DUBLIN CITY UNIVERSITY
SCHOOL OF ELECTRONIC ENGINEERING

Development of a WLAN Performance Calculator

Bronagh Connolly
January 2006

MASTER OF ENGINEERING

IN

TELECOMMUNICATIONS ENGINEERING

Supervised by Dr. S. Murphy

Acknowledgements

I would like to thank my supervisor Dr. Seán Murphy for his guidance during this project. Thanks also to my family for all their support.

Declaration

I hereby declare that, except where otherwise indicated, this document is entirely my own work and has not been submitted in whole or in part to any other university.

Signed: .... Date: ....

Abstract

The research community is currently very interested in the performance of the 802.11 networks, particularly the performance of the 802.11 Medium Access Control (MAC) mechanisms. A number of models for the 802.11 MAC have been proposed within the research community; these models capture the behaviour of the 802.11 MAC under different circumstances.

Throughout the course of this project a Graphic User Interface based tool that can be used to calculate the performance of an 802.11 MAC under different conditions was developed. A number of analytical models that capture the behaviour of the Wireless Local Area Network (WLAN) in different conditions are implemented.

The tool is written completely in Java with Jama and JFreeChart freeware used as the matrix manipulation and graphing mechanisms respectively.

The results of the tool are compared with the results given in the research papers to verify the correct operation of the tool.

The tool can be used as a basis for further research in this area or as a teaching aid, demonstrating how the WLAN reacts under varying conditions.

Table Of Contents

Table Of Figures

Introduction

Traditional Local Area Networks (LANs) link together using cables. This wired environment means that a user must be physically connected to a data outlet and this makes mobility difficult. Expanding the size of this type of LAN involves extra cabling which takes time to install, occupies space and increases overhead costs. This makes wired LANs expensive and difficult to install, maintain and modify.

The emergence of Wireless LAN (WLAN) technology in the late 1980s gave users better mobility and network flexibility. New stations could be added to the network with greater ease and less cost, and users had the freedom to work wherever they chose. However, there was limited or no interoperability between WLAN products from the leading vendors. To aid interoperability, the Institute of Electrical and Electronic Engineers (IEEE) set up a working group. In 1997 the IEEE 802.11 Wireless LAN standard was approved.

The 802.11 standard allows vendors to develop interoperable wireless LAN products for the global market. It defines an interface between wireless nodes and the wired world.

The research community is currently very interested in the performance of the 802.11 networks, particularly the performance of the 802.11 Medium Access Control (MAC) mechanisms. A number of models for the 802.11 MAC have been proposed within the research literature; these models capture the behaviour of the 802.11 MAC under different circumstances.

The objective of this project is to develop a Graphical User Interface (GUI) based tool that can be used to calculate the performance of an 802.11 MAC under different conditions. The tool can be used as a basis for further research in this area or as a teaching aid.

Chapter 1 gives an overview of the IEEE 802.11 standard. It briefly discusses the main standards in the 802.11 suites; particular emphasis is given to the 802.11e standard, which is the one detailing Quality of Service. Four models have been proposed that model the WLAN system under different circumstances. Chapter 2 gives an overview of these models. Chapter 3 describes how the performance calculator is implemented, the tools used and the testing involved. Chapter 4 gives sample snapshots of the calculator and describes the results of the project. Chapter 5 gives the project conclusion.

Chapter 1 - 802.11 Wireless LAN

This chapter gives an overview of the IEEE 802.11 standard, with particular emphasis on the MAC layer. It then gives a brief introduction into the 802.11a, 802.11b and 802.11e standards.

1.1 Topology

A WLAN consists of two network components, stations and Access Points (APs). A station is a device that can communicate to another station via the 802.11 protocols and an AP is a station that is connected to a wired network. These network components form two types of network, an Independent Basic Service Set and an Infrastructure Basic Service Set, see Figure 1.1.

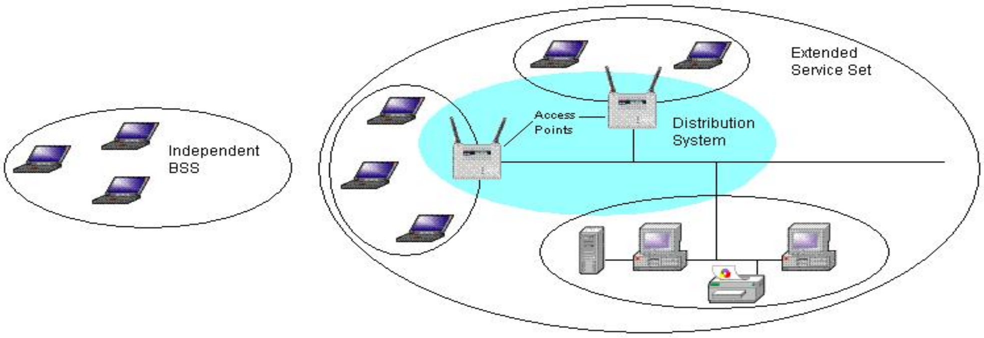


Figure 1.1 IEEE 802.11 Networks

Independent Basis Service Set (IBSS)

An IBSS consists of two or more stations that are wireless connected together. The stations communicate directly with each other and form an ad hoc network. They are not connected to the wired world.

Infrastructure Basic Service Set (Infrastructure BSS)

A BSS can connect to a wider network infrastructure via the AP. The AP increases the range of the wireless network by connecting multiple BSSs together and also interconnecting these BSSs to other networks, e.g. IEEE 802.3 Ethernet LAN. All data frames must be sent or received by the AP. The interconnection of BSS and other networks is known as the Distribution System and it allows the 802.11 networks to be as large as desired. The Infrastructure BSS is also known as the Extended Service Set.

1.2 Architecture

Two layers are defined in the 802.11 standard, the Physical (PHY) layer and the Medium Access Control (MAC) layer. The PHY layer is responsible for the transmitting of data from one station to another in a reliable way. The MAC layer is responsible for transmission channel allocation, addressing of Protocol Data Units and frame formatting. It also performs some functions that are normally associated with the higher layers (e.g. message fragmentation and reassembly and error recovery).

1.3 Physical Layer

Three different physical layer implementations are defined in the 802.11 standard - Direct Sequence Spread Spectrum (DSSS), Frequency Hopping Spread Spectrum (FHSS), and Infrared.

The two main implementations, DSSS and FHSS, use the spread spectrum technique. This technique spreads the narrow band radio signal over a wider band by using a spreading code that is known to both the transmitting and receiving sides. This increases the bandwidth of signals to be transmitted. The receiver performs the inverse operation and the spreaded signal is reconstructed in the narrow band along with narrowed noise. Spreading the signal over a wider frequency makes the signal more noise like and hence secure.

1.3.1 Direct Sequence Spread Spectrum

DSSS is the most widely used physical layer. It uses the 2.4 GHz Industrial, Scientific and Medical frequency band. In 802.11, the available bandwidth is divided into 11 sub-channels; each sub-channel is 11 MHz wide. (Adjacent BSSs prevent transmission on the same sub-channel by ensuring that their centre frequencies are at least 30 MHz apart.) Initially the data stream is divided into small pieces, known as data symbols, and each data symbol is allocated to a sub-channel. The data symbol is then multiplied with a higher data-rate bit sequence (known as a chipping code) to spread the energy of the signal over a larger bandwidth. In 802.11, the chipping code used is the 11-bit Barker code.

The 1 Mbps data rate uses Differential Binary Phase Shift Keying (DBPSK) to convert the digital signal into analog, whereas the 2 Mbps data rate uses Differential Quadrature Phase Shift Keying (DQPSK) modulation. DBPSK recognises two distinct phase shifts - 0º and 180º. DQPSK recognises four separate phase shifts - 0º, 90º, 180º and 270º.

Initially the 802.11 specifications only defined data rates of 1 and 2 Mbps. However in the 802.11b specification data rates of 5.5 and 11 Mbps were defined. These higher data rates mean that DSSS is the most widely used physical layer for WLAN.

1.3.2 Frequency Hopping Spread Spectrum

FHSS also uses the 2.4 GHz Industrial, Scientific and Medical frequency band.

Frequency Hopping is the spreading of the radio signal by transmitting a burst on one frequency channel for a short period of time and then hopping to another channel for another short period of time. This hopping occurs in a predefined pattern that is known by both the transmitting and the receiving sides.

In 802.11 each frequency channel in the hopping pattern occupies 1 MHz of bandwidth. The hopping pattern reduces the likelihood of a collision by one BSS transmitting on the same channel as another BSS.

The 1 Mbps data rate uses a 2-level Gaussian Frequency Shift Keying (GFSK) modulation where each data bit is mapped into 1 of 2 frequencies. The 2 Mbps data rate uses a 4-level GFSK modulation where 2 data bits are mapped into 1 of 4 frequencies.

FHSS only supports data rates of 1 and 2 Mbps. It is not widely supported because it can only support the lower data rates.

1.3.3 Infrared

Infrared is designed for indoor use only. It enables stations to receive line-of-site and reflected transmissions.

Infrared has never received interest and is considered obsolete.

1.4 MAC Layer

This section gives a brief overview of the MAC layer. It discusses the MAC frame structure and the mechanisms used to access the transmission channel.

1.4.1 Fragmentation

Fragmentation is the splitting up of a MAC Service Data Unit (MSDU) into multiple MAC Protocol Data Units (MPDUs). A MSDU is the data unit that is received from the upper layers of the WLAN application and its length can be up to 2304 bytes. A

MPDU is a fragment of the MSDU that is passed from the MAC layer to the physical layer for transmission. The MPDU contains header information, data and a 32-bit Cyclic Redundancy Check (CRC) for error detection. The MPDU also contains sequence information for its position within the MSDU.

By reducing the data unit sizes, the probability of a successful transmission is increased especially under poor channel conditions. However, under good channel conditions, fragmentation adds an extra overhead that reduces the throughput.

1.4.2 Frames

There are three types of frames used in the 802.11 protocol - management, control and data frames. Management frames handle the joining and leaving of a network, e.g. beacons. Control frames ensure that data is transmitted correctly, e.g. RTS, CTS and acknowledgments; and data frames are used to transfer the data between stations.

The MAC frame has the structure:

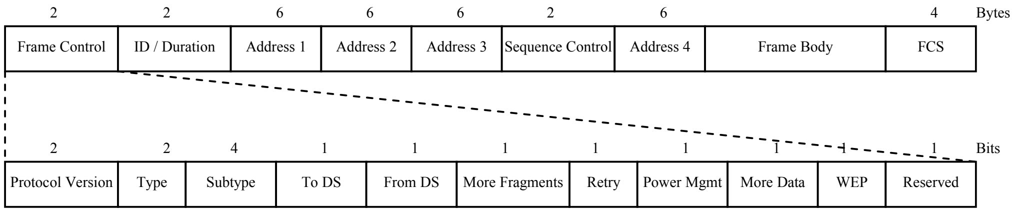


Figure 1.2 MAC Frame

The Frame Control field gives information about the current frame, i.e.

- Protocol Version - indicates the implemented version of the 802.11 standard.
- Type, Subtype - indicates the frame type, e.g. control, management or data.
- To DS - indicates whether a data frame is for the Distribution System.
- From DS - indicates whether a data frame is from the Distribution System.
- More Fragments - indicates whether there are more fragments of the current MSDU.
- Retry - indicates whether this is a retransmission of an earlier data or management frame.
- Power Mgmt - indicates whether the station is in power-save mode. Power-Save mode is when the station only supplies power to its timing circuits. It does not transmit or receive any data. The AP buffers any data for the “sleeping” station. Periodically the station “wakes up” and listens for beacons from the AP. The beacon will indicate whether there are any transmissions for

  the station. If there are, the station indicates to the AP that it can now receive transmissions.
- More Data - indicates to a station in power-save mode that it has data to receive.
- WEP (Wired Equivalent Privacy) - indicates whether an encryption algorithm is used.

The ID/Duration field is dual-purpose. If the station is in power-save mode then the field contains the station identifier; otherwise it contains a duration value. This duration value is used to update the stations Network Allocation Vector (NAV). The NAV indicates when the transmission channel will be idle and hence available for a new transmission.

The presence of some or all of the Address fields is dependent on the type of frame being transmitted. The Address fields indicate the addresses of the BSS identifier for the destination, the source, and the transmitter and receiver stations.

The Sequence Control field contains the MSDU sequence number and fragment number.

The Frame Body contains the actual data being transmitted.

The FCS (Frame Check Sequence) field contains a 32-bit checksum and is used for error checking.

1.4.3 Distributed Coordination Function

The 802.11 MAC layer uses the Distributed Coordination Function (DCF) to share the transmission medium between multiple stations. DCF relies on the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) mechansism to do this.

Carrier Sense Multiple Access (CSMA)

In CSMA, a station checks if the channel is idle before attempting transmission. If the channel is idle then that station waits a time interval. If, after this interval, the channel is still idle then the station begins its transmission. However, there is a high probability of a collision as several stations have been waiting to transmit. To counteract this collision avoidance is used.

Collision Avoidance (CA)

The collision avoidance mechanism uses a random backoff procedure. As stated above, when a busy channel becomes idle, all stations wait a time interval. Once this interval has elapsed, the stations wait a random backoff time interval also. Each station chooses a random backoff time interval from the range 0 to contention window size. The contention window size varies. Initially the range is from 0 to a user defined value or a physical layer defined value, CWmin. Each time a transmission fails, the contention window size doubles, up to a maximum user or physical layer defined value, CWmax. A backoff counter is used to indicate the elapsed time. If the channel is still idle once the backoff time has elapsed, then a new transmission can occur. However if the station senses the channel as busy during its backoff time it pauses the backoff counter and waits for the channel to be sensed as idle for a time interval again. Once it is sensed as idle the backoff counter resumes. Only when the backoff counter is 0 is that station allowed to transmit.

DCF enhances the CSMA/CA medium access mechanism via the Basic Access Mechanism and the RTS/CTS Access Mechanism. These are described in Sections 1.4.3.2 and 1.4.3.3 respectively. However DCF does have some limitations, these are:

- Once a station "wins" access to the medium, it may keep the medium for as long at it chooses.
- If there are many collisions, then the bandwidth is lowered.
- Different types of traffic are not prioritised, i.e. voice traffic should be treated as a high priority because it is real-time, whereas email should be treated as a low priority.
- There are no Quality of Service (QoS) guarantees.

1.4.3.1 Inter Frame Space

The Inter Frame Space (IFS) is the time interval between frames. The IFSs prioritise the access to the transmission channel. The node that waits the shortest IFS time will gain access to the channel first because the other nodes will still be waiting for their IFSs to elapse. The various IFS times are:

SIFS Short IFS. This is the shortest IFS and hence has the highest priority. It is used when sending control frames for setting up transmission channels and acknowledging successful transmissions.

PIFS PCF IFS. Only stations using the PCF access mechanism (see Section 1.4.4) use PIFS. When the PIFS time has elapsed, the station may take control of the channel for its transmission. The PIFS duration exceeds the SIFS duration by a slot time.

DIFS DCF IFS. Stations using either the PCF or DCF access mechanism both use DIFS. DIFS are used when transmitting data frames and management frames.

$$\text{DIFS duration} = \text{SIFS duration} + (2 \times \text{slot time})$$

EIFS Extended IFS is used in the DCF access mechanism when the physical layer has indicated to the MAC layer that there has been an unsuccessful frame transmission.

$$\text{EIFS duration} = \text{SIFS duration} + (8 \times \text{ACK Frame Length}) + \text{PHY Header Length} + \text{DIFS duration}$$

1.4.3.2 Basic Access Mechanism

When a station wants to transmit data, it senses the channel to see if it is idle. If the channel is idle then it waits a DIFS time interval. If, after this interval, the channel is still idle, then the station sends its data. When the data is successfully received, the receiving station waits a SIFS time interval before sending the acknowledgment. When the other stations receive the data transmission, they update their NAV so that they know when the channel will be available again. When the NAV time has expired, the other stations wait a DIFS time interval. Once this has elapsed the backoff procedure for collision avoidance is started in all stations.

Figure 1.3 illustrates a successful data transmission using the Basic Access mechanism.

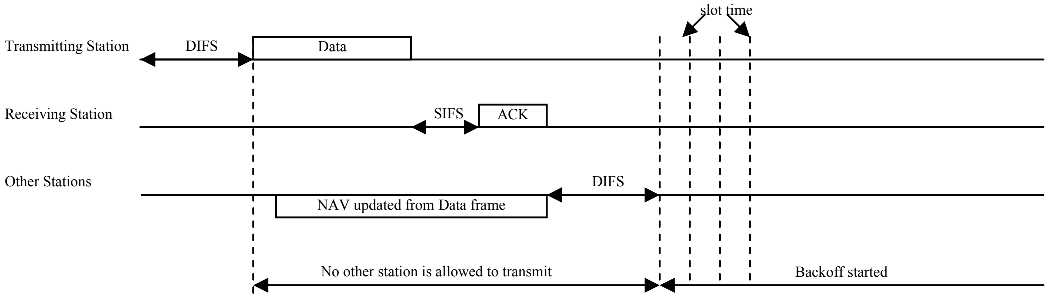
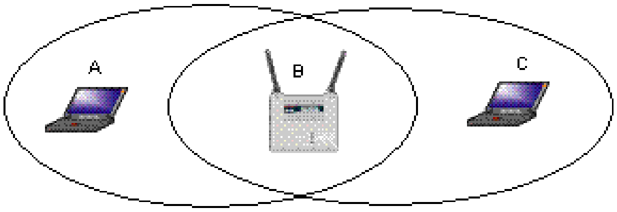


Figure 1.3 Basic Access Mechanism

A problem with the Basic Access mechanism is that of the “hidden node”. This problem occurs when two stations (A and C) are on either side of an AP (B). Station A may not be able to hear transmissions from station C (and vice versa) due to distance or an obstruction. Hence data transmissions from A to B can be interrupted with data transmissions from C to B.



The RTS/CTS Access Mechanism solves the hidden node problem by reserving the transmission channel before transmitting any data.

1.4.3.3 RTS/CTS Access Mechanism

The Request To Send/Clear To Send (RTS/CTS) Access Mechanism reserves the channel, before any data is transmitted. The benefit of this is that bandwidth is not wasted if a collision occurs.

For the RTS/CTS Access Mechanism, when a station wants to transmit data, it senses the channel to see if it is idle. If the channel is idle then it waits a DIFS time interval. If, after this interval, the channel is still idle, then the transmitting station transmits a RTS control frame. When the receiving station receives the RTS frame, it waits a SIFS time interval before transmitting a CTS control frame. When the transmitting station receives the CTS frame, it waits a SIFS time interval before transmitting data. When the data is successfully received, the receiving station waits a SIFS time interval before sending the acknowledgment.

Each time the other stations receive a transmission from either the transmitting or receiving stations, they update their NAV so that they know when the channel will be available again. When the NAV time has expired, the other stations wait a DIFS time interval. Once this has elapsed the backoff procedure for collision avoidance is started in all stations.

Figure 1.4 illustrates a successful data transmission using the RTS/CTS Access mechanism.

The RTS/CTS access mechanism is not without its problems. The additional overhead of temporarily reserving the channel is usually only used for the largest packets because retransmission would be expensive from a bandwidth point of view.

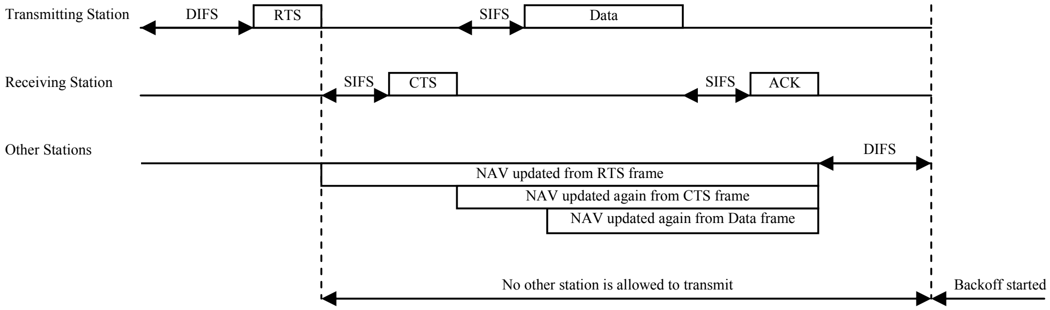


Figure 1.4 RTS/CTS Access Mechanism

1.4.4 Point Coordination Function

The Point Coordination Function (PCF) is another coordination function that is aimed at sessions thay require QoS. It is an optional capability and is only available in Infrastructure BSSs. The AP is known as the Point Coordinator (PC). PCF defines two periods, the Contention Free Period (CFP) and the Contention Period (CP). During CFP, the PC sends Contention Free-Poll packets to each station, one at a time, to give them the right to send a packet. During CP, the DCF access mechanism is used.

At the beginning of the CFP, the PC checks if the channel is idle. If the channel is idle then the PC waits a PIFS time interval. If, after this interval, the channel is still idle, the PC transmits a beacon management frame. Beacon frames are used to maintain synchronisation between the stations and to deliver protocol-related parameters, e.g. CWmin and CWmax. The PC generates beacon frames at regular intervals so that every station knows when the next beacon frame will arrive. Once the beacon frame has been transmitted, the PC waits a SIFS time interval. When this interval has elapsed, the PC initiates the CF transmission by sending either a ‘CF-Poll (No Data)’ frame, a ‘CF-Poll’ frame with Data or just a Data frame.

When a SIFS time interval has elapsed, a station that has PCF implemented can respond to the PC. The station can respond with either a ‘CF-Ack (No Data)’ frame or a ‘CF-Ack (Data)’ frame with Data depending on whether it has data to transmit or not. If the PC has not received a response from the polled station within a PIFS interval, it polls another station.The PC can terminate the CFP by sending a ‘CF-End’ frame.

Figure 1.5 shows the PC initiating the CFP; the first station does not respond to the PC within the PIFS period, so the PC polls another station. The second station sends a ‘CF-Ack (No Data)’ frame (as it has no data to transmit) and the third station sends a ‘CF-Ack (Data)’ frame as it has data to transmit. The PC then ends the CFP.

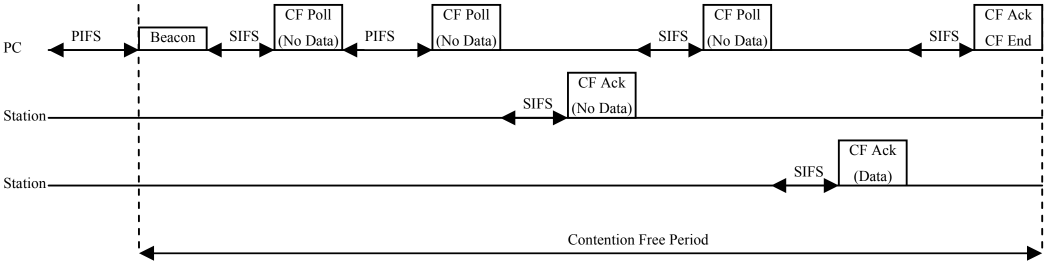


Figure 1.5 Contention Free Period in PCF

PCF allows the piggybacking of multiple frames into one frame, e.g. Data + CF-Ack + CF-Poll being sent in one frame. This improves efficiency by saving bandwidth. However, PCF has several limitations. These are:

- Once a polled station has been allowed to transmit by the PC, it may continue to transmit a full MSDU. The duration of the MSDU transmission is not under the control of the PC, which reduces the QoS to the other polled stations.
- Because the PC does not have control over the channel, it cannot transmit the beacon frame until the channel has been idle for a PIFS time interval. Thus the beacon frames may not arrive at regular intervals.
- Different types of traffic are not prioritised, i.e. voice traffic should be treated as a high priority because it is real-time, whereas email should be treated as a low priority.

1.5 802.11 Specifications

1.5.1 802.11a - OFDM in the 5 GHz Band

802.11a is a physical layer standard that specifies the operation in the 5 GHz band using orthogonal frequency division multiplexing (OFDM). OFDM is a method whereby the data signal is split into several channels each transmitting at different frequencies. OFDM splits the data signal into 48 separate sub-channels to provide transmissions of 6, 9, 12, 18, 24, 36, 48 or 54 Mbps. 6, 12 and 24 Mbps transmission rates are mandatory for all products.

The benefits of 802.11a are:

- Because of the high transmission rates it can support high-end applications involving video, voice, and the transmission of large images and files. It supports densely populated areas of users that have lower bandwidth needs.
- Because 802.11a operates in the 5 GHz band it avoids the interference from devices that use the 2.4 GHz band, e.g. cordless phones and Bluetooth devices.

The drawbacks of 802.11a are:

- 802.11a operates in the 5 GHz band but this means that it has a shorter range.
- It does not interoperate with 802.11b as 802.11b uses the 2.4 GHz band.

1.5.2 802.11b - High Rate DSSS in the 2.4 GHz band

The 802.11b standardises the physical layer for the higher data rates of 5.5 and 11 Mbps. DSSS is the only physical layer implementation in 802.11b as it is the only one that can support these higher data rates.

To increase the data rate, advanced coding techniques are employed. 802.11b uses Complementary Code Keying (CCK) to achieve the higher data rates. CCK uses a set of 64 eight-bit code words, rather than the Barker code specified in 802.11. As a set, these code words have unique mathematical properties that allow a receiver to correctly distinguish them from one another even in the presence of substantial noise and interference (interference can be caused by receiving multiple radio reflections within a building). The 5.5 Mbps rate uses CCK to encode 4 bits per carrier, while the 11 Mbps rate encodes 8 bits per carrier.

To support very noisy environments as well as extend the range, 802.11b WLANs allow data rates to be automatically adjusted to compensate for changes in the channel. Users connect at the full 11 Mbps rate, but as the device moves further away from the AP, the transmission rate lowers. The rate also lowers if there is a lot of interference. If the device moves back within the range or there is less interference, the connection will automatically speed up again. This is transparent to the user and the upper layers of the WLAN application.

1.5.3 802.11e - QoS

The 802.11e enhances DCF and PCF by introducing the Hybrid Coordination Function (HCF). The HCF defines two access mechanisms, the Enhanced Distributed Channel Access (EDCA) and Hybrid Controlled Channel Access (HCCA).

A BSS where all stations support HCF is called a QoS BSS (QBSS). The station that acts as the central coordinator of the other stations in the QBSS is known as the Hybrid Coordinator (HC). Like PCF, HCF defines a Contention Free Period and a Contention Period; EDCA is used in the CP only, HCCA is used in both periods.

HCF alleviates the PCF limitations by classing traffic according to its priority and defining Transmission Opportunities (TXOP). A TXOP allows a 802.11e station to transmit for a specified duration only. A TXOP can be either an EDCA TXOP or a HCCA TXOP (also known as a polled TXOP). The HC controls the TXOP in its QBSS, and announces the TXOP duration via the beacon frame.

Another enhancement of 802.11e is the prevention of frame exchange if that exchange cannot be completed before the next scheduled beacon transmission, this reduces the likelihood of unpredictable beacon delays.

EDCA

EDCA QoS support is provided by the use of Access Categories (AC) and having multiple priority queues in the same station. Queue prioritisation is allowed by varying the contention parameters. The AC contention parameters are:

- AIFS – arbitration IFS. The AIFS[AC] value must be at least a DIFS length. The smaller the AIFS[AC] value, the higher the priority.
- CWmin – minimum contention window size.
- CWmax – maximum contention window size.

There are four ACs that are labelled according to their traffic type. The 802.11e values for the ACs are:

	AC_VO (voice)	AC_VI (video)	AV_BE (best effort)	AC_BK (background)
AIFS	2	2	3	7
CWmin	3	7	15	15
CWmax	7	15	1023	1023

Contention based medium access is achieved in each priority queue by using different parameter values. The parameter values are decided and announced by the HC, via the beacon frame. The priority queues of the same AC in the different QBSS stations use the same EDCA parameter values. An EDCA priority queue begins a transmission when it senses the channel is idle after an AIFS[AC] time interval has elapsed.

When more than one priority queue of the same station attempts to access the channel, the highest priority queue wins the contention. The other queues treat the situation as if a collision has occurred.

EDCA also defines retry counters; when a frame transmission has failed the maximum number of times, the frame is discarded. This feature is beneficial in real-time applications where transmitting a frame too late is irrelevant.

Figure 1.6 shows different transmissions during the Contention Period. Station 1 has the highest priority (and hence the lowest AIFS value) and wins contention for the transmission channel. Station 2 has a higher priority than station 3 so it wins contention for the channel.

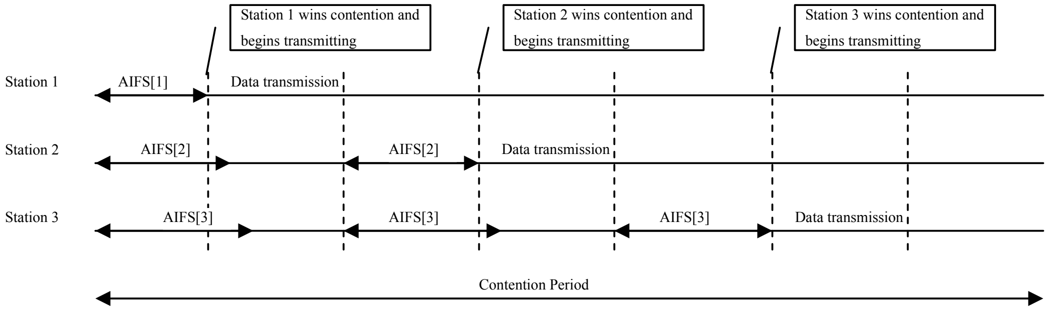


Figure 1.6 EDCA Access Mechanism

HCCA

The controlled medium access of the HCF extends EDCA by allowing the highest priority queue access to the transmission channel during the CFP and the CP.

During CP, the TXOP is determined under the EDCA rules (the highest priority queue wins the contention) or when a station receives a polling frame from the HC.

During CFP, the TXOP is determined when a station receives a polling frame from the HC. The CF-Poll frames can be transmitted by the HC after a PIFS time interval and without any backoff period.

Chapter 2 - Analytical Models of 802.11 WLAN

The performance of WLANs is evaluated through analytical models. Bianchi [5] introduced the most famous analytical model and some later models are built on this model, e.g. [6], [7], [8].

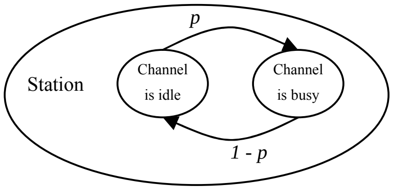
This chapter gives a clear description of the Bianchi analytical model. The other models implemented in the tool are discussed with regards to how they differ from the Bianchi model.

2.1 Bianchi Analytical Model

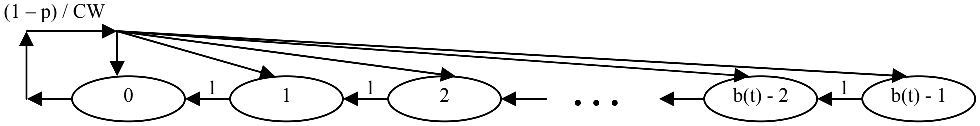
The Bianchi analytical model evaluates the saturation throughput performance of the 802.11 access mechanisms under different circumstances. Saturation throughput is the maximum load that the system can handle when it is in saturation conditions. In saturation conditions each station always has a packet ready for transmission and each packet always waits for a random backoff time before transmitting.

In DCF, the transmission channel can be viewed as being either “busy” (when a station is transmitting) or “idle” (when no station is transmitting). If the channel is busy then a station waits until it is idle before attempting transmission. Hence, the action of one station is dependent on the actions of other stations. However, the station can use the state of the channel to determine its behaviour independently of the other stations (i.e. if the channel is busy then the station knows that another station is transmitting).

Bianchi models the channel as a two state (busy/idle) Markov process. The channel changes from the idle state to the busy state with probability p and from busy to idle with probability $q = (1 - p)$. If the channel is busy, it will remain in the busy state with probability p. If the channel is idle, it will remain in the idle state with probability q, i.e.



A station will change state, depending on its current condition and the state of the channel. If the station is in backoff mode, it will decrement its backoff counter with probability *q* (if the channel is idle) or it will pause the counter with probability *p* (if the channel is busy). When the backoff counter reaches zero, the station transmits a packet (irrespective of whether it results in a collision or not). This situation can be modelled as:



where CW is the contention window size and b(t) represents the backoff counter value for a particular station at time t.

As discussed in Section 1.4.3, DCF uses a contention window. Each time a backoff procedure is utilised, the contention window size increases. Each increase is known as a backoff stage. Figure 2.1 shows the Bianchi Markov Chain Model where each state is represented by {s(t), b(t)}; s(t) is the backoff stage for a particular station and b(t) is the backoff counter value at time t. In this figure, i represents the backoff stage, m is the maximum backoff stage[1] and W_i is the contention window size for that i^{th} backoff stage.

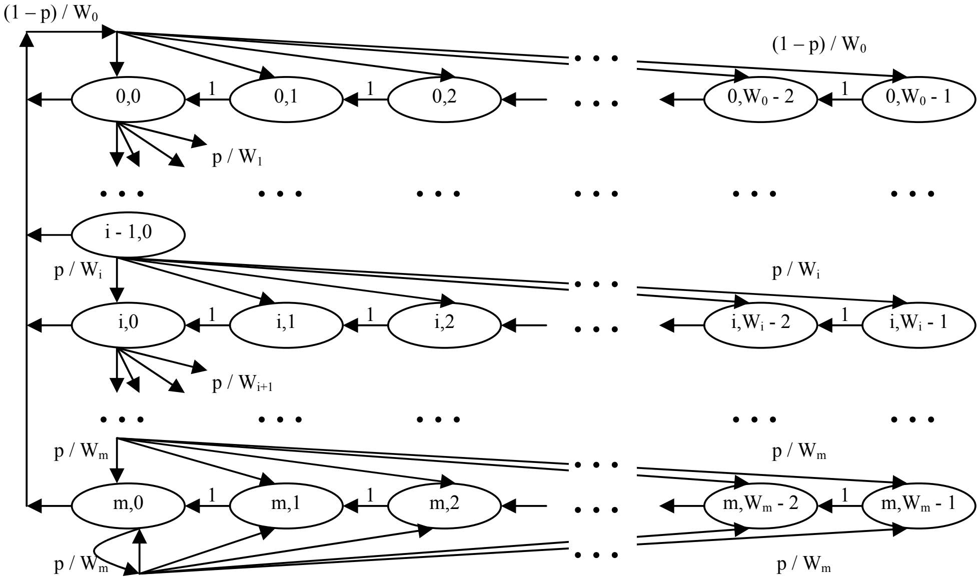


Figure 2.1 Markov chain model for the contention window size

---

[1] m is calculated such that $CWmax = 2^m CWmin$

In this Markov chain the only transitions that affect the model are:

- At the beginning of each slot time, the backoff time counter is decremented.
- After a successful packet transmission, a new packet transmission attempt occurs and the backoff stage is initialised to 0.
- After an unsuccessful transmission, the backoff stage is incremented.
- When an unsuccessful transmission occurs at the maximum backoff stage, the backoff stage value remains the same until the packet is successfully transmitted.

Each transition in the Markov chain is represented by the equations:

$$\left\{\begin{array}{lllll} P\{s(t+1), & b(t+1) \mid s(t), & b(t)+1\} = 1 & b(t) \in (0, W_{s(t)} - 2) & s(t) \in (0,m) \\ P\{0, & b(t+1) \mid s(t), & 0\} = (1-p) / W_0 & b(t) \in (0, W_0 - 1) & s(t) \in (0,m) \\ P\{s(t+1), & b(t+1) \mid s(t) - 1, 0\} & = p / W_i & b(t) \in (0, W_{s(t)} - 1) & s(t) \in (1,m) \\ P\{m, & b(t+1) \mid m, & 0\} = p / W_m & b(t) \in (0, W_m - 1) & \end{array}\right.$$

where s(t) is the backoff stage for a particular station at time t and b(t) is the backoff counter value at time t.

By obtaining a closed form solution of the Markov chain, the following equations are derived:

$$\tau = \frac{2\,(1-2p)}{(1-2p)\,(W+1) + p\,W(1-(2p)^{m})} \tag{1}$$

$$p = 1 - (1-\tau)^{n-1} \tag{2}$$

$$\text{therefore } \tau = 1 - (1-p)^{1/(n-1)} \tag{3}$$

where τ is the probability that a station transmits in a randomly chosen slot time, *p* is the collision probability, W is the actual contention window size, m is the maximum backoff stage and n is the number of stations.

By combining equations (1) and (3), the value of τ can be found. τ is then used to calculate the probabilities for successful transmissions and collisions; these are then used to calculate the throughput.

$$S = \frac{P_{tr}\,P_s\,E[P]}{(1-P_{tr})\,\sigma + P_{tr}\,P_s\,T_s + P_{tr}\,(1-P_s)\,T_c}$$

$$P_{tr} = 1 - (1-\tau)^{n}$$

$$P_s = \frac{n\,\tau\,(1-\tau)^{n-1}}{1-(1-\tau)^{n}}$$

where S in the system throughput, P_{tr} is the probability that there is at least one transmission in the considered time slot and P_s is the probability of a successful transmission. E[P] is the average packet payload size and σ is the slot duration.
T_s and T_c are the timings of a successful transmission and a collision respectively. These values are dependent on the access mechanism used.
For the basic access mechanism:

$$T_s = PHY_{HDR} + MAC_{HDR} + E[P] + SIFS + \delta + ACK + DIFS + \delta$$

$$T_c = PHY_{HDR} + MAC_{HDR} + E[P^*] + DIFS + \delta$$

PHY_{HDR} and MAC_{HDR} are the physical layer and MAC layer header lengths in bits. SIFS and DIFS are as discussed in Section 1.4.3.1. δ is the propagation delay and ACK is the time it takes to transmit an ACK frame. E[P*] is the average length of the longest packet payload involved in a collision, however in this model all packets are assumed to be of equal length so the E[P] packet size is used.
For the RTS/CTS access mechanism:

$$\begin{aligned} T_s &= RTS + SIFS + \delta + CTS + SIFS + \delta + PHY_{HDR} + MAC_{HDR} + E[P] + \\ &\qquad SIFS + \delta + ACK + DIFS + \delta \end{aligned}$$

$$T_c = RTS + DIFS + \delta$$

where RTS and CTS are the time it takes to transmit RTS and CTS frames respectively.

The steps involved in calculating the system throughput using the Bianchi analytical model are to initially calculate the control and data frame durations and the IFS durations (based on the data rate supplied by the user and the chosen physical layer implementation). Then the values for T_s and T_c are calculated, followed by the calculation of τ. Using τ, the values for P_s and P_{tr} are calculated. Then using the T_s, T_c, P_s and P_{tr} values, the system throughput is calculated.

2.2 Ergen & Varaiya Model

The Ergen & Variaya model [6] models the situation where stations transmit at different data rates.
The model is based on the Bianchi model, the main difference being how the models define an event. In the Bianchi model an event is an empty slot, a transmission or an

empty slot that assumes no consecutive transmissions. In the Ergen & Variaya model an event is either an empty slot or a retransmission.

The probability of transmission, τ, is:

$$\tau = \frac{1}{\dfrac{(1-2p)(W+1)+p\,W(1-(2p)^{m})}{2\,(1-2p)(1-p)}}$$

and the throughput calculation is:

$$S = \frac{P_s\,E[P]}{(1-P_{tr})\,\sigma + T_s + T_c}$$

$$P_{tr} = 1-(1-\tau)^{n}$$

$$P_s = n\,\tau\,(1-\tau)^{n-1}$$

All values, except the T_s and T_c ones, are the same as for the Bianchi model. In Ergen & Varaiya, the T_s and T_c values are the average durations of a successful transmission and a collision respectively. These durations are dependent on the data rates and the access mechanism used.

The average duration of a successful transmission is calculated by averaging the successful transmission duration values of each station, since only one station is involved in a successful transmission.

$$T_s = \frac{P_s}{n} \sum_{i=1}^{D} n^i\,T^i_s$$

where D is the number of different data rates and T^i_s is the average time the channel is sensed as busy due to a successful transmission.

The average duration of a collision is determined by the average collision duration of the station with the lowest data rate, the number of other stations involved in the collision and how often they are involved.

$$T_c = \sum_{i=1}^{n-1}\sum_{j=1}^{D}\sum_{k=1}^{n^j}\left[\, n-k-\underset{i}{\sum_{l=1}^{j-1}}\ n^l \right] \times T^j_c\,\tau^{i+1}\,(1-\tau)^{n-1-i}$$

where T^j_c is the average time the channel is sensed as busy by each station due to a collision.

The T^i_s and T^i_c durations are dependent on the access mechanism used.

For the basic access mechanism:

$$T^i_s = PHY_{HDR} + MAC_{HDR} + E[P] + SIFS + \delta + ACK + \delta + DIFS$$

$$T^i_c = PHY_{HDR} + MAC_{HDR} + E[P] + \delta + EIFS$$

where EIFS is as discussed in Section 1.4.3.1.

For the RTS/CTS access mechanism:

$$T^i_s = RTS + SIFS + \delta + CTS + SIFS + \delta + PHY_{HDR} + MAC_{HDR} + E[P] + SIFS + \delta + ACK + \delta + DIFS$$

$$T^i_c = RTS + \delta + EIFS$$

All values are the same as for the Bianchi model.

The steps involved in calculating the system throughput using the Ergen & Varaiya analytical model are to initially calculate the control frame durations and the IFS durations (based on the data rate supplied by the user and the chosen physical layer implementation). Then the value of τ is calculated. Using τ, the values for P_s and P_{tr} are calculated. Then the values for T_s and T_c for the stations transmitting at the high data rate and the stations transmitting at the low data rate are calculated. Using the T_s, T_c, P_s and P_{tr}, values, the system throughput is calculated.

2.3 Ziouva & Antonakopoulos Model

The Ziouva & Antonakopoulos model [7] is an extension of the Bianchi model, the difference being that it has an extra state {-1, 0}. This extra state represents the situation whereby the backoff procedure is not invoked because the backoff counter is 0 and the channel is sensed as idle for DIFS time interval.

The collision probability, p, and probability that the channel is busy, p_b, are used to calculate the probability of transmission, τ.

$$\tau = \frac{2(1-p_b)(1-2p)}{2(1-p_b)^2(1-2p)(1-p) + (p_b + p(1-p_b))(1-2p)(W+1) + pW(p_b + p(1-p_b))(1-(2p)^m)}$$

$$p = 1 - (1-\tau)^{n-1} \qquad \text{therefore } \tau = 1 - (1-p)^{1/(n-1)}$$

$$p_b = 1 - (1-\tau)^n$$

The throughput calculation is:

$$S = \frac{P_s\, E[P]}{E[\Psi] + P_s T_s + (1 - P_s) T_c}$$

$$P_s = \frac{n\, \tau\, (1 - \tau)^{n-1}}{1 - (1 - \tau)^n}$$

$$E[\Psi] = \frac{1}{p_b} - 1$$

where $E[\Psi]$ is the average number of consecutive idle slot times before a transmission takes place due to the backoff procedure. All other values are the same as for the Bianchi model.

The steps involved in calculating the system throughput using the Ziouva & Antonakopoulos analytical model are to initially calculate the control and data frame durations and the IFS durations (based on the data rate supplied by the user and the chosen physical layer implementation). Then the values for T_s and T_c are calculated (the timings are then converted from seconds to slot times), followed by the calculation of τ. Using τ, the values for p_b and P_s are calculated. Then using p_b, $E[\Psi]$ is calculated. Using the T_s, T_c, p_b, P_s and $E[\Psi]$ values, the system throughput is calculated.

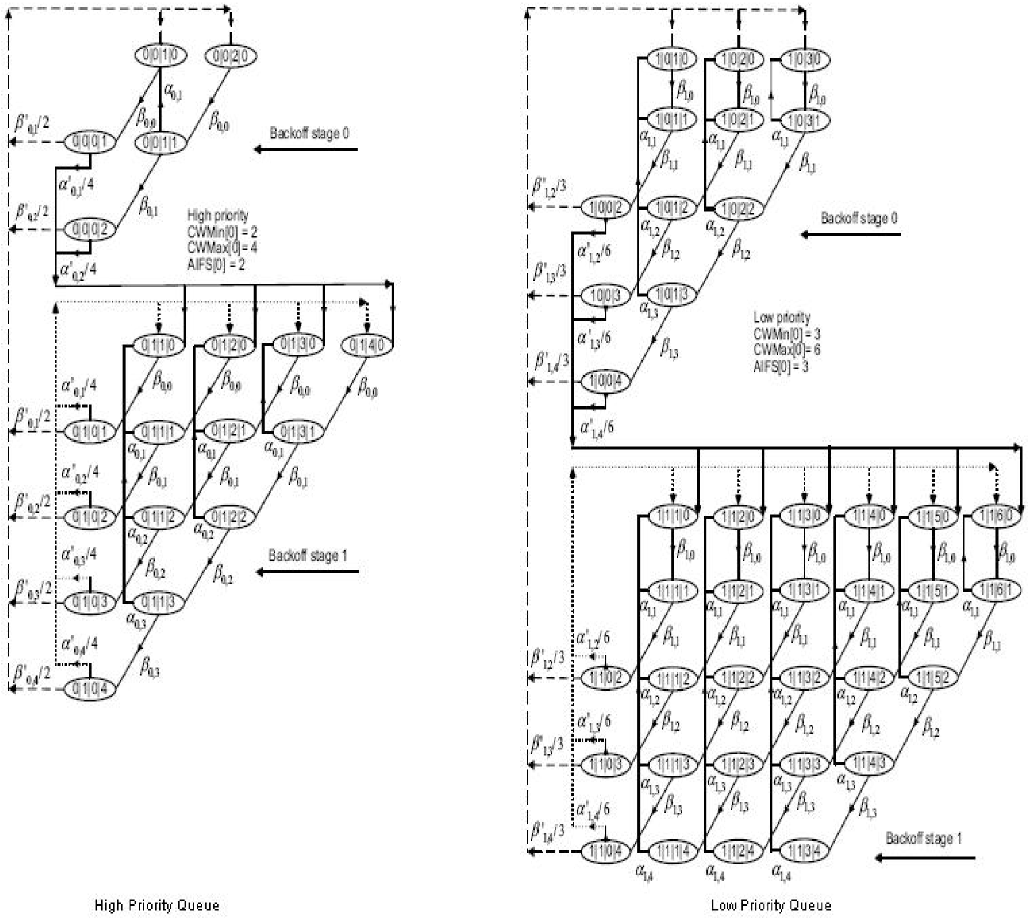
2.4 Tao & Panwar Model

The Tao & Panwar model [8] is based on a hybrid version of the 802.11 DCF mechanism, called Enhanced DCF (EDCF).

In EDCF, each station can have multiple queues that buffer packets of different priorities; the layers above the MAC assign the priorities to the packets. Each priority is given its own AIFS value, which is the number of slot times a packet of that priority must wait (after a SIFS interval) before it can transmit or begin the backoff procedure. Because several priority queues exist within the one station, there is a possibility of packet collisions from different queues but from the same station (i.e. when the backoff counters reach zero for several queues simultaneously). In this situation, the highest priority queue wins contention.

The Tao & Panwar analytical model is a three dimensional Markov model which takes all of the major quality of service features, defined in 802.11e, into consideration. It is an extension of the Bianchi model. The three processes are $\{s^c(t), b^c(t), l^c(t)\}$ where $s^c(t)$ is the backoff stage for the priority c queue, $b^c(t)$ is the backoff time counter for the priority c queue and $l^c(t)$ is the physical time slot.

Figure 2.2 shows example Markov chains for a high and low priority queue.



High Priority Queue

Low Priority Queue

Figure 2.2 Example Markov chains for different priority queues, from [8]

Each transition in the Markov chain is represented by the following equations:

$$\begin{cases} p(c,i,j,0) = q_{c,i} + \sum_{v=1}^{\min\{Wic+Dc,0-j,\ WMax\}} \alpha_{c,v}\, p(c,i,j,v) & \text{when } j \in [1, W_{ic}] \\ p(c,i,j,k) = \beta_{c,k-1}\, p(c,i,j,k-1) & \text{when } j \in [1, W_{ic}],\ k \in [1, D_{c,0}] \\ p(c,i,j,k) = \beta_{c,k-1}\, p(c,i,j+1,k-1) & \text{when } j \in [0, W_{ic}-1]; \\ & k+j \le W_{ic} + D_{c,0}; \\ & k \in [D_{c,0}+1, \min\{W_{ic} + D_{c,0}, WMax\}] \end{cases} \qquad (4)$$

$$
\begin{cases}
q_{c,0} = \sum_{v=0}^{m[c]} \sum_{k=Dc,0+1}^{\min\{WMax,\ Wvc+Dc,0\}} \dfrac{\beta'_{c,k}\, p(c,v,0,k)}{W_{min}[c]} & \\[2ex]
q_{c,i} = \sum_{k=Dc,0+1}^{\min\{WMax,\ W(i-1)c+Dc,0\}} \dfrac{\alpha'_{c,k}\, p(c,i-1,0,k)}{W_{ic}} & 1 \le i \le m[c]-1 \\[2ex]
q_{c,m[c]} = \sum_{k=Dc,0+1}^{\min\{WMax,\ (Wmax[c]/2)+Dc,0\}} \dfrac{\alpha'_{c,k}\, p(c,m[c]-1,0,k)}{W_{max}[c]} + & \\[2ex]
\qquad \sum_{k=Dc,0+1}^{\min\{WMax,\ Wmax[c]+Dc,0\}} \dfrac{\alpha'_{c,k}\, p(c,m[c],0,k)}{W_{max}[c]} &
\end{cases}
\tag{5}
$$

$$
\tau_{c,k} =
\begin{cases}
0 & k \in [0, D_{c,0}] \\[2ex]
\dfrac{\sum_{i=v}^{m[c]} p(c,i,0,k)}{\sum_{i=v}^{m[c]} \left[\sum_{j=0}^{Wic+Dc,0-k]} p(c,i,j,k) \right]} & v \in [0, H[c]],\ H[c] \ge 0;\ k \in [W_{(v-1)c}+D_{c,0}+1,\ \min\{WMax, W_{vc}+D_{c,0}\}] \\[4ex]
\dfrac{\sum_{i=H[c]+1}^{m[c]} p(c,i,0,k)}{\sum_{i=H[c]+1}^{m[c]} \left[\sum_{j=0}^{Wic+Dc,0-k]} p(c,i,j,k) \right]} & k \in [W_{H[c]c}+D_{c,0}+1,\ WMax\};\ H[c] \ge 0
\end{cases}
\tag{6}
$$

$$
\begin{cases}
\beta_{c,k} = (1-\tau_{c,k})^{N-1} \times \prod_{u=0}^{c-1} (1-\tau_{u,k})^{N} \prod_{u=c+1}^{C-1} (1-\tau_{u,k})^{N} \\
\beta'_{c,k} = \prod_{u=0}^{c-1} (1-\tau_{u,k})^{N} \prod_{u=c}^{C-1} (1-\tau_{u,k})^{N-1} \\
\alpha_{c,k} = (1-\beta_{c,k}) \\
\alpha'_{c,k} = (1-\beta'_{c,k})
\end{cases}
\tag{7}
$$

In these equations C is the maximum number of priorities and N is the number of stations. Wmin[*c*] and Wmax[*c*] are the minimum and maximum contention window sizes for priority *c* packets and m[*c*] is the maximum backoff stage for priority *c* packets.

$W_{ic} = 2^{i} \times Wmin[c]$, if i = -1 then $W_{ic} = 0$.

$D_{j,i} = D[j] - D[i]$, where D[*c*] is the AIFS value for the priority *c* packet.

WMax is defined as $\min_{c \in [0, C-1]} \{D[c] - D[0] + Wmax[c]\}$.

H[c] is the largest integer for each priority that satisfies $W_{H[c]c} + D_{c,0} \le WMax$.

Equations (5), (6) and (7) form a non-linear system of equations. This solution of the non-linear system is discussed in Section 2.4.1.

The saturation throughput for priority *c* packets, S_c, is:

$$S_c = \frac{Ps_c \, E[P]}{Pb_c \, \sigma + (1 - Pb_c) \, Ts_c}$$

where Ts_c is the average duration of a successful transmission and is the same as the T_s value in the Bianchi model, as is E[P] and σ. Ps_c and Pb_c are the probability of a priority having a successful transmission and the probability of a priority being in backoff respectively. They are represented by the equations:

$$Ps_c = \sum_{k0=1}^{WMax-1} \Big[P\{k = k_0\} \cdot N \; \tau_{c,k0} \, (1 - \tau_{c,k0})^{N-1} \times \prod_{u=0}^{c-1} (1 - \tau_{u,k0})^{N} \times \prod_{u=c+1}^{C-1} (1 - \tau_{u,k0})^{N-1} \Big]$$

$$Pb_c = \sum_{k0=0}^{WMax-1} \Big[P\{k = k_0\} \times \prod_{u=0}^{C-1} (1 - \tau_{u,k0})^{N} \Big]$$

$$P\{k=k_0\} = \begin{cases} \sum_{i=0}^{m[c]} \Big[\sum_{j=1}^{Wic} p(c,i,j,k_0) \Big] & k \in [0, D_{c,0}] \\ \sum_{i=v}^{m[c]} \Big[\sum_{j=0}^{Wic+Dc,0-k} p(c,i,j,k_0) \Big] & v \in [0, H[c]], H[c] \geq 0; \; k \in [W_{(v-1)c}+D_{c,0}+1, \min\{WMax, W_{vc}+D_{c,0}\}] \\ \sum_{i=H[c]+1}^{m[c]} \Big[\sum_{j=0}^{Wic+Dc,0-k} p(c,i,j,k_0) \Big] & H[c] \geq 0; \; k \in [W_{H[c]c}+D_{c,0}+1, WMax] \end{cases}$$

The steps involved in calculating the system throughput using the Tao & Panwar analytical model are to initially calculate the control and data frame durations and the IFS durations (based on the data rate supplied by the user and the chosen physical layer implementation). Then the value of T_s is calculated. The probability values for each state in the Markov model are then calculated. Using these probabilities, the value of τ is calculated. Using τ, the values for Pb_c and Ps_c are calculated. Then using the T_s, Pb_c and Ps_c values, the system throughput is calculated.

2.4.1 Solving the Non-Linear System of Equations

In the Tao & Panwar analytical model, a method to calculate the probabilities for every state in the Markov model is required. Equations (5), (6) and (7) in Section 2.4 form the non-linear system of equations which gives these probabilities.

The Tao & Panwar research paper ([8]) recommends the use of Matlab to solve the non-linear system of equations. However, Matlab is incapable of solving the size of non-linear system that the Tao & Panwar model can produce when the contention window size is large.

A sparse matrix approach and the Newton Raphson method were both implemented in the WLAN performance calculator to try and solve the non-linear system of equations.

2.4.1.1 Sparse Matrix Approach

Tao suggests the use of a sparse matrix to solve the non-linear system. Each transition state in the Markov chain is represented as a transition probability in the sparse matrix. The transition probability sparse matrix is recursively generated until the correct solution is found.

The steps involved in solving the system of non-linear equations using the sparse matrix approach are to initially estimate the probability value for each state in the Tao & Panwar system, matrix A. Generate matrix T, the transition matrix that is based on A and equations (5), (6) and (7) in Section 2.4. Generate matrix T^T, the transpose matrix of the transition matrix. Calculate matrix B such that $B = A \times T^T$. Calculate the probability value for each state in the Tao & Panwar system based on matrix B. Store the output in matrix A. Repeat until an accurate solution is found.

2.4.1.2 Newton Raphson Method

The Newton Raphson method is an efficient algorithm for converging to a root. The problem is solved by initially guessing an approximate solution to the function. The next guess can be solved by using this guess and a derivative of the function. This iteration continues until the correct solution is found.

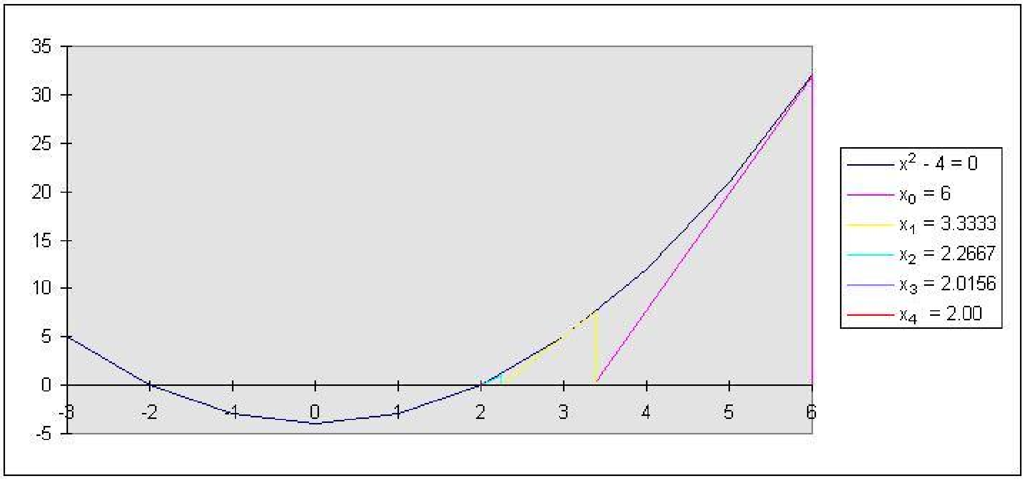
E.g. The function to be solved is $f(x) = x^2 - 4$.

The derivative of the function is $f'(x) = 2x$.

The initial guess is $x = 6$.

n	x_n	f(x)	f'(x)	$x_{n+1} = x_n – (f(x) / (f'(x))$
0	$x_0 = 6$	$f(x) = 6^2 – 4 = 32$	f'(x) = 2×6 = 12	$x_1 = 3.3333$
1	$x_1 = 3.3333$	f(x) = 7.1111	f'(x) = 6.6666	$x_2 = 2.2667$
2	$x_2 = 2.2667$	f(x) = 1.1377	f'(x) = 4.5333	$x_2 = 2.0156$
3	$x_3 = 2.0156$	f(x) = 0.0629	f'(x) = 4.0313	$x_3 = 2.00$

This can be represented graphically by:



Newton Raphson can be generalised to solve non-linear systems of equations. A graphical representation is shown in Figure 2.3. The solid curves represent f(x,y) and the dashed lines represent g(x,y). The solutions are where f(x,y) and g(x,y) intersect.

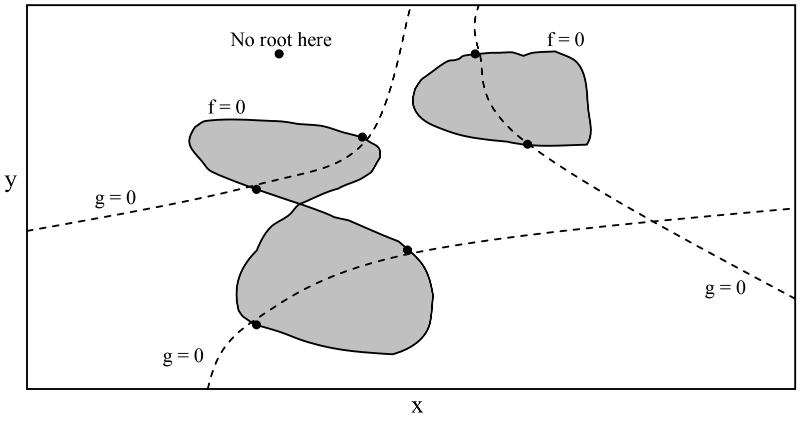


Figure 2.3 Solution of two non-linear equations in two unknowns

In matrix notation this is written as:

$$F(x + \delta x) = F(x) + J \bullet \delta x + O(\delta x^2)$$

where F represents the system of equations, x represents the guesses and J represents the matrix of partial derivatives (i.e. the Jacobian matrix). The new guesses are calculated from $x_{new} = x_{old} - \delta x$.

The matrices can be solved using LU decomposition. LU decomposition is the splitting of a matrix into its lower and upper triangles and then solving the triangles, i.e. Given a matrix A,

$$L \bullet U = A$$

where L is the lower triangular (only has elements on the diagonal and below) and U is the upper triangular (only has elements on the diagonal and above).

$$\begin{pmatrix} l_{11} & 0 & 0 & 0 \\ l_{21} & l_{22} & 0 & 0 \\ l_{31} & l_{32} & l_{33} & 0 \\ l_{41} & l_{42} & l_{43} & l_{44} \end{pmatrix} \bullet \begin{pmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ 0 & u_{22} & u_{23} & u_{24} \\ 0 & 0 & u_{33} & u_{34} \\ 0 & 0 & 0 & u_{44} \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix}$$

The decomposition can be used to solve the linear set

$$A \bullet x = (L \bullet U) \bullet x = L \bullet (U \bullet x) = b$$

by initially solving for the vector y such that

$$(L \bullet y) = b$$

and then soving

$$(U \bullet x) = y.$$

The advantage of LU decomposition is that the solution of the triangular set of equations is quite easy.

The Newton Raphson method is not always accurate. The guessed solution can be completely incorrect and as further guesses are based on the incorrect guess, the further solutions are also incorrect (in fact becoming more incorrect with each iteration). The Newton-Raphson method can also get trapped in a loop.

The steps involved in solving the system of non-linear equations using the Newton Raphson method are to initially estimate the probability value for each state in the Tao & Panwar system, x. Generate F, the set of functions that will calculate each state in the system. Generate J, the set of partial derivatives of F (use the newly generated version of F, the old version of F, the newly calculated version of x (or if this is the first attempt, the estimated version of x) and the old version of x to do this).

F, J and x then form the inputs into the Newton Raphson method, this involves solving the system of linear equations by LU decomposition. The outputs of the method are stored in x, where they are checked for convergence. If converged then a solution has been found, otherwise the steps are repeated.

Chapter 3 - System Implementation

This chapter describes the implementation of the WLAN performance calculator. It describes the Java classes developed for the tool and the external packages used in the implementation.

3.1 Overview

The WLAN Performance Calculator consists of a graphical front end, with the back end performing the complex mathematical operations required. The calculator is developed using the Java 2 SDK Standard Edition (J2SE) development environment [12] and the freeware packages Jama [14] and JFreeChart [15].

The Java class structure of the WLAN Performance Calculator is:

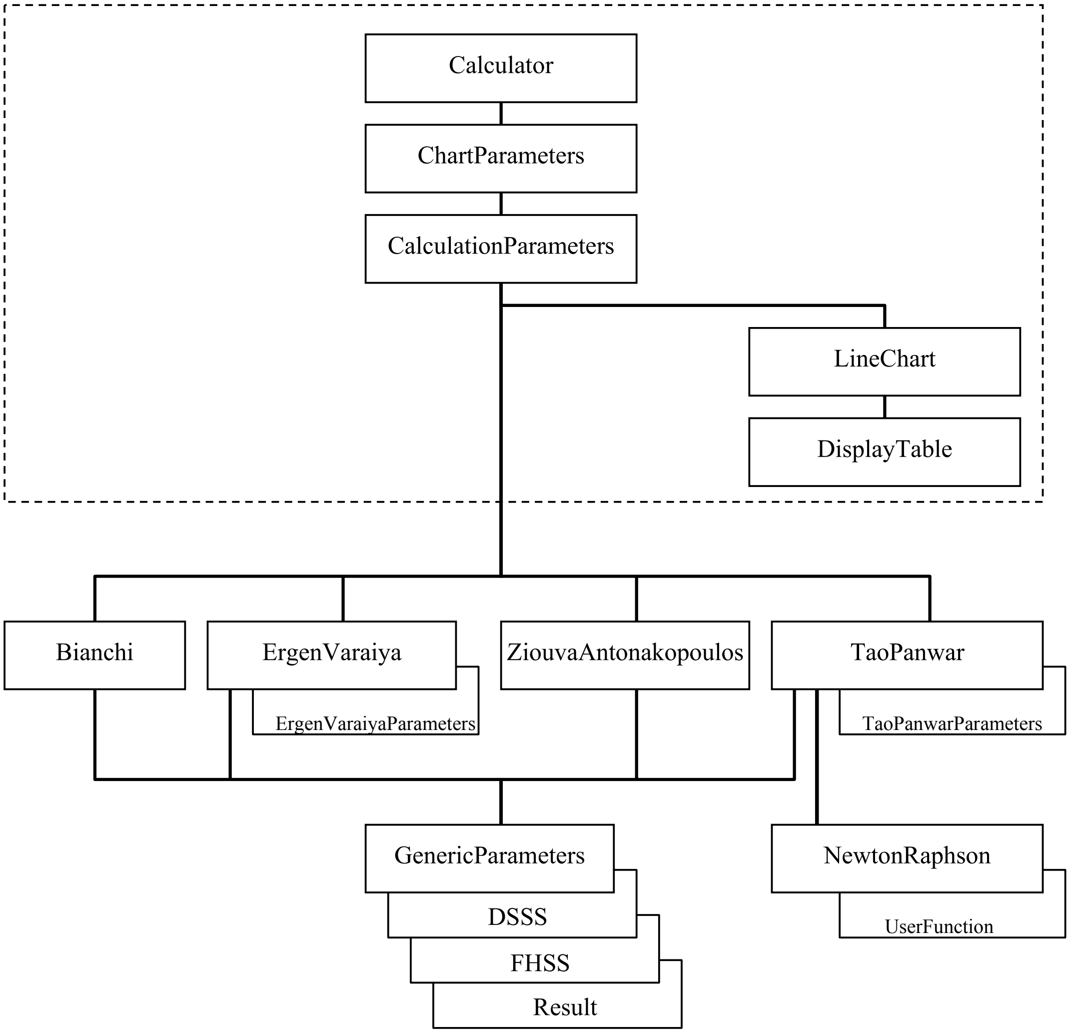


Figure 3.1 WLAN Performance Calculator Java Class Structure

3.2 Java Classes

3.2.1 Calculator Class

This GUI class displays the title screen of the tool. The offered functions are:

Function	**Description**
New	Start a new calculation.
Open	Open a file from a previous calculation. The allowed file types are .TXT (shows results in tabular form).
Exit	Exit the application.

3.2.2 ChartParameters Class

This GUI class displays the Chart Parameters screen. This screen allows the user to input what parameters they would like to chart and how they would like the chart to look. The chart parameter options are:

Chart Parameter	**Description**
X Axis	The item to be displayed on the X-axis of the chart. Options are Number of Stations, Packet Size. Only one selection is allowed.
Y Axis	The item to be displayed on the Y-axis of the chart. Options are Basic Throughput, RTS/CTS Throughput, and Collision Probability. Multiple selections are allowed.
Legend	The text to be displayed as the chart legend.
Chart Type	The type of chart to be displayed. Options are Line Chart. Only one selection is allowed.
Model	The analytical model to be used for the calculation. Options are Bianchi, Ergen & Varaiya, Ziouva & Antonakopoulos and Tao & Panwar. Only one selection is allowed.

3.2.3 CalculationParameters Class

This GUI class displays the Calculation Parameters screen. This screen allows the user to input the parameters that form the basis of the calculation.

The different analytical models require different parameters. The analytical model is chosen as part of the ChartParameters GUI and so the CalculationParameters GUI will change accordingly. There is a set of core parameters that are required irrespective of which analytical model has been chosen. These are:

Calculation Parameter	**Description**
Physical Layer	Physical layer type. Options are DSSS, FHSS. Infrared is considered as obsolete and not implemented in this tool.
CWMin	Minimum size of contention window.
CWMax	Maximum size of contention window.
Data Rate (Mbps)	Data rate used for control frames.[2]
Packet Size (bits)	Average packet size.
Number of Stations	Number of stations.
Minimum, Maximum, Increment	Used for generating X-axis data. Specifies the range that will appear on the chart and the increment.

The default values of these parameters are based on the DSSS physical layer implementation.

If the Bianchi model is chosen, then the set of core parameters are the only ones required.

If the Ziouva & Antonakopoulos model is chosen, then the set of core parameters are the only ones required.

---

[2] This data rate is also used to transfer data frames for the Bianchi, Ziouva & Antonakopoulos and Tao & Panwar models. The Ergen & Varaiya model uses different data rates to transfer data frames.

If the Ergen & Varaiya model is chosen, the extra parameters are:

Ergen & Varaiya Calculation Parameter	Description
High Data Rate	High data rate used to transmit data frames only.
Low Data Rate	Low data rate used to transmit data frames only.
Minimum, Maximum, Increment	Number of stations transmitting at high data rate. Remaining stations in the range will transmit at low data rate. The user specifies the range and the increment. E.g. Minimum = 1, Maximum = 11, Increment = 2. Model calculates the throughput for: 1 station transmitting at high data rate, 10 stations at low data rate; 3 stations transmitting at high data rate, 8 stations at low data rate; … 11 stations transmitting at high data rate, 0 stations at low data rate.

If the Tao & Panwar model is chosen, the extra parameters are:

Tao & Panwar Calculation Parameter	Description
Priority	Default of 1 and 2.
AIFS	AIFS value for this priority.
CWMin	Minimum size of contention window for this priority.
CWMax	Maximum size of contention window for this priority.

3.2.4 LineChart Class

This GUI class displays the results of the calculation in a line chart.

It generates the line chart using the chart parameter inputs and the results of the calculations. It allows the user to display the results in tabular form and to save the results to a .TXT, .PNG or .JPG file.

The LineChart class integrates with the JFreeChart library to display the charts, this is explained further in Section 3.4.

3.2.5 DisplayTable Class

This GUI class displays the results of the calculations in tabular form.

It generates the table using the results of the calculations.

3.2.6 Bianchi Class

This class performs that calculations based on the calculation parameters supplied by the user and the Bianchi analytical model.

3.2.7 ErgenVaraiya Class

This class performs that calculations based on the calculation parameters supplied by the user and the Ergen & Varaiya analytical model.

3.2.8 ZiouvaAntonakopoulos Class

This class performs that calculations based on the calculation parameters supplied by the user and the Ziouva & Antonakopoulos analytical model.

3.2.9 TaoPanwar Class

This class performs that calculations based on the calculation parameters supplied by the user and the Tao & Panwar analytical model.

The TaoPanwar class integrates with the Jama library to perform matrix manipulation, this is explained further in Section 3.3.

3.2.10 NewtonRaphson Class

This class performs the Newton Raphson method for solving non-linear system of equations. The Tao & Panwar analytical model supplies the system of equations.

3.2.11 UserFunction Class

This class is used by the Newton Raphson class to store the equations and the derived equations.

3.2.12 GenericParameters Class

This class stores the set of core calculation parameter values that are required irrespective of which analytical model has been chosen. The values are inputted by the user via the Calculation Parameters screen.

This class provides access functions for each parameter in the set of core parameters.

3.2.13 ErgenVaraiyaParameter Class

This class stores the set of Ergen & Varaiya calculation parameter values that are required for this analytical model. The values are inputted by the user via the Ergen & Varaiya Calculation Parameters screen.

This class provides access functions for the extra parameters required by the Ergen & Varaiya analytical model.

3.2.14 TaoPanwarParameter Classes

This class stores the set of Tao & Panwar calculation parameter values that are required for this analytical model. The values are inputted by the user via the Tao & Panwar Calculation Parameters screen.
This class provides access functions for the extra parameters required by the Tao & Panwar analytical model.

3.2.15 Result Class

This class stores the result of the calculation. The values stored are Number of Stations, Packet Size, Collision Probability, Throughput using the Basic Access mechanism and Throughput using the RTS/CTS Access mechanism.
This class provides access functions for each result value.

3.2.16 FHSS, DSSS Classes

These classes store the default values for these physical layer implementations.

3.2.17 ExampleFileFilter Class

This class filters out the file types that it does not know about. It is used by the Calculator class to filter out all files except the .TXT files. It is used by the LineChart class to filter out all files except the .TXT, .PNG or .JPG files.
This class was not developed as part of this project. It is free software and was downloaded from [13].

3.3 Jama

Jama is a free Java class library used for matrix manipulation. It provides operations for basic numerical linear algebra. It constructs and manipulates real, dense matrices; provides operations to access sub-matrices and individual matrix elements. The basic arithmetic operations provided in Jama include matrix addition and multiplication and calculating transpose matrices.
Jama also provides five fundamental matrix decompositions:

- Cholesky Decomposition of symmetric, positive definite matrices
- LU Decomposition (Gaussian elimination) of rectangular matrices

- QR Decomposition of rectangular matrices
- Eigenvalue Decomposition of both symmetric and nonsymmetric square matrices
- Singular Value Decomposition of rectangular matrices

The WLAN Performance Calculator interfaces with the Jama library in the TaoPanwar class, when solving the non-linear system of equations produced by the Tao & Panwar analytical model.

3.4 JFreeChart

JFreeChart is a free Java class library for generating charts. It supports several charts types including line and area charts, scatter plots, bar charts and pie charts (2D and 3D). It interfaces with any type of data. Charts can be exported to .PNG and .JPG file types.

JFreeChart uses the JCommon collection of classes. The JCommon library includes text utilities, user interface classes for displaying information about applications, custom layout managers, a date chooser panel and serialization utilities.

The WLAN Performance Calculator interfaces with the JFreeChart library in the LineChart class. The first step to creating a chart from the results of the calculation is to convert the results into the JFreeChart format. The JFreeChart format is a simple data set where each point on the chart is represented by the actual value, which category it belongs to (the categories are selected from the Y-axis options on the ChartParameters screen) and the item that is to be displayed on the X-axis. Then the look and feel of the chart is created and the axis labels and ranges are set.

Chapter 4 - Results

The goal of the performance calculator is to be able to implement the models and achieve the same results given in [5], [6], [7] and [8].

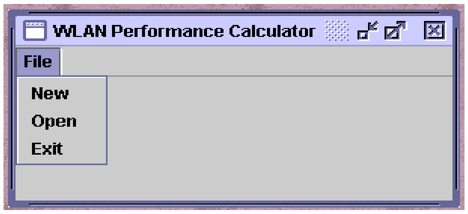
4.1 GUI Snapshots

4.1.1 Title screen

The ‘New’ option displays the Chart Parameters screen.

The ‘Open’ option displays the standard Java file selection screen.

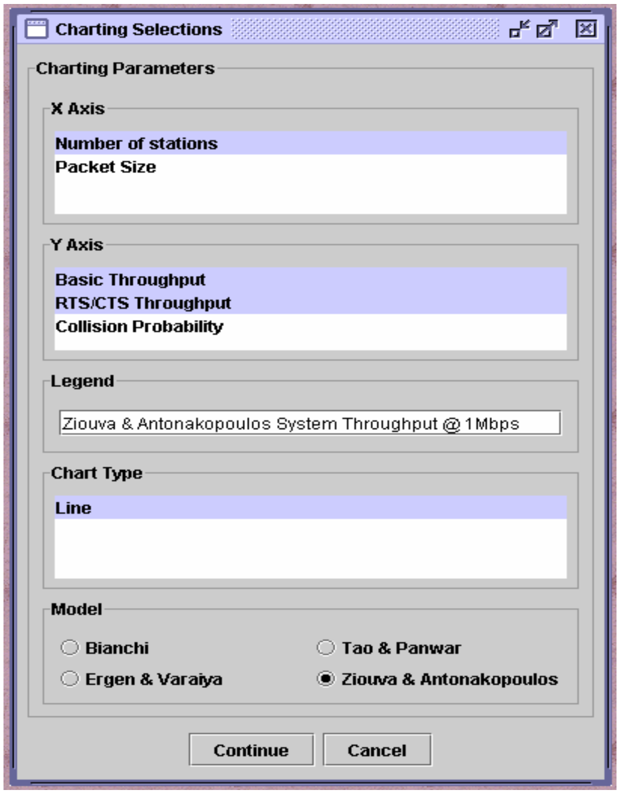
The ‘Exit’ option exits the application.



4.1.2 Chart Parameters screen

The ‘Continue’ option displays the Calculation Parameters screen.

The ‘Cancel’ option closes the screen.



X Axis: Number of stations (selected), Packet Size

Y Axis: Basic Throughput (selected), RTS/CTS Throughput (selected), Collision Probability

Legend: Ziouva & Antonakopoulos System Throughput @ 1Mbps

Chart Type: Line (selected)

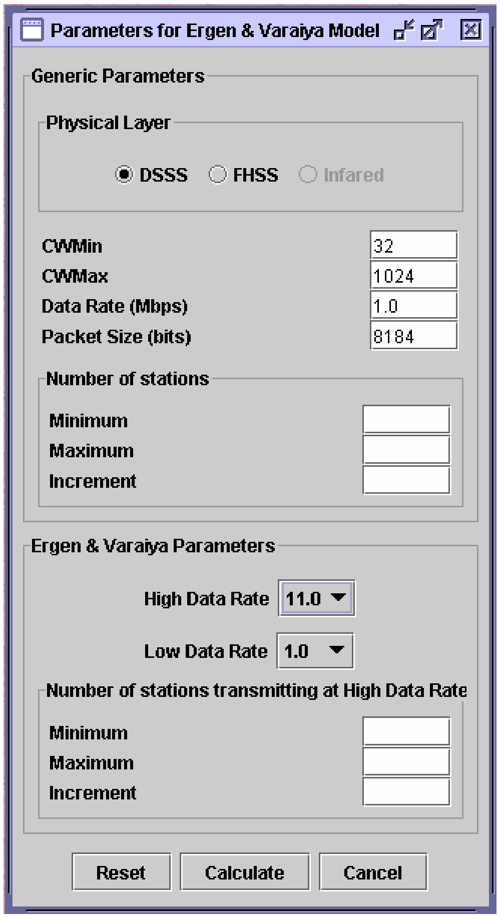
Model:
○ Bianchi ○ Tao & Panwar
○ Ergen & Varaiya ● Ziouva & Antonakopoulos

4.1.3 Calculation Parameters screen

The 'Reset' option initialises the parameters to their default values.

The 'Calculate' option performs the calculation and then displays the Chart Results screen.

The 'Cancel' option closes the screen.



Parameters for Ergen & Varaiya Model

Generic Parameters

Physical Layer

● DSSS ○ FHSS ○ Infared

CWMin	32
CWMax	1024
Data Rate (Mbps)	1.0
Packet Size (bits)	8184

Number of stations

Minimum	
Maximum	
Increment	

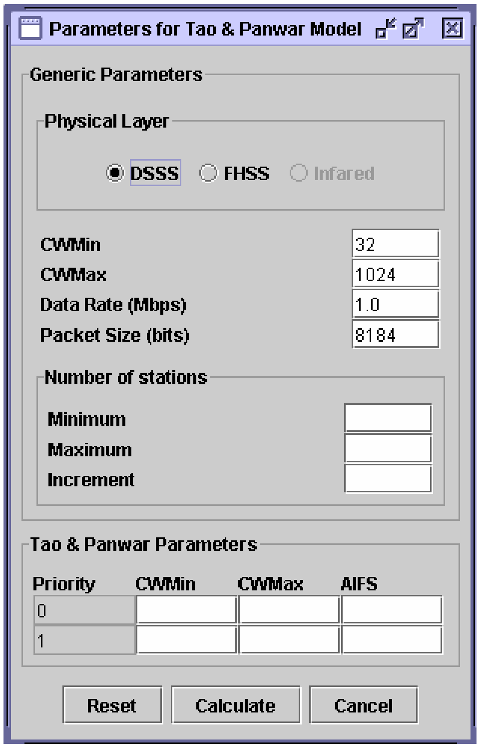
Ergen & Varaiya Parameters

High Data Rate: 11.0

Low Data Rate: 1.0

Number of stations transmitting at High Data Rate

Minimum	
Maximum	
Increment	



Parameters for Tao & Panwar Model

Generic Parameters

Physical Layer

● DSSS ○ FHSS ○ Infared

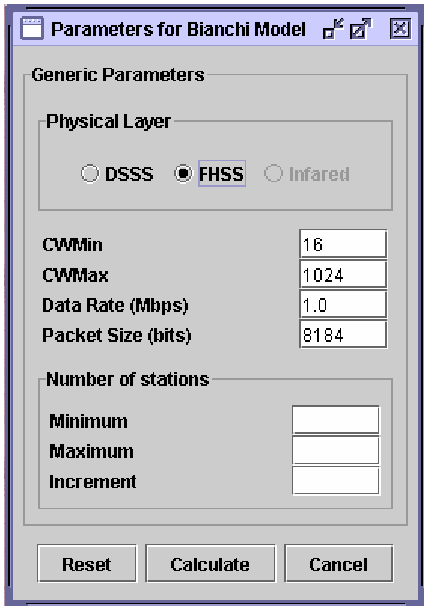
CWMin	32
CWMax	1024
Data Rate (Mbps)	1.0
Packet Size (bits)	8184

Number of stations

Minimum	
Maximum	
Increment	

Tao & Panwar Parameters

Priority	CWMin	CWMax	AIFS
0			
1			



Parameters for Bianchi Model

Generic Parameters

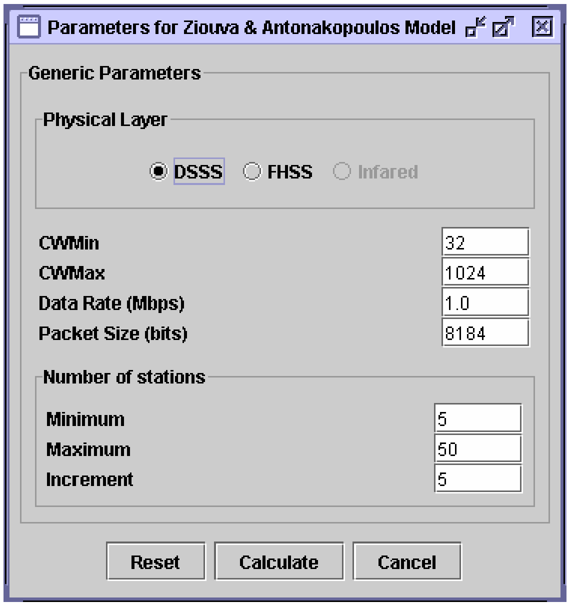
Physical Layer

○ DSSS ● FHSS ○ Infared

CWMin 16
CWMax 1024
Data Rate (Mbps) 1.0
Packet Size (bits) 8184

Number of stations

Minimum
Maximum
Increment



Parameters for Ziouva & Antonakopoulos Model

Generic Parameters

Physical Layer

● DSSS ○ FHSS ○ Infared

CWMin 32
CWMax 1024
Data Rate (Mbps) 1.0
Packet Size (bits) 8184

Number of stations

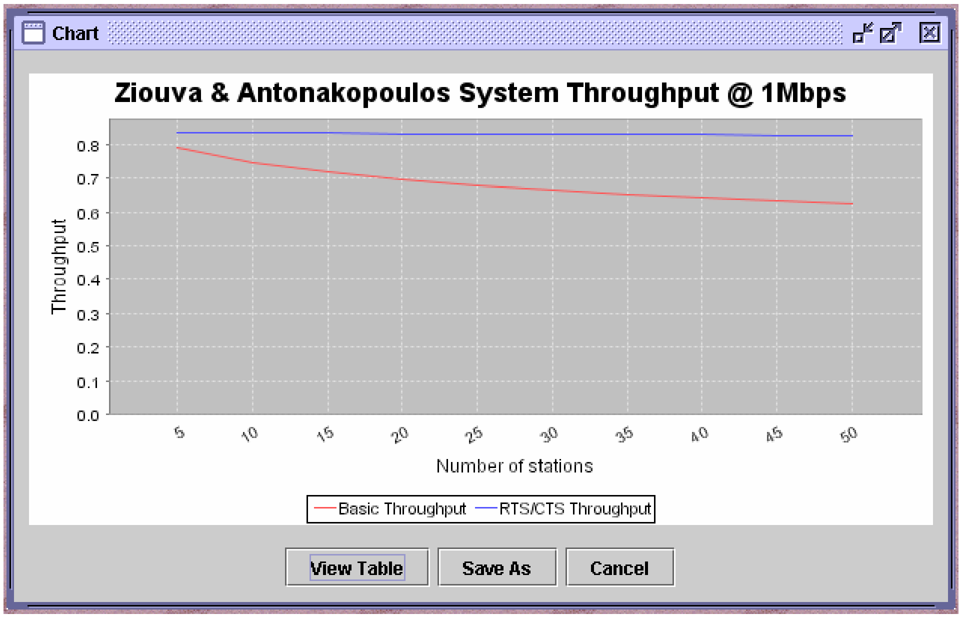
Minimum 5
Maximum 50
Increment 5

4.1.4 Chart Results Screen

The ‘View Table’ option displays the results in tabular form, see Section 4.1.5.

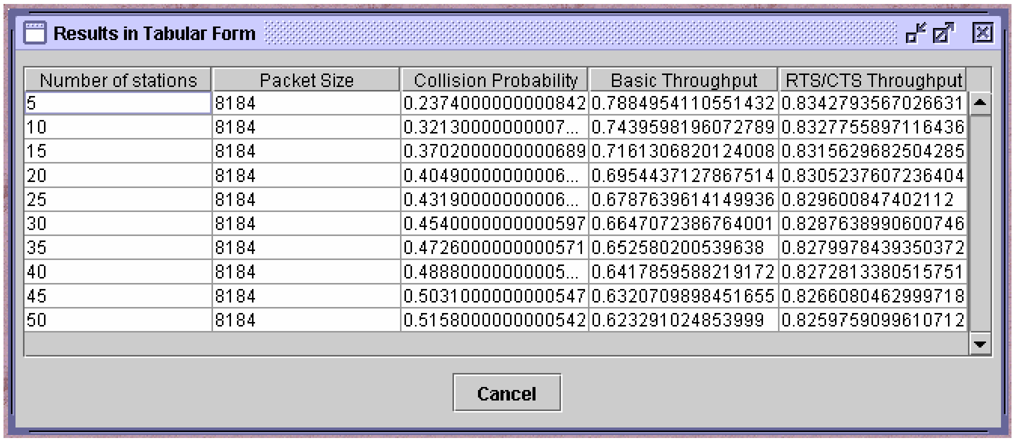
The ‘Save As’ option allows the results to be saved to file; the standard Java file selection screen is used.

The ‘Cancel’ option closes the screen.



4.1.5 Table Results Screen

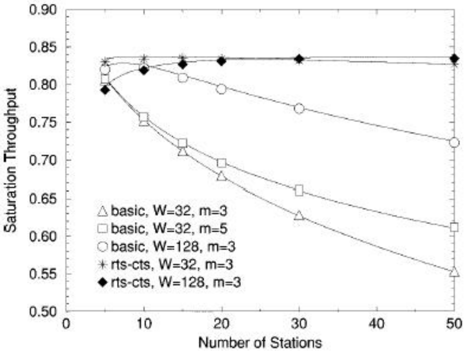
The ‘Cancel’ option closes the screen.

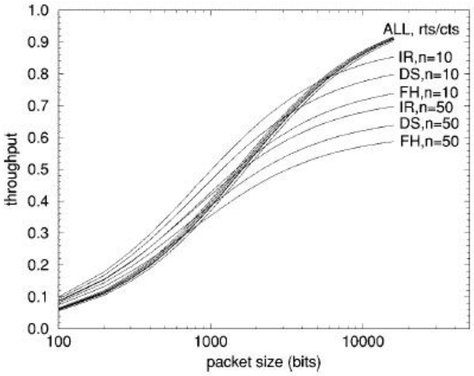


Number of stations	Packet Size	Collision Probability	Basic Throughput	RTS/CTS Throughput
5	8184	0.2374000000000842	0.7884954110551432	0.8342793567026631
10	8184	0.32130000000007...	0.7439598196072789	0.8327755897116436
15	8184	0.3702000000000689	0.7161306820124008	0.8315629682504285
20	8184	0.40490000000006...	0.6954437127867514	0.8305237607236404
25	8184	0.43190000000006...	0.6787639614149936	0.829600847402112
30	8184	0.4540000000000597	0.6647072386764001	0.8287638990600746
35	8184	0.4726000000000571	0.652580200539638	0.8279978439350372
40	8184	0.48880000000005...	0.6417859588219172	0.8272813380515751
45	8184	0.5031000000000547	0.6320709898451655	0.8266080462999718
50	8184	0.5158000000000542	0.623291024853999	0.8259759099610712

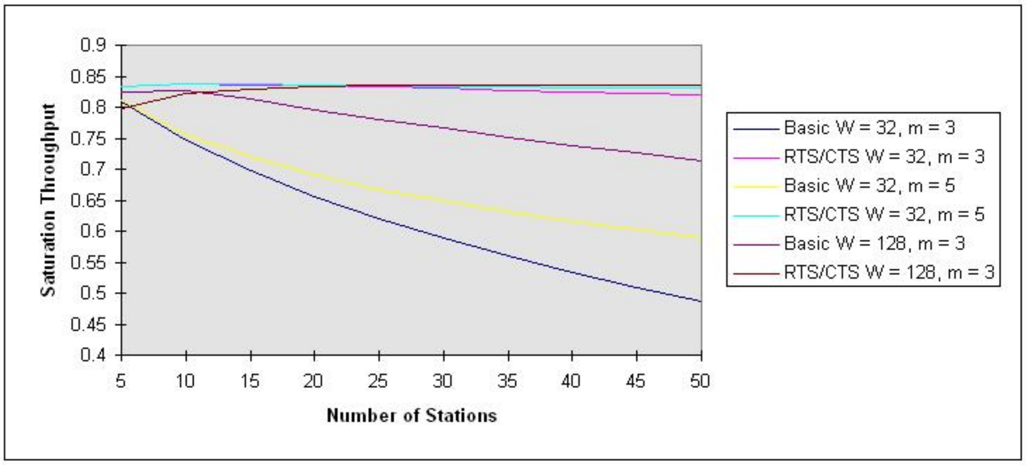
4.2 Bianchi Model Results

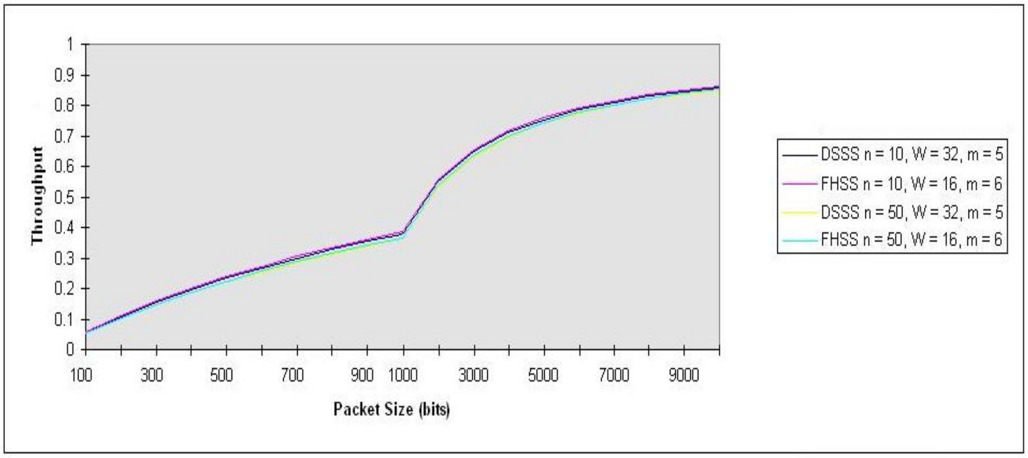
The Bianchi model simulation results using the FHSS physical layer parameters given in [5] are:





The results[3] from the WLAN Performance Calculator using the FHSS physical layer parameters are:



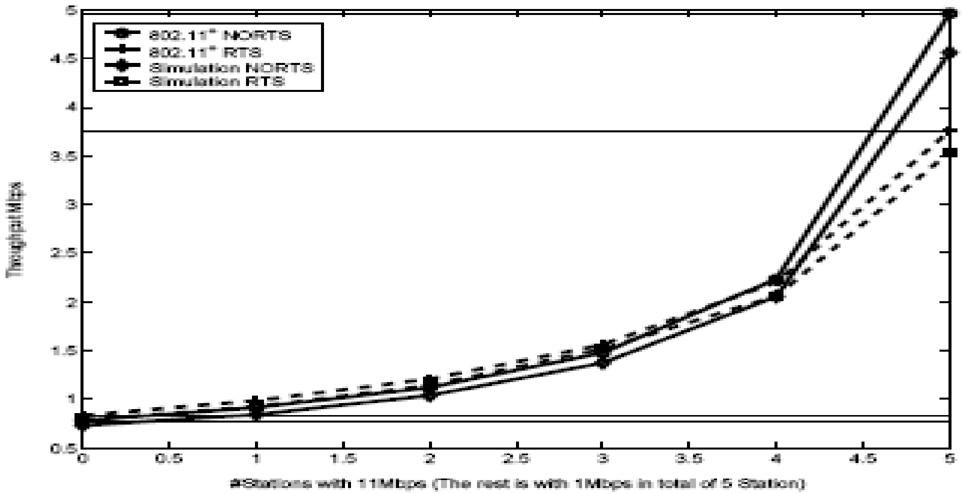


---

[3] The results are not displayed using the performance calculators charting feature because it is limited to showing the basic and RTS/CTS access mechanisms throughput, for one W and m combination only.

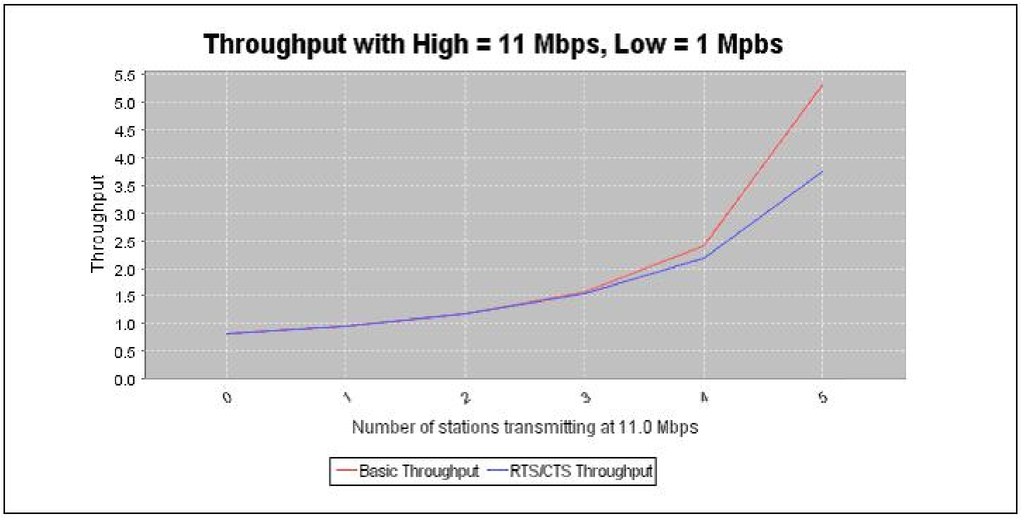
4.3 Ergen & Variaya Model Results

The Ergen & Variaya model simulation results using the DSSS physical layer parameters given in [6] are:



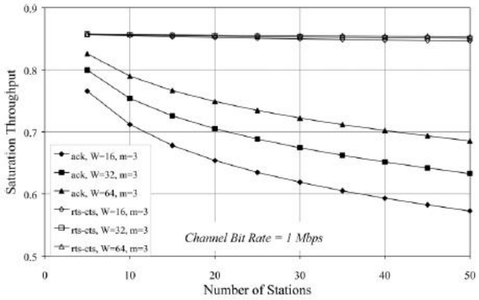
Throughput

The results from the WLAN Performance Calculator using the DSSS physical layer parameters are:

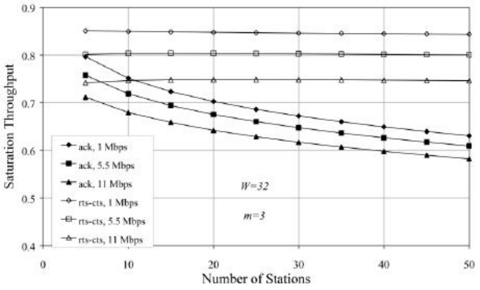


4.4 Ziouva & Antonakopoulos Model Results

The Ziouva & Antonakopoulos model simulation results given in [7] are:

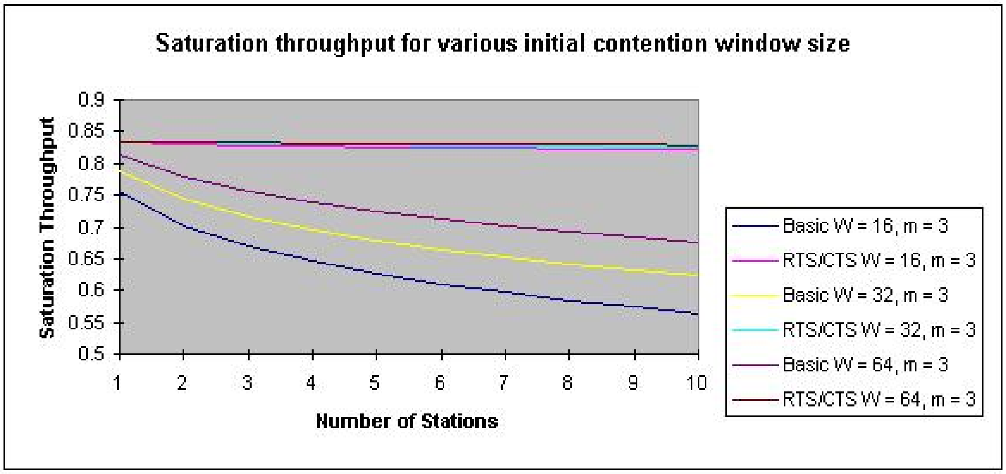


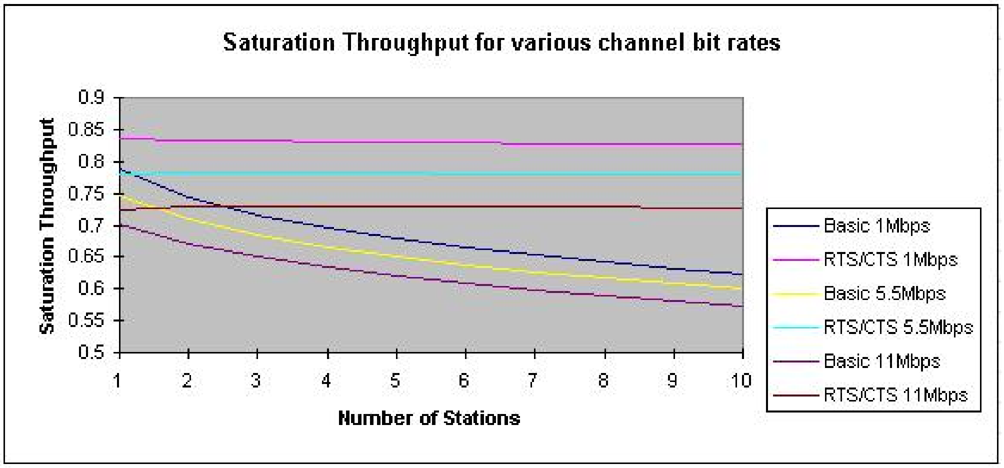
Saturation throughput for various initial contention window sizes.



Saturation throughput for various channel bit rates.

The results[3] from the WLAN Performance Calculator using the DSSS physical layer parameters are:

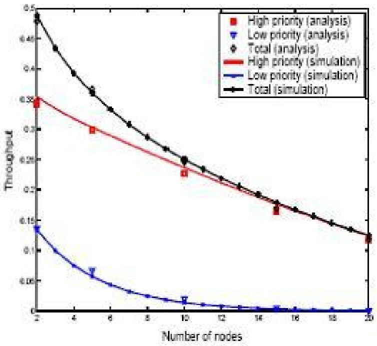




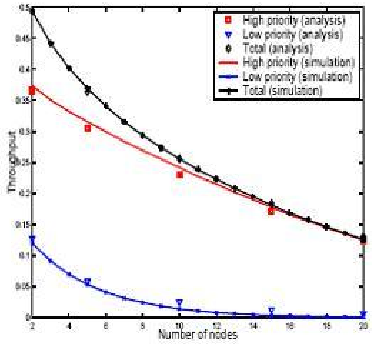
4.5 Tao & Panwar Model Results

As part of the implementation of the Tao & Panwar analytical model, the Newton Raphson method for converging to a root was also implemented. This implementation was also tested and the results are given in Section 4.5.1.

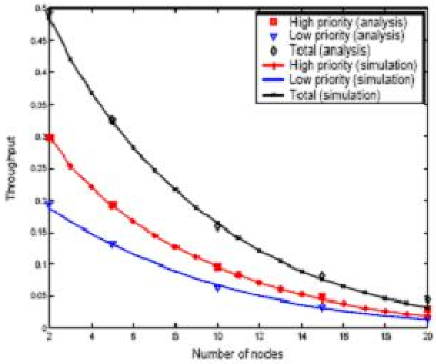
The Tao & Panwar model simulations use the DSSS physical layer parameters.
The simulations use 2 different priorities, a high priority and a low one. They simulations vary the contention window size and the AIFS values. The results given in [8] are:



Two priorities with different AIFS values: $CWMin/Max[0] = 8/16$, $CWMin/Max[1] = 8/16$, $AIFS[0] = 2$, $AIFS[1] = 3$



Two priorities with different CWMin, CWMax and AIFS: $AIFS[0] = 2$, $AIFS[1] = 3$, $CWMin/Max[0] = 8/16$, $CWMin/Max[1] = 10/20$



Two priorities with different CWMin and CWMax. $AIFS[0] = 2$, $AIFS[1] = 2$, $CWMin/Max[0] = 8/16$, $CWMin/Max[1] = 10/20$

The implementation of the Tao & Panwar model was unsuccessful because of the inability to solve the system of non-linear equations. Both the sparse matrix approach and Newton Raphson method had problems in solving the non-linear system.
The sparse matrix approach failed because the transition matrix was set up incorrectly.
The Newton Raphson method was unable to handle large systems of equations and also because the initial guess must be approximately correct. A simple Tao & Panwar

system that consists of 2 priorities, the contention window size ranging from 8 to 16 for both priorities and AIFS values of 2 and 3 will generate a system of 1156 equations with 1156 unknowns. This simple system is too large for the Newton Raphson method to solve.

4.5.1 Newton Raphson Method Results

The Newton Raphson implementation was tested by using a small system of non-linear equations - 3 equations with 3 unknowns.

The equations and the derived equations that were tested are:

$f(x)$	$f'(x)$
$2x^2 - y^2 + z + 3 = 0$	$4x - 2y + 1$
$x^2 + 2y^2 - 5z - 11 = 0$	$2x + 4y - 5$
$x^2 - 3y^2 + 2z^2 - 6 = 0$	$2x - 6y + 4z$

One possible solution to these equations is $x = 2$, $y = 4$ and $z = 5$.

The results from the Newton Raphson implementation:

Initial Guess			Final Result			n
x	y	z	x	y	z	
2.5	4.5	5.5	2.000	4.000	5.000	6
1.5	3.5	4.5	1.999	3.999	5.000	7
1.0	4.0	5.0	1.999	3.999	4.999	8
1.0	1.0	1.0	-2.000	-4.000	5.000	640

where n is the number of iterations of the method that it took to achieve the final result.

These results show that even in a simple system a number of iterations are required to solve the system. If the initial guess is not close to the actual solution then the volume of iterations increases dramatically.

Chapter 5 - Conclusion

This report has given an overview of the project “Development of a WLAN Performance Calculator”.

The first chapter gave an overview of the IEEE 802.11 standard. It described the topology of an 802.11 system and the architecture. The physical layer was described with the emphasis on the two main physical layer implementations – DSSS and FHSS. The MAC layer was described with regards to the MAC frame structure and mechanisms used to access the transmission channel - the Point Coordination Function and the Distributed Coordination Function. The chapter then briefly described the 802.11a and 802.11b specifications. The 802.11e specification was then described in detail, as it is the one detailing the Quality of Service requirements.

The second chapter gave an overview of the analytical models implemented in the performance calculator. The Bianchi analytical model was described in detail, as this forms the basis for the rest of the models implemented in the calculator. The Markov model used in the Bianchi throughput analysis is explained and how this model is used to calculate the throughput. This chapter then described the Ergen & Varaiya analytical model and explained the equations used to calculate the throughput. The Ziouva & Antonakopoulos analytical model and the related equations were then explained. The final analytical model described in this chapter is the Tao & Panwar model. This analytical model incorporates most of the QoS features by using the DCF hybrid, EDCF, which was also described in this section. The models equations were also explained. Some of these equations form a non-linear system. The methods implemented to solve the non-linear system (i.e. the sparse matrix approach and the Newton Raphson method) were also discussed.

The third chapter described how the performance calculator was implemented. It gave an overview of the Java class structure of the performance calculator. It described the function of each class. Where the class related to a GUI it described the options allowed for that GUI.

The fourth chapter gave sample screenshots of each screen of the performance calculator. It then also gave the expected results and the actual results for each analytical model. The implementations of the Bianchi, Ergen & Varaiya and Ziouva & Antonakopoulos analytical models were completed successfully, as can be seen from the graphs. The Tao & Panwar model implemented both the sparse matrix approach and the Newton Raphson method for solving the non-linear system of equations that it generated. The Tao & Panwar analytical model was not implemented successfully. This is due to the inability of solving the system of equations generated by the model. In a simple system the Tao & Panwar can generate approximately 1100 equations with 1100 unknowns, and this proved to be too much for the Newton Raphson method. The sparse matrix approach was unsuccessful because the transition matrix was not set up correctly.

Although this project was mostly successful, the WLAN Performance Calculator could be further improved by:

- Allowing other chart types. The tool is currently limited to displaying a line chart and other chart types would aid readability and usability.
- Merging the results of several tests and displaying in one chart. Section 4.2 and Section 4.4 use several different W and m combinations for each test. In the current implementation it is not possible to display these results in one chart.
- Successfully implementing the Tao & Panwar model using the sparse matrix approach.

References

[1] "ANSI/IEEE Std 802.11, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, June 2003.

[2] Crow, Brian P., Widjaja, Indra, Kim, Jeong Geun and Sakai, Prescott T. "IEEE 802.11 Wireless Local Area Networks", *IEEE Communications Magazine,* pp 116-126, September 1997.

[3] Pagtzis, Theo, Kirstein, Peter and Hailes, Steve "Operational and Fairness Issues with Connection-less Traffic over IEEE 802.11b", *IEEE International Conference on Communications*, pp1905-1913, June 2001.

[4] Mangold, Stefan, Choi, Sunghyun, Hiertz, Guido R., Klein, Ole and Walke, Bernhard "Analysis of IEEE 802.11E for QoS Support in Wireless LAN", *IEEE Wireless Communications*, pp40-50, December 2003.

[5] Bianchi, Giuseppe "Performance Analysis of the IEEE 802.11 Distributed Coordination Function", *IEEE Journal on Selected Areas in Communication*, pp535-547, March 2000.

[6] Ergen, Mustafa and Varaiya, Pravin, "Throughput Formulation and WLAN Optimization in Mixed Data Rates for IEEE 802.11 DCF Mode", *IEEE Global Telecommunications Conference Workshops,* pp266-269, December 2004.

[7] Ziouva, Eustathia and Antonakopoulos, Theodore "CSMA/CS performance under high traffic conditions: throughput and delay analysis", C*omputer Communications,* pp313-321, February 2002.

[8] Tao, Zhifeng and Panwar, Shivendra "An Analytical Model for the IEEE 802.11e Enhanced Distributed Coordination Function", *The 13th IEEE Workshop on Local and Metropolitan Area Networks,* pp39-44, April 2004.

[9] Bing, Benny *High-Speed Wireless ATM and LANs,* Artech House, 2000, pp74-103.

[10] Stallings, William *Wireless Communications and Networ*ks, Prentice Hall, 2002, pp449-477.

[11] Press, William H., Teukolsky, Saul A., Vetterling, William T., Flannery and Brian P. *Numerical Recipes in C*, Cambridge University Press, 2002, pp43-50, pp379-382.

[12] Sun Microsystems Inc., “Java 2 SDK Documentation - Version 1.4.2” http://java.sun.com/j2se/1.4.2/docs/index.html, 2003. (6th August 2005).

[13] KickJava, http://www.kickjava.com/ (6th November 2005).

[14] Mathworks & National Institute of Standards and Technology (NIST), “Jama: A Java Matrix Package - Version 1.0.1” http://math.nist.gov/javanumerics/jama/, 1999. (3rd January 2006).

[15] JFreeChart, http://www.jfree.org/jfreechart/index.php. (6th August 2005).

Abbreviations and Acronyms

ACK	Acknowledgement
AIFS	Arbitration IFS
AP	Access Point
BSS	Basic Service Set
CFP	Contention Free Period
CP	Contention Period
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
DCF	Distributed Coordination Function
DIFS	DCF IFS
DSSS	Direct Sequence Spread Spectrum
EDCA	Enhanced Distributed Channel Access
EIFS	Extended IFS
FHSS	Frequency Hopping Spread Spectrum
GUI	Graphical User Interface
HCCA	HCF Controlled Channel Access
HCF	Hybrid Coordination Function
IEEE	Institute of Electrical and Electronic Engineers
IFS	Inter Frame Space
LAN	Local Area Network
MAC	Medium Access Control (layer)
MPDU	MAC Protocol Data Unit
MSDU	MAC Service Data Unit
NAV	Network Allocation Vector
PC	Point Coordinator
PCF	Point Coordination Function
PHY	Physical (layer)
PIFS	PCF IFS
QoS	Quality of Service
RTS/CTS	Request To Send/Clear To Send
SIFS	Short IFS
WLAN	Wireless LAN