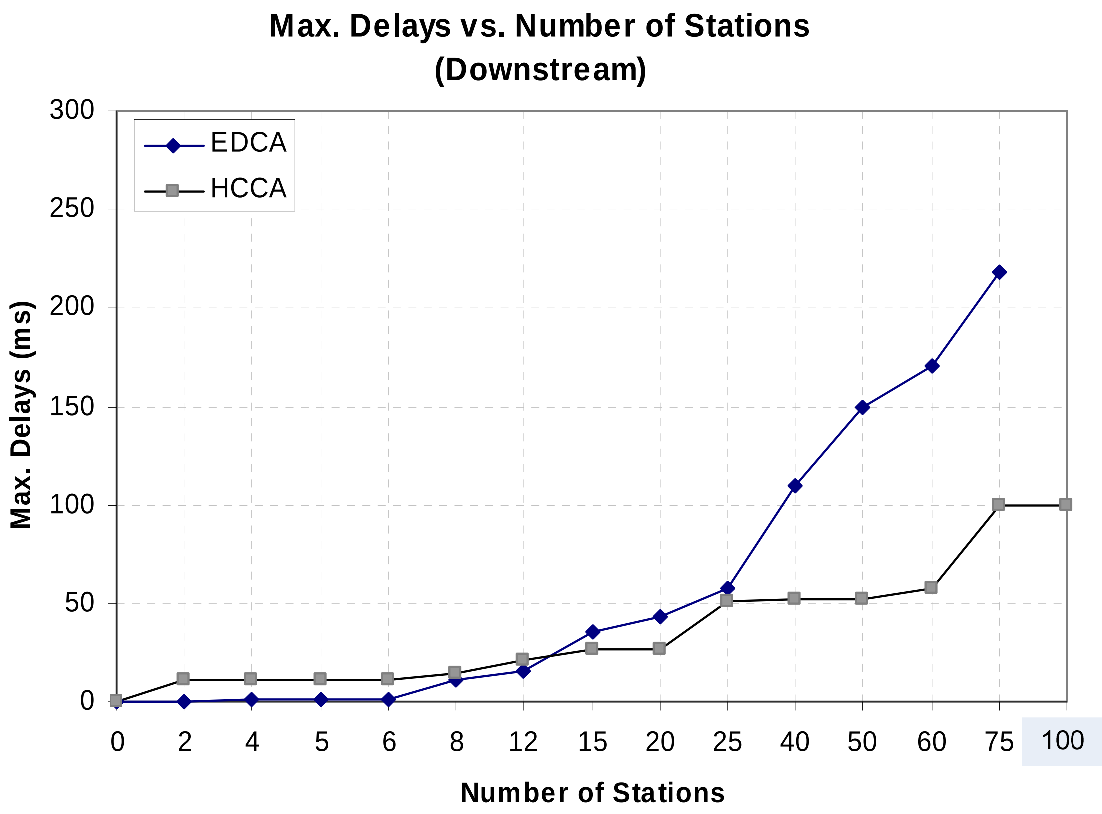


Figure 6.4: Maximum Delay vs. Number of Clients (Downstream)

Figure 6.4 shows the maximum delays for the HCCA and the EDCA in the downstream direction. It can be seen that EDCA is more feasible for small number of stations while as the number of stations increases an improvement of HCCA performance can be observed. The gap between HCCA and EDCA expands dramatically as the number of stations exceeds 25. The further observation of the graph shows that the EDCA delay values are almost twice as much as the HCCA delays.

6.2.2 Second Scenario

In the second scenario we considered best-effort traffic in the network in order to evaluate the effect of low priority traffic on the real-time traffic. Similar to the first scenario the delay statistics for HCCA and EDCA are compared in terms of upstream and downstream delays. It is expected that the best-effort traffic would not affect the real-time traffic at all with the HCCA as there is isolation between real-time traffic and best-effort traffic. The throughput of HCCA for the best effort traffic might initially increase until a threshold value is reached and then decrease due to more collisions.

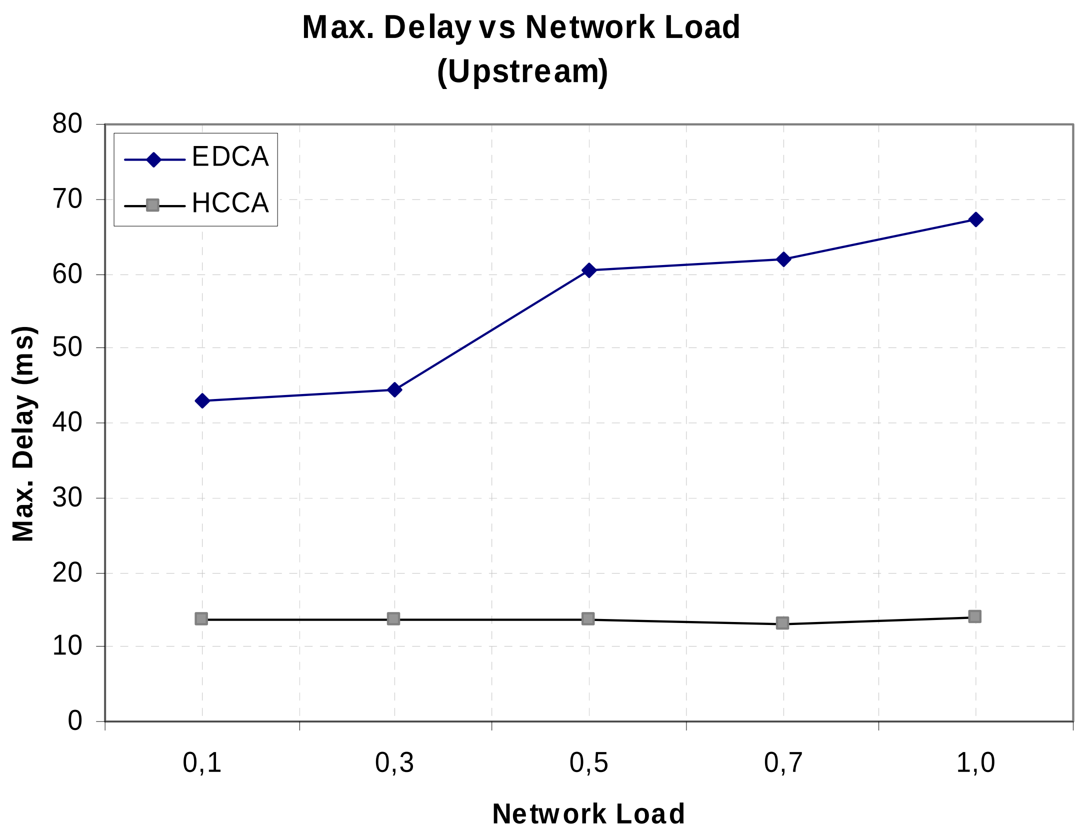


Figure 6.5: Maximum Delay vs. Network Load (Upstream)

Figure 6.5 shows the maximum delays depending on the network load. From here we can see that when we use HCCA there is a clear isolation between real-time traffic and non real-time traffic and the increasing network load due to the best effort traffic has no significance effect on the maximum delays whereas with EDCA the delay increased with the network load. The upstream delays of EDCA are more then twice of HCCA because in case of EDCA all stations are trying to transmit at the same time competing with the BE traffic and other high priority traffic.

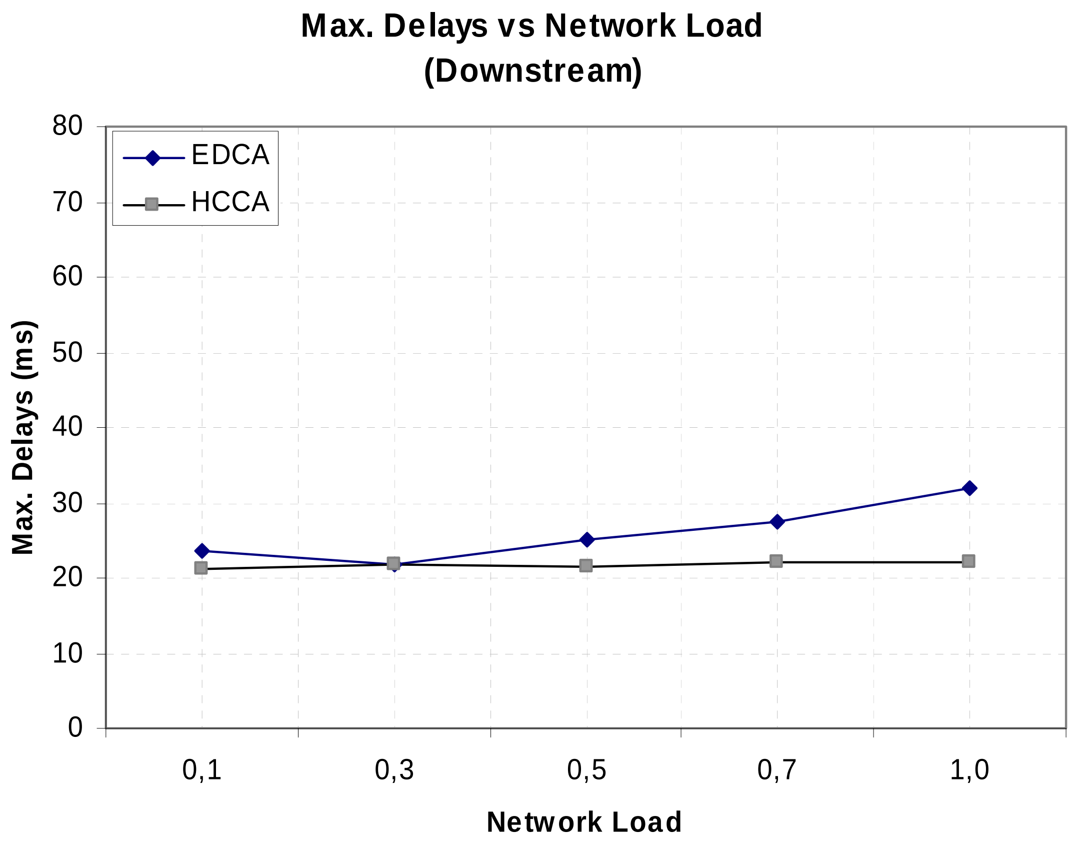


Figure 6.6: Maximum Delay vs. Network Load (Downstream)

Figure 6.6 shows the downstream traffic where the difference between HCCA delay and EDCA delay has decreased because here with EDCA the AP has to compete mostly with the BE traffic and rarely with high priority traffic. At some point it can be observed that the delay decreased even though the load was raised. This is due to the selected representation of the results as maximum values and their rare occurrence.

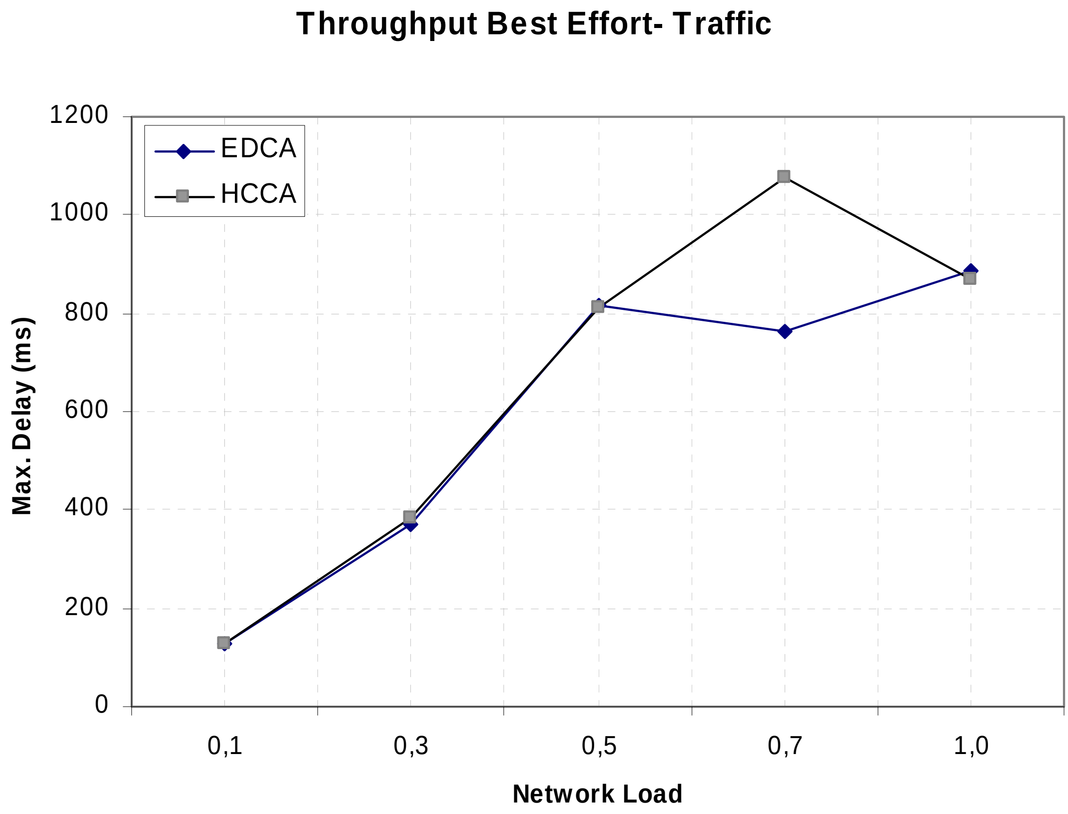


Figure 6.7: Avg. Throughput of BE Traffic

Figure 6.7 shows the throughput of BE with EDCA and HCCA in which the graph shows the average throughput as a function of the network load. The throughput in terms of OPNET is the number of packets that are successfully received at the receiving end calculated as bit/s. In case of EDCA, the throughput increases initially with the network load but after a value of 50% network load starts decreasing. The throughput of HCCA for the best effort traffic initially increases until a threshold value is reached and then decrease due to more collisions. The maximum throughput value for HCCA is at 70% which shows the efficiency of HCCA with BE traffic as well. .

7 Conclusions and Future Work

This master thesis evaluates the performance of HCCA in industrial environments and compares these results with EDCA by means of a simulation case study. Moreover the basic principles of IEEE 802.11, the 802.11e amendment and the HCCA implementation for OPNET have been discussed in detail.

The results from the first scenario prove that there can be more stations accommodated in an HCCA environment compared with EDCA, because of its highly deterministic polling mechanism. It can be seen that as the number of stations in EDCA, transmitting frames with the same priority, increments it leads to a raise in the corresponding channel load. In contrast to this statement, the EDCA is more feasible and efficient for setups having a small amount of clients. According to the results achieved, it can be concluded that the HCCA is suitable for provision of real-time guarantees for certain industrial applications.

The results from the second scenario show that HCCA deployment causes a clear isolation between real time and non real time traffic, i.e. the increased network load due to a best effort traffic load does not affect the delay parameters of real time traffic. The reason for this isolation is the CAP period, which separates real time and non real time traffic. From another point of view considering best effort traffic, the throughput plot elucidates that the best effort throughput is higher for HCCA with channel load larger than 50%. This clearly demonstrates that the HCCA mechanism provides a better service to best effort data providing a higher throughput than EDCA.

The future work might include the implementation of an admission control mechanism for HCCA. The scheduling algorithm for the HCCA in the present model is quite simple. For a better performance, a more efficient scheduling algorithm should be taken into consideration. Furthermore, we will take different channel conditions into consideration, e.g. a time variant channel due to movements and multipath environments, to have a more realistic setup.

References

[1] ISO/IEC 8802-11: 1999, \Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications. Standard reaffirmed in june 2003", 1999.

[2] OPNET, University Program [Online], (Oct. 2005). Available at: http://www.opnet.com/services/university

[3] Y. Pourmohammadi Fallah, H. Alnuweiri, A controlled-access scheduling mechanism for QoS provisioning in: IEEE 802.11e Wireless LANs ACM international workshop on Quality of Service and Security in Wireless and Mobile Networks Q2SWinet ’05, pp. 122–129.

[4] IEEE P802.11e/D13.0-2005, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications. Amendment: Medium access control (MAC) Quality of Service enhancements", 2005.

[5] C. Cicconetti *, L. Lenzini, E. Mingozzi, Giovanni Stea, Design and performance analysis of the Real-Time HCCA scheduler for IEEE 802.11e WLANs.

[6] Z.A. Muhamad, N.T. Suzuki, S. Tasaka, A Packet Scheduling Scheme for Audio-Video Transmission with IEEE 802.lle HCCA and its Application-Level QoS Assessment.

[7] Qiang Ni, Performance Analysis and Enhancements for IEEE 802.11e Wireless Networks

[8] A. Grilo, M. Macedo, and M. Nunes, "A Scheduling Algorithm for QoS Support in IEEE 802.1 le Networks," IEEE Wireless Communications, pp. 36-43, Jun. 2003.

[9] N. Ramos, D. Panigrahi, and S. Dey, "Dynamic Adaptation Policies to Improve Quality of Service of Multimedia Applications in WLAN Networks," in Proc. Int'l. Wksp. Broadband Wireless Multimedia, San - Jose, CA, 2003.

[10] D. Gao, J. Cai, and K.N Ngan, "Admission Control in IEEE 802.lle Wireless LANs," IEEE Networks, Jul/Aug. 2005.

[11] W.F Fan, Danny H.K. Tsang, and B. Bensaou, "Admission Control for Variable Bit Rate Traffic in IEEE 802.1 le WLANs," in Proc. IEEE LANMAN '04, Mill Valley, CA, 2004.

[12] N. Krommenacker and V. Lecuire, Building Industrial Communication Systems based on IEEE 802.11g wireless technology", 2005.

[13] C. Casetti, C.-F. Chiasserini, M. Fiore, and M. Garetto, Notes on the inefficiency of 802.11e HCCA", September 2005.

[14] J. Jasperneite and P. Neumann, Measurement, Analysis and Modelling of Real-Time Source Data Traffic in Factory Communication Systems", in IEEE 3rd Int. Workshop on Factory Communication Systems (WFCS '00), Sept. 2000, pp. 327-334.

[15] K. Sugun Pradeep, H. Trsek, and J. Jasperneite, Potential of the HCCA scheme defined in IEEE 802.11e for QoS enabled Industrial Wireless Networks", Master Thesis 2006.

[16] M. Felser, Real-time Ethernet - Industry Prospective", volume 93, June 2005, pp. 1118-1129.

[17] http://www.isa.org/Content/ContentGroups/InTech2/Departments/Industry_Views/200411/Wireless_meshes_with_industrial_automation.htm

[18] 802.11 Wireless Network – The Definitive Guide by Matthew S. Gast

[19] S. Choi et al., "IEEE 802.11e Contention-based Channel Access (EDCF) performance Evaluation," Proc. IEEE ICC, Anchorage, AK, May 2003.

[20] S. Mangold et al., "IEEE 802.11e Wireless LAN for Quality of Service," Proc. Euro. Wireless, Florence, Italy, Feb. 2002

[21] Q. Ni, L. Romdhani, and T. Turletti, "A Survey of QoS Enhancements for IEEE 802.11 Wireless LAN," Wiley J. Wireless and Mobile Comp., vol. 4, no. 5, Aug. 2004, pp. 547–66.

[22] http://grouper.ieee.org/groups/802/11/

[23] Analysis of IEEE 802.11e for QoS support in Wireless LANs by Stefan Mangold, Philips Research Sunghyun Choi, Seoul National University Guido R, Ole and Bernhard, Aachen University.

[24] Data Communication and Networks by Williams Stallings 6 Edition.

[25] http://en.wikipedia.org/wiki/Realtime_system

[26] The Art of Computer System Performance Analysis by Raj Jain 1991

[27] H.Trsek, "Investigation of IEEE 802.11e standard and its performance in an industrial environment based on a wireless lan access point with embedded linux", Master Thesis Sept.2005.

[28] http://en.wikipedia.org/wiki/Simulation#Engineering_.28Technology.29_simulation_or_Process_simulation

[29] OPNET Tutorials

[30] S.P.Karanam, "Potential of HCCA scheme defined in IEEE 802.11e for QoS enabled Industrial Wireless Networks", Master Thesis 2006.

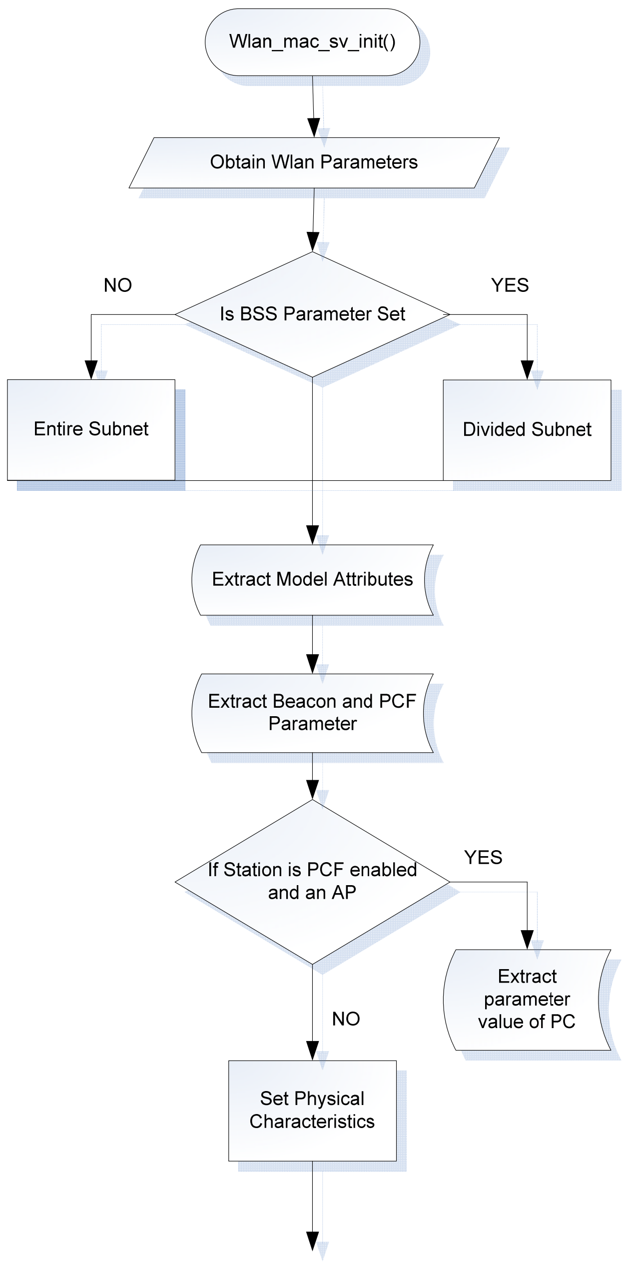
Abbreviations

ACK	Acknowledge Frame
AIFS	Arbitration Interframe Space
AIFSN	Arbitration Interframe Space Number
AP	Access Point
BSS	Basic Service Set
CFP	Contention Free Period
CP	Contention Period
DCF	Distributed Coordination Function
DSSS	Direct Sequence Spread Spectrum
EDCA	Enhanced Distributed Channel Access
EDCF	Enhanced Distributed Coordination Function
ESS	Extended Basic Service Set
ESSID	Extended Service Set Identifier
FTP	File Transfer Protocol
HCCA	HCF Control Channel Access
HCF	Hybrid Coordination Function
IBSS	Independent Network
LLC	Logical Link Control
MAC	Medium Access Control Layer
MPDU	MAC Protocol Data Unit
MSDU	MAC Service Data Unit
OSI	Open Systems Interconnection
PC	Point Coordinator
PCF	Point Coordination Function
PDF	Probability Density Function
PHY	Physical Layer
PLC	Programmable Logic Controller
PLCP	Physical Layer Convergence Protocol

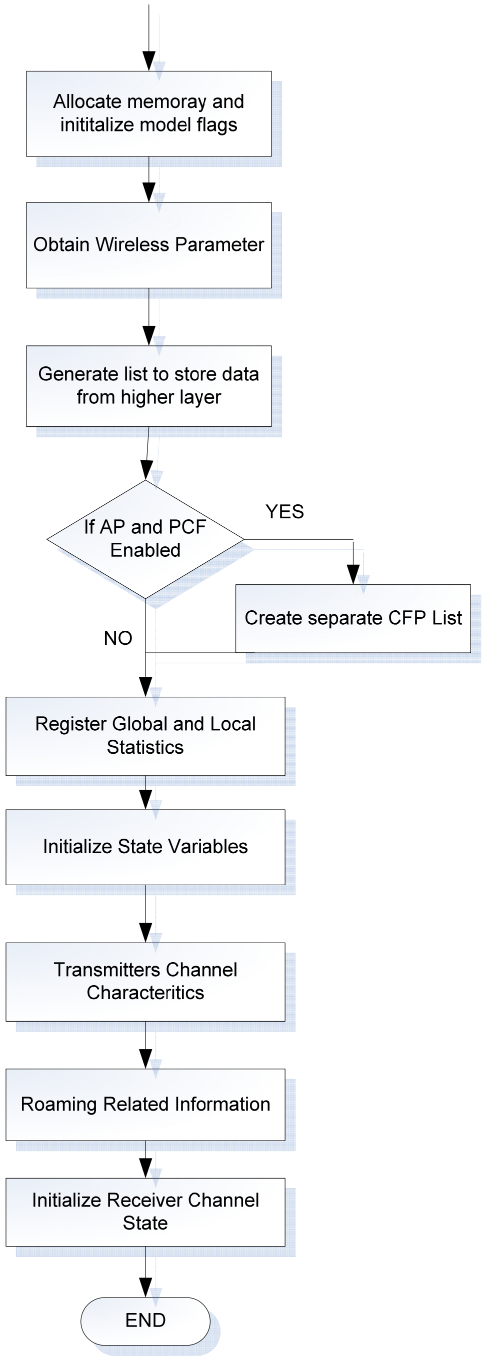
PMD	Physical Medium Dependent
PPDU	PLCP Protocol Data Unit
PSDU	PLCP Service Data Unit
QoS	Quality of Service
QSTA	QoS Enabled Station
SAP	Service Access Point
STA	Station
TSPEC	Traffic Specification
TXOP	Traffic Oppotunity
WLAN	Wireless Local Area Network
VOIP	Voice over IP

Appendix A: Flow Charts

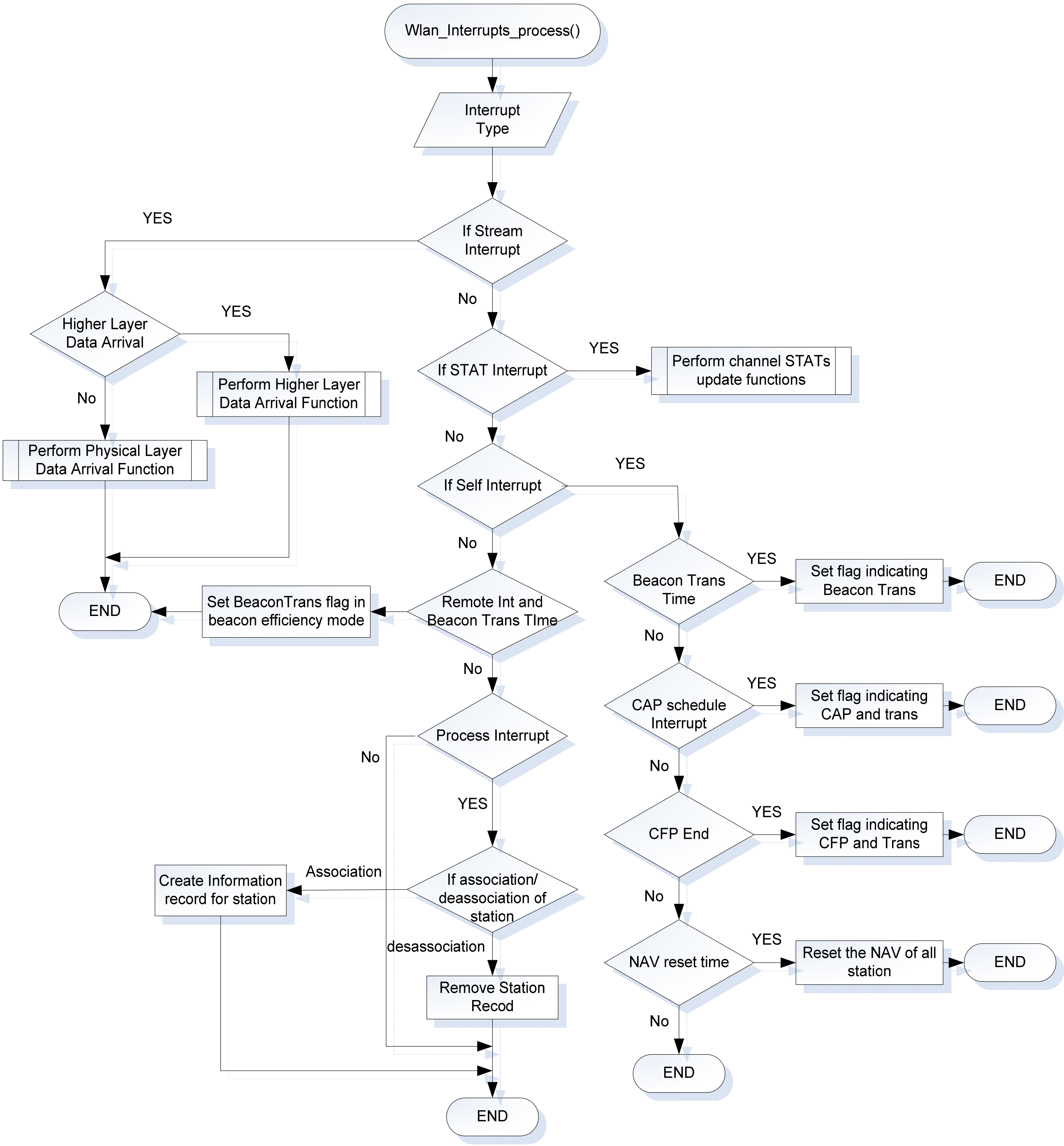
Initialize State Variables



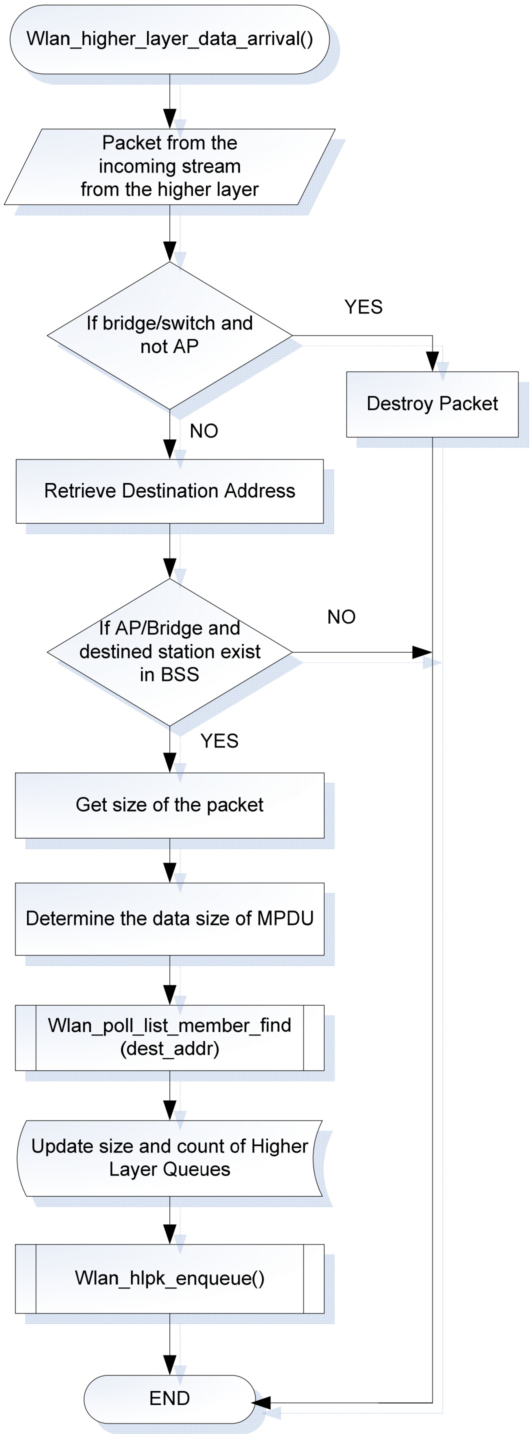
Initialize State Variables Cont..



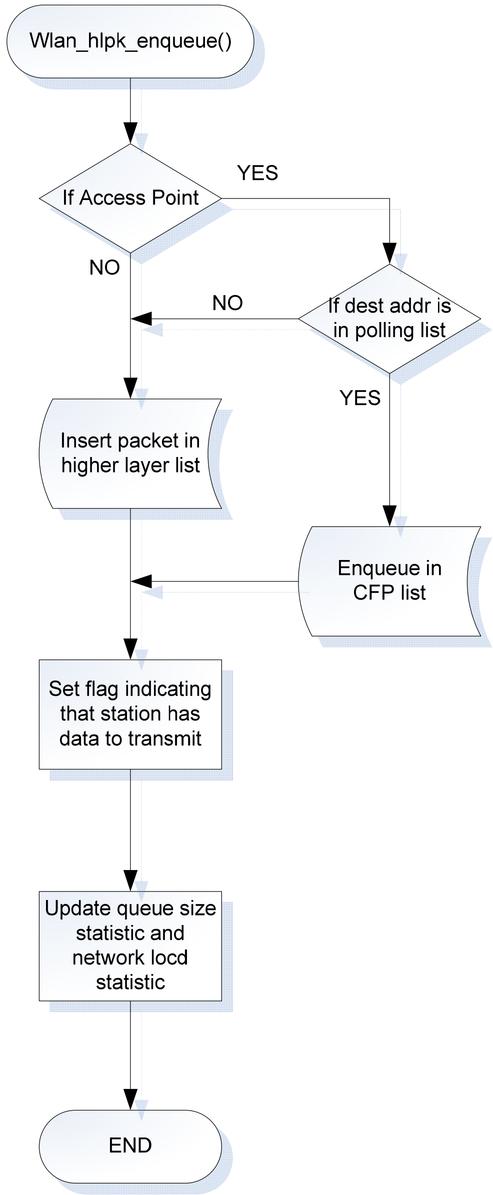
Interrupts Process



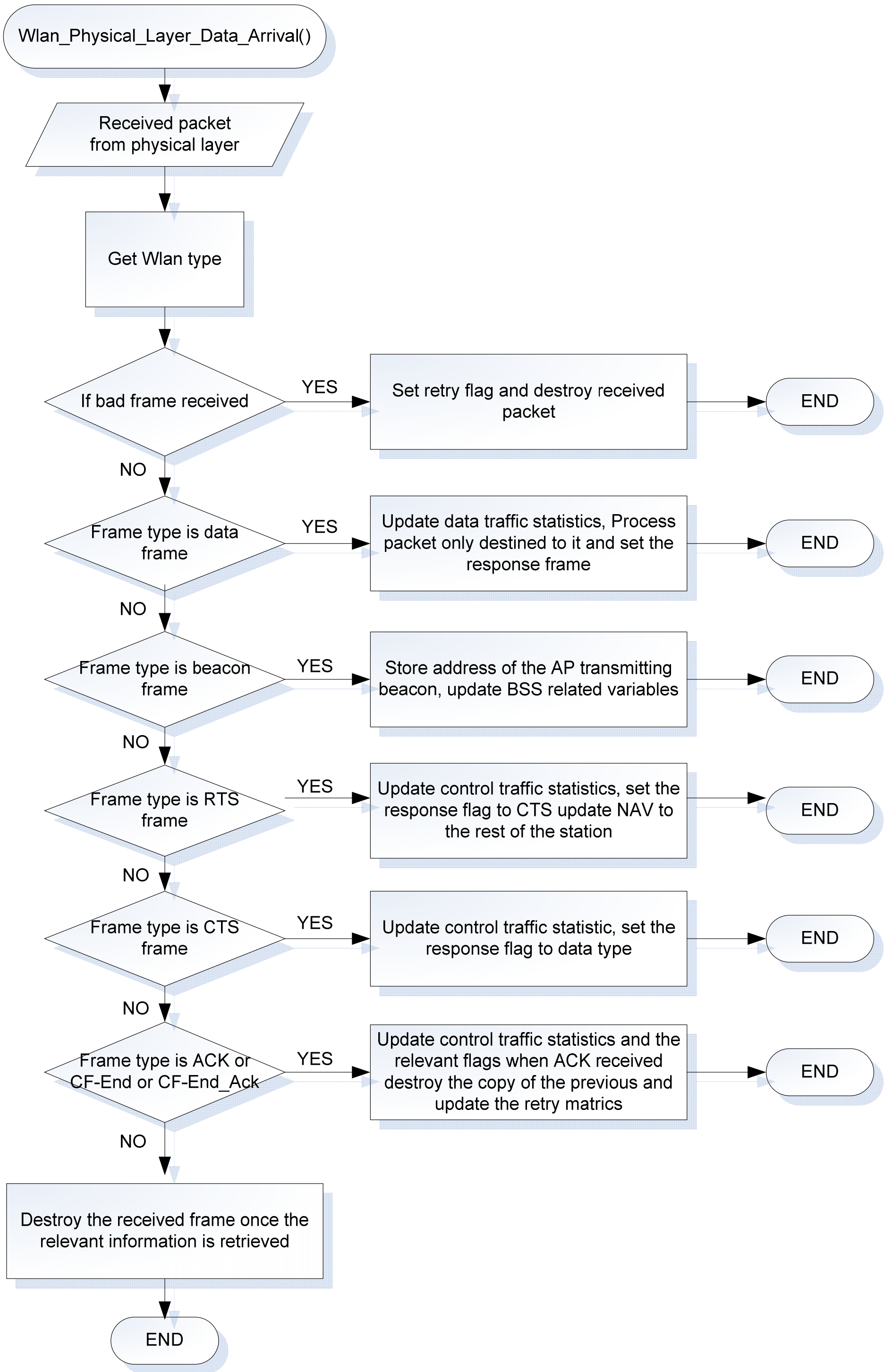
Higher Layer Data Arrival



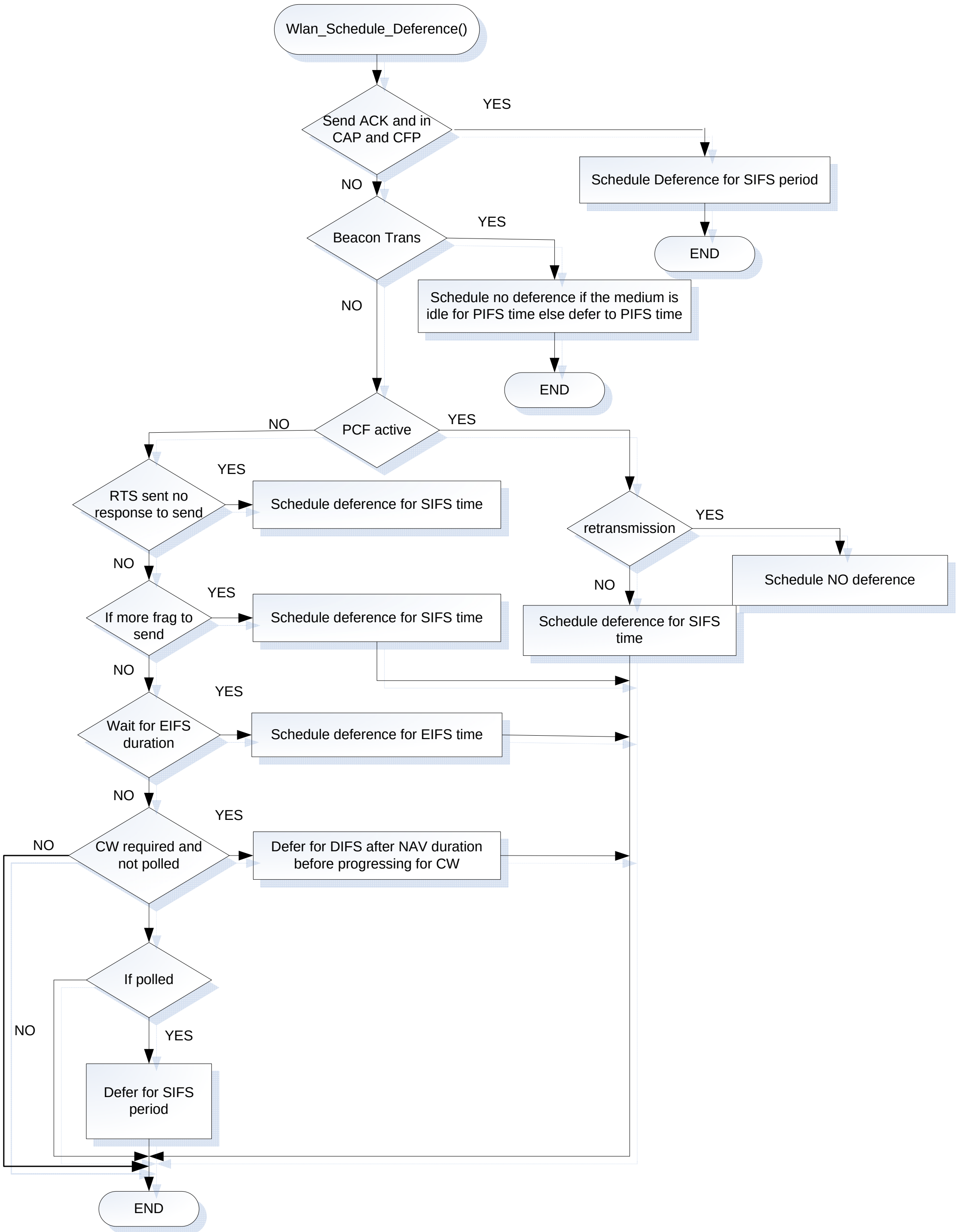
Higher layer Packet Enqueue



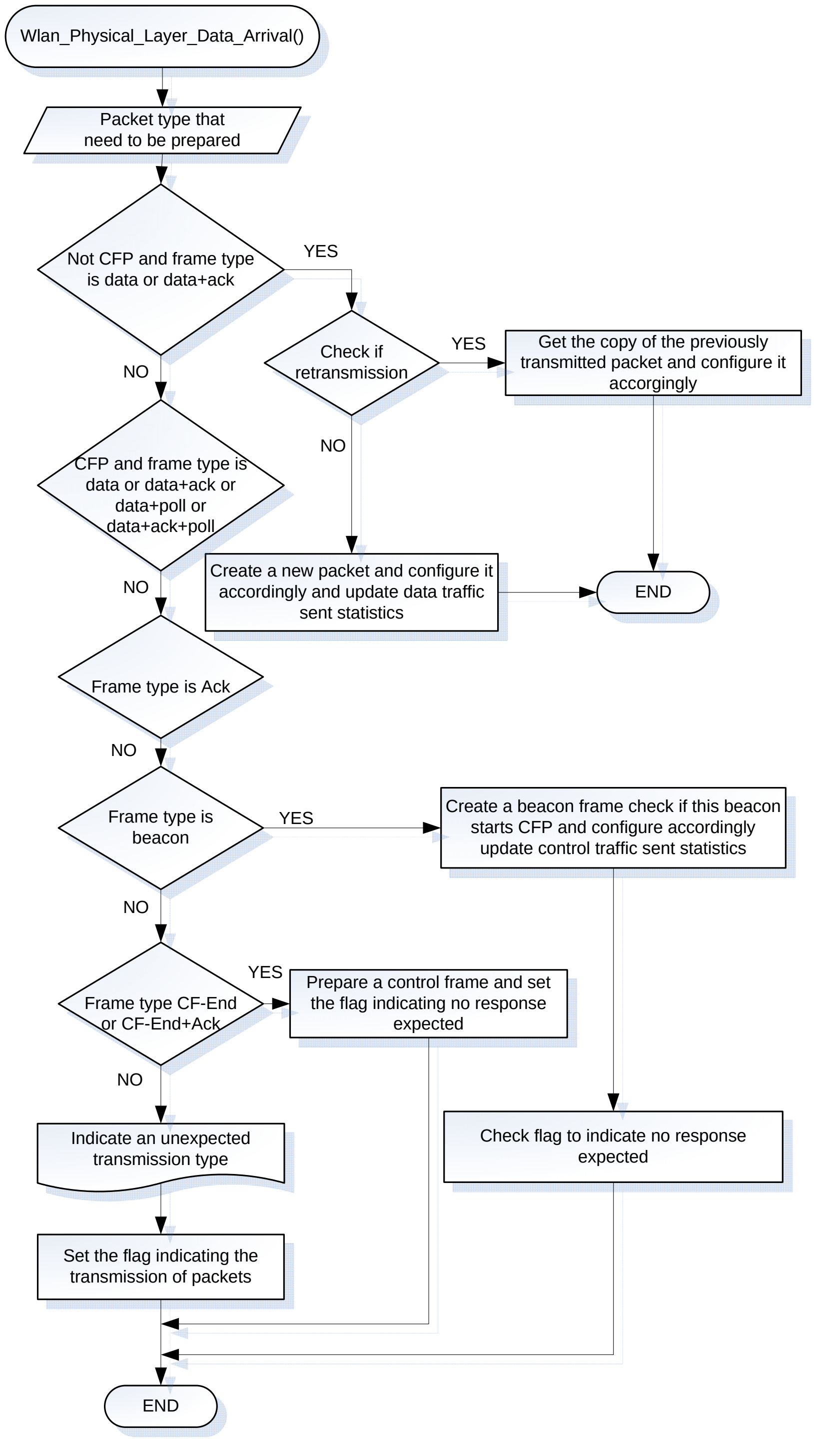
Physical Layer Data Arrival



Schedule Deference



Prepare Frame To Send



Appendix B: Detailed Results

- ## First Scenario

Upstream Delay		
	HCCA	EDCA
QSTA	Delay (ms)	Delay (ms)
0	0	0
2	6,83	4,34
4	6,83	6,08
5	8,60	6,19
6	8,70	6,71
8	9,70	11,78
12	11,01	24,33
15	13,32	50,00
20	20,18	52,87
25	21,05	68,73
40	31,99	173,03
50	40,64	201,00
60	42,64	220,00
75	60,54	268,17
100	79,84	

Downstream Delay		
	HCCA	EDCA
QSTA	Delay in mSec	Delay in msec
0	0	0
2	11,11	0,33
4	11,13	0,67
5	11,14	0,67
6	11,12	0,67
8	13,93	10,67
12	21,12	15,18
15	26,12	35,00
20	26,60	42,69
25	51,04	57,39
40	51,88	109,68
50	51,88	150,00
60	58,11	170,00
75	100,13	218,61
100	100,13	

• Second Scenario

UpStream Delay		
	HCCA	EDCA
BE	Delay in msec	Delay in msec
0,1	13,62	43,01
0,3	13,66	44,54
0,5	13,72	60,30
0,7	13,00	62,00
1,0	14	67,28

Downstream Delay		
	HCCA	EDCA
BE	Delay in mSec	Delay in msec
0,1	21,12	23,47
0,3	21,94	21,94
0,5	21,40	25,21
0,7	22,15	27,32
1,0	22,15	31,95

Throughput		
	HCCA	EDCA
BE	Delay in mSec	Delay in msec
0,1	127	128
0,3	384	370
0,5	812	818
0,7	1077	764
1,0	870	887

Index of the appended CD