Performance Improvements to the 802.11 Wireless Network Medium Access Control Sub-layer

A thesis presented in partial fulfilment of the
requirements for the degree of

Masters of Engineering
in
Computer Systems Engineering

at Massey University, Palmerston North,
New Zealand.

Michael Philip Morrison

2005

ABSTRACT

This thesis presents the outcome into the research and development of improvements to the 802.11 wireless networking medium access control (MAC) sublayer. The main products of the research are three types of improvement that increase the efficiency and throughput of the 802.11 protocol.

Beginning with an overview of the original 802.11 physical layer and MAC sub-layer standard, the introductory chapters then cover the many supplements to the original standard (including a brief on the future 802.11n supplement). The current state of the 802.11 MAC sub-layer is presented along with an assessment of the realistic performance available from 802.11. Lastly, the motivations for improving the MAC sub-layer are explained along with a summary of existing research into this area.

The main improvement presented within the thesis is that of packet aggregation. The operation of aggregation is explained in detail, along with the reasons for the significant available throughput increase to 802.11 from aggregation. Aggregation is then developed to produce even higher throughput, and to be a more robust mechanism. Additionally, aggregation is formally described in the form of an update to the existing 802.11 standard.

Following this, two more improvements are shown that can be used wither with or without the aggregation mechanism. Stored frame headers are designed to reduce repetition of control data, and combined acknowledgements are an expansion of the block acknowledgement system introduced in the 802.11e supplement.

This is followed by a description of the simulation environment used to test the three improvements presented, such as the settings used and metrics created. The results of the simulations of the improvements are presented along with the discussion. The developments to the basic improvements are also simulated and discussed in the same way.

Finally, conclusions about the improvements detailed and the results shown in the simulations are drawn. Also at the end of the thesis, the possible future direction of research into the improvements is given, as well as the aspects and issues of implementing aggregation on a personal computer based platform.

ACKNOWLEDGEMENTS

Firstly, I would like to thank my supervisor and co-supervisors - in no particular order, Firas Al-Ali, Amal Punchihewa and Liyanage De Silva. They have given me guidance throughout, and have been invaluable. Without them this thesis and the research that it concludes would have been impossible.

Secondly I would like my family and my friends. I have unfortunately not had much time to see you all during the past year. I wish to thank you for the support that you have given me without the slightest hesitation. Thank you so much.

I would also like to personally thank Matthew Sinclair who, as a fellow research student and as a long time friend, has provided everything I could wish for in a friend and colleague. You have always been a faithful companion, and may it continue for a long time to come. Good luck in your future endeavours.

Last, and definitely not the least, I wish to give my warmest thanks to my long suffering partner. She has endured my venture into this research without the slightest hesitation. I am eternally indebted for her support and love given throughout. Keren, I love you.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF EXCERPTS

1 INTRODUCTION

The introduction to this thesis covers both the literature survey, and the background information, beginning with the general background and scope of the research. Then the remaining introduction is divided into three separate and distinct chapters following the overall introduction that provide a more detailed look at the facts of wireless networks.

The first of these covers the Institute of Electronic and Electrical Engineers (IEEE) 802.11 family of wireless local area network standards, where they come from and where are they headed. Secondly, the current 802.11 Medium Access Control (MAC) layer is analysed to find where and why it needs attention, and thirdly a summary of the existing ideas proposed for the improvement of the MAC sub-layer is presented.

1.1 BACKGROUND

The initial intention for this research was formulated during the years leading up to the conclusion of my undergraduate course. I developed an interest in wireless networks, and began to look beneath the surface, seeking the ability to understand the features of wireless networks - in particular the IEEE 802.11 standard used for Wireless Local Area Networks (WLAN).

Perhaps the most interesting facet of this topic was the use of 802.11's ad-hoc mode to create a wireless mesh network. Wireless mesh networks need no existing infrastructure – rather they simply and solely use the stations (wireless network devices) themselves. If two stations are out of range of each other, they form a route through other stations.

Many of these networks have been created across the globe. These have been largely private undertakings by groups of friends, neighbourhoods, municipal councils and many other organisations. They exist to improve connections between the members of these groups, or in many cases to create connections where none existed before.

However, while the connections are created, the throughput offered over the connections is often not great, and can be much lower than the notional maximum speed. This research was initially started to understand why the notional speeds were never attained in real world deployments. Once the reasons were found and understood, it was hoped that improvements could be found that would rectify the problem and provide a better service to the users of WLANs.

The main problem this research is targeted towards is the improvement of the throughput of 802.11 wireless networks. The improvements investigated within the research are also presented alongside some other improvements to the 802.11 standard that are focused upon other performances metrics – for example, greater priority for multimedia traffic.

The improvements detailed within this research, and the other improvements mentioned, have the goal of improving the user experience of 802.11 wireless networks. All of the improvements are part of an ongoing push for better performance, greater efficiency, improved robustness and stability and an overall maturity for the collection of 802.11 standards.

1.2 CONTENT OF THESIS

Firstly, the overall 802.11 standard is introduced as part of the IEEE's group of 802 networking standards. This is followed with an overview of the general operation of 802.11, as well as descriptions for the wide range of physical and MAC sub-layer supplements. This includes the 802.11a, 802.11b and 802.11g physical layer extensions, and the 802.11e MAC sub-layer extension. Also, the next generation version – 802.11n – is discussed.

After this, the need for improving the MAC sub-layer is detailed. This includes a look at the realistic performance delivered by the current set of extensions, and a look at the typical traffic distribution likely over a wireless network. This is followed by the current state of research into improving the MAC sub-layer, as well as an assessment of chipsets that claim to implement improvements.

Then the main improvement targeted in this research – packet aggregation – is introduced as a mechanism to significantly reduce the overheads of the 802.11 operation, and thus improve throughput. Three chapters are dedicated to aggregation, with the first detailing the two main types of aggregation and their respective operation, as well as how packet queues operate within an aggregator. This is followed by a chapter dealing with the development of specific parts of the aggregation mechanism in order to gain efficiency and cope with interference. The third chapter formally describes aggregation as an update to the existing 802.11 standard using the same formal description methods as the standard specification.

Then two more improvements to the MAC sub-layer are presented. Combined acknowledgements reduce the overhead incurred by the acknowledgement system of 802.11, thus giving an increase in the possible maximum throughput. A stored frame header system aims to improve the performance of static wireless links, such as a fixed point to point wireless link.

The design and details of the simulation environment for the assessment of the improvements is described. This is followed by a presentation and discussion of the results of the simulation of the improvements.

Finally, conclusions are made both against the objectives of the research presented by the thesis, and about the outcome of the individual improvements. Also included are comments about the future direction of this research.

2 STATE OF 802.11 STANDARDS

This research is focused on the re-design of the MAC sub-layer used in IEEE 802.11 wireless networks, with the aim of improving the data throughput of wireless networks. The very first task to undertake for the research was to understand the existing IEEE 802.11 wireless networking standards and protocols. This included the original 802.11 standard (first ratified in 1997 and extended in 1999) [1], as well as the three main supplementary extensions of this standard - 802.11a [2] and 802.11b [3] that were approved in 1999, and 802.11g [4] that was approved in 2003. There are also several other extensions that have all focused on either improving some feature of 802.11, or have added functionality to the existing standards.

As well as this, investigations as to the future direction of the 802.11 WLAN standards were carried out. The next major milestone is the proposed IEEE 802.11n standard [5] that targets higher data transfer rates than all existing standards, and has just commenced the standards development process. The standard is expected to be ratified by the relevant IEEE standards committee in 2006.

2.1 INTRODUCTION TO IEEE 802 AND IEEE 802.11

The IEEE 802 family of standards define standards for local and metropolitan area networks. More precisely, these standards are designed to provide telecommunications and information exchange between systems over LANs (local area networks - privately owned, covering areas of a few square kilometres at maximum, with systems between 10 and 150 metres apart) and MANs (metropolitan area networks - privately or publicly owned, covering all or most of a city, with systems typically a few kilometres apart).

2.1.1 OSI Basic Reference Model

All of the IEEE 802 standards operate within the Open Systems Interconnection (OSI) Basic Reference Model (ISO/IEC 7498-1:1994) as defined by the International Organisation for Standardisation (ISO). The OSI model divides networking technology into seven hierarchical layers to reduce design complexity (see Figure 2.1). IEEE 802 standards deal with the lowest two layers - the physical and data-link layers.

Some of the standards within the IEEE 802 family define Management and overall Architecture (for both physical and data-link layers together), as well as Logical Link Control and Bridging (forming the upper part of the data-link layer) (see Figure 2.2). The rest of the standards define medium access technologies, and the associated physical media technologies. Each of these standards is designed for a specific application, with new standards often in development. Examples of these applications are Ethernet (802.3), Token Ring (802.5) and Wireless Personal Area Networks (802.15).

Figure 2.1 shows the basic layout of the OSI basic reference model. At the very bottom is the actual physical medium across which information is transmitted, and at the very top are the software applications where the need for communications arises. The model itself does not specify any services, protocols or implementation details to be used in any layer. Rather, it outlines the functions that each layer should be able to carry out. Standards that fit into this model, such as the 802 standards, specify the services that need to be implemented in order to carry out these functions.

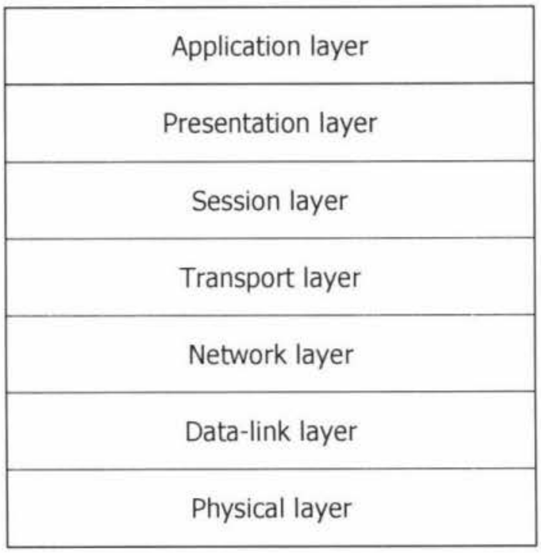


Figure 2.1: The OSI basic reference model

Figure 2.2 shows the relationship of several of the different 802 standards, and how they fit within the physical and data-link layers of the OSI basic reference model. As the diagram shows, the 802.11 subsection of standards lies over the physical layer and the bottom half of the data-link layer (also known as the Medium Access Control sub-layer). The basic operation of each subsection such as 802.11 is to take the packets from the Network layer via the Logical Link Control as MAC Service Data Units (MSDUs - henceforth referred to in the thesis as "packets"), and encapsulate them into MAC Protocol Data Units (MPDU), or frames (henceforth referred to in the thesis as "frames"), that can be sent over the physical layer.

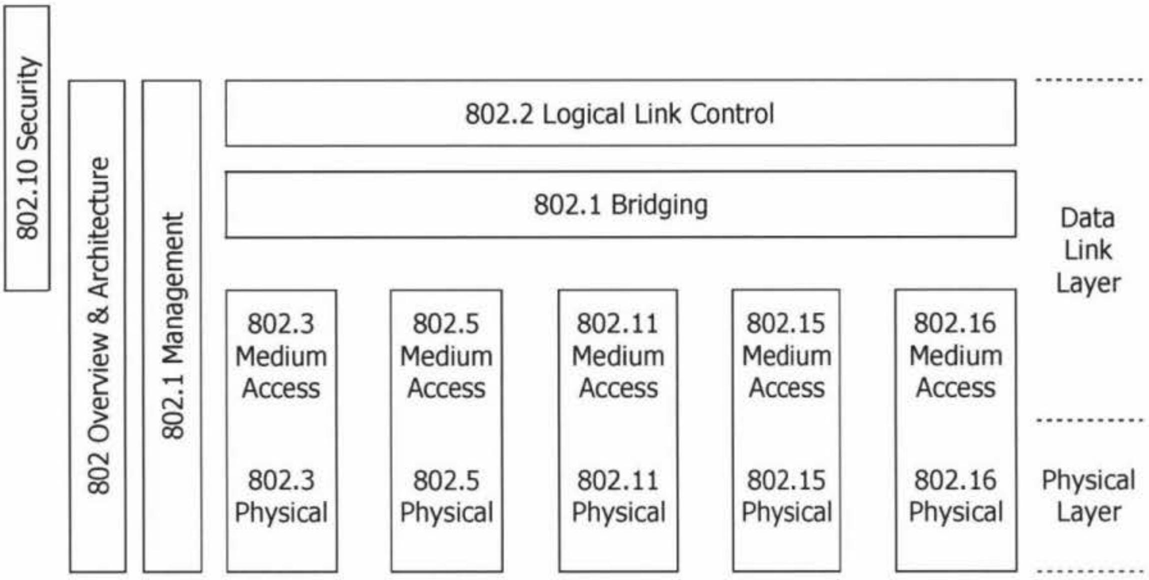


Figure 2.2: IEEE 802 Standards Family as from [4]

2.1.2 IEEE 802 Part 11

802.11 – more formally known as Part 11 of the IEEE 802 family of standards - defines the specifications for the Medium Access Control (MAC) and the Physical Layer (PHY) used in WLANs. This standard was originally ratified by the IEEE in 1997 (IEEE Std 802.11-1997), with a new edition released in 1999 (IEEE Std 802.11-1999). The primary change in the new edition of the standard was the removal of redundant management items, and changing the management systems to comply with the Simple Network Management Protocol (SNMP). The only other changes were minor document alterations.

The 802.11 standard defines the protocols and interconnections of stations via the "air" interface – using signals transmitted over either infrared or radio. These stations are connected in a LAN using the Carrier Sense, Multiple Access protocol with Collision Avoidance (CSMA/CA) to share the wireless medium used. There are two modes of operation supported by the MAC: stations operate under control of a central access point or management station (henceforth referred to in the thesis as "access point"), called "infrastructure mode"; and with all stations connecting with each other independently of any central station, called "ad hoc mode".

Whether operating in infrastructure or ad-hoc mode, stations are controlled by a coordination function. This is by default a distributed operation, but it may also be centralised in the case of infrastructure mode. Due to the nature of radio transmission, it is impossible to accurately define a collision domain in 802.11 as in its popular cousin Ethernet (802.3). Instead, 802.11 defines the Basic Service Set (BSS) – which is the group of stations controlled by a single coordination function. This definition is often extended to say that a BSS is a group controlled by a centralised coordination function (control by an access point), and an Independent Basic Service Set (IBSS) is a group controlled by a distributed coordination function.

In conjunction with this is a BSS Basic Rate Set that contains the set of data transmission rates that all stations in the BSS can use to receive and transmit data. All physical layer and MAC sub-layer headers or frames will use a rate from this set unless it is specified on a per-connection basis. This is achieved though a handshake mechanism where both sender and receiver may agree on a rate outside of the Basic Rate Set, thus forming an Operation Rate Set.

The protocol includes several services including authentication, association (and reassociation), optional encryption procedures, power management to reduce power consumption for mobile devices, and both distributed and point coordination functions that control the transfer of data. The standard also includes the definition of the Management Information Base (MIB). The MIB comprised the objects, attributes, actions and notifications required to manage a station using the IEEE 802.11 protocol. The functions of the MAC protocol are also formally described using the Specification and Description Language (SDL).

2.2 802.11 PHYSICAL LAYER

In the original IEEE 802.11 PHY, the infrared implementation supported a Basic Rate Set of 1 and 2 Mbps. The radio implementation also supported the same 1 and 2 Mbps data rates, using either frequency-hopping spread spectrum (FHSS), or direct sequence spread spectrum (DSSS). Both versions of the radio implementation operated in the 2.4 GHz Industrial, Scientific and Medical (ISM) band. The 2.4 GHz band is one of three ISM frequency bands, which were originally set aside for non-commercial industrial, scientific or medical use internationally. The amendments 802.11a, 802.11b and 802.11g all extend the operation of the original 802.11 PHY to allow much higher data rates.

2.2.1 802.11b

The basis of the 802.11b supplement – named the Higher-speed Physical Layer Extension in the 2.4 GHz Band - is the extension of the transmission bit rate. Minor changes were made to the MAC and the MAC/PHY service access point (SAP) to accommodate the addition of extra transmission rates, but otherwise the MAC was left unchanged. The bulk of the 802.11b supplement is in the extension of the DSSS transmission scheme. This 'higher speed' extension added 5.5 and 11 Mbps data rates to the 1 and 2 Mbps rate already in the Basic Rate Set, with a complementary code keying (CCK) modulation scheme providing better performance, and less susceptibility to multipath-propagation interference than the modulation in use by the original 802.11 standard.

The physical layer headers and preambles where kept unchanged to allow interoperability with the existing 802.11 standard. Lastly, several optional features were specified that were intended to be implemented once they became economically feasible. These options included a packet binary convolutional coding (PBCC) modulation scheme to replace CCK, a shorter PHY preamble mode, and provision for channel agility.

2.2.2 802.11a

The 802.11a supplement, like 802.11b, specified an extended transmission system, with only wording changes to the MAC standard to accommodate for the extra data rates specified. 802.11a introduced the use of a different frequency band – the 5 GHz ISM band – compared with the 2.4 GHz of 802.11 and the 802.11b supplement. As well as using a different band, 802.11a introduced orthogonal frequency division multiplexing (OFDM).

As the frequency band was entirely different to existing 802.11 PHYs, 802.11a supports a completely different Basic Rate Set of 6, 9, 12, 18, 24, 36, 48 and 54 Mbps. Also, most of the timing details were reduced to be more suited to the transmission rates specified. The OFDM scheme for 802.11a uses 52 sub-carriers modulated using either binary phase shift keying (BPSK), quadrature phase shift keying (QPSK), 16-quadrature amplitude modulation (16-QAM) or 64-QAM. Also, the use of forward error correction (FEC) coding is made, utilising convolutional coding at rates of $^1/_2$, $^2/_3$, or $^3/_4$.

The shift to operating 802.11a wireless networks in the 5 GHz band offers two main advantages. The first is that many other devices use the 2.4 GHz band (significant examples are Bluetooth wireless networks, many types of digital cordless telephones, and garage door openers), whereas the 5 GHz is relatively uncluttered in terms of spectrum usage. Secondly, the physical layer can achieve a higher bit rate using the same modulation scheme compared to hardware using a lower frequency.

2.2.3 802.11g

The third and final PHY supplement to the 802.11 standard is 802.11g. This supplement was designed to take some of the advantages of the 802.11a PHY and combine them with the less expensive 802.11b equipment. In this way, 802.11g remained backwards compatible with the large installed base of 802.11b hardware, but also delivered the higher data rate previously only available with 802.11a.

All of the 802.11b data rates and transmission schemes are included in the 802.11g PHY, as well as the addition of the Basic Rate Set rates as 802.11a. Also, a two extra transmission rates were added, bringing the size of the Basic Rate Set up to 14, as follows: 1, 2, 5.5, 6, 9, 11, 12, 18, 22, 24, 33, 36, 48, and 54 Mbps. The rates added from 802.11a use a modulation mode known as DSSS-OFDM, while the two extra rates exclusive to 802.11g (22 and 33 Mbps) use the PBCC modulation mode specified in 802.11b for later use. The hybrid DSSS-OFDM mode uses the same DSSS system as 802.11b for the physical layer preamble and header, with OFDM used for the actual transmission of the MAC sub-layer data.

The 802.11g PHY caters for back-compatibility with 802.11b networks. If an 802.11b station enters into an 802.11g network, the network automatically will revert the Operational Rate Set to the same as the 802.11b Basic Rate Set, ensuring full compatibility. 802.11g incorporates a number of changes to the MAC sub-layer to handle the extended set of rates and the backwards compatibility. However, these changes are solely to the timing and station capability report features of the MAC, and as such the changes do not change the underlying MAC mechanism.

2.2.4 The next standard - 802.11n

The proposed IEEE 802.11n standard targets higher data transfer rates than is previously available from the existing standards. This standard has just entered the development phase, and is expected to be completed and ratified some time in 2006, and will specify changes to both the physical layer and the MAC sub-layer. The stated aim for 802.11n is to have at least 100 Mbps at the MAC sub-layer service access point, with an envisaged physical data rate of at least 160 Mbps to achieve this throughput. The specification of a data throughput at the MAC service access point is new for 802.11 standards – previously, only the physical layer transmission rates have been stated, which are always higher than the actual data throughput achievable.

As the standard is still in the early stages of development, little can be said about its details. Almost certain though is that the PHY will be based upon Multiple Input, Multiple Output (MIMO) OFDM. MIMO is the use of more than one antenna at both the receiving and transmitting stations – set up in such a way that multiple transmissions can be made simultaneously. The OFDM system will be similar in principle to that used in 802.11a, but with better signal processing power for better performance. It is also likely that 802.11n will use the 5 GHz band, for the same reasons as 802.11a.

2.2.5 Graphical Comparison of Physical Layer Standards

Figure 2.3 shows the relationship of all of the physical layer standards in the 802.11 family, with the frequency band utilised, maximum data rates and approximate maximum likely throughput above the MAC sub-layer all shown (details about these figures will be given in section 3). The original 2 Mbps 802.11 standard specified in 1997 used the 2.4 GHz ISM band, and was superseded by the 11 Mbps 802.11b standard in 1999. 802.11b was further superseded by 802.11g in 2003 that offered a maximum data rate of 54 Mbps. 802.11a is the only current 802.11 standard to use 5 GHz, and has offered 54 Mbps since 1999, but at a much higher cost. The future 802.11n standard is also likely to use the 5 GHz frequency band to deliver a maximum data rate well over 160 Mbps to guarantee greater than 100 Mbps throughput above the MAC sub-layer, and be backwards compatible with 802.11a.

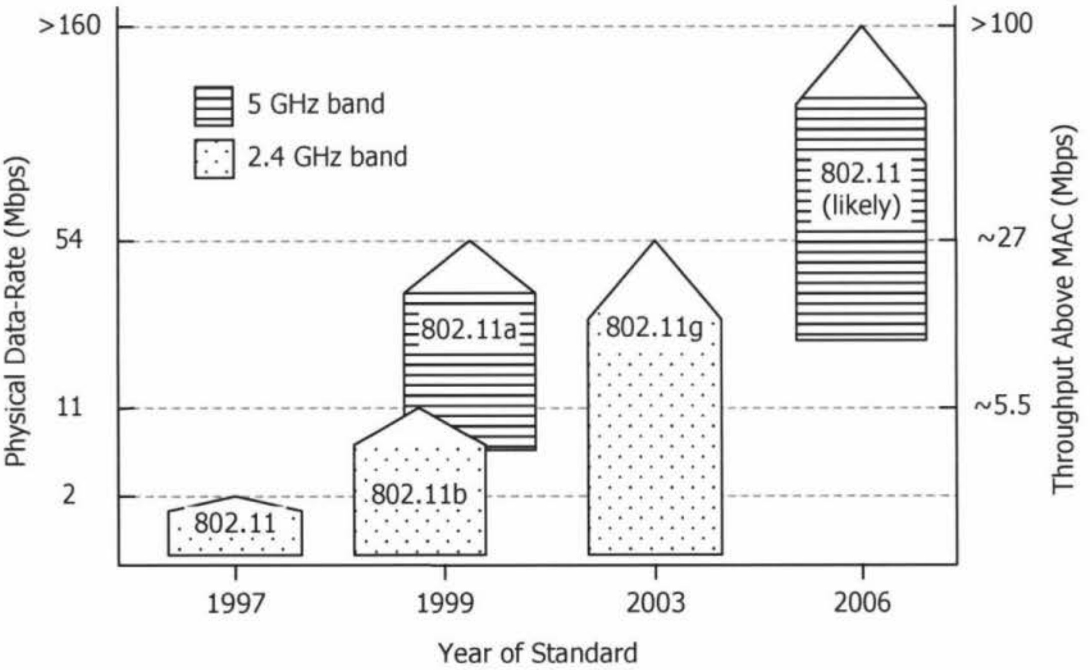


Figure 2.3: Comparison of the different 802.11 Physical layer standards

2.2.6 Other Physical layer Supplements

In addition to the three major physical layer supplements, other supplements have also been ratified by the IEEE. The first of these was 802.11d in 2001 [6]. This did not change the technology, but extended the MAC to control the physical layer operation in a way that would conform to regulatory environments other than those originally specified in the 802.11 standard.

Then in 2003, 802.11h was finalised, covering the management of spectrum and transmission power levels for 802.11 WLANs operating in the 5 GHz band (e.g. 802.11a, and in the future 802.11n) in Europe [7]. This standard resolves issues that had arisen from European Union rules on power levels for consumer equipment in the 5 GHz band.

Most recently, in late 2004, 802.11j was ratified [8]. This supplement covered the operation of an 802.11 physical layer over 4.9 GHz and 5 GHz. The standard specified new operating modes that use slightly lower frequencies than 802.11a. Additionally, the 802.11j has the ability to switch between 802.11a and 802.11j frequencies, as well as change the channels spectral characteristics and specific indoor/outdoor operation modes to improve performance. These frequency bands were set aside in 2002 in Japan for use with 802.11 networks, and thus 802.11j is primarily targeted at Japan, but other countries have also decided to designate these bands for use with 802.11j too.

2.3 802.11 MAC SUB-LAYER

The 802.11 MAC sub-layer was specified in the original 802.11 standard from 1997. Since then, there have been no major changes to the MAC sub-layer. In fact, other than the minor changes outlined in the previous sub-sections to cater for new and diverse physical layers, and two amendments that add security (802.11i) and quality of service (802.11e) features on top of the existing operation (these are covered in more detail later), there have been no changes whatsoever. This means that the underlying operation of the MAC sub-layer has remained unchanged for 7 years, while the physical layer has improved considerably.

2.3.1 MAC Sub-layer Services

The MAC sub-layer performs 9 services in total – 6 to support the delivery of frames across a WLAN, and 3 to support access and confidentiality. To perform the services, three distinct frame types are used – data frames, control frames and management frames. There are also two different categories of services – station services (4 of the services), and distribution system services (the other 5 services).

The 5 distribution system services (DSS) are provided by access points to the other stations in the WLAN. The distribution system (DS) is used to combine two or more BSSs into an Extended Service Set (ESS). Each BSS has one access point, and the DS is essentially the network between these access points. The main DSS is Distribution – the passing of any message from a station through the DS to another station. This also includes the case when an access point relays a message from one station in a BSS to another station in the same BSS. The second

DSS is Integration – where the destination is not contained within another BSS, but rather in an integrated wired network that is accessed via a portal. This includes messages to and from the Internet via an access point or a portal on the DSS.

The last three DSSs are Association, Reassociation and Disassociation. Association links a station with a particular access point, which the DS also uses as a mapping (i.e. a message intended for a particular station is sent to its associated access point). Reassociation handles mobility – where a station moves from one access point to another on the same DS. Disassociation simply removes the association of a station from a particular access point.

The 4 station services are more basic than the DSSs, and are provided by all stations on a WLAN, including access points. The first two station services are Authentication and Deauthentication. Authentication aims to bring the assumed physical security of a wired link to wireless networks. This is done solely on a link-level basis, and no communication will be made unless a mutual level of authentication is made. The 802.11 standard does not specify a specific authentication scheme, but does however specify an optional shared key system – Wired Equivalent Privacy (WEP).

Since the original 802.11 standard was released, both the basic authentication system and the WEP system were found to be relatively insecure. This prompted the creation of the 802.11i supplement, which was ratified in 2004 [9]. This primarily specified the inclusion of Advanced Encryption Standard (AES) and a key integrity system to greatly enhance the security of 802.11. 802.11i did not change any other feature of the MAC sub-layer.

Deauthentication is simply a notification service, and is initiated either by a station leaving the network, or by an access point wishing to terminate a connection with a station. Deauthentication, as a notification service, cannot be refused.

The third station service is Privacy. WEP (and later the 802.11i improvements), in the same way as with authentication, can be used to encrypt all transmissions. All stations and access points will have this service, but it is disabled by default.

The last station service is perhaps the most important – MSDU Delivery. This service is responsible for delivering the packets (MSDU), over the wireless medium. The MAC sub-layer must create a frame that encapsulates the packet, adding a header and Frame Check Sequence (FCS) in the process. This is then sent to the physical layer for transmission.

This last service is the one focused on primarily in this research, as the aim of the research is to improve the data throughput of WLANs – or in other words, to improve the speed and efficiency of the MAC sub-layer MSDU Delivery station service.

2.3.2 Medium Access and Coordination Functions

The fundamental technique used by the 802.11 MAC sub-layer to access the medium is a distributed coordination function (DCF) called Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA). All stations in any 802.11 network have an implementation of this DCF, and this is used for all ad-hoc WLANs (IBSS is the technically correct term, but for readability the term 'ad-hoc' is used). WLANs set up to use infrastructure mode may also use DCF, but have the option of using a centralized coordination function, which is described later. In addition, irrespective of coordination function used, all stations use an immediate positive acknowledgment system where retransmission is attempted by the sender if no acknowledgement frame is received. The acknowledgement frame is sent after a time period called the Short Inter-frame Space (SIFS). Inter-frame spaces are used extensively throughout the MAC timing systems to give known time gaps independent of the data rate in use.

The DCF defines two access mechanisms used in transmission, basic CSMA/CA access and a Request-To-Send/Clear-To-Send (RTS/CTS) 'handshake'. For the basic CSMA/CA mechanism, a station with a frame to transmit senses the activity of the wireless medium both physically and virtually – physically by checking the transmitter/receiver state, and virtually by means of a Network Allocation Vector (NAV, a prediction of future traffic on the network based on duration information in control frames received previously). If the medium is found to be idle, the station can transmit. If the medium is sensed to be busy, the station monitors the medium until it becomes idle for more than a Distributed (coordination function) Inter-Frame Space (DIFS) time period.

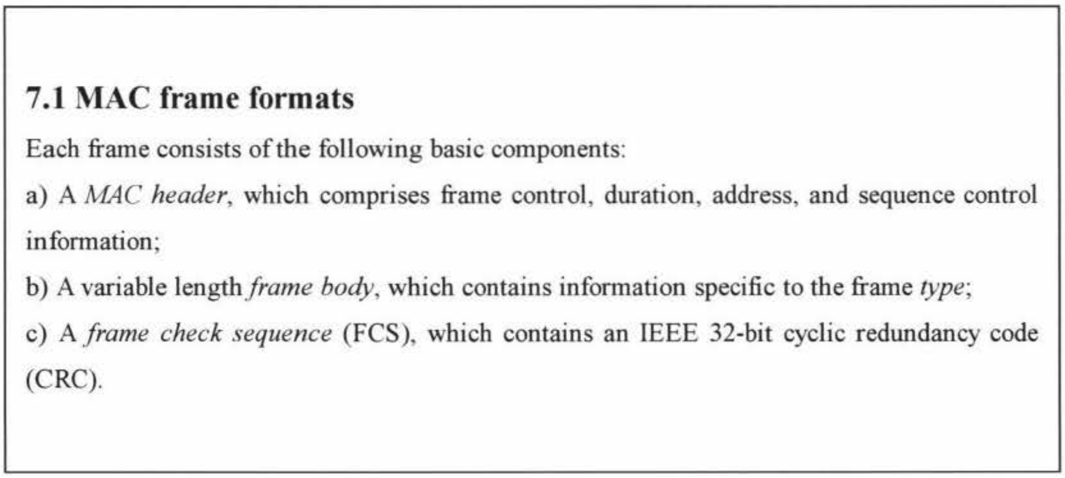
To minimise collisions the station then waits a random backoff interval before transmitting the data frame. This is part of the collision avoidance measures of CSMA/CA. This backoff interval is selected randomly between 0 and the stations Contention Window (CW) parameter. CW always takes values of integer powers of 2, minus 1. Initially this is set at *CWmin* – specified in the standards as 15 for 802.11, 802.11a, 31 for 802.11b, and either 15 or 31 for 802.11g (15 in a pure 802.11g network; 31 in a mixed 802.11g/802.11b network). In the event of an unsuccessful transmission, the value of CW is set at the subsequent integer power of 2, minus 1, up to a maximum value *CWmax* – specified as 1023 for the original 802.11 standard and all supplements. Therefore, the total set of values that CW can take is {15, 31, 63, 127, 255, 511, 1023} for all networks apart from 802.11b where *CWmin* is 31.

This basic DCF scheme can be extended based upon the packet size being sent. This is controlled by a variable in the MIB – dot11RTSThreshold. The packet size is compared to this threshold, and if it is larger, a Request to Send/Clear to Send (RTS/CTS) reservation scheme is used. In this scheme, short RTS and CTS frames are exchanged to reserve the medium prior to the transmission of the data frame. The RTS/CTS scheme shortens the collision duration (collisions only happen over the period of the short RTS and CTS frames) and also copes with hidden stations.

As well as DCF, 802.11 specifies a point coordination function (PCF). As opposed to DCF, the implementation of PCF is optional. PCF provides contention-free channel access, and is controlled using polling by the access point (therefore PCF cannot be used for ad-hoc networks). PCF, if used, has higher precedence by way of the PCF interframe space (PIFS) being less than the DIFS. Once the access point has control of the medium, it polls each station on a polling list, and grants channel access if it is needed. If PCF is used, there are alternating periods of PCF (with no contention) and DCF (with contention).

2.3.3 Frame Formats

Section 7.1 of the 802.11 standard defines the basic components of the MAC sub-layer frames, as shown below.



7.1 MAC frame formats

Each frame consists of the following basic components:

a) A *MAC header*, which comprises frame control, duration, address, and sequence control information;

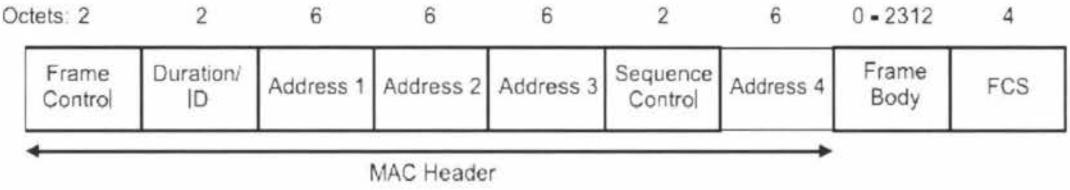
b) A variable length *frame body*, which contains information specific to the frame *type*;

c) A *frame check sequence* (FCS), which contains an IEEE 32-bit cyclic redundancy code (CRC).

Excerpt 2.1: Basic components of a MAC sub-layer frame – from 802.11 [1]
(Note: numbers in the excerpt are from the source and do not refer to the thesis)

For example, a MAC data frame is a packet passed to the MAC by the upper layers in the OSI model (the frame body) encapsulated in a header (the MAC header) and trailer (the FCS).

The general frame format including the MAC header and the FCS is described diagrammatically in the standard as shown in Figure 2.4.



Octets:	2	2	6	6	6	2	6	0 - 2312	4
	Frame Control	Duration/ ID	Address 1	Address 2	Address 3	Sequence Control	Address 4	Frame Body	FCS

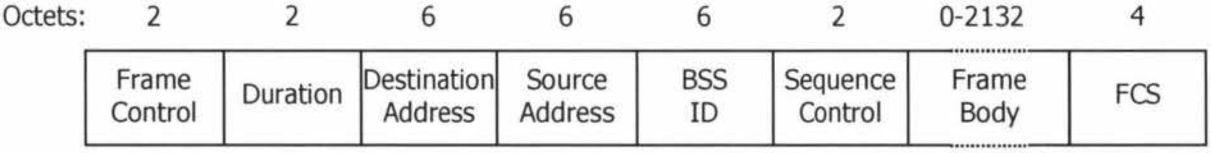
Figure 2.4: General frame format as from [1]

The frame body is specified to have a maximum size of 2312 octets. In my simulations I used a maximum of 1500 octets, as this is the maximum packet size used in Ethernet (802.3) networks, and is the default setting on almost all wireless networking equipment for compatibility in case of mixed media networks.

The frame formats for Control and Management frames does not always contain all of the fields shown in this diagram. For instance, the RTS frame only uses the Frame Control, the Duration, Address 1 (receiver address), Address 2 (transmitter address) and the FCS. All Control and Management frame will contain at least the Frame Control, the Duration/ID and the FCS.

Data frames will contain all of the fields shown, with the exception of the Address 4 field (This is only contained if the frame has both the 'From DS' (from distribution system) and 'To DS' (to distribution system) bits in the Frame Control set to 1 – specifying that the frame is being sent using Wireless Distribution System (WDS), which is an optional wireless backbone system for frame distribution between access points. From this point forward, I will not consider the use of WDS.).

The layout of a generic MAC sub-layer data frame, sent entirely within a wireless network BSS is shown in Figure 2.5.



Octets:	2	2	6	6	6	2	0-2132	4
	Frame Control	Duration	Destination Address	Source Address	BSS ID	Sequence Control	Frame Body	FCS

Figure 2.5: MAC sub-layer Data frame (within BSS)

The destination address (DA) and source address (SA) are the MAC addresses of the MAC sub-layer entities receiving and transmitting the frame, respectively. (Note: these are not equivalent to the DA and SA used in several network layer routing protocols used for routing in wireless networks, such as AODV.)

This format is for a Data frame with both the 'From DS' and 'To DS' bits set to 0. This applies to all data frames sent within the group of stations controlled by a single coordination function (this applies to all frames sent within an IBSS, or ad-hoc network, and some frames sent within a BSS, or infrastructure network). However, it does exclude frames sent via an access point to another station within the local BSS.

Even if a frame is being relayed by an access point to a station in the same BSS, it is still logically sent across the DS. Therefore, if a frame is being sent to an access point, the 'To DS' bit would be set to 1, and the address fields would be in the order: BSS ID, SA and DA. If the frame is being sent from an access point, 'From DS' would be set to 1, and the address fields would be in the order: DA, BSS ID and SA.

2.3.4 802.11e

Aside from the 802.11i extension to the security services of the MAC sub-layer, there has only been one other supplement targeted at the MAC sub-layer. This second supplement is 802.11e, which is in its final draft and is due for ratification, is aimed at improving the quality of service (QoS) in wireless networks.

QoS is needed to deliver requirements imposed by certain applications – most significantly multimedia applications such as streaming video and voice calls. Simply increasing the date rate may not be sufficient to meet these requirements, so some form of prioritisation mechanism must be built into the MAC sub-layer. 802.11e specifies such mechanism, and realizes these by introducing two new coordination functions to replace the original DCF and PCF.

The new coordination functions are enhanced DCF (EDCF) to replace DCF, and hybrid coordination function (HCF) to replace PCF. These coordination functions ensure that services can differentiate transmissions by means of priority or a requested level of QoS. In EDCF, each station can have four Access Categories (AC) – Voice, Video, Best effort and Background – with priority in that order from Voice down to Background. Over these four ACs, up to eight User Priorities can be assigned to applications in use, with each belonging to an AC. Each AC is given an Arbitration Inter-frame Space (AIFS) that is always longer than DIFS, and equal to or longer than any higher priority AC.

This essentially creates up to four virtual stations at each station, with each AC acting as a station. All ACs on all stations contend for the medium in the same way as the DCF function described in the previous sub-section. If there is a local collision, the AC with the highest priority will transmit, and the lower ACs act as if a real collision has occurred. If there is a collision in the wireless medium, it is handled in the same way as normal. Once an AC at a station has access, it then has a Transmit Opportunity (TXOP) that is a specified time interval over which the station/AC can transmit data frames with SIFS time in between.

HCF is the centralized version of EDCF, and has a Hybrid Controller (HC) that has a higher priority than all ACs in the BSS. The HC has the ability to transfer traffic to the other stations at any time, and also allocate TXOPs to these stations in a similar method to PCF granting channel access. HCF (like PCF) also alternates between contention periods (EDCF) and HCF-assisted Controlled Access Periods (CAP). The HC also has the added ability to alter the timing of these periods, and also change the length of TXOPs given to stations.

802.11e also specifies a Block Acknowledgement scheme, whereby several frames can be acknowledged at the MAC sub-layer level in a single acknowledgement control frame. However, this must first be setup through a sequence of handshake control frames, and is only an optional implementation within 802.11e. Compounding this further is that 802.11e itself is optional, and block acknowledgements require HCF to be used – thus the scheme cannot be used for ad-hoc networks, or by stations that do not have 802.11e functionality.

On a more technical level, the block acknowledgement can only be used to acknowledge frames sent within an uninterrupted Data Block, which is limited in length by the length of the TXOP of the sending station. Although the block acknowledgement itself is not required to be sent immediately (i.e. it can wait until the next TXOP), the acknowledgement can only apply to frames sent in a single TXOP.

3 WHY IMPROVE THE MAC SUB-LAYER

The first section in this chapter covers the realistic numbers associated with the throughput of 802.11 networks. Following this is a look at the effect of the 802.11e extension to the performance of the MAC. Lastly the efficiency of the current MAC sub-layer is analysed.

3.1 REALISTIC 802.11 PERFORMANCE

3.1.1 Data Rate and Throughput

The 54 Mbps 'maximum data rate' of 802.11a/11g and 11 Mbps for 802.11b is the maximum possible speed that data in a frame can be transmitted at across the wireless medium. However, much lower rates are used for the Physical Layer Control Protocol (PLCP) data e.g. headers and preambles [2, 3, 4]. Therefore, the 'maximum data rate' that is associated with each 802.11 extension is only used for the frames sent across the network, with relatively long periods of slower rate PLCP data.

The maximum "throughput" is therefore much lower than the data rates used. For the evaluation of networking protocols, the throughput is defined as the data rate at the MAC sub-layer Service Access Point (SAP). The SAP is the interface that all packets from the layers above must go through when sent or received by the MAC sub-layer. At this point in the OSI model, none of the physical layer headers or MAC sub-layer headers count towards the throughput, but time spent transmitting them is considered. Also, time spent on any other MAC sub-layer services (see 2.3.1) is taken into account as far as number of bits per second transmitted.

A common misconception in this regard is that the MAC sub-layer control and management frames are also transmitted at lower rates than data frames. However, section 9.6 "Multirate Support" of the 802.11 standard (incorporating changes up to and including those in the 802.11g extension) specifies that:

> Some PHYs have multiple data transfer rate capabilities that allow ... dynamic rate switching. The algorithm for rate switching is beyond the scope of this standard.
>
> All control frames that initiate a frame exchange shall be transmitted at one of the rates in the BSS Basic Rate Set.
>
> Data and/or management MPDUs with a unicast receiver address shall be sent on any supported data rate selected by the rate switching mechanism.
>
> A STA responding to a received frame shall transmit [any response] at the highest rate in the BSS basic rate set that is less than or equal to the rate of the immediately previous frame in the frame exchange sequence (as defined in 9.7) and that is of the same modulation type as the received frame.

Excerpt 3.1: 802.11 Multirate support – from Section 9.6 of standards up to and including 802.11i [1 - 4]

This means that the control frames *may* be transmitted at lower rates, but this will only be caused by the rate switching algorithm chosen by a manufacturer in their network equipment or software. As this research is focused on improving the standard as a whole, the author has assumed that all control and management frames are sent at the same rate as data frames.

However, the fact that just the PLCP data is sent at a lower rate is very significant. It means that the actual data throughput for an 802.11 wireless network is much lower than the maximum data transmission rate, as the maximum data rates are not used for all transmissions.

3.1.2 Calculating the Maximum Throughput

The maximum throughput for 802.11 networks can be calculated by using a basic transactional model that assumes all transmissions use the highest possible data rate, and all transmissions are successful and error-free [10]. Also, the model assumes that there is no contention for the wireless medium, and therefore no CSMA/CA backoff periods. Although this model is too oversimplified for the real world, it still gives a valid indication of the maximum possible throughput available over 802.11.

To work out the throughput, it is necessary to calculate the timing characteristics of a maximum length TCP data packet, plus a returning TCP acknowledgement packet. By default, the maximum packet size is 1500 bytes, as although 802.11 networks can handle higher packet sizes, Ethernet has a maximum size of 1500 bytes, so most operating systems and 802.11 network devices limit the payload size to 1500 bytes in case an 802.11 network is bridged to an Ethernet LAN at some point. A TCP acknowledgement packet is 40 bytes in length.

Each TCP transaction (i.e. both the TCP data packet and the TCP acknowledgement packet) requires the station to sense the medium is free for a DIFS before sending the packet within a frame, followed by a SIFS and the positive acknowledgement control frame. On top of each TCP packet, 8 bytes of Logical Link Control and Sub-Network Access Protocol headers are added, as part of the overall 802 network system. Then finally, a 24 byte MAC sub-layer header and a 4 byte Frame Check Sequence trailer are added. In total, an extra 36 bytes is added to all packets to encapsulate it into a frame.

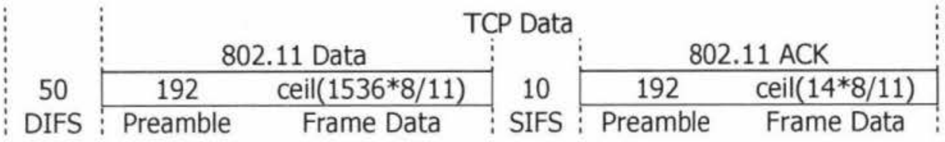
For 802.11b, the basic timing information is as follows [3]:

Table 3.1: Selected 802.11b timing details

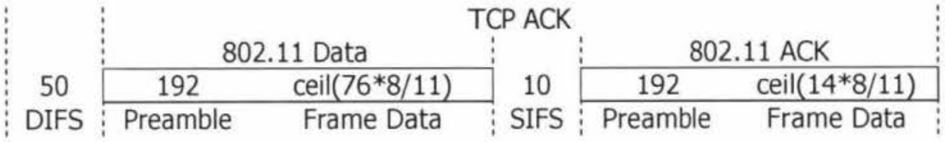
Preamble	192 μs
Slot time	20 μs
SIFS	10 μs
DIFS	50 μs
Symbol size	8 bits
Symbol duration	8/11 μs

As the symbol duration is a fraction of microseconds, the total data transmission duration is rounded up to the nearest microsecond, as implemented in the 802.11b standard. Also, the preamble time shown is known as the "long preamble", as 802.11b also specifies a "short preamble". However, the long preamble is used by default, and implementation of the short preamble is optional.

Using the simple transactional model, and the 802.11b timing information, the calculation for the time taken for a single TCP transaction (data frame and acknowledgement) can be made. The sequence of MAC frames sent across the network is shown in Figure 3.1. As the symbol size is 8 bits, the byte-size of the TCP data can be used.



(a) TCP data frame from source to destination



(b) TCP acknowledgement from destination to source

Figure 3.1: TCP data packet and acknowledgement over 802.11b

Summing up the times in this model gives:

Table 3.2: Time per 802.11b transaction

TCP data packet		
Data frame symbols		1118 μs
TCP acknowledgement		
Data frame symbols		56 μs
MAC sub-layer		
SIFS x 2	20	
DIFS x 2	100	
Acknowledgement x 2	22	
		142 μs
Physical layer		
Preamble x 4		768 μs
Time per transaction		2084 μs

At 2084 μs per 1540 byte transfer (1500 bytes of TCP data plus 40 bytes of TCP acknowledgement), 479 transfers per second are possible. This gives a maximum possible throughput of 5.76 Mbps for 802.11b.

For the 802.11a physical layer supplement, the basic timing information is shown in Table 3.3 below [2]:

Table 3.3: Selected 802.11a timing details

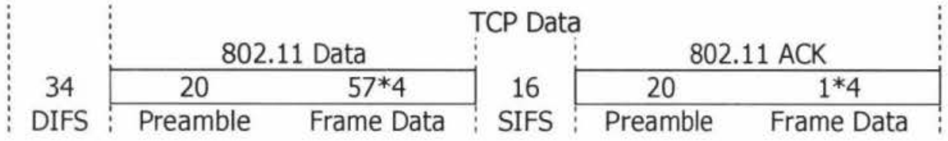
Preamble	20 μs
Slot time	9 μs
SIFS	16 μs
DIFS	34 μs
Symbol size	216 bits
Symbol duration	4 μs
Tail bits	6 bits

The physical layer preamble is made up of a 16 μs training sequence (a sequence of known symbols with set time lengths, rather than a series of bits) plus one 4 μs symbol carrying necessary information. The 6 tail bits are part of the OFDM modulation scheme used by the 802.11a supplement. The symbol size for 802.11a is 216 bytes - therefore the number of symbols per MAC sub-layer frame can be found by using the formula: ceil(bits/216).

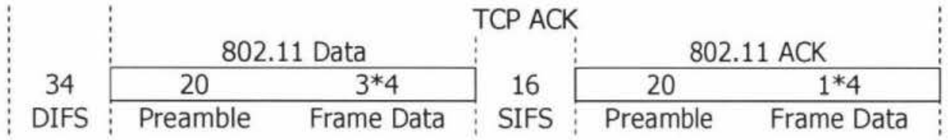
Table 3.4: Number of 802.11a symbols per MAC frame

	Bytes	Bits	Symbols
TCP Data	1536	12288 + 6	57
TCP ACK	76	608 + 6	3
MAC ACK	14	112 + 6	1

Using the simple transactional model, and the 802.11a timing information, the values used to calculate the time taken for a single TCP data frame and acknowledgement as shown in Figure 3.2.



(a) TCP data frame from source to destination



(b) TCP acknowledgement from destination to source

Figure 3.2: TCP data packet and acknowledgement over 802.11b

Summing up the times in this model gives:

Table 3.5: Time per 802.11a transaction

TCP data packet		
Data frame symbols		228 µs
TCP acknowledgement		
Data frame symbols		12 µs
MAC sub-layer		
SIFS x 2	16	
DIFS x 2	68	
Acknowledgement x 2	8	
		108 µs
Physical layer		
Preamble x 4		80 µs
Time per transaction		428 µs

At 428 µs per 1540 bytes of actual data (counting both TCP data and TCP acknowledgement), it is possible to complete 2336 transfers per second, thus giving a maximum possible throughput of 27.44 Mbps for 802.11a.

In summary, the maximum theoretical throughput for data at the MAC sub-layer SAP for 802.11b is 5.76 Mbps, and 27.44 Mbps for 802.11a and 802.11g.

In regard to the simplicity of the model used, there are several factors that must be considered when deciding what these figures mean in reality, but have been left out for simplicity. The most important of these is that there is no contention for the medium. This means that the model only describes the throughput for a single point-to-point wireless link, and assumes that no other station is transmitting at all. This is an 802.11 network in its most basic form, and the highest throughput will be seen in this configuration.

Also, the model omitted the backoff procedure for simplicity of calculation. The random backoff period of CSMA/CA is taken after the initial DIFS, and will further lengthen the time per transfer by a few slot times. When the network is in contention, the variable CW (see 2.3.2) typically increases exponentially, giving the random backoff time a greater effect on the overall transfer duration.

Lastly, the complexity of using TCP over wireless networks is ignored. Firstly, the TCP protocol uses a sliding window approach to acknowledgements that means that a data packet does not have to be followed immediately by an acknowledgement. This means that the model probably overemphasizes the effect of upper layer acknowledgements. Also, other protocols such as UDP do not require acknowledgements at all, which would markedly increase the throughput calculated from this model. On the other hand, the 802.11 MAC sub-layer cannot distinguish between related upper layer packets. Thus, a returning acknowledgement can collide with the next data packet belonging to the same TCP stream – even on the simple single point-to-point link used in this model.

3.1.3 Realistic Packets

Irrespective of the shortcomings of the model used, however, the maximum throughput speeds calculated above are only obtainable if the packets contained within each frame are close to, or equal to, the maximum allowable packet size (typically 1500 bytes). Realistically however, in a typical network, a large proportion of frames have a much smaller size.

In a twelve week study of a wireless LAN at Stanford University's Gates Computer Science Building, Baker and Tang examined over 78 million frames and found that over 70% of packets were smaller than 200 bytes [11]. Figure 3.3 shows the cumulative packet distribution found from the study. The packet distribution shows the large percentage of packets that were relatively small. Looking at the bytes transferred across the network, the packets with sizes less than 200 bytes (which accounted for greater than 70% of the packet count) only account for about 15% of the bytes transferred.

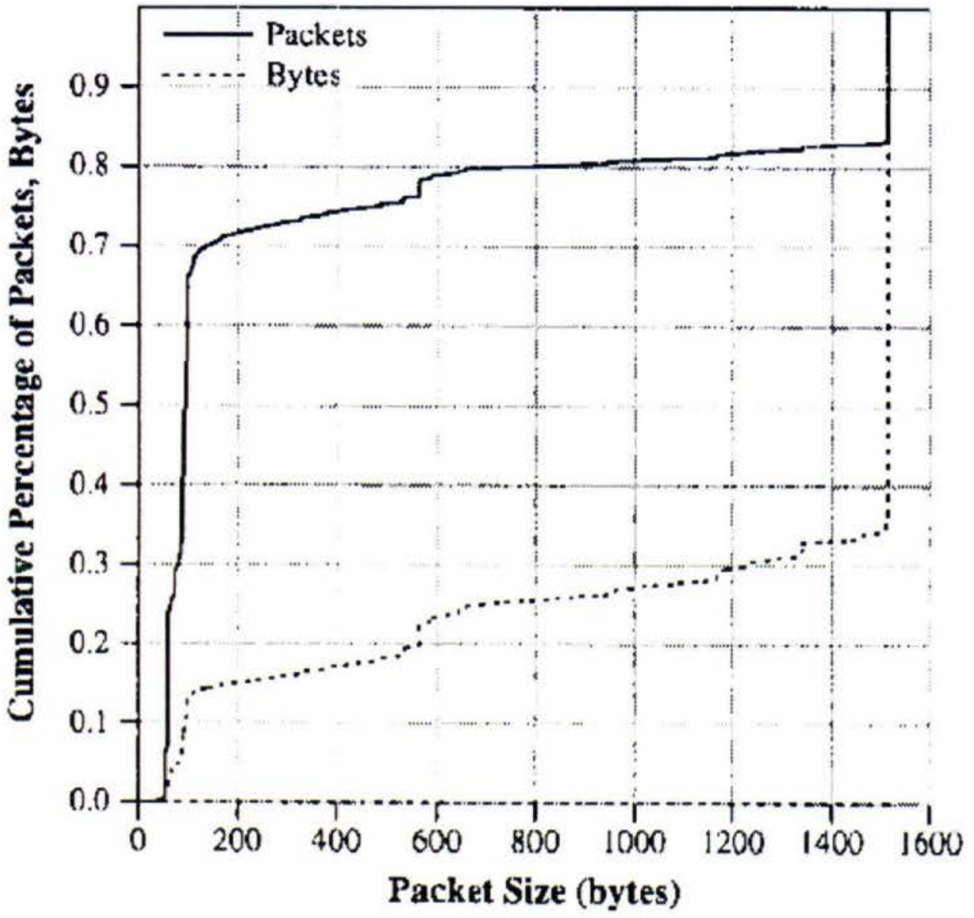


Figure 3.3: Packet Distribution as from Baker & Tang [11]

With an average packet size that is lower than the maximum packet size, the throughput drops accordingly, as the per packet overhead percentage increases. This is because every packet, irrespective of its size, requires the same headers and preambles. Therefore, the absolute per-packet overhead is identical for each packet transmitted. When the size of each packet is factored in however, the percentage per packet overhead is inversely proportional to the size of the packet. Hence, a lower average packet size gives a greater percentage per packet overhead.

In terms of the statistics of this study, the average packet size was 590 bytes, yet the median packet size is only approximately 150 bytes! The higher average can be explained by looking at the percentage of bytes transferred in packets that are the maximum size available. Even though these packets make up just over 15% of the total packets sent, they contribute to approximate 65% of the total bytes sent over the network.

In a study further expanding on that mentioned above, Dartmouth College's wireless network was studied over eleven weeks by Kotz and Essien [12]. This study covered 476 access points that were spread all 161 buildings on the campus, and recorded over 350 million frames either originating at or destined for a wireless station on the network. Although no packet distribution statistics were given, it was possible from the results to determine that the average packet size for all packets sent across the wireless network was just over 650 bytes. This is slightly larger than the average in the Tang and Baker study, but still illustrates that the average is much smaller than that needed to sustain throughputs close to those calculated in the previous sub-section.

The Kotz and Essien study looked closely at the protocols used over the wireless network. The predominant protocol is IP (Internet Protocol) – accounting for 99.7% of packets – with the difference made up almost entirely of ARP (Address Resolution Protocol) and AppleTalk. TCP packets made up 94.3% of the IP data sent, and UDP packets made up a further 5.0%, with the rest being made up of ICMP, IGMP, PIM, RSVP, and NARP protocols. The fact that the majority of packets are TCP means that there are a large number of small acknowledgement frames, as opposed to UDP which does not make use of acknowledgements.

One obstacle that was found during the research was the lack of relevant published studies into the sizes and distribution of packets sent across typical wireless networks. Although some studies have been made, it is not a very substantial number. . Also, often the studies carried out tend to have a narrow focus on a particular aim (such as the performance of a particular radio transmitter [13] or the installation and management of a WLAN [14]), rather than statistics such as packet size statistics or applications used, as noted by Baker & Tang in their paper.

3.2 EFFECT OF THE 802.11E EXTENSION

To deliver quality of service (QoS) over 802.11 WLANs 802.11e introduced the ability for the coordination functions to allow differentiation of packets with various priority levels transmitted from Access Categories (AC) within the stations (see 2.3.4). 802.11e uses the distributed EDCF to implement the differentiation by changing the timing values for the different priority traffic, or the centralised HCF to combine EDCF with the ability of a central controller to alter the priority given to the stations.

However, the 802.11e standard does not change the overall network throughput in the case of EDCF. Although the higher priority ACs usually see decreased delays and an increase in throughput, lower priority ACs in turn will experience longer delays and decreased throughput. Simulations of the draft version of 802.11e have shown that the EDCF coordination function can decrease the efficiency of the 802.11 protocol by as much as 30%, to achieve the differentiation needed to provide QoS, and does not increase the throughput of an 802.11 wireless network [15].

HCF on the other hand can be used to markedly increase the overall network throughput by introducing a higher proportion of contention free periods, thus increasing the proportion of frames transmitted without error. However, while this does increase the overall network throughput, it typically needs to be fined tuned for specific applications rather than offering an increase in the general situation [16].

The same simulation referred to above also found that HCF can be used to increase the overall network throughput by up to about 20%. This is because HCF is a very efficient way of controlling the wireless channel – as illustrated by the fact that the QoS enhancements are achieved with negligible overhead increases over PCF, thus the significant throughput enhancement. However, it cannot be used in ad-hoc wireless networks, and the state system used reduces the robustness of the 802.11 protocol [15].

3.3 MAC SUB-LAYER EFFICIENCY

3.3.1 Overhead Percentage

Figure 3.4 shows the transmission of a TCP acknowledgement packet over an 802.11b wireless network (with the backoff period omitted for simplicity, as in the transmission model used in 3.1.2). In this transmission, only the MAC frame data – the TCP acknowledgement packet plus its MAC sub-layer header, and the subsequent MAC sub-layer acknowledgement control frame – is sent at the maximum 11 Mbps, while everything else uses the lowest speed – 1 Mbps – for backwards compatibility and decreased susceptibility to errors. In the example shown, the maximum data rate is only used about 13% of the time.

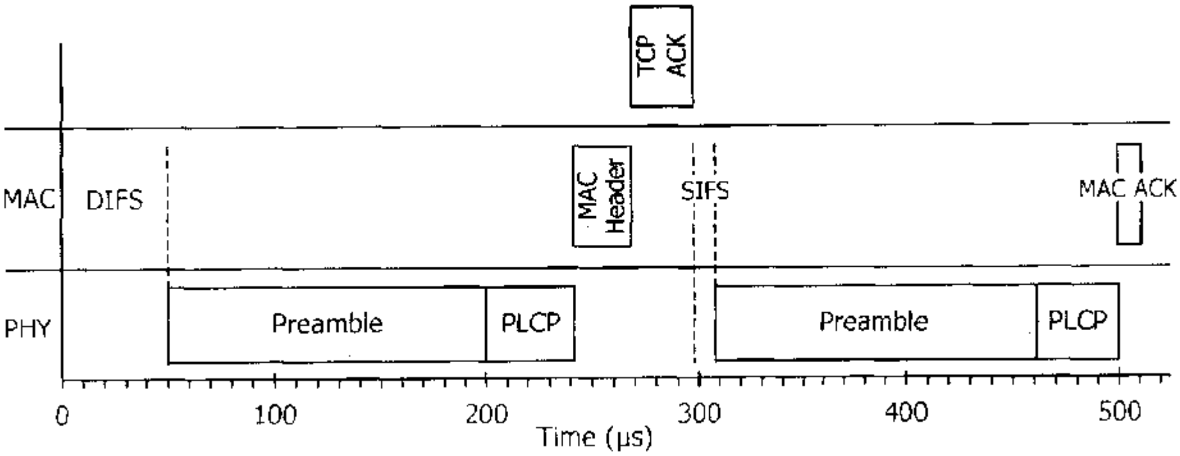


Figure 3.4: TCP ACK Transmission showing Physical and MAC overheads of 802.11b (refer to Figure 3.1 (b) for exact timing details)

Often further compounding the high percentage of overheads is the backoff procedure. If the wireless medium is in use, a station wishing to transmit defers until the medium is not busy, and then waits a DIFS time periods *plus* a random backoff deferral period. The length of this deferral period is selected at the end of the DIFS as described below, and decrements at every slot time. If the deferral period is already non-zero (i.e. the last backoff period did not decrement fully), the existing period is used, thus giving a station that has been previously deferred a higher chance of transmission.

The random backoff interval is located after the initial DIFS waiting period, and is selected as an random integer number multiplied by the slot time – with the slot time being 20 µs for 802.11b and 802.11g/802.11b mixed mode and 9 µs for 802.11a and pure 802.11g. The integer number is selected randomly from the interval [0, CW], with the controlling variable CW typically increasing with contention for the medium (see 2.3.2).

In terms of 802.11b, one slot time is 20 µs – twice the length of the SIFS period in Figure 3.4 – and the minimum value of CW is 31 (see 2.3.2). Therefore, an average backoff period will be about 15 slots, which equates to 300 µs. However, the station with the lowest backoff will be the station that first tries to send, thus in terms of the example shown in Figure 3.4, the backoff period for the station transmitting would be significantly smaller than 15 slot times. For comparison, a backoff period of 8 slot times would equate to a drop of 25% in the percentage time the maximum data rate is used in the example shown above.

Figure 3.5 below shows the same transmission of a TCP acknowledgement packet over an 802.11a network for comparison.

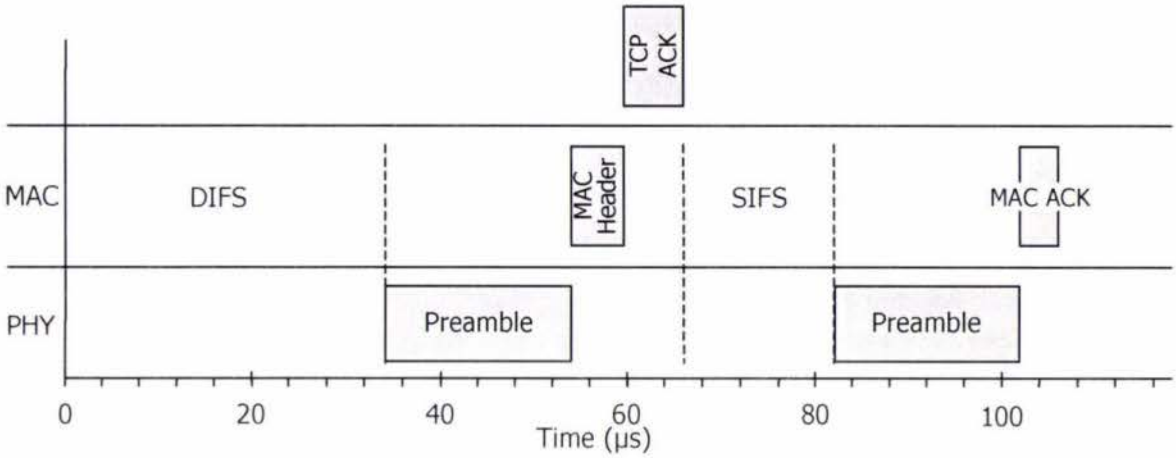


Figure 3.5: TCP ACK Transmission showing Physical and MAC overheads of 802.11a (refer to Figure 3.2 (b) for exact timing details)

In the case of 802.11a, the interframe spaces cause the greatest overhead. The preamble in 802.11a is only about 10% of the length of the preamble used in 802.11b, and the maximum data rate is 54 Mb/s compared with 11 Mbps. The slot time is decreased from 20 µs to 9 µs, but the DIFS is only 32% smaller than the DIFS of 802.11b, and the SIFS is 60% larger. Therefore, the DIFS and SIFS have a pronounced effect on the percentage of time the maximum data rate is used.

In the example above, the maximum data rate is used 15% of the time – slightly higher than for the same transaction over an 802.11b network. However, the DIFS and SIFS alone account for 47% of the time taken for the transaction. With the random backoff scheme, the overheads from waiting for the medium to be free are even higher. For example, a 5 slot backoff period increases the overheads for waiting for the medium (DIFS, SIFS and backoff) up to 63%.

3.3.2 Efficiency Results

Simulations have been carried out as part of the development process for the future 802.11n supplement that have looked at the efficiency of 802.11e. The 802.11e MAC sub-layer was tested using several of the proposed physical layer configurations and modulation schemes for 802.11n. The 802.11e MAC sub-layer was set up with block acknowledgements, up to a limit of 16 frames. These simulations show that with the physical layer modes giving over 140 Mbps, the 802.11e MAC sub-layer is at best 64% efficient, and in some cases only 54% efficiency [17].

More importantly though is a study by Xiao and Rosdahl that looks at the throughput and delays from the 802.11 MAC protocol as the data-rate is increased [18]. An upper limit for the throughput, and a lower limit for delay, was proven, and backed up by simulation results. This is a very important conclusion, as it means that without improving the MAC sub-layer, no significant improvements in throughput can be made after a certain point.

For example, the throughput was tested with packets of 1000 bytes (smaller than the typical maximum, but higher than the average packet sizes stated in 3.1.3). Using the 802.11a physical layer the maximum achievable throughput is 24.7 Mbps. With an infinitely high data rate, the maximum achievable throughput is only 50.7 Mbps, with a simulated 5400 Mbps physical layer achieving about 48 Mbps. Therefore, without any improvements, the targeted 100 Mbps throughput for 802.11n will be impossible.

4 EXISTING IMPROVEMENTS

The majority of existing improvements to the MAC sub-layer deal with the quality of service system (e.g. 802.11e), the security and authentication services (e.g. 802.11i, WPA), or modifications to the medium access protocol. These improvements do not deal with the outright throughput of wireless networks, but rather deal with the provision of particular services or with the improvement in their delivery and effectiveness.

Therefore these types of improvement are not covered in the review of existing improvements. Outside of these improvements there are only two main existing identified areas of improvement to the 802.11 MAC sub-layer.

4.1 IMPROVING RTS/CTS MECHANISM

One already identified area of improvement to the MAC sub-layer is in the value of the dot11RTSThreshold MIB variable. This threshold controls whether the RTS/CTS handshake is used for data transfer over the wireless channel [see 2.3.2]. This handshake has a marked effect on the throughput as it can greatly increase overheads.

If a frame to be sent is smaller than dot11RTSThreshold bytes, then the basic access mechanism is used. If the frame is larger than dot11RTSThreshold bytes, then a RTS/CTS handshake is used. The RTS/CTS handshake is inefficient for transmitting smaller frames, as the relative overhead of the RTS/CTS handshake is increased with small frame payloads. Secondly, the RTS/CTS handshake is inefficient when only a small number of stations are in the current service set, but very efficient with a large number of stations.

The effect of the value of dot11RTSThreshold has been studied on numerous occasions [19,20,21,22]. All of these studies found that the dot11RTSThreshold should be calculated dynamically to improve the performance of the wireless network. The main finding that is consistent over all research is that the dot11RTSThreshold should be scaled according to the number of stations in the network. As the number of stations increases, the optimal value for dot11RTSThreshold decreases.

If the number of the contending stations is relatively small ($n \approx 5$), the basic CSMA/CA access mechanism achieves equivalent packet delay compared to the RTS/CTS mechanism, for all packet size values. This shows that the RTS/CTS mechanism is not advantageous in small size networks due to the low collision probability. On the contrary, when the network size n

increases to 25 stations and the packet size exceeds a specific threshold (~250 bytes), the RTS/CTS mechanism achieves a lower packet delay value compared to the basic access. This threshold decreases to about 150 bits for large network scenarios ($n \approx 50$). In this case, the RTS/CTS mechanism is particularly beneficial due to the increase of the collision probability [23].

One study [20] examined the performance of the network with different dot11RTSThreshold values as a function of load. The study found that the optimal throughput occurred for dot11RTSThreshold values of less than 200 bytes, and that any dot11RTSThreshold value fewer than 200 bytes gave almost identical throughput performance. The conclusion for the study recommended that the dot11RTSThreshold should be set to 0 i.e. always uses the RTS/CTS mechanism.

As well as having a dynamic dot11RTSThreshold, some researchers have proposed other changes to the RTS/CTS access mechanism. One proposal is that stations only reply with a CTS frame to an RTS frame received with a specified level of signal strength. This is designed to deal with the fact that – especially in ad-hoc networks – the signal power needed to interrupt a frame transfer is often much lower than the signal power needed to deliver a frame successfully [24].

Another proposed RTS/CTS improvement is a "fast forward" scheme for TCP streams across multi-hop wireless networks. Currently, for a multi-hop route across an 802.11 network, the data packet and subsequent acknowledgement must reserve the medium for the first hop between the source and the first intermediate station. At this first intermediate station, the acknowledgement frame is sent back to the source, then to send the data packet onwards along the route the station must re-contend for the medium to reserve it – often incurring significant latency.

The "fast forward" scheme combines the acknowledgement that travels back to the source with the CTS that reserves the medium for the next hop in the route. In this way, the data packet is "fast forwarded" through the network. This eliminates the contention for this data packet (as other stations will wait at least a DIFS period), slightly reduces the control overhead (one less control frame, as the acknowledgement and CTS are combined into one frame), and lowers the overall per-packet overhead (as there is no DIFS or backoff before the physical layer preamble; see Figure 3.4 and Figure 3.5 for more details) [25].

4.2 PACKET AGGREGATION

Another primary target for improving the performance of the data-link layer in wireless networks is packet aggregation. Currently, the overheads introduced by the MAC frame header are quite substantial, especially for smaller frame payloads. For 802.11b, there is an approximate average gap between frames of 1100 'byte-times', and the gap is larger still for 802.11a [26]. Also, for internet traffic over wireless, the problem is made worse due to the nature of internet traffic – 65% of packets are less than 100 bytes [26].

There are currently two wireless networking chipsets on the market that use proprietary aggregation methods. These chipsets are 802.11a/802.11g compliant chipsets with several extensions. One such chipset is the TurboCell from KarlNet [26]. This chipset is used in wireless access points, bridges and routers. TurboCell uses 'super frame aggregation', where packets are aggregated into a 'super frame'. Transmission of the frame occurs either when it has been in a queue for a certain period, or a queue is full [27].

A second chipset to use reputedly use frame aggregation is the Atheros Super-G chipset. This chipset uses 'bursting' and 'fast frames' [28]. Bursting treats each medium access similar to a TXOP (as in the 802.11e supplement discussed in section 2.3.4), and allows the station to transmit several frames at once with no IFS between (compared to a SIFS period between frames in 802.11e's TXOP). The 'fast frames' mechanism supposedly employs frame aggregation. However, the 'fast frames' mechanism only uses an extra negotiation handshake between a station and its associated access point to agree on a different payload size [28]. Therefore, this mechanism probably does not use aggregation, but because more details on the chipset are kept as trade secrets, this cannot be stated with absolute certainty. However, for the same reason, this implementation cannot be given much merit in the context of this research. Also, it will not work for ad-hoc wireless networks.

These chipsets have a major drawback, aside from any negative aspects from the actual technologies used. The extensions to 802.11 and its supplements that have been implemented in these chipsets are non-standard, and as such are not shared by any other manufacturers, irrespective of whether the details are made public. This means that the wireless network must be made up entirely of the one set of products offered by the company. If any other product is used – either another manufacturers or an older model – the network will operate solely as specified in the 802.11 standards. Some research also suggests that the extensions may violate interference rules within the ISM bands.

However, the only apparent ongoing research into the area of packet aggregation in 802.11 wireless networks is that of Jain et al [27]. Their proposed aggregation method is named 'congestion triggered aggregation'. Congestion triggered aggregation works on the principle that a stream of small packets will cause queuing at the transmitter, as opposed to the KarlNet TurboCell implementation where aggregation occurs after a forced delay period.

5 AGGREGATION

Firstly the aims and ideas behind aggregation are stated, and then the two forms of packet aggregation identified in the existing improvements section are explained in more detail. A comparison of the two types of aggregation is also given. Lastly, development of the architecture and control of the queue within the aggregator is presented.

5.1 BASIC AGGREGATION

As discussed in various preceding sections, the current form of the 802.11 MAC sub-layer has a high per-packet overhead. Predominantly this is due to the fact that the same overheads are incurred independent of the packet size, and the fact that in a wireless network in general use will have an average packet size that is much smaller than the maximum packet size available.

The aim of packet aggregation is to reduce the per-packet overheads. The operation of packet aggregation is to include several smaller packets into the payload of a single frame (see Figure 5.1). In this way, the overhead percentage is now measured against the payload size – on a per-frame basis as opposed to a per-packet basis.

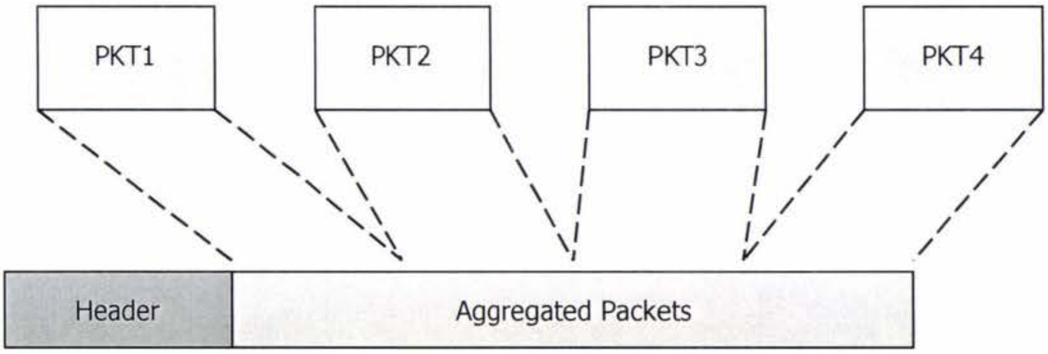


Figure 5.1: Aggregating several packets into a single frame

In this way, only a single MAC sub-layer header, a single physical layer preamble, and a single medium access process is used to transmit multiple packets - significantly reducing the overhead per packet. Additionally, this system both increases the frame size and sends several packets for each RTS/CTS handshake if used, thus significantly reducing the inefficiency of the RTS/CTS mechanism.

In this basic form of aggregation, the aggregated frame is sent to a single destination station. A possible system that may provide even more advances would be a system that sends an aggregated frame to more than one station, with one or more packets contained within that frames payload destined for each station that the frame is directed to. However, such a scheme was considered slightly beyond the scope of this research, but is considered to be a recommended direction of development should this work be continued.

5.1.1 Operational Details

Each packet aggregated into the frame payload requires its own control information. A total of 6 octets per packet has been allocated for this purpose as follows (see Figure 5.2). Each sub-header contains a 16 bit CRC packet check sequence (PCS), a 2 octet sequence control field and a length field. The PCS field allows each individual packet to be checked for errors. If only one packet out of several aggregated has an error, only that single packet needs to be re-transmitted.

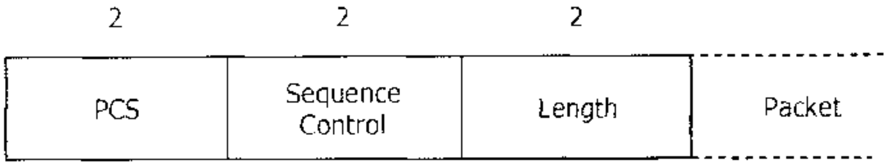


Figure 5.2: Proposed Aggregation Sub-Header

Each packet sent over an existing 802.11 wireless network has a sequence control field of 2 bytes. The aggregation sub-header format keeps the sequence control field for each packet. All operations that use the existing sequence control are therefore unaffected. Finally the length field only needs 12 out of the 16 bits left to represent all packet lengths (maximum allowable is 2132 bytes), with the last 4 bits reserved for future use.

Also, a subtype will be added to the existing set of subtypes in the Data type. This subtype will be named "Data (aggregated)", and will take up one of the reserved Data subtype values. It is possible that future extensions to aggregation would also require separate subtypes – such as a combined acknowledgement, or selective repeat scheme.

5.2 AGGREGATION TYPES

5.2.1 Forced Delay Aggregation

The form of aggregation used by KarlNet in their TurboCell wireless network firmware has become known as forced delay aggregation [26]. Although internal details are not publicly available, a whitepaper for TurboCell does give an indication on its operation. This is the easiest form of packet aggregation to implement, but on the other hand forces a delay on all packets (hence the name).

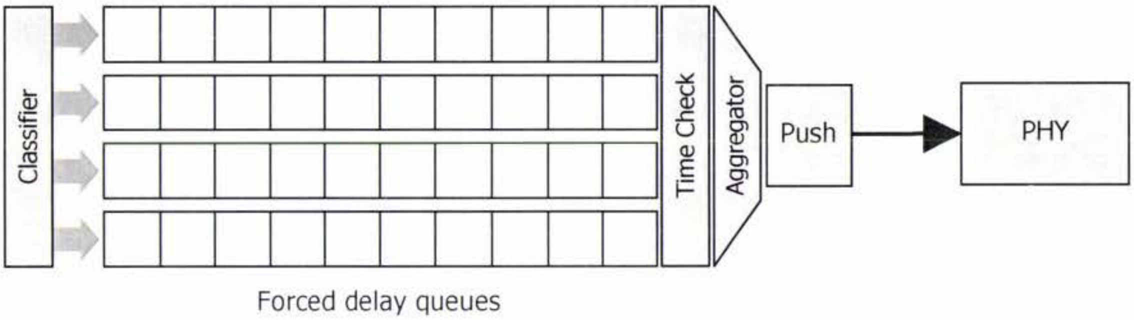


Figure 5.3: Forced delay aggregation mechanism

Figure 5.3 shows the basic functional layout of a forced delay aggregation MAC sub-layer. From the information available on the TurboCell chipset, it is not clear as to what the queueing architecture is. However, the discussion on the chipset assumes that there are multiple queues. The study by Jain et al. also modelled the TurboCell implementation as having multiple queues [27].

As the packets arrive at the MAC SAP, they are classified according to their destination address into queues. In forced delay aggregation, the queues are time-bounded, with packets sent if enough packets are present to comprise a maximum length payload, or if they have been present in the queues for a set length of time. What this actually means in reality is that all packets are kept from being pulled from the queue by the aggregator until they have undergone a set delay period, unless there are enough packets to comprise a full frame payload.

The purpose of delaying the packets in the queues is an attempt to increase the amount of aggregation. The forced delay aggregation mechanism is designed to let packets queue up, so that there is an increased chance that a higher number of packets will be aggregated, thus increasing the reduction in overhead percentage.

If there is any queue that can provide a full payload, the aggregator will use that queue. Else, the aggregator chooses the queue that has the packet with the earliest timestamp to aggregate out of. Once the packets have been aggregated into a frame, it is 'pushed' to the transmitter fro sending in the same way as a normal 802.11 data frame would be.

Looking closer at the description of 'super frame aggregation' in the TurboCell whitepaper, the aggregation mechanism seems to only be implemented in wireless bridges and routers. This precludes the benefits of TurboCell aggregation from being used in ad-hoc networks. If solely implemented at bridges and routers, it also means that the end users will not see the increases in throughput – but there will be less outage due to the distribution system being overloaded.

Also, the whitepaper names a single 'super-frame buffer'. With reference to the assumption made about the queueing architecture, this would give another means of implementing the 'super-frame' – have only one queue. This would mean that all packets are 'aggregated' within a 'super-frame' complete with their accompanying MAC sub-layer headers. This system is not technically packet aggregation, and is very similar to the Atheros Super G chipset's 'fast frames' to – which is itself coincidentally like the TXOP of 802.11e.

5.2.2 Congestion Triggered Aggregation

The other form of aggregation – congestion triggered aggregation – was identified by Jain et al. in their study of aggregation [27]. This method of aggregation looked at the reason for wishing to perform aggregation. In an idle network, with only intermittent transmissions, there is no need for aggregation as there is amble bandwidth available. Therefore all packets should be sent as per normal. However, once the network comes under heavier traffic conditions, congestion occurs, and some or all station will experience a backlog of packets at the wireless card.

It is at this point that aggregation would see the most benefits, as it can clear larger portions of backlog than by sending one packet at a time. Therefore, as the aggregation only occurs in times of congestion, the name congestion triggered aggregation was given.

This method is harder to implement than that used by the TurboCell chipset, as the physical layer must essentially 'pull' the frames from the aggregator. Once the physical layer has confirmed that the last packet has been sent, a signal is sent to the aggregator to create a new frame for the physical layer to send. This frame may contain only one packet – thus it will be identical to an existing 802.11 frame – or it may contain several packets aggregated into a single frame payload.

Figure 5.4 below shows the basic operational design of a congestion triggered aggregator. Each incoming packet is classified according to its destination address. When the physical layer is free to send another frame, it 'pulls' an aggregated frame from one of these queues that is made up of one or more packets. If no queue contains any packets, the transmitter remains idle.

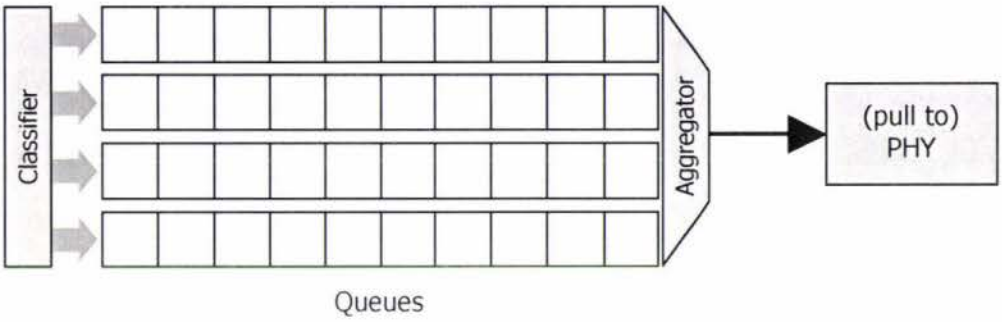


Figure 5.4: Congestion triggered aggregation mechanism

In this way, the incurred delay of the TurboCell chipset's aggregation mechanism is avoided, as intermittent packets are sent as soon as the transmitter is available. In times of congestion, the only delay is that caused by the congestion in the network – which is the case for existing 802.11 MAC, and is minimised when aggregation is in place as it will alleviate the congestion more rapidly by sending more data than is possible with the existing MAC.

5.2.3 Comparing the Two Types of Aggregation

The research of Jain et al. compared the TurboCell's forced delay method (with the assumption that a multi-queue architecture was used) to the congestion triggered aggregation mechanism. The comparison between the two types of aggregation was performed in simulation, and involved 3 different scenarios, as outlined below:

1) Suburban client-server scenario: 5 servers, each with 5 clients, representing isolated client-server applications, such as streaming music server, or interactive gaming. Ad-hoc network with omni-directional antennae.

2) Suburban client-server-gateway scenario: the same as the first scenario, but each server now also has a gateway to the Internet for web traffic. Ad-hoc network with omni-directional antennae.

3) Rural scenario: 2 access points, with 4 stations connecting through each one, with much larger distances than in the first two scenarios. There is also a two hop 'backbone' wireless connection between the access points, which also provides gateway access to the Internet.

The traffic used to compare the two aggregation mechanisms was data captured from a local wireless Internet Service Provider, as the research was partly directed at providing a higher performance wireless network in the area. The throughput and median packet latency results are presented below in Table 5.1 and Table 5.2.

Table 5.1: Summary of throughput achieved as from [27]

Scenario	Existing 802.11 (No Aggregation)	Forced Delay Aggregation	Congestion Triggered Aggregation
Client-Server	360 kbps	510 kbps	626 kbps
Gateway	409 kbps	535 kbps	663 kbps
Rural	202 kbps	326 kbps	393 kbps

Table 5.2: Summary of median packet latency observed as from [27]

Scenario	Existing 802.11 (No Aggregation)	Forced Delay Aggregation	Congestion Triggered Aggregation
Client-Server	0.605 s	0.675 s	0.153 s
Gateway	0.985 s	1.120 s	0.078 s
Rural	0.894 s	0.673 s	0.165 s

From the results of the experiments, congestion triggered aggregation proved to deliver higher throughput, and much lower median latency than existing 802.11. The forced delay aggregation also delivered a higher throughput than the existing 802.11 MAC sub-layer, but actually increased the median packet latency for two of the scenarios, and only slightly reduced the latency for the other scenario.

5.3 QUEUES IN AGGREGATION

5.3.1 Queueing Control

For the forced delay method of aggregation used by the TurboCell chipset (assuming a multi-queue architecture), the decision on which queue to aggregate packets from is straightforward. The queue that has the packet at its head that has the earliest timestamp gets packets aggregated from it, and 'pushed' to the transmitter.

Congestion triggered aggregation, on the other hand, does not have a choice of queue implicit in its operation. Therefore, any queueing control algorithm may be used to choose which queue packets should be aggregated from. At low traffic levels, there will be very little depending on which queue is chosen, particularly as the MAC sub-layer will be acting essentially identically to the current 802.11 MAC sub-layer.

However, as the traffic load on the network and its stations increases, the choice of queueing control algorithm may have a marked effect on the throughput. The most efficient method of queueing control will deliver the best throughput in combination with packet aggregation. Therefore in combination with the best method of aggregation, a decision on which technique is used to control the packet queues is also needed.

Another important factor for choosing queuing architectures is the required serving characteristics of the incoming packets. The main attribute that incoming packets need minimised is the packet delay – therefore the choice of queueing algorithm must also take into account the average and maximum packet delays of the algorithms.

Also, the length of the queues is important. The longer a queue or buffer is, the larger the packet delay will be. This is most significant for connections where the packets have a relatively short timer. However, if the queue is too short, then packets may be dropped. A too larger queue will drive up cost due to memory requirements.

Elements of Quality of Service (QoS) like those catered for in 802.11e can be easily introduced with multiple queues, as higher priority applications will have their packet queues serviced more often. However, if this is the case, a fairness policy must also be used to ensure that the lower priority queues are still serviced.

5.3.2 Different Control Algorithms

Several different queuing control algorithms have been identified for use in the aggregation mechanism. The control algorithms identified are as follows:

1) Round-robin queueing – This is the most simple queueing algorithm. Starting with the first queue to have a packet in, the algorithm simply iterates through the packet queues one by one. This means that all queues will get even access to the physical layer.

2) Priority queueing – Each queue is assigned a priority. The algorithm simply chooses the queue that is non-empty and has the highest priority. This is the similar to the 802.11e QoS differentiation in that certain queues (similar to Access Categories) will always get access to the physical layer if it is needed.

3) Weighted fair queuing – This algorithm is a combination between Priority queueing and Round-robin queueing. Every queue gets a turn as in Round-robin, but each queue's turn is weighted in time, meaning the higher priority queues get more time to transmit in their turns, and the lower priority queues get smaller amounts of time.

4) Load based – Simply chooses the queue with the largest packets weighting, or alternatively, the queue with the largest number of bytes in. This queueing algorithm would be good in making sure that packets are not dropped due to any lack of queueing space.

5) First-in first-out – Simply a first-in, first-served queue overall for several queues. This is the queueing algorithm that is used in the forced delay form of aggregation used in the TurboCell chipset, i.e. choose the queue with the packet that has earliest timestamp at its head.

6) Maximise aggregation – Also similar to the TurboCell operation, this algorithm seeks to maximise the gains from the aggregation scheme. This is accomplished by choosing the queue with the largest number of smaller packets, as this would provide the biggest decrease in overheads.

6 EXTENDING AGGREGATION

The basic form and structure of the aggregation mechanism presented in the previous chapter can be extended both for possible greater performance. This research focuses on the queueing systems involved in aggregation, and improvements to it – with two possible improvements to the aggregation mechanism explained in this chapter.

On the other hand, aggregating several packets into a single frame can increase the chance of a transmission failing in the presence of periodic interference to the wireless medium. Also presented in this chapter is a scheme that attempts to improve the number of successful transmissions in the event of periodic interference.

6.1 QUEUEING ACCESS

6.1.1 Look-Ahead Queueing

The default method of removing packets from the aggregator queues is to simply take the packets from the head of the queue. This would mean that the packet previously in the second position would now be the new head of the queue. In this way, only the packets near the front of the queue are the only packets available for aggregation.

However, this may lead to inefficient cases where aggregation does not fully achieve the performance improvements possible. An example is a queue with a small packet followed by maximum size packet, followed by several smaller packets. Using the default method of accessing the queues for the aggregator will result in the small packet being placed in the payload, but then looking at the next packet – the maximum sized packet – will result in the conclusion that no more packets can fit within the remainder of the payload.

A look-ahead queue access mechanism would solve this. Look-ahead queueing adds the ability to look further into the queue, beyond the head of the queue. Using this access method in the above example will result in the aggregator looking beyond the maximum sized packet, and finding the several small packets behind it in the queue. These small packets could then fill up the rest of the payload left after the first small packet is aggregated. In this example, a look-ahead queue access method would reduce the number of frames needed to send the data by one frame.

6.1.2 Indexed Queueing Machine

Another possible method of improving the queueing access methods combined with aggregation is to changed the way the packets are inserted into the queue (as opposed to the change in the way they are removed as in look-ahead queueing).

This idea is based on a system named Indexed Queuing Machine (IQM) – originally developed for removing delays caused by data dependencies in processors [29]. Rather than using the traditional way of looking ahead at instructions to find something to do when a processor is doing something, IQM predicts the dependency and controls the order of instructions on the instruction queue. In IQM, the new instructions can be put anywhere at all on the instruction queue, but are always only taken from the front of the queue. This aimed to leave the complexity of the queue access at a minimum, but add complexity into the queue insertion.

A simplified variation of this could be used in conjunction with the aggregation mechanism. It is known that the aggregation of packets will always occur up to a maximum number of bytes. Therefore, the input mechanism to the queue could look at the first N bytes of the queue, and if a large packet is overlapping the maximum aggregation size, and a small packet arrives, it could be added to the front of the queue.

This IQM based mechanism will only have the usual pull (from the head) function, with a standard push (onto the tail) and a push onto the head of the queue. In comparison, the initial look-ahead mechanism needs a push (onto the tail) function as usual, with a pull (from the head), and a pull from within the queue.

However, this method may give slightly lower performance than the look-ahead method. This is because of the fact that it looks at the first N bytes of the queue. Even with an almost full queue, a small packet could come along and be added to the head of the queue. Therefore, I need to perform a simulation of this method to see the effects it has on packet delay and throughput. Also, with first-in first-out queueing control, pushing a packet onto the head of the queue will mean the first-in first-out algorithm will have to be extended to look at all packets in the queue to retain the same functionality.

The choice of two schemes to improve the queueing is beneficial, as any real world implementation of aggregation may render one of the schemes impossible to implement. In this case, the other scheme may be used to provide the improvement instead.

6.2 TEMPORARY QUEUEING PRIORITY

6.2.1 Load based temporary queueing priority

Another possible improvement to the queue architecture would be to have a temporary priority system for queue's that are close to overload. This would prevent the dropping of packets at the location of the aggregation queues. This system recognises the fact that the memory available for the queues will typically be limited. Therefore, it is possible that with very high traffic loads the queues will approach saturation, and any subsequent packets will be dropped.

The temporary priority scheme looks for queues that are almost full. These queues are defined to be those that cannot fit a maximum sized packet if it were to be sent down by the network layer. The system gives a queue identified as full immediate priority over others to send one aggregated frame. Afterwards, operation of the aggregation mechanism returns to normal.

Although the packets will not be lost completely, (they will be resent by the network layer protocol, e.g. TCP), they will be delayed until they timeout at the network layer. In the case of TCP, the timeout is typically 3 seconds – significantly longer than the time scales that the MAC sub-layer operates on. Therefore, it is beneficial to prevent dropping packets from the queues.

This system can only be used for some of the queueing control algorithms. The load based queueing control algorithm already would give precedence to the fullest queue – so temporary queueing priority is essentially performing a backup load based algorithm upon the queues. However, the other queueing control algorithms (see 5.3.2) can all use this system to try and prevent unnecessary dropping of packets.

6.2.2 Time based temporary queueing priority

Conversely, a time based system could be implemented for the load based queueing control. With the priority for full queues, the purpose is to prevent packets from being dropped. For load based queueing, there is a smaller chance that packet are dropped as it always chooses the fullest queue. If one of the queues had a heavy traffic load, it would be served a greater amount of the time by the load based queueing control. This means that there is a possibility that queues with lower loads will then suffer greater delays, and if the load is low enough, no service at all.

Therefore, the load based queueing algorithm can be extended by using a temporary priority in queues where packets have been delayed too long. In the same way that the principle used in load base queueing can be applied as a backup to the first-in first-out queueing control, the same applies in the opposite sense that the first-in first-out principle can be applied to load based queueing.

6.3 PAYLOAD SIZE BACKOFF

The basic aggregation mechanism can also be improved with a backoff scheme for the size of the payload in case of transmission failure. Due to the nature of the wireless medium, longer frames are more prone to errors, simply because the data transfer last longer, and many errors are timed based (therefore, longer transmission time equals longer probability of errors). Such errors require the frame to be retransmitted if the number of errors is beyond the repairing abilities of the 32 bit CRC system used for the FCS.

A payload backoff would lower the upper limit for the frame payload size due to aggregation. This is designed to improve the performance of the network under periods of interference. For instance, the current aggregation mechanism assumes a constant 1500 bytes as the payload size. In the case of a failed transmission of an aggregated frame, the maximum size would be backed off to half that – 750 bytes. The payload size limits would continue to be backed off with each failure, until a successful transmission, upon which the limit is reset to the maximum value of 1500 bytes.

This system could be modified to allow one failed transmission and only backoff the payload size after the second failure. This would mean that a single independent error with duration less than that of a single frame does not cause an unnecessary performance decrease. Similarly, the return to the maximum payload size after a successful transmission could be delayed until two consecutive successful transmissions. Not allowing the payload size to be set back at maximum aims to ensure a lower rate of frame transmission failures. Both of these differences can only be assessed with full scale field trials of aggregation with a payload size backoff system.

Additionally, intelligent control of the payload size could also be used to counter longer duration interference effects that can often plague wireless networks. This would be most useful if some form of link quality or signal quality measurement could be utilised.

The payload size backoff system is a simple extension to aggregation that should reduce the number of failed transmissions when the wireless network undergoes interference. A failed transmission in this case adds to the overhead, so preventing repeats of failures will therefore reduce the overhead.

7 STANDARDISING AGGREGATION

This chapter looks at specifying the operational details of the aggregation mechanism, so that it can be integrated into the current 802.11 standard and its documentation. This also gives the aggregation system presented here the advantage of complete compatibility with existing 802.11 systems, which means that aggregation should be able to be implemented in future as just a software patch or a firmware upgrade (or both) – but still utilising the existing wireless networking hardware. This would mean a performance with no major capital outlay and no loss on previous investment.

Firstly, the changes to the MAC data frame format that are necessary to support the aggregation scheme are specified. Then the aggregation extension is looked at with respect to the current 802.11 standards formal definition, and where aggregation fits within the current MAC sub-layer formal description.

7.1 FRAME FORMAT

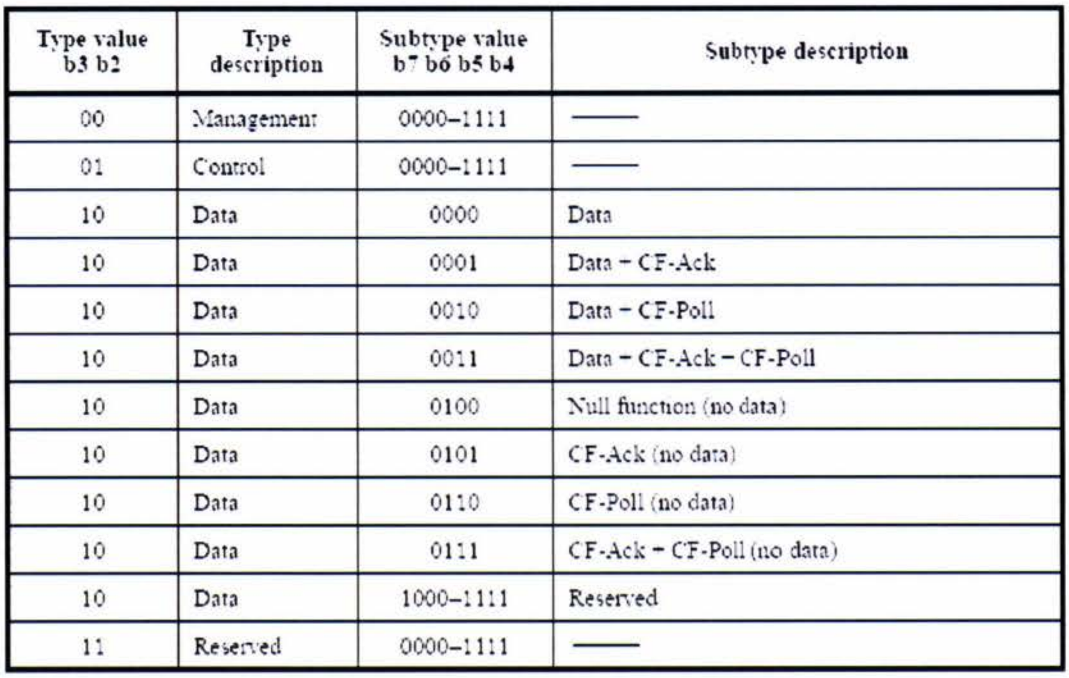
7.1.1 Frame Control and Frame Type

For complete backwards compatibility with existing 802.11 wireless networks, the entire MAC header needs to be left unchanged – in particular the field boundaries. For a MAC sub-layer frame sent with several packets aggregated in its payload, the format of the Frame Control field of the MAC header would be left unchanged (see Figure 2.5: MAC sub-layer Data frame (within BSS) for the current Data frame format).

Every frame sent by the MAC sub-layer has a Type and a Subtype. The Type and Subtype are indicated within the Frame Control fields – bits 2 and 3 for Type, and bits 4 through 7 for Subtype. The 3 Types are Management, Control and Data. Aggregation only deals with Data frames, so the other two types remain unchanged. However, a new Data Subtype would need to be introduced.

The Type bits would remain as '10' to designate a Data frame. The Subtype currently has a reserved range of 1000 to 1111 for Data frames. The current designations for Data Subtypes in the 802.11 standard are shown in Table 7.1.

Table 7.1: Data Subtypes used in 802.11 [1]



Type value b3 b2	Type description	Subtype value b7 b6 b5 b4	Subtype description
00	Management	0000–1111	——
01	Control	0000–1111	——
10	Data	0000	Data
10	Data	0001	Data + CF-Ack
10	Data	0010	Data + CF-Poll
10	Data	0011	Data + CF-Ack + CF-Poll
10	Data	0100	Null function (no data)
10	Data	0101	CF-Ack (no data)
10	Data	0110	CF-Poll (no data)
10	Data	0111	CF-Ack + CF-Poll (no data)
10	Data	1000–1111	Reserved
11	Reserved	0000–1111	——

Therefore, a value of 1000 may be used to designate an aggregated frame with the Subtype description of 'Data (aggregated)'. Other Subtypes may need to be defined when the basic aggregation scheme is extended further. All other bits within the Frame Control (e.g. the 'More Fragments' bit) field are set as per the normal operation for all other Data frames.

7.1.2 Duration, Address and BSS ID Header Fields

The Duration field, Address fields and the BSS ID field will also be set as per usual for a Data frame. The Duration is set according to what period the network is under (i.e. contention, or contention-free) and the size of the full frame (including header and FCS). The BSS ID field either contains the BSS ID of the IBSS in ad-hoc mode, or the address of the local access point (the access point will use its own address if it is sending frames) in infrastructure mode.

All packets aggregated into the Frame Body will have different sequence numbers. This means that some form of the Sequence Control field will need to be included into the frame sub-headers. The Sequence Control field is defined in the 802.11 standard as shown in the following excerpt.

7.1.3.4 Sequence Control field

The Sequence Control field is 16 bits in length and consists of two subfields, the Sequence Number and the Fragment Number.

7.1.3.4.1 Sequence Number field

The Sequence Number field is a 12-bit field indicating the sequence number of an MSDU or MMPDU. Each MSDU or MMPDU transmitted by a STA is assigned a sequence number. Sequence numbers are assigned from a single modulo 4096 counter, starting at 0 and incrementing by 1 for each MSDU or MMPDU. Each fragment of an MSDU or MMPDU contains the assigned sequence number. The sequence number remains constant in all retransmissions of an MSDU, MMPDU, or fragment thereof.

7.1.3.4.2 Fragment Number field

The Fragment Number field is a 4-bit field indicating the number of each fragment of an MSDU or MMPDU. The fragment number is set to zero in the first or only fragment of an MSDU or MMPDU and is incremented by one for each successive fragment of that MSDU or MMPDU. The fragment number remains constant in all retransmissions of the fragment.

Excerpt 7.1: Sequence Control field definition – from the 802.11 standard [1]
(Note: numbers in the excerpt are from the source and do not refer to the thesis)

As a possible extension to the aggregation scheme, it may be possible that fragments of packets could be aggregated if space is available within the frame payload, thus maximizing the usage of the payload space available. Therefore, the entire Sequence Control field should be included in the sub-headers within the frame payload, even if this research does not implement this extension, for possible future work on aggregation.

The Sequence Control field in the full MAC header will refer to the first packet aggregated into the frame's payload. Each subsequent sub-header's Sequence Control will refer to the immediately following packet.

7.1.3 Sub-header Packet Check Sequence

Due to the fact that the frame sizes are much greater under aggregation than with existing 802.11, there is a higher probability that an error will occur during the transmission of the frame. In the existing 802.11 standard such an error would mean that the entire frame would have to be re-transmitted. However, using aggregation, all but one of the packets aggregated may have been sent without errors, but using the existing system means that any error not repairable through the FCS CRC-32 system in a single packet would mean that the whole frame needs to be re-transmitted.

Therefore, it would be beneficial to add an individual packet check sequence (PCS) field into every sub-header. The PCS would use the standard CRC-16 system (16 bit CRC with polynomial of $x^{16} + x^{15} + x^{2} + 1$) be calculated on just the packet preceding the sub-header. With this system, when the frame is received, and the overall FCS shows that an error has occurred somewhere over the entire packet, the individual packets can be checked against their individual PCS. For instance, consider a frame containing 4 aggregated packets. The frame would then contain the following relevant sections:

Header	Packet 1	Sub-header 1	Packet 2	Sub-header 2	Packet 3	Sub-header 3	Packet 4	FCS

Figure 7.1: Example Data frame, including sub-headers, with 4 packets aggregated

The CRC-32 value of the entire frame is checked against the FCS received, and an error is detected. By calculating the CRC-16 value for each packet, and comparing it against the PCS contained in the following received sub-header, each individual packet can be checked for errors.

For example, the PCS is calculated for Packet 1, and matches the PCS contained in sub-header 1 (which also contains the sequence control for Packet 2). This means that although the frame overall has an error, Packet 1 has been received without error. Following on with the example, Packet 2 may have been received without error as well, but Packet 3's PCS doesn't match the PCS contained in sub-header 3. This means that Packet 3 needs to be re-transmitted. If the error is not found once all sub-headers are checked, it is inferred that the last packet will be the source of the error.

If the error is found in a packet before the last two, the sequence can continue to check the remaining packets as well. However, in this scheme, if the error occurs in the second to last packet, the last packet cannot be checked, as it does not have a following PCS. However, the PCS system could be designed so that it can be compared against the overall FCS to overcome this problem. Referring to the example, this would work by using the mathematical properties of the CRC-32 and CRC-16 systems, and by comparing the three PCSs with the FCS a result for the final aggregated packet may be found.

7.1.4 Sub-header Length Field

The final component of the frame sub-header will be a length field. It is impossible to use a delimiting sequence in data frames, as the data can possibly contain a similar sequence of bits. Therefore, a Length field needs to be included in the sub-header. The fact that more than one packet will be added means that the existing operation of 802.11 MAC and PHY layers is not enough to decode a received packet.

Currently, the only length parameters used in the 802.11 standard are:

1) TXVECTOR LENGTH – passed between the MAC and PHY when the frame is transmitted, as part of the PHY-TXSTART.request primitive, which informs the PHY as to how many octets the MAC entity is wishing to send, and

2) RXVECTOR LENGTH – passed between the PHY and MAC when the frame is received, as part of the PHY-RXSTART.indicate primitive, which informs the MAC sub-layer as to how many octets the PHY will pass up from the received frame.

This system works as there is only ever a single packet (or a single fragment of a packet) within the payload of a frame. However, with aggregation, there can be two or more packets within the frame payload. The information regarding the length of the packets contained within the frame payload is only needed at the MAC sub-layer. Therefore, a Length field needs to be added to both the sub-header, and the overall MAC frame header.

The overall length of the frame is already contained within the RXVECTOR LENGTH as identified above. The Length field in the frame header will correspond to only the first packet contained within the frame payload. The Length field will indicate the number of octets of data until the first sub-header. This sub-header will have a Length field that will have the length of the second packet included in the frame payload. Thus each sub-header includes a Length field corresponding to the following packet, as well as the Sequence Control field for the following packet.

The maximum Length value that is needed will be limited by the maximum size of the frame. If only a single frame is sent, the Sub-Type of the Data frame will not be set to "Data (aggregation)", and no Length fields will be used. However, a packet that is only a few octets short of the maximum size may possibly be aggregated with a very small packet (or a small fragment of a packet) – thus the Length field needs to be able to indicate the maximum length value of the frame payload.

From the 802.11 standard, this maximum length will be 2132 octets. Thus a 12 bit Length field ($2^{12} = 4096$) will be sufficient to encode the packet length within the Sub-Header. However, this bit-length is not equivalent to a whole number of octets. All other MAC header fields for 802.11 are a whole number of octets in length; therefore the Length field should be set to 16 bits (2 octets). The extra 4 bits allow for either the later extension of the aggregation system, the possibility of greater packet lengths being used, or if extra header information is required.

7.1.5 Overall Frame and Sub-header Formats

In summary, there is an extra Length field of 2 octets added to the Data MAC header if the Sub-Type is "Data (aggregation)". Each Sub-header will consist of a Sequence Control field, a Packet Check Sequence (PCS) field, and a Length field. In keeping with the existing FCS system, the PCS should be placed directly after the packet that it corresponds to. The Sequence Control field will be identical to that used in the Header, and thus is 2 octets in length. The Length field in each Sub-Header will be the same size as the extra field in the Header – 2 octets. The PCS will also be 2 octets in length (16 bits for CRC-16 remainder). The proposed Sub-header for use in an aggregation is as shown inFigure 7.2, and the proposed extension of the overall frame header – when the Subtype is 'Data (aggregation)' is shown in Figure 7.3.

Octets:	2	2	2	
Packet	PCS	Sequence Control	Length	Packet

Figure 7.2: Proposed aggregation sub-header format

Octets: 2	2	6	6	6	2	2	0-2132	4
Frame Control	Duration	Destination Address	Source Address	BSS ID	Sequence Control	Length (1st pkt)	Frame Body	FCS

Figure 7.3: Proposed extension to frame format

7.2 MAC SUB-LAYER FORMAL DESCRIPTION

7.2.1 SDL Description of 802.11 MAC Sub-layer

The 802.11 standard and its supplements formally describe the MAC sub-layer using a language called SDL (Specification and Description Language). SDL is a graphical language, and describes the MAC operation as a set of finite state machines that run concurrently. The SDL specification of the MAC sub-layer describe its behaviour, the generation of frames, the service primitives and their parameters, and the interaction between the MAC sub-layer and the MAC Management Information Base (MIB). It also describes the data types used in all of the MAC operations.

The SDL description is contained in Annex C of the 802.11 standard, which is entitled "Formal description of MAC operation" [1]. There are two main sections of this Annex, after the introduction section – Annex C.3 "State machines for MAC stations", and Annex C.4 "State machines for MAC access point".

For all extensions and supplements to the 802.11 MAC sub-layer, the SDL diagrams must be updated. Therefore, the changes necessary to include aggregation into the 802.11 MAC sub-layer were developed. They have been described in the same format as the changes made in the current supplements to the 802.11 MAC sub-layer. The changes have only been described for the SDL descriptions contained within Annex C.3 of the 802.11 standard that apply to MAC stations. The changes to the MAC access point descriptions in Annex C.4 of the 802.11 standard are identical.

The following section describes the changes to the diagrams, which have been included for completeness. The reader needs to refer to the diagrams in the 802.11 standard to understand the context of the changes.

7.2.2 Modifications to SDL Diagrams

In "Process MSDU_from_LLC" (SDL diagram named "Msdu_from_LLC_1b(1)") change the name diagram name to "Msdu_from_LLC_1b(2)". Modify the section of the diagram shown in Figure 7.4 to that shown in Figure 7.5.

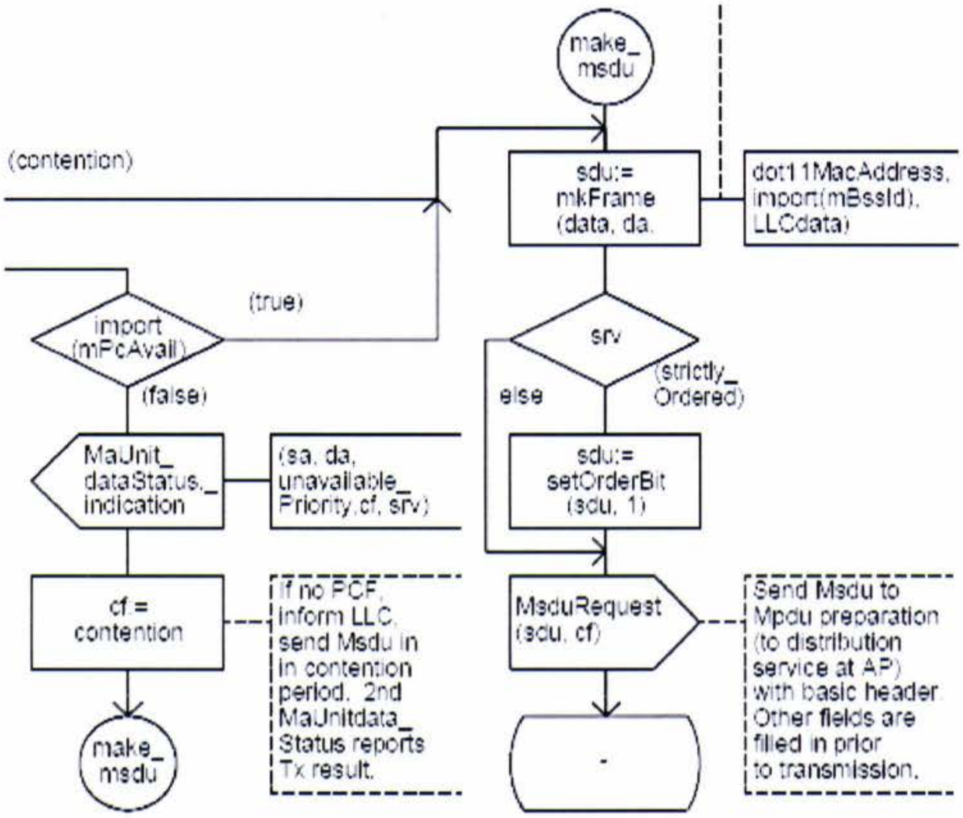


Figure 7.4: Existing section of Msdu_from_LLC_1b as from [1]

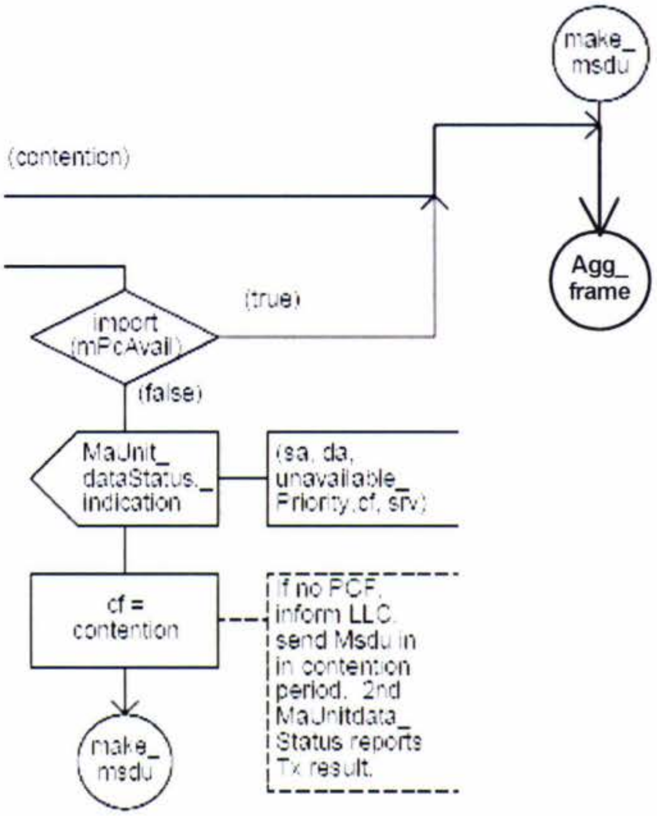


Figure 7.5: Modified section of Msdu_from_LLC_1b

Add the diagram section shown in Figure 7.6 as a new diagram within the process "Process MSDU_from_LLC", with the diagram name "Aggregate_2b(2)".

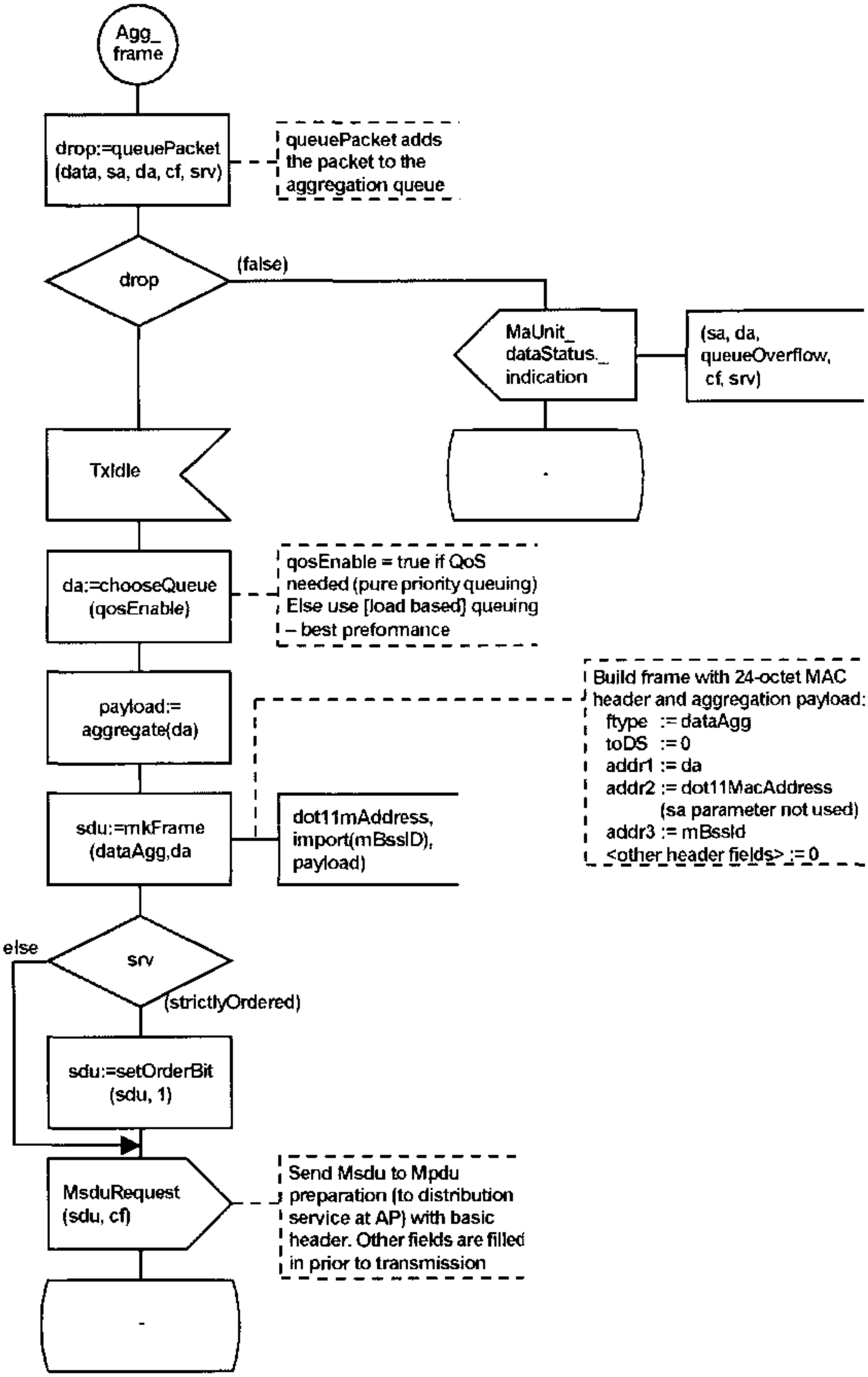


Figure 7.6: New SDL diagram Aggregate_2b(2)

Change the SDL diagram of the block named "Protocol_Control_STA" (diagram "sta_CTL_1c(1)") to the diagram shown in Figure 7.7.

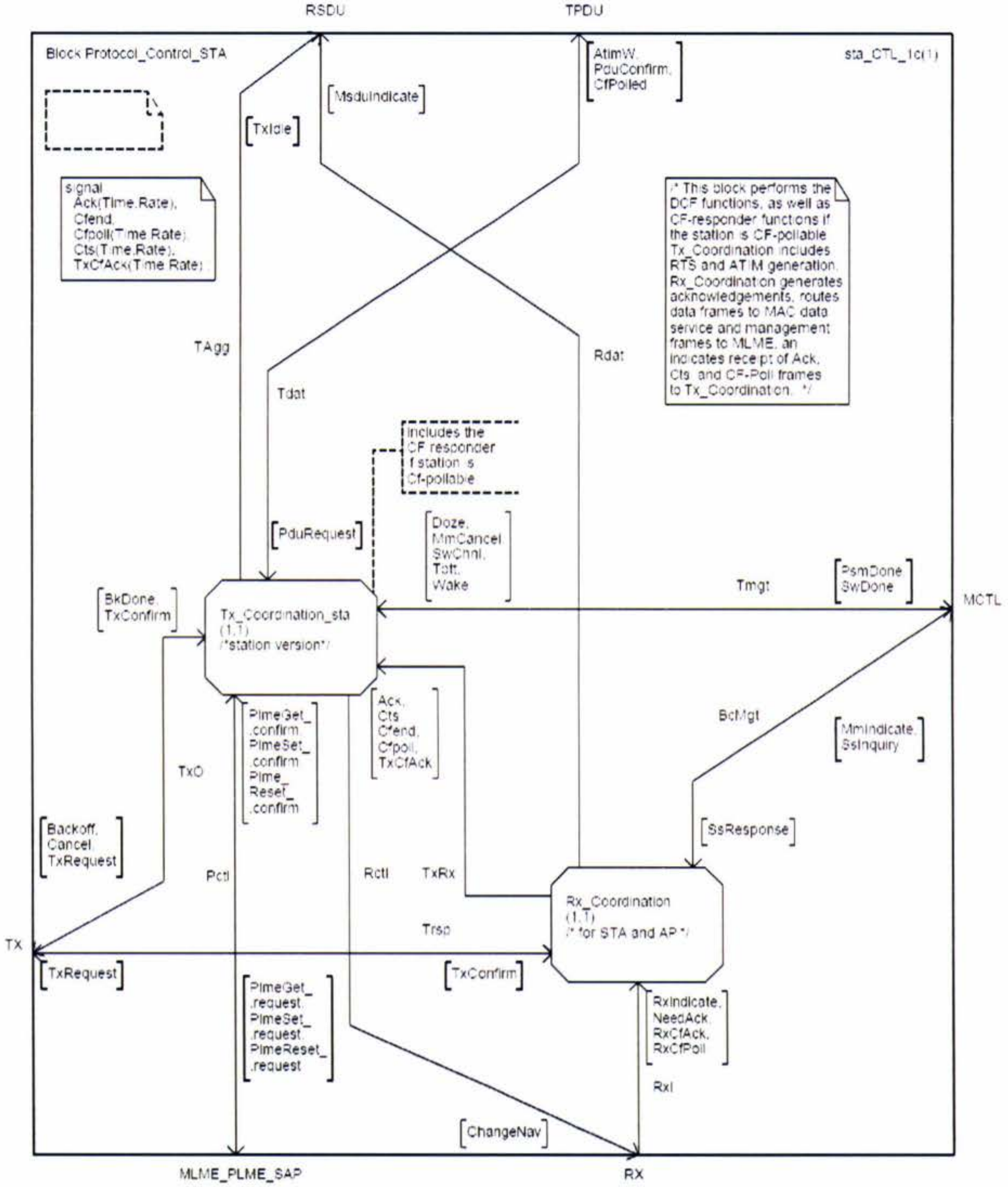


Figure 7.7: Modified block Protocol_Control_STA

Within the Tx_Coordination_STA process shown in Figure 7.7, one of the process states is "TxC_Idle". This state is only entered when the previous frame transfer has concluded (including any subsequent frames transmitted or returning frames to be received). By adding a signal "TxIdle" to be sent along the path "TAgg", the Aggregator within the process

Msdu_from_LLC (refer Figure 7.6) is then notified that the last packet has been sent successfully, and the next packets may be aggregated and sent. This implements the 'pull' operation of the aggregation mechanism (see subsection 5.2.2).

The TxIdle signal is generated at the start of the TxC_Idle state, and changes the part of the diagram "sta_tx_idle_2d(10)" (from the process Tx_Coordination_sta) shown in Figure 7.8 to the diagram shown in Figure 7.9.

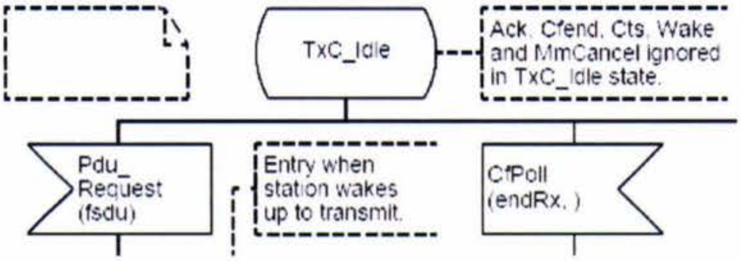


Figure 7.8: Existing part of diagram sta_tx_idle_2d(10) as from [1]

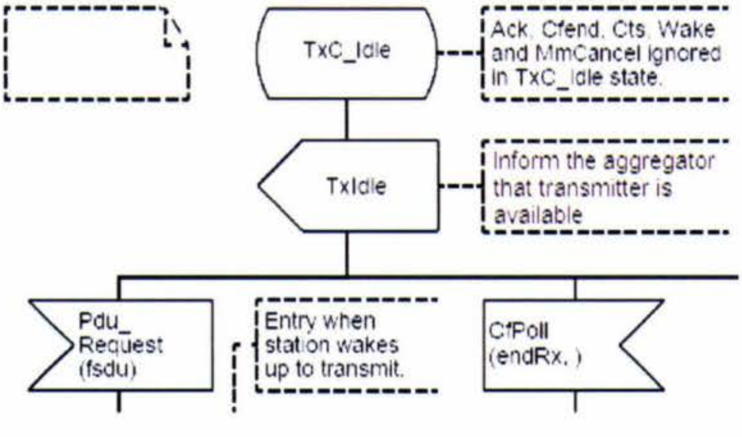
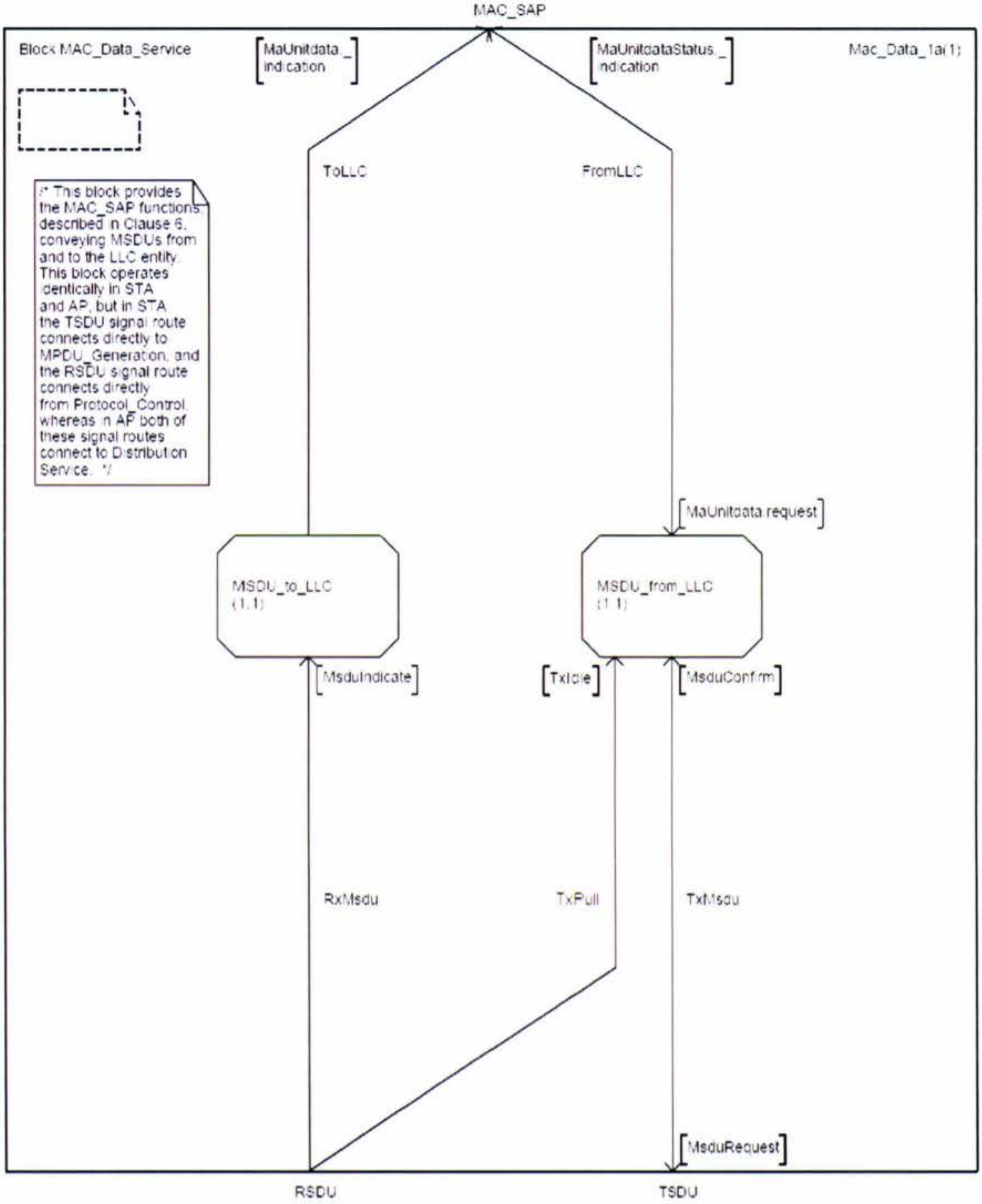


Figure 7.9: Modified part of diagram sta_tx_idle_2d(10)

Lastly, the diagram of the block "MAC_Data_Service" (named "Mac_Data_1a(1)") is changed to the diagram shown in Figure 7.10 to complete the change necessary to add the signal TxIdle.



/* This block provides the MAC_SAP functions described in Clause 6, conveying MSDUs from and to the LLC entity. This block operates identically in STA and AP, but in STA the TSDU signal route connects directly to MPDU_Generation, and the RSDU signal route connects directly from Protocol_Control, whereas in AP both of these signal routes connect to Distribution Service. */

Figure 7.10: Modified diagram of block MAC_Data_Service

8 COMBINED ACKNOWLEDGEMENTS

In this chapter, the development of 802.11e's block acknowledgment into a more general scheme is presented. The combined acknowledgement system takes elements of the aggregation system already presented, and applies it to the 802.11 acknowledgement scheme in order to reduce the overheads incurred.

8.1 802.11E BLOCK ACKNOWLEDGMENT

Currently, the default operation of 802.11 is to acknowledge every frame received. This requires an entire transaction, including reserving the wireless medium, and no data transmission can occur in this period. While the 802.11e extension does add a Block Acknowledgment control frame, it is optional, requires extra negotiation, and requires the network to be operating in the contention free HCF mode. This means that it cannot be used for ad-hoc networks, or by stations without the extra 802.11e capability.

However, a block acknowledgement scheme is very useful, as it lowers the number of control frames sent. Control frames are important as they are needed as part of the coordination of the network, but they take up the wireless medium without sending any actual useful information. Therefore it will be beneficial to have a scheme similar to block acknowledgements available in all 802.11 wireless networks.

8.2 COMBINED ACKNOWLEDGEMENTS

The authors propose a 'combined' acknowledgement system. The same acknowledgement frame header as existing 802.11 will still be used, but will be followed by the sequence control fields of each frame that is being acknowledged, as shown in Figure 8.1. As the sequence number of each frame is independent of whether the frame was aggregated, was sent alone, or what order it was sent, it can be used to uniquely identify the frame.

2	2	6	2	2		2	4
Frame Control	Duration	Receiver Address	Sequence Control (pkt 1)	Sequence Control (pkt 2)	......	Sequence Control (pkt n)	FCS

Figure 8.1: Combined acknowledgement frame format

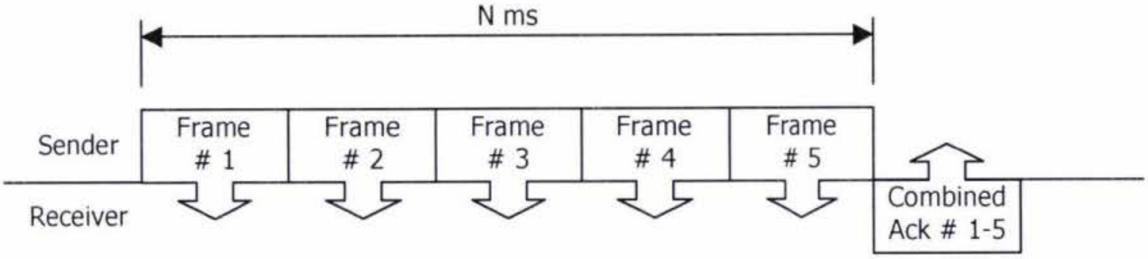


Figure 8.2: Example of time triggered combined acknowledgement

This system will not require the station to be operating in any specific control mode, and could be used for both aggregated packets and packets sent individually. Block acknowledgements work over a certain time duration in which multiple frames are sent. A similar system is used for this system a combined acknowledgement frame is only triggered after either a certain number of frames or a specified time period, which ever occurs first. The time limit and number of packets would be defined as part of the 802.11 Management Information Base.

Figure 8.2 shows a simplified example of several frames transmitted within the specified time period, after which a combined acknowledgement is sent back to the sender from the receiver. In total, this sequence takes 6 separate transactions that include all MAC sub-layer and physical layer elements such as a medium access and a physical layer preamble. Using the existing 802.11 acknowledgment system, this same process would take 10 separate transactions.

It is necessary to specify both a time limit and a packet limit for the acknowledgement trigger. The time limit would occur in periods of low traffic, where the acknowledgement timeout period may elapse before enough packets reach the receiver to reach the specified number of packets. Relying solely on a time limit system would mean that a large number of packets would have to be buffered at the sender under periods of heavy traffic load in case they were not transmitted correctly.

8.3 EXTENSIONS

8.3.1 Integrating Combined Acknowledgements with Aggregation

In essence, the combined acknowledgement control frame is the same as the existing 802.11 acknowledgement, but with the sequence control fields of the multiple packets being acknowledged "aggregated" as payload. A variant of the scheme could be used with the aggregation system detailed in chapters 5 through 7.

In the aggregation scheme, each packet has an individualised Packet Check Sequence (see 7.1.3) to check for errors in individual packets. In this scheme, the receiver needs a method of informing the sender that a packet was not received correctly. With the initial combined acknowledgement system, this would have to wait until the next timeout or packet count before an acknowledgement is sent.

In the event of individual packet errors in an otherwise error-free frame, the combined acknowledgement system can be triggered early to send a combined acknowledgement up to and including the current frame. In this way, the erroneous packet can be retransmitted as soon as possible.

8.3.2 Link Quality

A possible extension to the acknowledgement scheme would be to add some form of link quality measurement. This could be implemented directly by adding a link quality field to the acknowledgement header, or indirectly by assessing the proportion of successful transmission over a specified time period.

This technique would be very useful for frames that failed to transmit – as the transmitter of these frames could use information about the link quality in a control scheme e.g. a payload size backoff scheme for aggregation, to improve the chances of correct reception when it the frames are resent.

9 STORED FRAME HEADERS

As a further improvement to the MAC sub-layer operation, our research sought to utilise the fact that several frames may be sent between two stations that share very similar headers in each direction. For instance, a file transfer involves splitting the file into several small packets, which are sent one by one to be reassembled at the receiver. In this transfer, the same header will be used for each packet apart from the sequence number. If the header is remembered at the receiver from the first packet, it does not need to be resent.

Although the header is only a small fraction of the payload size, it is often sent several times over – in many cases it is used continuously. This means reducing the header size may have a significant effect on throughput. For example, storing the frame header for the transfer of a large file should have some effect, as the header size will be repeated a large number of times.

This scheme was designed with systems that use a slower transmission speed for the MAC header. Originally it was thought that this was common practice, but with further research it was unclear at what speed most common 802.11 devices use to transmit the MAC header, due to the closed nature of these devices. Therefore, it is assumed that the header is sent at the highest available transmission rate – as this is the highest possible rate for the MAC header according to the standard. See Excerpt 3.1 (page 19) and its associated comments for more information.

9.1 CONTROL SYSTEM

9.1.1 Global Control System

The frame headers need to be stored at all stations contained in the BSS, as both the receiver *and* the non-receivers need to know the header. In the case of an RTS/CTS handshake, the receiver knows beforehand that it is going to receive a frame. But if the RTS/CTS system is not used, the receiver has no way of knowing that is about to receive a frame – and similarly the non-receivers do not know the next frame is not destined for them. Hence all stations must be aware of the stored headers, to at least determine whether the frame is destined for them, or some other station.

Therefore, there is a need for a global control system responsible for managing which headers are stored. This control system would have to decide which headers should or should not be stored. This will need information about the types of transfers occurring, and preferably information as to if and how long each transfer will be repeating.

This information cannot be determined at the MAC sub-layer however. Therefore, the stored frame header system must be integrated into the quality of service differentiation system. Quality of service provides a certain level of service across the wireless network to certain types of traffic, and in a similar way, would set the frame headers that are stored to those of the highest priority transfers. However, this means that the system can only be used with the 802.11e coordination functions described in 2.3.4, which means that it would suffer from the problems identified in 3.2.

Another possible method is to make stored frame headers user or application controlled. In a user controlled system, the user of each device on the wireless network would nominate a single application that would benefit from stored frame headers, such as a file transfer or streaming video in the wireless network device settings. If application controlled, the applications running on the device would negotiate between themselves to decide which has the greatest need for benefit from stored frame headers.

The application or user nominating each candidate for the stored frame header would also include an estimate of duration, traffic load and possibly other basic details. In this way, each station on the network would have a candidate for a stored frame header. All stations in the BSS would then be entered into a single control round with an algorithm used to decide which of the candidates would be successful depending on the basic information given. This system could be used for both ad-hoc and infrastructure networks.

The main drawback is that each control round is an added overhead, but depending on the duration between control rounds, the overhead may be negligible. It also requires some knowledge on the part of the user or application, and may not be easily compatible with many applications.

Lastly, the stored frame header system could be implemented most simply by adding it solely as a user controlled system. This would be only used for wireless network links that are long lived – such as a wireless backbone network. In the example of a backbone system, each intermediate wireless station will typically only have an uplink and a downlink. Each of these stations would only observe 4 varieties of headers.

Therefore, for a fully user operated system, during the setup stage of each station, the user would enter the address of both the uplink and the downlink station, and then set a flag enabling the stored frame header system. This would be done at each station.

9.1.2 Complexity

Another issue with the stored header scheme is complexity in regards to how many headers should be stored, and the control system that determines which headers should be stored. A simple form of storing frame headers is to keep all headers. This is simple to implement and can be controlled with a single bit in the existing control fields, with no need for a global control system, as all stations will simply store all headers. However, this scheme will not be very efficient in large networks as the number of possible headers increases exponentially due to the combinations of transmitters and receivers. This is impractical as an extra header field would have to be introduced to allow for the unique identification and also possibly memory limitations.

Storing a limited number of headers requires a more complicated control scheme such as the two outlined previously, with the possibility of extra handshakes as well. However, if the number of stored headers is small enough, it may be possible to include the information within the current header fields. For example 3 bits could be used to identify 8 stored headers.

9.2 FRAME FORMATS

The stored frame header would need to transmit the 2-byte Frame Control fields of the header, as this would be used to identify that the frame is one which has had its header fields stored. A new data subtype "Stored Header" would be created. Other than the Frame Control, only the Sequence Control field is needed, giving a header size of only 4 bytes (see Figure 9.1a), compared with 24 for the full header! If the payload is a different length but otherwise meets the criteria for a stored header, a different subtype would be used, and the duration field also included for a 6-byte header (Figure 9.1b).

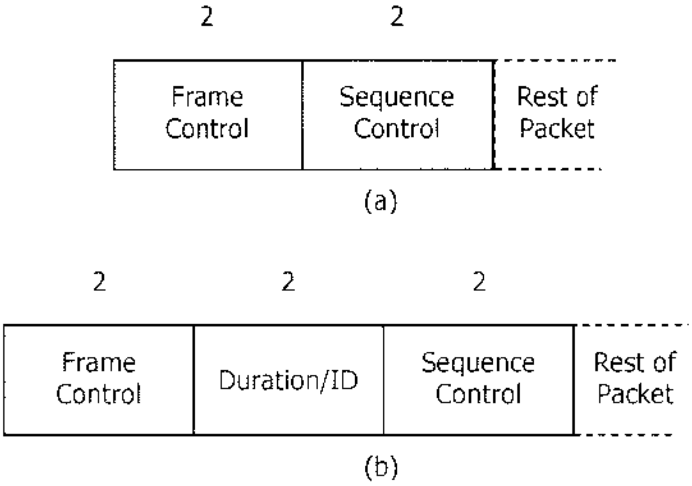


Figure 9.1: Reduced Header Sizes of Stored Header System

This part of the system is implemented by using a unique Data frame Subtype (with a description of 'Repeat Header (initial)'), along with a unique identifier for this particular header. The permission to send this type of frame, along with the value of the identifier, would have previously been granted by the control system. Upon transmission of this frame type, *all* stations will store the header in memory either as a new stored header, or replacing the header that previously had used that identifier.

When receiving a subsequent Data frame Subtype with a description of 'Repeat Header (repeat)', the stations will then be able to look up the header corresponding to the identifier of the repeat header, and then use the data contained in the header that was stored earlier combined with the new fields in the reduced header to recreate the full frame header as if it had been sent with the frame initially.

9.3 BENEFITS OF STORED HEADERS

As part of the 802.11n standard extension, certain types of data transfer are designated as requiring a certain level of performance. For example, streaming video and audio requires low delay and strict ordering. To achieve this, 802.11n defines modes of operation that can deliver the required performance for several likely usage scenarios. It is proposed that some of these usage scenarios – such as streaming video – would benefit greatly from the repeated header scheme as they involve repeatedly sending similar data to a single receiver.

The common home applications that would benefit from this scheme are file transfers and streaming multimedia and games. Also, backhaul wireless links that uses 802.11 would benefit, as the number of receivers will be minimal for the static links of a backhaul link.

10 SIMULATION

The simulation environment used was MATLAB. The first thing noticed in the simulation design stage was the lack of (wireless) network simulation tools that could cope with variable packet sizes and aggregation of packets. Most simulation tools required a set packet size for all packets in a particular simulation iteration.

Some network simulators do allow variable packet sizes within simulation iterations, but to allow aggregation of these packets requires programming software modules, and secondly these simulators do not have very good measurement and statistics recording within the MAC sub-layer.

Therefore, it was decided to use MATLAB, as the researchers involved had a strong grounding in MATLAB, and ground-up design of the aggregation mechanism was necessary anyway.

It was also initially thought that at least a simple version of aggregation could be written in VHDL, to look at aggregation targeted as an embedded solution. However, it was found that without access to an existing 802.11 MAC sub-layer in VHDL, the aggregation mechanism could not be simulated properly.

10.1 PACKET GENERATOR

The first thing to be completed for the purposes of simulation was to write a simple MATLAB function that generated traffic to send across a simulated 802.11 wireless network. This worked by generating a stream of packets with various sizes. These packets were a simulation of what a typical wireless network card would see at its MAC SAP.

The packet size cumulative distribution function found by the Stanford University Computer Science building wireless network study was used [11]. This was divided up into 15 separate packet size 'bins'. The packet generation function randomly selects a packet size from each section according to a probability that roughly matches the packet size cumulative distribution function. Figure 10.1 shows the cumulative distribution function of the packets from the generator used in the simulations.

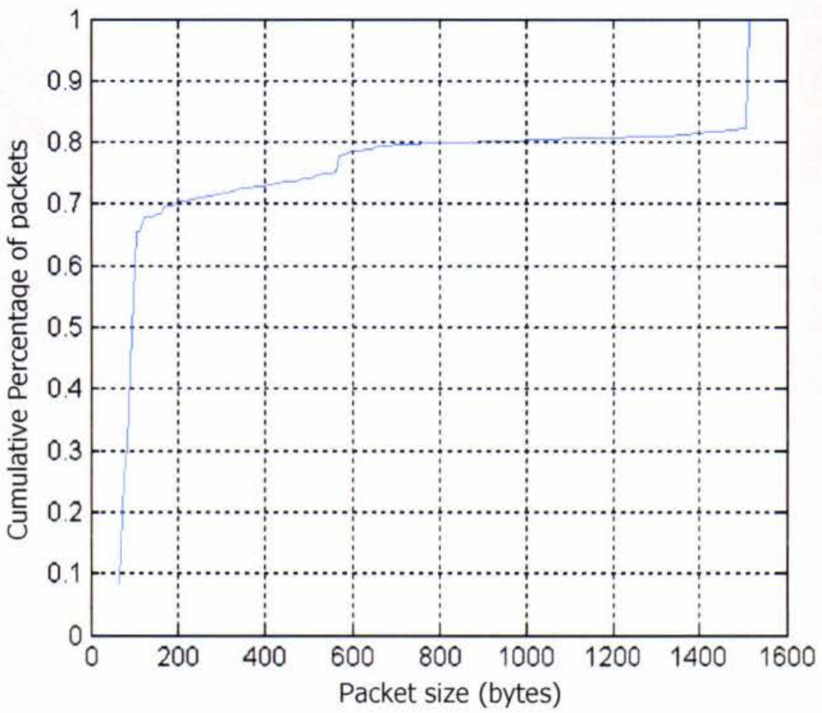


Figure 10.1: Cumulative packet size distribution function of packet generator (compare with Figure 3.3, Tang and Bakers findings)

A set of packet arrivals at the MAC SAP is generated using the packet generator. An average packet inter-arrival rate is passed to the packet generator, which then generates a set of packets per simulation iteration for each station being simulated. Each packet has an arrival timestamp, a packet size and a random destination address. The destination address is randomly selected, with an even amount of packets overall going to each station simulated in the network.

The packets recorded in the study by Tang and Baker were a mixture of both connection-oriented and connectionless protocols at the network layer. To the wireless network MAC sub-layer, the acknowledgements associated with the connection-oriented protocols' packets and the packets from both the connection-oriented and connectionless protocols appear as "data". The packet distribution presented by Tang and Baker, and thus the packet distribution used in the generator, includes the acknowledgements.

The expected average packet size for the packet generator can be found by calculating the expected value of the packet size generation function. The expected value is 3194 bits (399.25 bytes). Using this value, the average data rate per second can be calculated using the average inter-arrival rate. The table below shows the typical inter-arrival rates used per station in the simulations, and their associated data rates.

Table 10.1: Packet inter-arrival rates per station and approximate data rates

Packet inter-arrival rate	Data rate (at MAC SAP)
500	1.52 Mbps
1000	3.05 Mbps
1500	4.57 Mbps
2000	6.09 Mbps
3000	9.14 Mbps
4000	12.18 Mbps
5000	15.23 Mbps

10.2 SIMULATION DETAILS

Outlined below are the operational details of the simulation environment used for the results presented in this thesis.

10.2.1 Simulation Operation

The simulation steps through each iteration using a global timer. The step for the algorithm is based on the slot timing of the physical layer used (802.11a or 802.11b) for the simulation. This slot time is used in the 802.11 physical layer standards to synchronize the beginning of most of the MAC operations. The slot times for the 802.11a and 802.11b standards are given in Table 3.3 and Table 3.1 respectively.

At each step, any packets 'scheduled' to arrive (using the timestamp from the packet generator), are classified into a queue according to there destination address. Also at each step, the DCF coordination function is used to control the stations that belong to the simulated network. The full DCF operation has been implemented from the 802.11 standard.

In the same time step, if the transmitter is ready to transmit, the next queue is determined using the queueing control mechanism (see 5.3), and then packets are aggregated and sent from that queue.

The simulation repeats several times to enable more representative averages for the statistics generated. Each set of packet arrivals generated by the packet generator is used 5 times, and for each combination of stations and packet inter-arrival rates, 3 different sets of packet arrivals are generated, making a total of 15 simulation iterations per combination.

10.2.2 Simulation Architecture

The simulation uses a dynamic group of packet arrays to model the aggregation queues. Each queue is simulated as an array, with the group of array representing the set of queues. Each packet is stored with its length in bytes, its arrival time and also an arrival number that records the order in which the packets enter for the purposes of time analysis.

The DCF algorithm was implemented by using a global medium free flag, and a global medium access flag that worked similar to a token in Token based forms of networking. The medium free flag has a similar effect to the Network Allocation Vector (NAV) indicator used in 802.11. The NAV is a flag at each station which, when set, prevents a station from transmitting until the end of the NAV period. In the simulation, if the medium free flag is false, all stations defer in the same way as if there was a global NAV. If the flag is true, any stations that have packets to send will contend for the right to send a packet using the CSMA/CA algorithm, which consists of a constant wait period (DCF interframe space, DIFS) followed by the random backoff period.

The medium access flag is then set to the first station whose backoff timer gets to zero, which then allows that station to send by entering into a sending loop where the aggregation mechanism operates, and sends an aggregated frame. If the medium is not free, all stations do nothing until it is free. If a station has to wait to select the right queue first, it still retains access to the medium. The medium free flag is set back to true at once a station has finished transmitting.

In the case of a collision, two stations request access to the medium, but due to the collision, neither gets access to the medium. The simulation still sets the medium free flag to false for the period of the longest transmission of the stations involved in the collision. This because 802.11 wireless network interfaces cannot sense a collision, and will only discover the error when the acknowledgement control frame times out.

10.2.3 Simulation Statistics and Metrics

Statistics are generated each iteration of the simulation. Some statistics are generated on a per station basis and some are based on the overall wireless network. The set of per station statistics gene is shown below:

Table 10.2: Station only statistics generated in simulation

Metric	Unit
Transmissions failed	-
Average packet delay	µs
Median packet delay	µs
Maximum packet delay	µs
Average queue length	bytes
Average queue length	packets
Maximum queue length	bytes
Maximum queue length	packets
Station throughput	Mbps

The number of transmissions failed records the number of transmissions that did not complete successfully due to simulated errors from the simulation error model. In cases where the error model is not used, this statistic will always be 0.

The average, median and maximum packet delays are calculated at the end of each simulation iteration. An array of all of the packet's delay times is calculated by subtracting each packets final sending time (recorded as the time when the frame that the packet was aggregated into begins a successful transmission) from the arrival time of the packet at the aggregator queues.

As the queue implementation in the simulation is fully dynamic, there is no set limit to the queue length as there may be in a real-world chipset. In the real world, if the queue is kept onboard the wireless network card or in an embedded system, there will only be a limited amount of memory space available. If the queue is kept in system memory on a computer with an operating system, there may be limitations imposed by the operating system on the amount of memory that the wireless network card driver can utilise.

Therefore, an informative statistic is the average and maximum queue lengths of each station during the simulation. This gives an indication as to the amount of memory that would be required in a real-world implementation for the full benefits of aggregation. The queue length statistics are given in both bytes and packets. For the issue of memory limitations, the byte length of the queue needs to be known. In circumstances where the implementation is on a full computer system with an operating system, however, a network driver's memory usage may be measured in terms of length of the data structure used to represent the queue, which is the same as the number of packets in the queue.

Lastly, the throughput seen by each station is also measured.

The set of statistics generated for the overall network in the simulation are shown below in Table 10.3.

Table 10.3: Overall network statistics generated in simulation

Metric	Unit
Total bytes (top MAC)	bytes
Total bytes (top PHY)	bytes
Byte efficiency	- (%)
Overall send time	ms
Simulation time	ms
Time utilisation	- (%)
Overall throughput	Mbps

The total number of bytes at the top of the MAC sub-layer refers to the total number of bytes presented to the MAC sub-layer at the MAC SAP by the layers above it. This number is the total over all data given to the MAC sub-layer in the form of packets, which also includes all header data from the above layers. The number is used internally within the simulation for throughput calculations, and is at the same position as where the throughput target is defined for in the 802.11n proposal [5].

The total number of bytes at the top of the physical layer is the total number of bytes given to the physical layer device to send by the MAC sub-layer as frames. This includes all of the data given to the MAC sub-layer, plus the MAC sub-layer headers. Also, if a frame has to be retransmitted, the data counts every time. This is due to the nature of the 802.11 MAC sub-layer, where there is no way that the system knows whether the frame has transmitted successfully until no acknowledgement frame is received within the required time.

The byte efficiency of the MAC sub-layer is a percentage figure showing how many bytes where presented to the physical layer in order to send the packets across the network as frames.

The simulation time is simply the time length between when the first data transmission is attempted, and the time when the last data transmission is acknowledged successfully. The overall send time is the time spent sending data, as opposed to the time spent waiting for the medium free, backoff periods, or any other operations associated with the coordination function.

The time utilisation is a percentage figure of how much time was spent sending data during the simulation.

The overall network throughput is probably the most important metric – as this is the throughput achieved by the network *above* the MAC sub-layer. This figure is directly comparable to the actual throughput calculated in 3.1.2, and the throughput target for the future 802.11n standard.

10.3 ERROR MODEL

802.11 networks, like all wireless telecommunications, is subject to several types of interference that cause errors ranging from single bit errors that are easily detected and fixed through to total loss of communication. Therefore, some from of error model is essential in any simulations of 802.11 wireless networks.

The errors that are most significant and most likely to affect the improvements in the thesis are relatively long periods of interference. In 802.11 networks, especially those operating in the 2.4 GHz ISM band, this is typically caused by other devices using the same frequency band, or an adjacent band causing spectral overlap. For example, many cordless telephones use the 2.4 GHz band, and if present in the vicinity of an 802.11b or 802.11g network can cause interference.

The typical occurrence of these errors is random and sporadic, but the duration is relatively long, especially when compared with the time scale used for most 802.11 operations. Therefore the basic error model used in the simulations is to have a relatively long period of errors occur at long random intervals. The error periods were implemented to be constant error rate over the time of the error period. At each transmission, a random probability function is used to generate an outcome of whether each transmission has failed or not. The probability is scaled by the duration of the packet, as longer packets in these periods are more prone to errors.

11 SIMULATION RESULTS AND DISCUSSION

11.1 QUEUEING ACCESS

The first simulations ran in this research were those for the different queueing access methods as described in 6.1. It was considered that the targeted implementation of aggregation would be software based, and would therefore it would allow the implementation of either look-ahead queueing or an indexed queueing machine.

It was decided to run the queueing access simulations before the queueing control algorithm simulations, as it would mean that the queueing control simulations would have to be completely re-run. This is because better queueing access would at least significantly change the performance of aggregation, and possibly change the effectiveness of one or more queueing control algorithms.

The simulations were run over the combinations of number of stations and packet arrival rates per station shown in Table 11.1. The aggregation mechanism was used with the load based algorithm for the queueing control. It was assumed that the relative results for the queue access simulation was independent of the queueing control used. The error model was disabled for these simulations.

Table 11.1: Combinations for queue access simulations, with approximate data rates in Mbps

	Number of stations transmitting		
Pkts/s	4	8	12
500	6.1	12.2	18.3
1000	12.2	24.4	36.6
1500	18.3	36.6	54.8

Figure 11.1 shows the overall throughput results for the three simulations carried out for the 4 station network. There is essentially no difference between the normal queue access and the two extensions to the queue access. This is because the aggregation mechanism is congestion triggered, and no congestion occurs at the lower traffic levels.

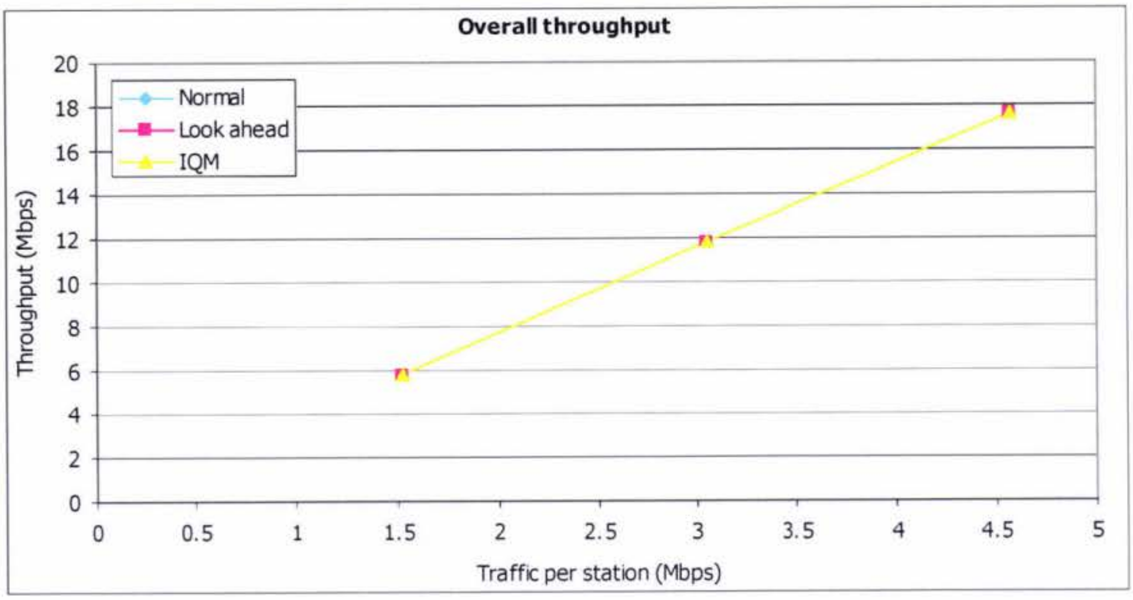


Figure 11.1: Overall throughput for queueing access simulation with 4 stations

The simulations carried out with 12 stations, however, have much higher traffic rates that will cause congestion over a typical 802.11a wireless network. Figure 11.2 below shows the overall throughput of the 12 station network simulations. When the congestion triggers the aggregation mechanism, the look-ahead queueing access method clearly gives higher throughput. The indexed queueing machine (IQM) system improves upon the performance of the normal queueing system, but not with as great as an effect as the look-ahead method.

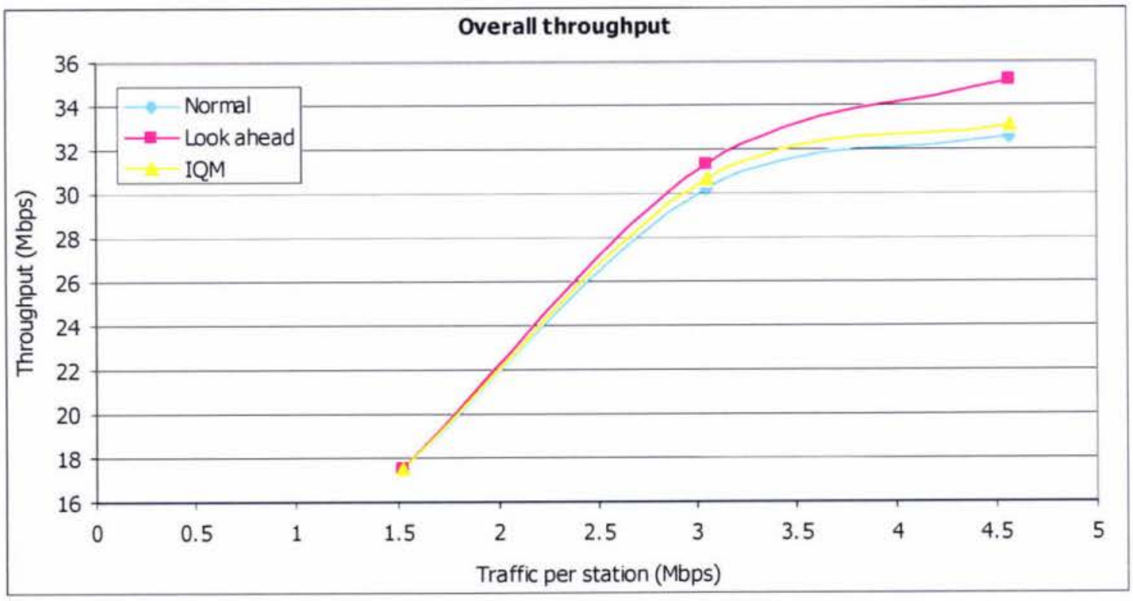


Figure 11.2: Overall throughput for queueing access simulation with 12 stations

The greater performance of look-ahead queueing is due to the system allowing the aggregation mechanism to access a larger number of queued packets. As the congestion increases, the queues start increasing in size. The look-ahead mechanism allows the aggregator to look through the entire queue, if needed, in the event of extra space being available at the end of the payload. The IQM system can only place new packets at the head of the queue. This means that the extra space left at the end of the payload can only be filled with the new packets arriving since the last successful transmission, rather than all of the packets still waiting to be transmitted.

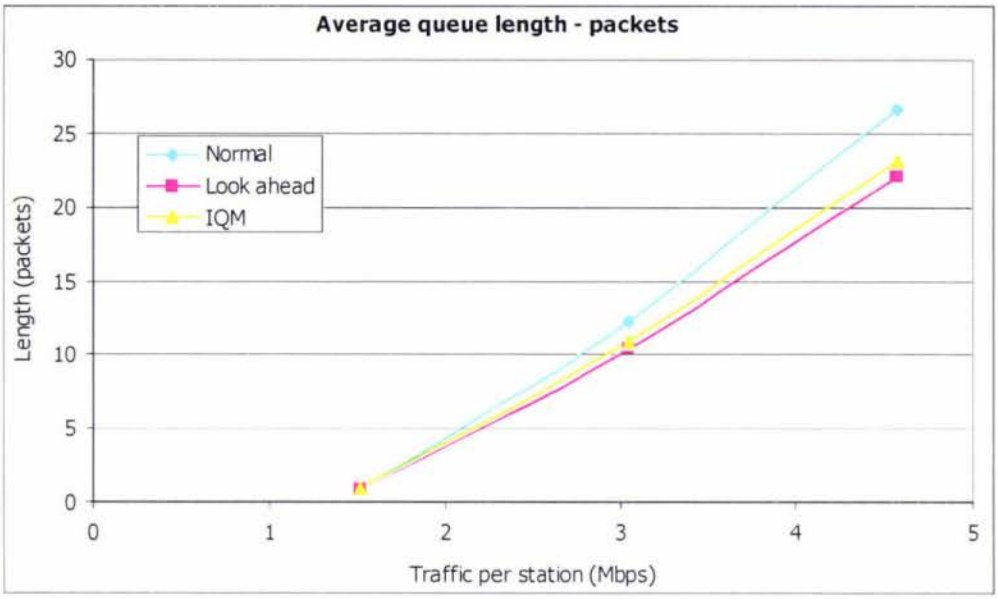


Figure 11.3: Average queue length for queueing access simulation with 12 stations

Looking at the average queue length in packets, it can be seen that the look-ahead system reduces the average number of packets in the queue slightly more than the IQM system. As both of the systems target small packets to aggregate into the end of the payload, a small difference in the number of packets can make a significant difference in throughput. This is because smaller packets affect the overhead percentage more than larger packets – so aggregating a higher number of smaller packets will achieve a greater increase in throughput through reduction in the overhead percentage.

From the result of these simulations, the look-ahead queue access system was chosen for all of the subsequent simulations. If, in implementation of an aggregation mechanism, it is too difficult or too costly to implement look-ahead queue access, the IQM system could possibly be used to still to enhance the performance increase of aggregation.

11.2 SIMULATION OF QUEUEING ALGORITHMS

To evaluate the different queueing control algorithms for the aggregation mechanism as described in 5.3.2, the wireless networks were simulated using the 802.11a physical layer. The simulations were run with networks containing different numbers of stations that were all using one particular queueing control algorithm. The same networks using the existing 802.11 MAC sub-layer were also simulated for comparison purposes between all of the forms of aggregation.

All queueing control algorithms use the look-ahead queueing access method that was found to give greater performance in the previous section. The error model was disabled for these simulations.

Operational details of the simulation architecture and implementation can be found in Chapter 10 and the sections contained within. The approximate data rates per station are shown in Table 10.1. The queueing algorithms were simulated over networks of 1, 2, 4, 6, 8, 12, 16 and 20 stations, giving the combined traffic loads (in Mbps) shown below in Table 11.2.

Table 11.2: Approximate combined traffic loads in Mbps for queueing algorithm simulations

	Number of stations transmitting							
Pkts/s	1	2	4	6	8	12	16	20
500	1.5	3.1	6.1	9.1	12.2	18.3	24.4	30.5
1000	3.1	6.1	12.2	18.3	24.4	36.6	48.7	60.9
1500	4.6	9.1	18.3	27.4	36.6	54.8	73.1	91.4
2000	6.1	12.2	24.4	36.6	48.7	73.1	97.4	
3000	9.1	18.3	36.6	54.8	73.1	109.7	146.2	
4000	12.2	24.4	48.7	73.1	97.4	146.2		
5000	15.2	30.5	60.9	91.4	121.8	182.8		

The higher traffic loads used in the simulation (e.g. 50 Mbps and above) are well above the maximum possible throughput that the 802.11a physical layer and 802.11 MAC sub-layer can deliver in their current state (see 3.1.2 for calculated maximum throughput). This was designed to test the limits of both the aggregation mechanism, and the existing 802.11 MAC sub-layer. Also, because the aggregation is congestion triggered, such high traffic loads will maximise the congestion of the network, which should show the true effect of aggregation.

11.2.1 Low traffic loads

For the purposes of comparing the queueing algorithms in use with congestion triggered aggregation, the low traffic loads are those that all within the range of the existing 802.11 MAC sub-layer combined with the 802.11a physical layer. To illustrate the performance under "low" traffic loads, the simulations run with 1 and 2 stations transmitting (for any average packet per second setting), as well as the group of simulations with 500 packets per second inter-arrival rate (for any number of stations) are presented. Note that the set of approximate combined traffic loads for the group of 2 station simulations and that for the group of 500 packets per second simulation is the same; therefore these simulations will should show very similar results.

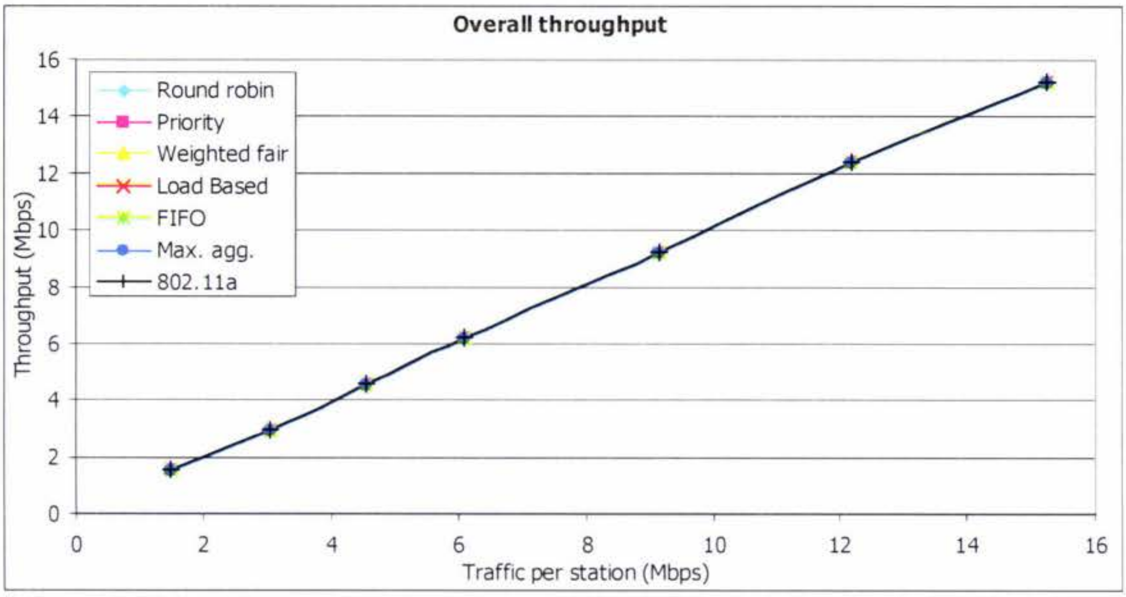


Figure 11.4: Overall throughput for queueing control simulation with 1 station

At all data rates used with one station transmitting, there is no significant difference between the throughput performance of the queueing algorithms (Figure 11.4), or between aggregation and the existing 802.11 MAC sub-layer. At these low traffic loads, the existing MAC sub-layer is actually slightly more efficient than the aggregation mechanism, although the difference in efficiency is not enough to cause any significant difference in throughput.

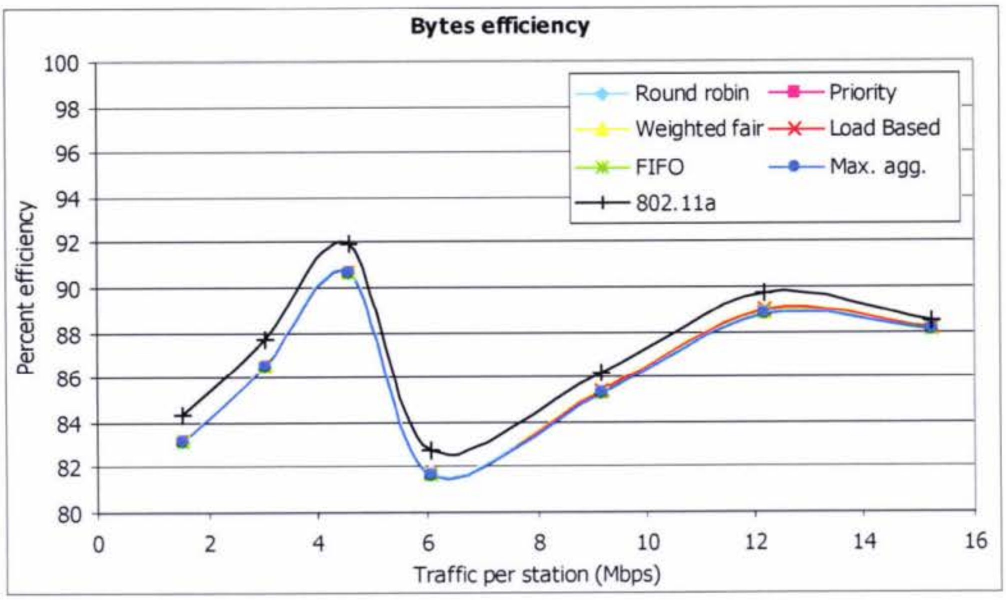


Figure 11.5: Byte efficiency for queueing control simulation with 1 station

The only difference between the queueing controls with only one station transmitting is in the packet delay statistics. Figure 11.6 shows the median packet delay for the round-robin queueing is consistently higher than that of the other five queueing control algorithms. Also, as the traffic load at the single station starts to approach the calculated limit of 802.11a (see 3.1.2), the existing MAC sub-layer starts to exhibit a much higher median packet delay.

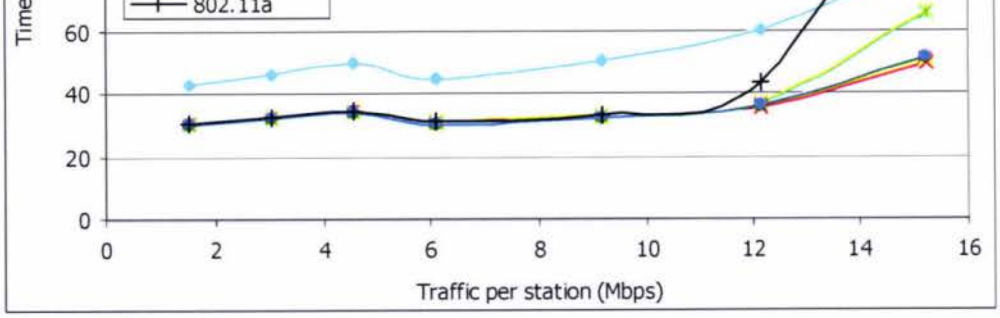


Figure 11.6: Median packet delay for queueing control simulation with 1 station

When 2 stations are transmitting at once, the lower traffic rates per station (below 20 Mbps for both stations combined) exhibit the same results as for one station. However, for the higher data rates, a significant difference between the aggregation mechanism and the existing 802.11 MAC sub-layer appears, as illustrated in the overall throughput results shown in Figure 11.7. The existing MAC sub-layer appears to hit a limit of approximately 20 Mbps. This limit is lower than the 27.44 Mbps calculated in sub-section 3.1.2 due to the average packet size being much smaller than 1500 bytes, as well as the DCF model used in the simulation being more realistic. Small differences also start to appear between the throughputs for each of queueing control, but the differences are not large enough to draw conclusions.

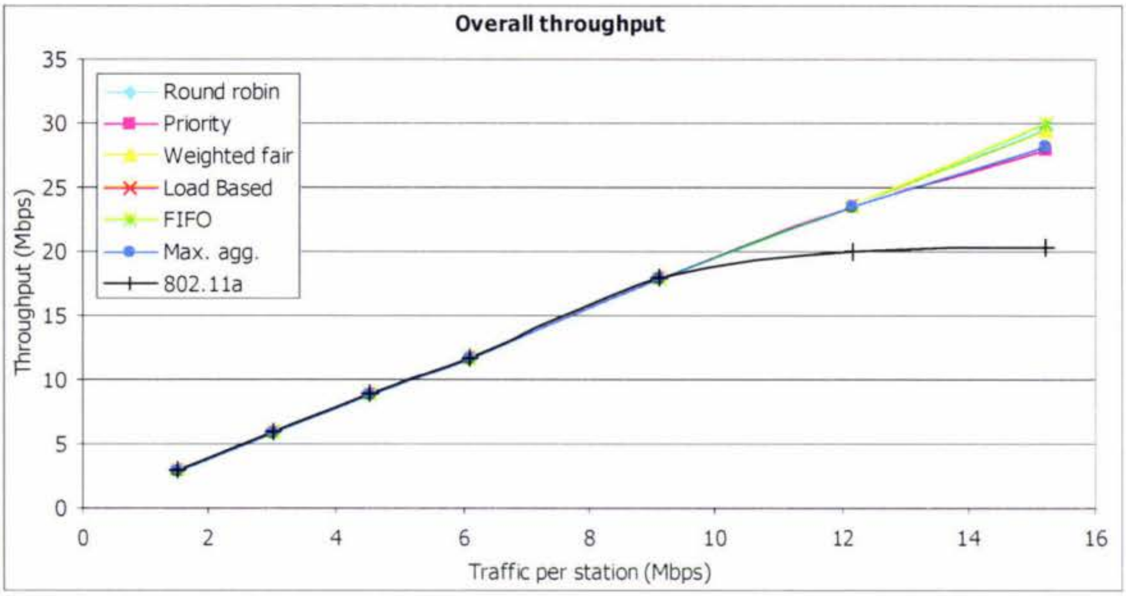


Figure 11.7: Overall throughput for queueing control simulations with 2 stations

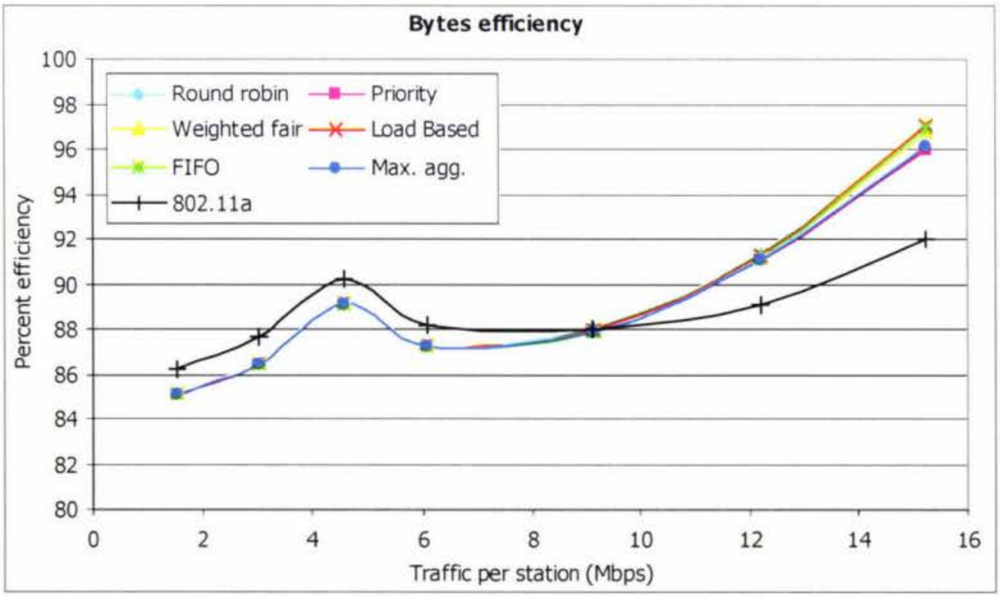


Figure 11.8: Byte efficiency for queueing control simulations with 2 stations

Figure 11.8 shows the byte efficiency for all of the aggregation MAC sub-layers become significantly greater than that of the existing 802.11 MAC sub-layer. As the network becomes congested with the higher traffic loads, the efficiency of the aggregation mechanisms improves upon its efficiency when the network does not have any congestion.

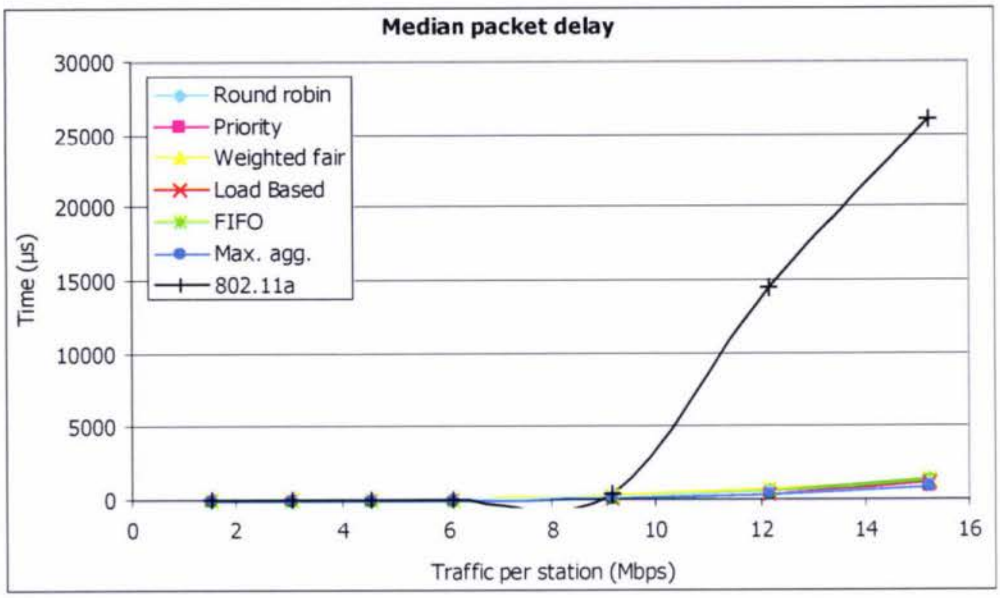


Figure 11.9: Median packet delay for queueing control simulations with 2 stations

The median packet delay for the existing 802.11 MAC sub-layer with the combined data rates over 20 Mbps is orders of magnitude greater than that for the aggregation MAC sub-layers at the same combined traffic rates (Figure 11.9).

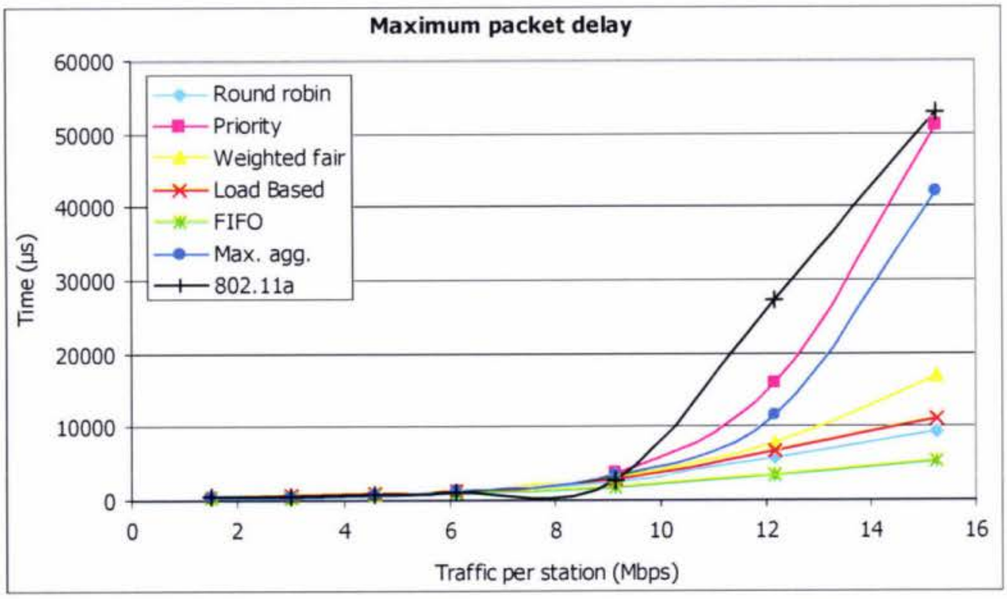


Figure 11.10: Maximum packet delay for queueing control simulation with 2 stations

Figure 11.10 shows that there is a large difference in the maximum packet delay for the different queueing control algorithms. Most notably, the priority and maximise aggregation algorithms have large maximum delays, even though their median packet delays were comparable to the other algorithms. The difference lies in how these algorithms handle the queues differently.

Priority queueing favours the queues that have the highest priority, therefore the packets in those queues will have lower packet delays, but the packets in the other queues will experience longer delays – hence the large maximum packet delay. The maximise aggregation algorithm favours queues with more small packets in, therefore causing longer delays in queues with larger packets.

The statistics for the 500 packet per second inter-arrival rate (over any number of stations) are almost identical to the statistics from the two stations transmitting group of simulations. Although the combined data rate of packets to send for the 20 station simulation at 500 packets per second is the same as the highest data rate for the 2 station simulation, the overall throughput seen is lower. This is because there are more collisions in the larger network. The existing 802.11 MAC sub-layer once again appears to be limited at about 20 Mbps.

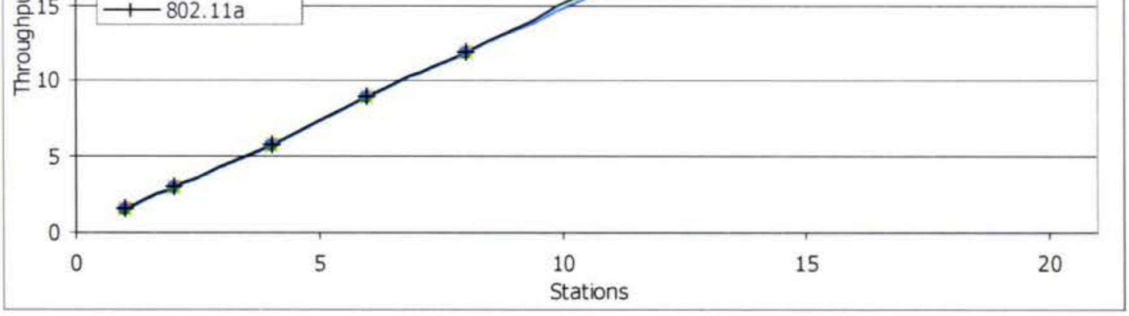


Figure 11.11: Overall throughput for queueing control simulation with inter-arrival rate of 500 packets per second at each station

11.2.2 Medium traffic loads

Medium traffic loads have been defined for these results as groups of simulations that have the lower combined traffic loads over all stations that is well within the limits of the existing MAC sub-layer when using the 802.11a physical layer, but have higher combined traffic loads that are near the limit of the aggregation MAC sub-layer using the 802.11a physical layer.

These simulations are those with 4 and 6 stations transmitting (for all packets per second settings), and the simulations with 1000, 1500 and 2000 packets per second inter-arrival rates for any number of stations. For these groups of simulations, the 4 stations and the 1000 packets per second groups share the same set of combined data rates as well as very similar results, as do the 6 station and 1500 and 2000 packet per second simulations. Due to the similarity, only the 4 station and 6 station results are displayed, as the conclusion drawn apply to all of the simulation groups that have medium traffic loads.

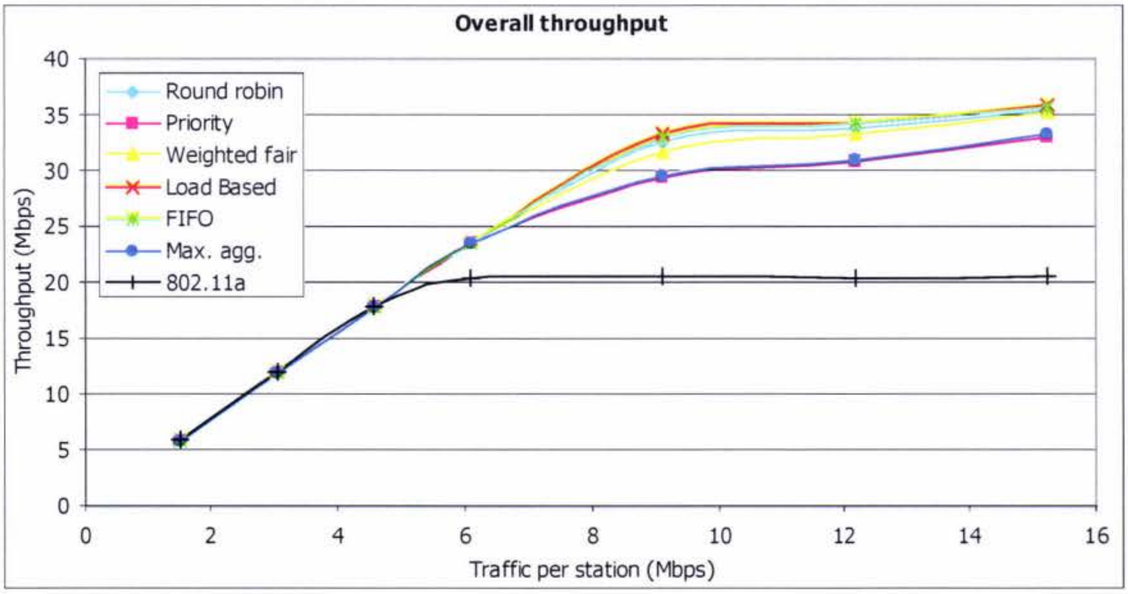


Figure 11.12: Overall throughput for queueing control simulation with 4 stations

In the overall throughput for simulations with 4 stations transmitting (Figure 11.12), again the existing MAC sub-layer limits at just over 20 Mbps. The aggregation MAC sub-layers appear to be approaching a limit as well, with the highest performing queueing control algorithms at around 35 Mbps overall throughput. Again at the lower levels, where there is now congestion, no significant difference is observed between the aggregation and existing MAC sub-layers.

Figure 11.13 shows a magnified view of the previous graph, focusing on the area where the aggregation MAC sub-layers level off. Here the priority and maximise aggregation queueing control algorithms have lower overall throughput. It must be noted that this graph shows the overall throughput. The priority scheme will give excellent throughput for some queues individually – and will, for example, choose a high priority queue with one small packet over a lower priority queue with several packets that can be aggregated.

Next worst is the weighted fair queueing algorithm, as this has elements of priority queueing still favours some queues over others. Then the round-robin queueing algorithm is next, and performs almost as well as the top two queueing algorithms – load based and first-in first-out. These two have very similar performance, although that for load based is slightly better.

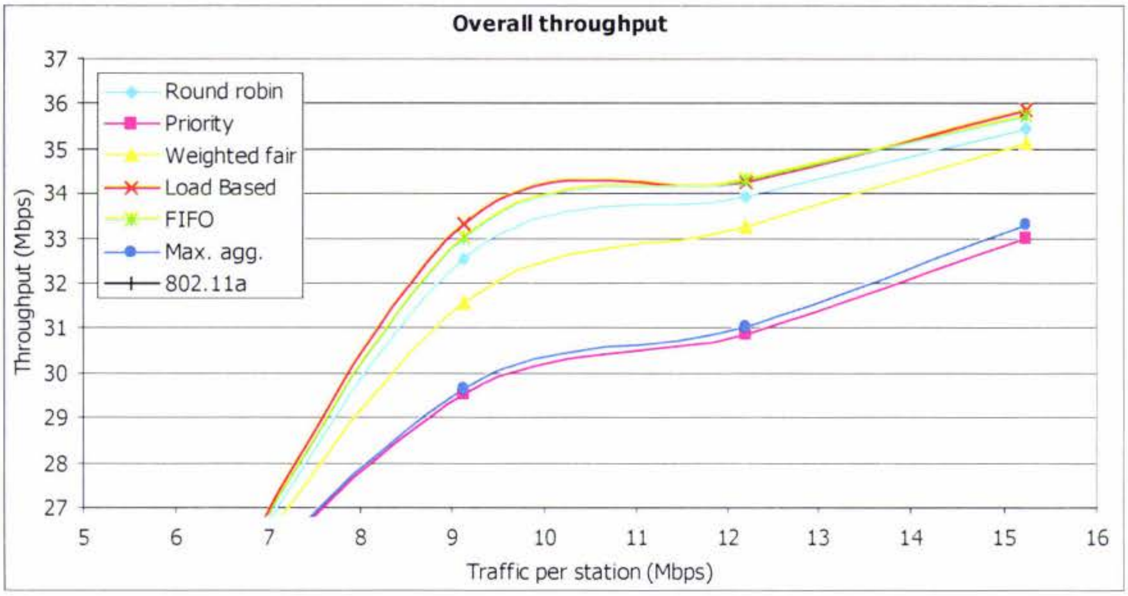


Figure 11.13: (magnified) Overall throughput for queueing control simulation with 4 stations

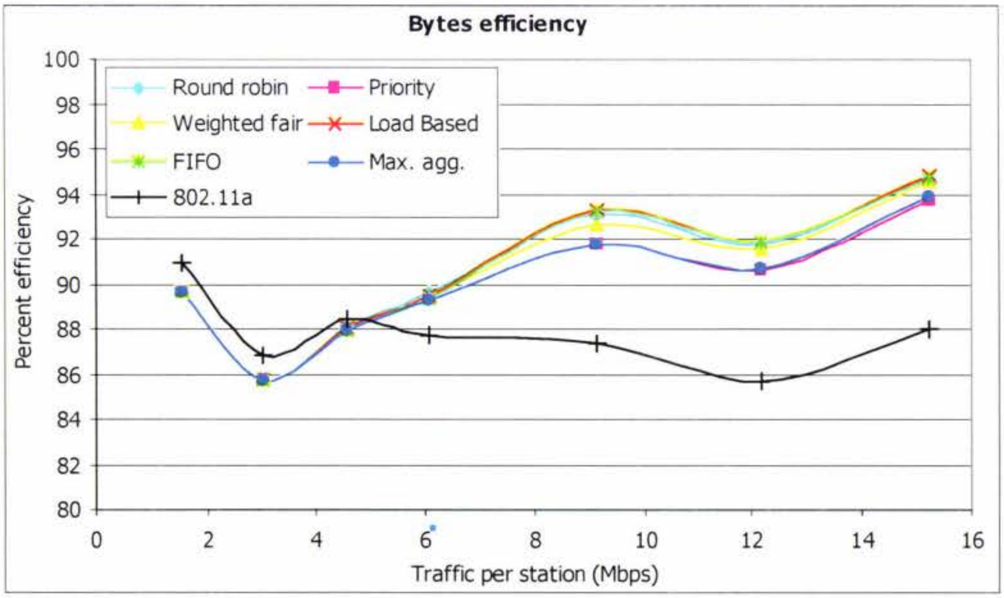


Figure 11.14: Byte efficiency for queueing control simulation with 4 stations

Looking at the byte efficiency of the different MAC sub-layers as shown in Figure 11.14, the difference between the existing MAC and the aggregation MACs efficiency is increasing. Also, the efficiency of the priority and maximise aggregation algorithms are slightly lower. Figure 11.15 shows the median packet delay (note that the existing MAC sub-layer packet delay is not fully shown, as it is so much greater). Here the priority and maximise aggregation, although performing worst in terms of throughput, perform best in median packet delay.

For priority this is due to the fact that the higher priority queues will be served as soon as possible, just reducing the packet delay for those queues enough to outweigh the increased packet delay for the lower priority queues. The maximise aggregation algorithm favours the queues that can transfer the most packets first – thus tending to move a larger number of smaller packet very quickly, reducing the *median* packet delay as the median is based upon the number of packets. In terms of the higher throughput algorithms, the load based queueing control performs slightly better than the first-in first-out control.

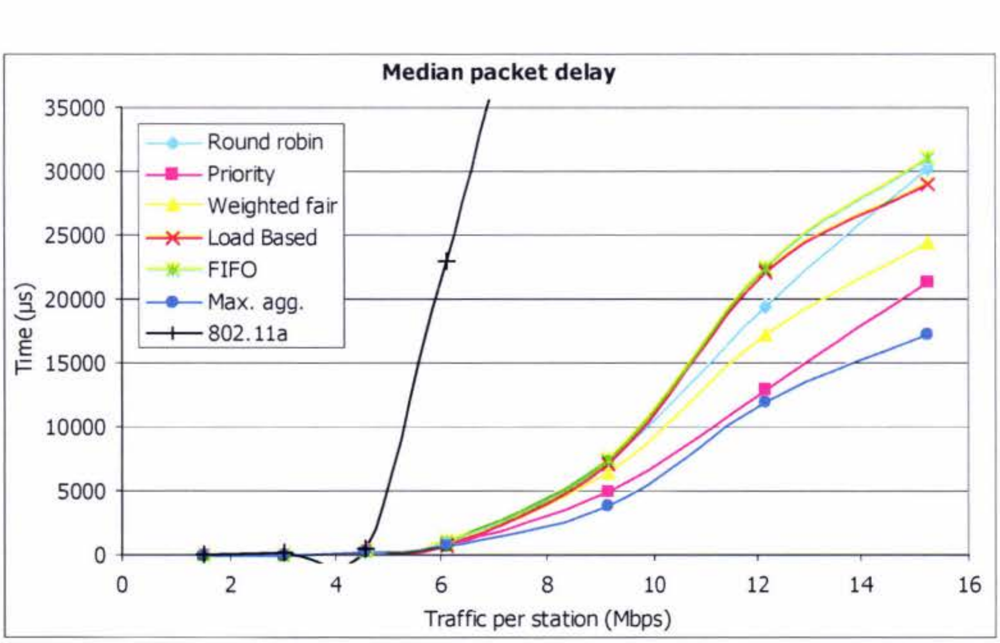


Figure 11.15: Median packet delay for queueing control simulation with 4 stations

Looking at the average packet delay (Figure 11.16), the priority and maximise aggregation queueing algorithms now perform the worst, due to a significantly number of longer packet delays (from low priority queues and queues with larger packets respectively). All of the other queueing controls all showed about the same average packet delay, although the weighted fair queueing was slightly higher.

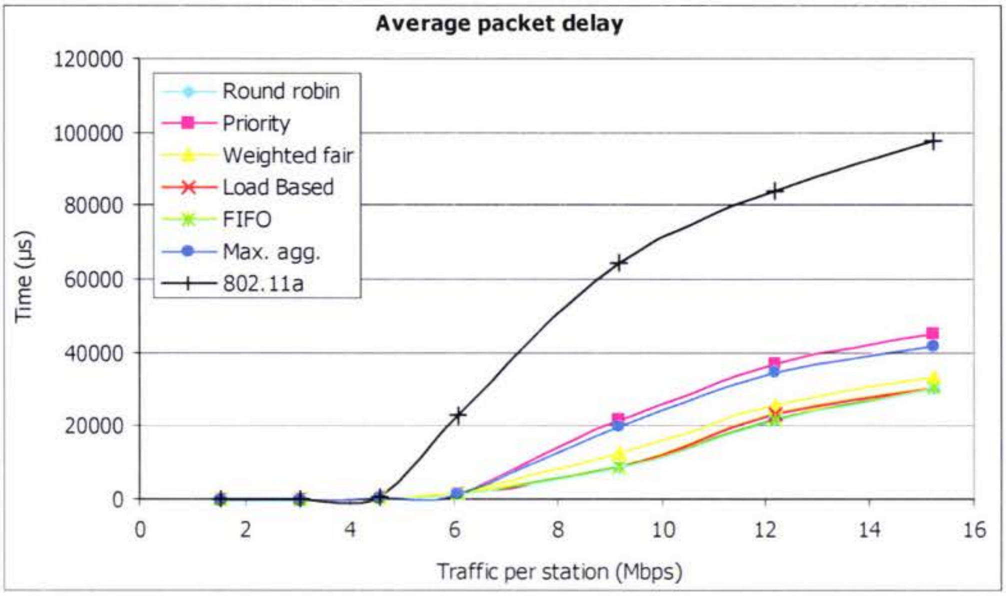


Figure 11.16: Average packet delay for queueing control simulation for 4 stations

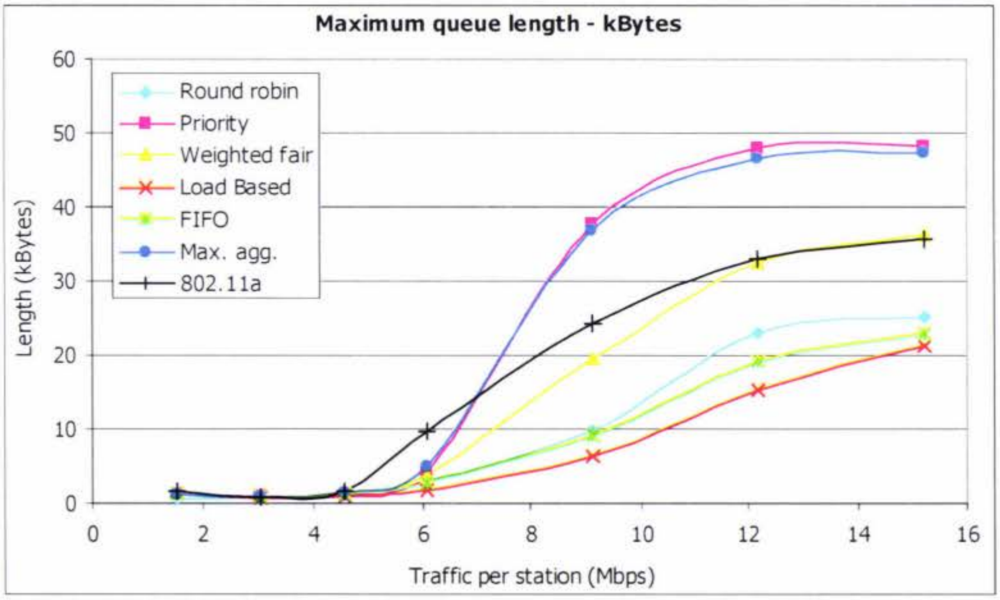


Figure 11.17: Maximum queue length for queueing control simulation for 4 stations

Considering that the set of combined data rates for this group of simulations is likely to be typical of the traffic that an implementation of aggregation would see, it is worth looking at the queue lengths. Figure 11.17 shows the maximum queue lengths used over the course of the simulations. For the load based control algorithm at the highest combined data rate, the longest that any queue got to was only just over 20 kB.

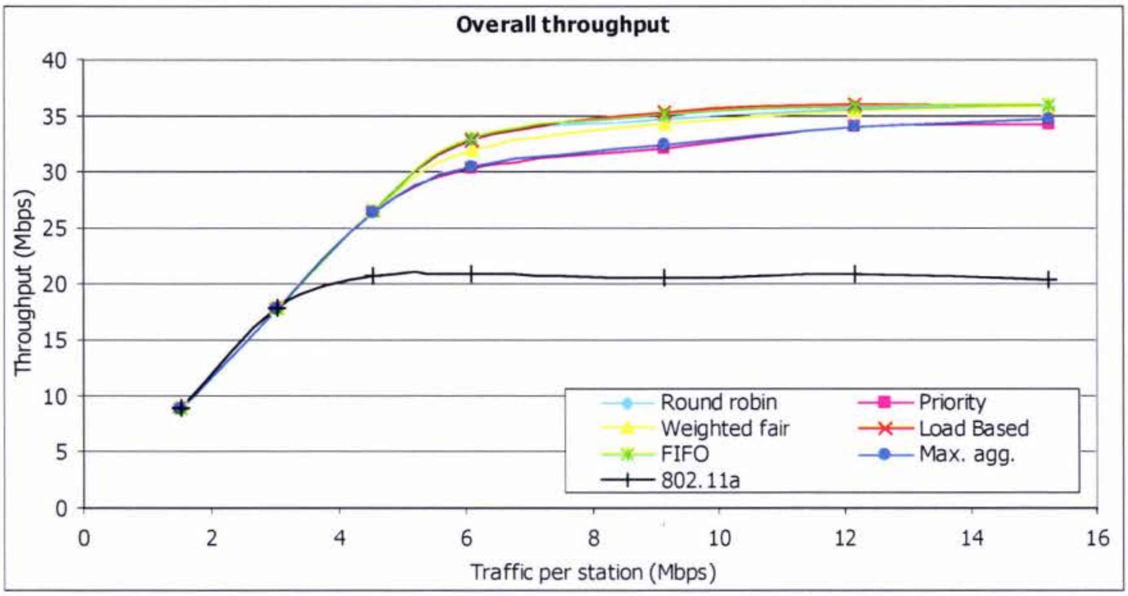


Figure 11.18: Overall throughput for queueing control simulations with 6 stations

Figure 11.18 above shows the overall throughput for the 6 station simulations. The highest three traffic per station setting for this simulation represent combined data rates of about 55, 73 and 91 Mbps respectively – higher than the physical layer bitrate, let alone the throughput ability. This level of traffic means that even the aggregation mechanism is now limited, but at an overall throughput over 35 Mbps, compared with a throughput of just over 20 Mbps for the existing MAC sub-layer.

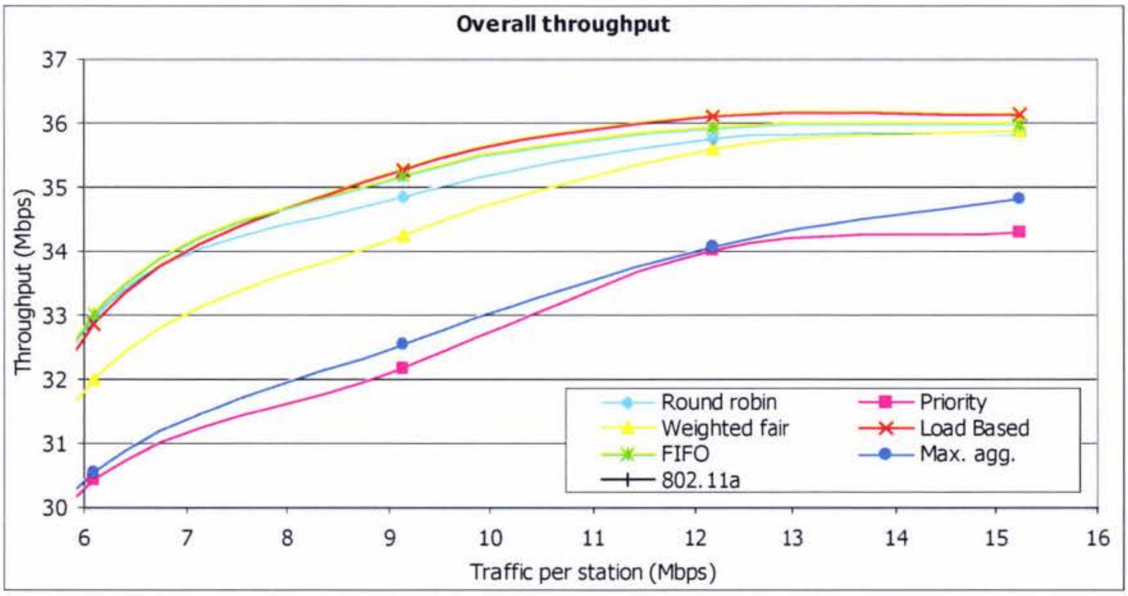


Figure 11.19: (magnified) Overall throughput for queueing control simulations with 6 stations

The magnified graph in Figure 11.19 shows that the load based algorithm is the best in terms of throughput, although only by a small amount. It seems as if the other queueing algorithms will also limit at a similar overall throughput level. However, these simulations use an essentially unlimited size for the packet queues. The queueing control algorithms should logically all approach the same limit once the combined traffic load is high enough, as with a high enough traffic load, any queue chosen by the queueing algorithm will be able to have the optimal aggregated payload to fit around the packet at the head of the queue. At these high loads, the average number of packets aggregated per frame will be the same for all of the queueing control algorithms.

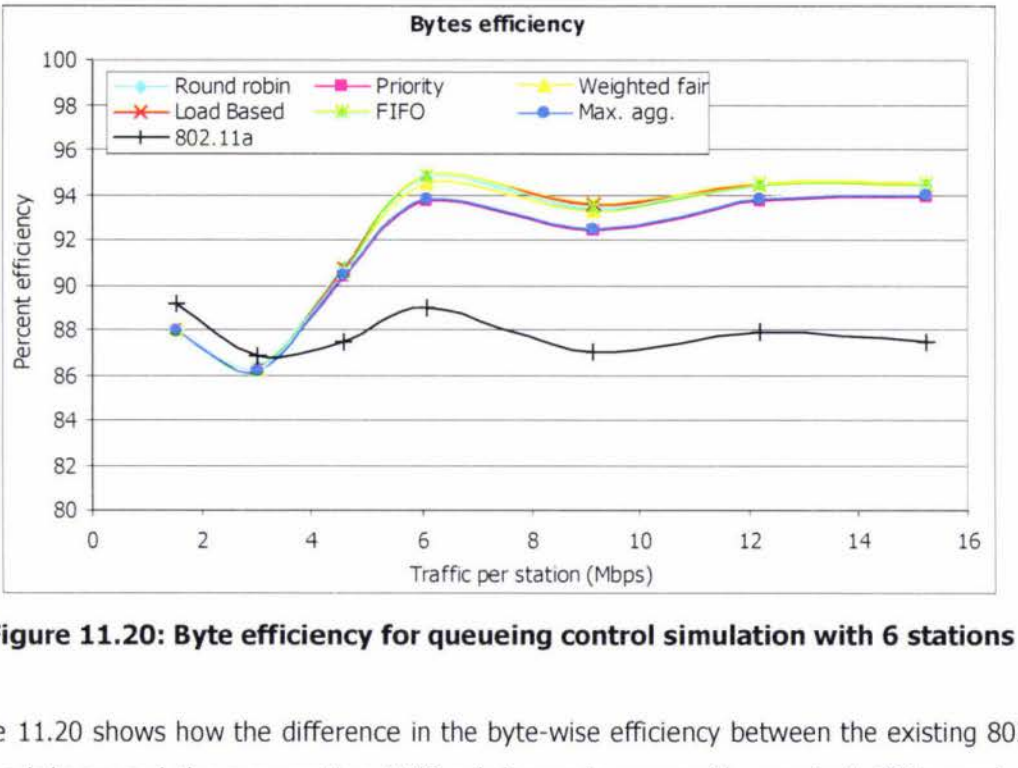


Figure 11.20: Byte efficiency for queueing control simulation with 6 stations

Figure 11.20 shows how the difference in the byte-wise efficiency between the existing 802.11 MAC sub-layer and the aggregation MAC sub-layers is reasonably constant. This graph also illustrates how the performance differences between the different queueing control algorithms closes up with the high traffic loads. The top four algorithms are within 0.2% of each other, and the two slower controls (priority and maximise aggregation) are within 1% of the other four.

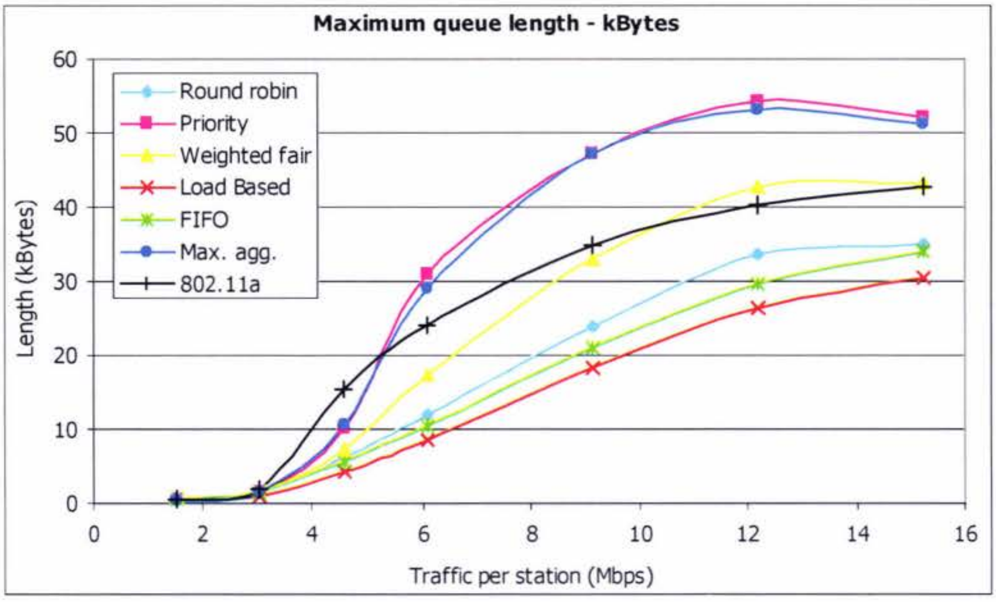


Figure 11.21: Maximum queue length for queueing control simulation with 6 stations

Looking at the maximum size of the packet queues, the load based algorithm is clearly the most efficient in terms of memory, due to the fact that the longest queue is always the one given service. The poor service given by the unfair queueing algorithms is obvious in this figure by the fact that the maximum queue length is greater than the much slower existing 802.11 MAC sub-layer.

In terms of packet delay, there is no difference to that of the 4 station simulations in terms of relative performance between the different queueing controls, but there is a slight increase in magnitude for all algorithms.

11.2.3 High traffic loads

In the simulations, very high traffic loads were used to see how the aggregation mechanism handled situations of overload. The combined data rates for these high traffic rates reached over twice the data rate of the 802.11a physical layer. The graphs presented in this section to illustrate the performance at high traffic loads are those from simulations with 12 stations.

Figure 11.22 shows the overall throughput, and clearly shows the limits of both the existing and aggregation MAC sub-layers. As described previously, all of the aggregation algorithms limit at essentially the same throughput of about 36 Mbps. As shown in Figure 11.23, the byte efficiency of all six queueing control algorithms converges to the same value. Once again, the packet delay for the algorithm changes slightly in magnitude, but not in relative performance.

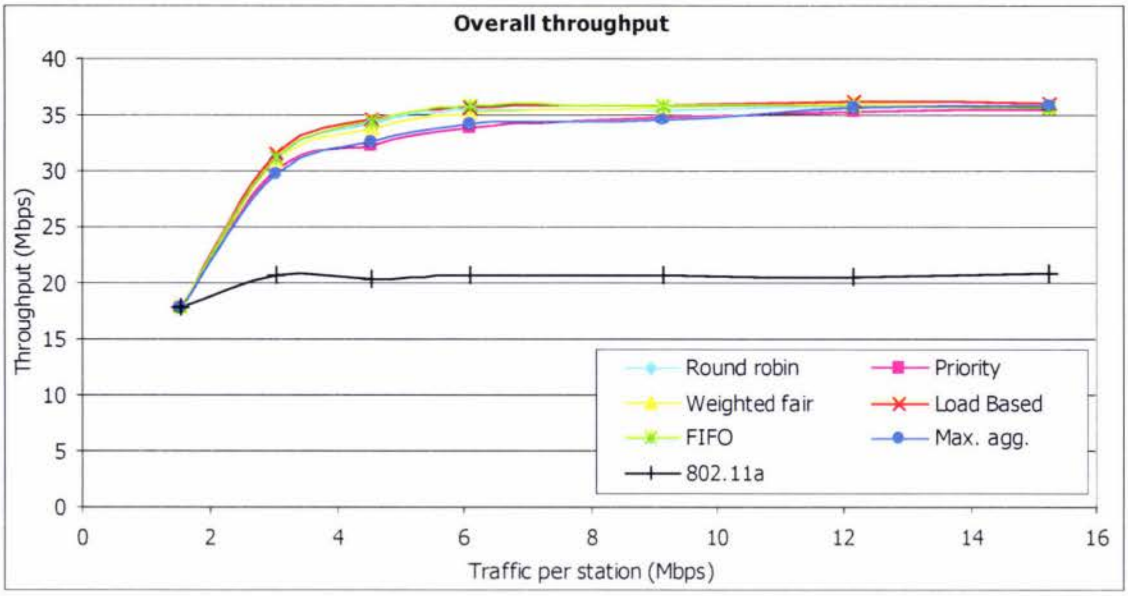


Figure 11.22: Overall throughput for queueing control simulations with 12 stations

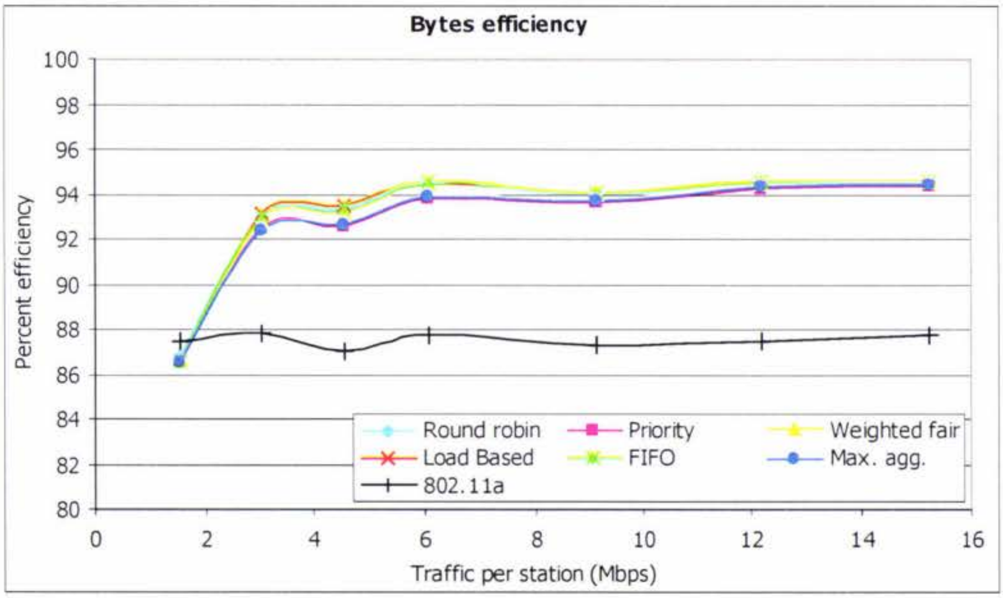


Figure 11.23: Byte efficiency for queueing control simulations with 12 stations

11.2.4 Summary

In summary, the best queueing control algorithm was the load based algorithm. It consistently provided the highest throughput and efficiency results, although not by a large margin. However, it clearly has the best queue usage performance, which is likely to be critical in many implementation environments where memory is often scarce, and thus load based should be used.

In terms of packet delay, the statistics are different depending on whether the median packet delay is used, or the average and maximum packet delays are used. In terms of median packet delay, which is typically used to assess packet delays in wireless networks as it is not affected by outliers as much, the priority algorithm is clearly the best.

However, this is because this algorithm unfairly favours certain packets and sends them with minimal delay. This can be useful in particular applications – mostly multimedia and voice applications – but in terms of general network performance it is not useful. Therefore, it would be best to have load based as the basic aggregation queueing control, with priority queueing implemented to complement the quality of service features of 802.11e.

In this way, if quality of service is not needed or activated, the load based queueing control will deliver the best performance fairly, but when multimedia applications require quality of service differentiation the priority queueing control can be used to provide low packet delay to that application. Due to the benefits of aggregation, the lower priority applications should still see better performance than with the existing 802.11 MAC sub-layer without quality of service.

11.3 AGGREGATION AND 802.11B PHYSICAL LAYER

Simulations were also carried out to find out the improvements possible while using the 802.11b physical layer. The 802.11b physical layer used much lower data rates than the 802.11a and 802.11g physical layers. The maximum data rate for 802.11b is 11 Mbps, and the calculated maximum throughput is 5.76 Mbps (see 3.1.2). These simulations used the combinations of packet arrival rates and number of stations shown in Table 11.3. Only the load based queueing control algorithm was used, as these simulations are simply to find the level of improvement in 802.11b, so we only considered the best performing queueing algorithm.

Table 11.3: Loads for 802.11b simulations, with approximate data rates in Mbps

	Number of stations transmitting			
Pkts/s	1	2	4	6
500	1.5	3.1	6.1	9.1
1000	3.1	6.1	12.2	18.3

The 500 packets per second group of simulations gave the throughput results shown in Figure 11.24. We can see that the existing 802.11 MAC sub-layer combined with the 802.11b physical layer is limited to about 4 Mbps. This is lower than the calculated 5.76 Mbps as the average packet size is much lower than 1500 bytes, and so offers a more realistic overall maximum throughput for an 802.11b wireless network. The aggregation MAC sub-layer gets a substantially higher throughput – almost 7 Mbps for the highest traffic load in this group of simulations – well above even the 5.76 Mbps seen with very large packets.

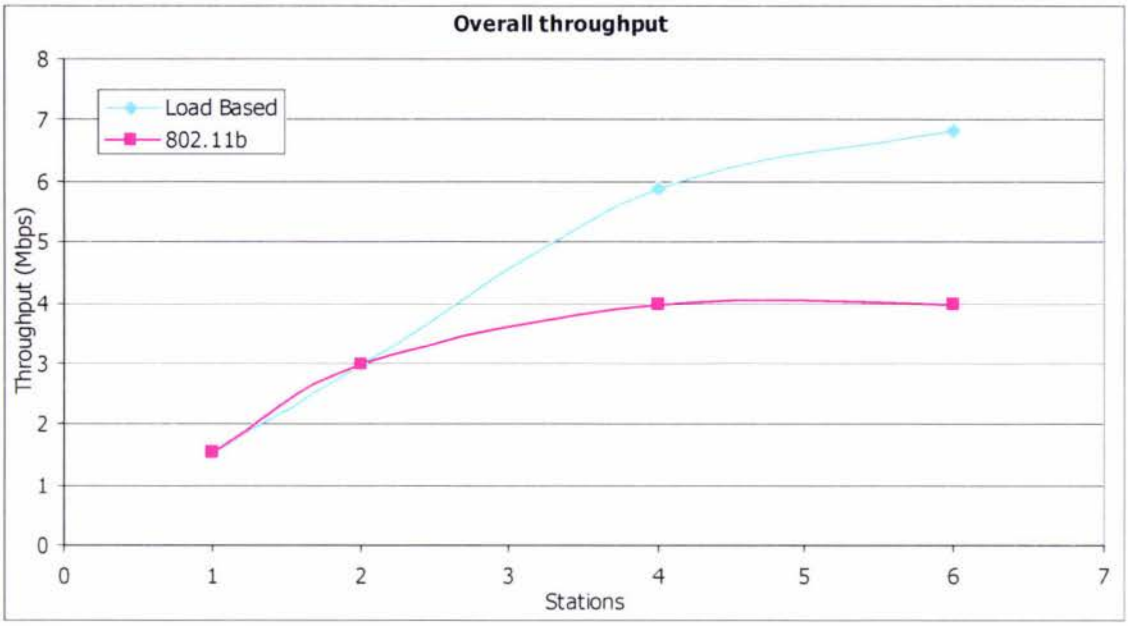


Figure 11.24: Overall throughput for 802.11b simulations with 500 packets per second per station

The throughput results for the group of simulations with 1000 packets per second are shown in Figure 11.25. Here the combined traffic load is enough to show that the throughput for the aggregation MAC sub-layer limiting to just above 7 Mbps.

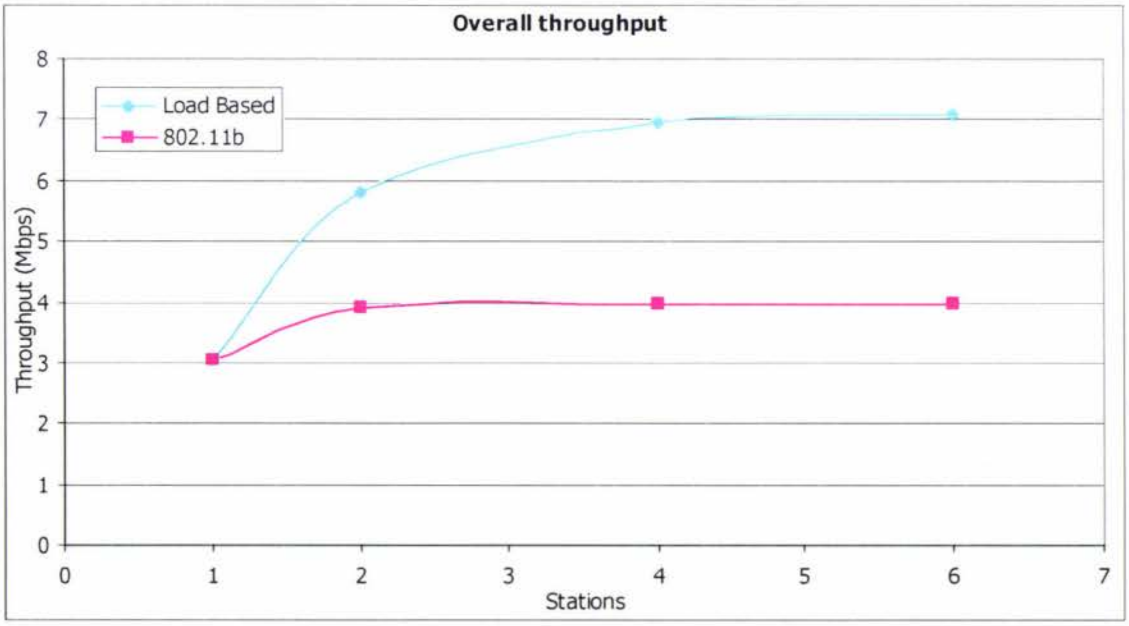


Figure 11.25: Overall throughput for 802.11b simulations with 1000 packets per second per station

11.4 LOAD BASED QUEUEING: BYTE LOAD OR PACKET LOAD

The load based queueing used so far in the simulations has been based on the number of bytes in the queues. However, the load can also be measured by the number of packets, which may possibly improve performance, as well better memory usage in implementation where the queues are limited by number of packets in the queues rather than the total size of the queues.

Figure 11.26: Overall throughput for load based simulation with 2 stations

At the lower traffic rates, there is no difference between the two types, as shown in Figure 11.26. In this simulation, the traffic loading is well within the maximum throughput for aggregation seen in the previous queueing simulations. There is also no significant difference in any of the other metrics used in the simulations.

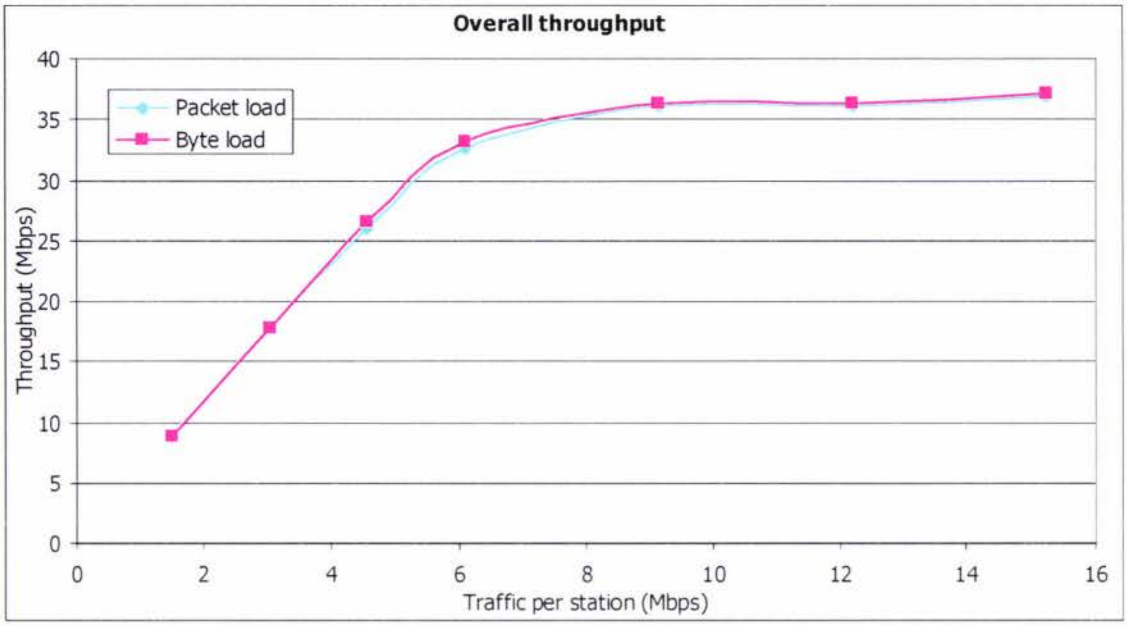


Figure 11.27: Overall throughput for load based simulations with 6 stations

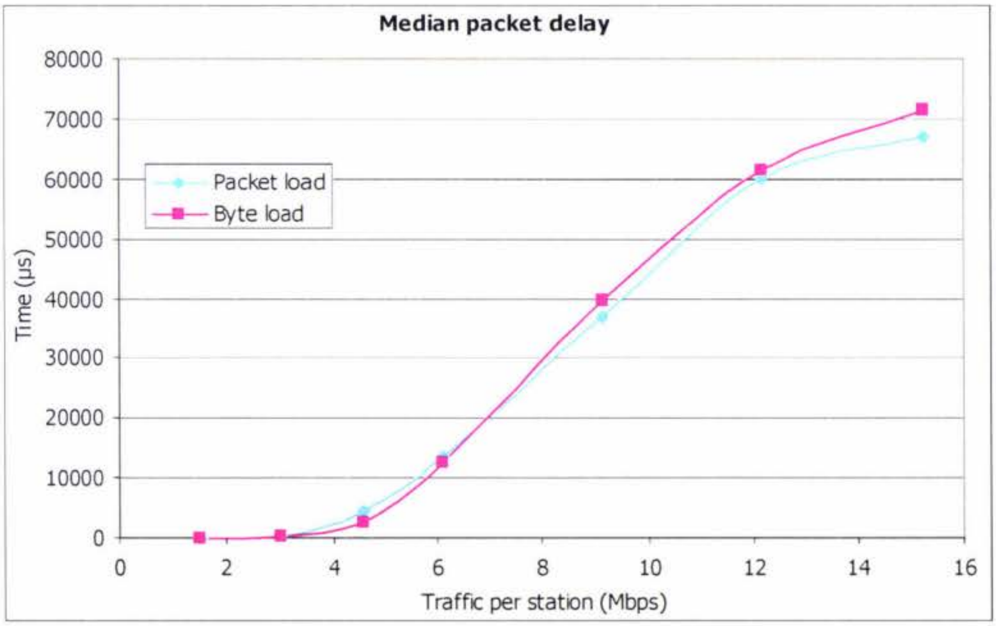


Figure 11.28: Median packet delay for load based simulations with 6 stations

When the traffic load is increased, the byte load based algorithm has a very slight advantage over the packet load algorithm (Figure 11.27). However, the median packet delay statistics (Figure 11.28) showed that packet load algorithm gave slightly lower packet delays. The difference in which algorithm is better comes down to the metrics used. Byte load based has a higher throughput as throughput is measuring the bytes transferred, and the byte load based algorithm, which over time will have less bytes waiting in the queues, will send more bytes quicker than the packet load algorithm. The same principle applies to the packet delays, where the packet load algorithm will have a lower number of packets waiting in the queues, and will send a more packets quicker than the byte load algorithm.

The difference in the queue sizes in terms of bytes and packets can be seen in Figure 11.29 and Figure 11.30 respectively, with byte load having a lower average length in terms of bytes, and packet load having a lower average length in terms of packets.

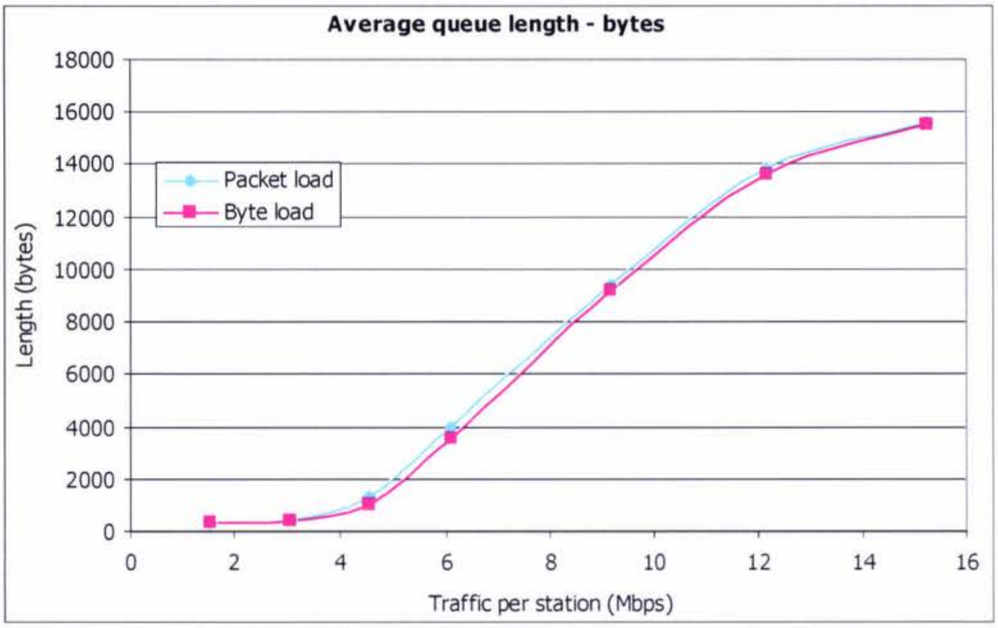


Figure 11.29: Average queue length (bytes) in load based simulation with 6 stations

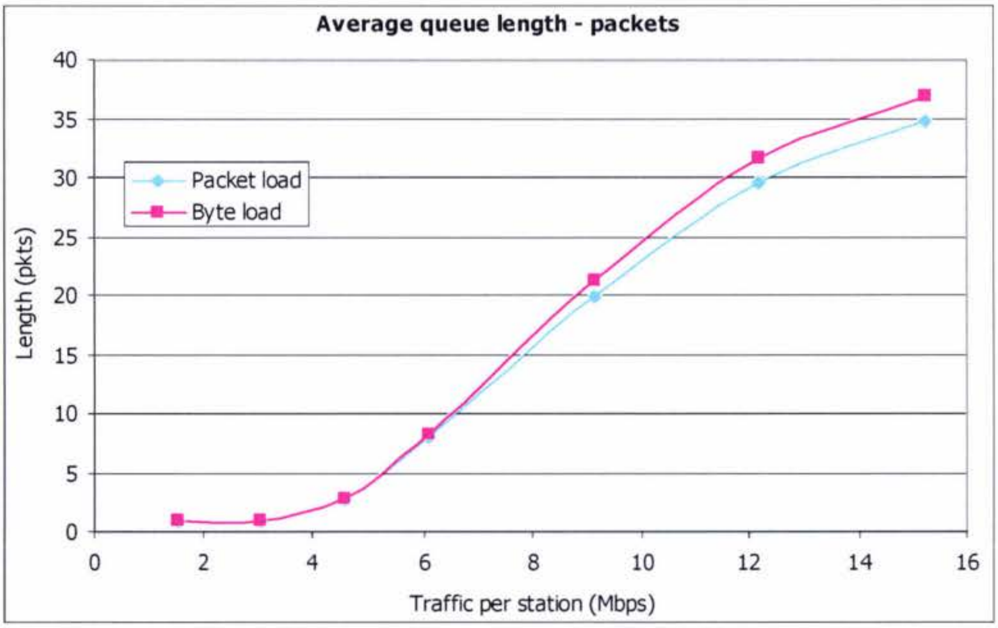


Figure 11.30: Average queue length (packets) in load based simulations with 6 stations

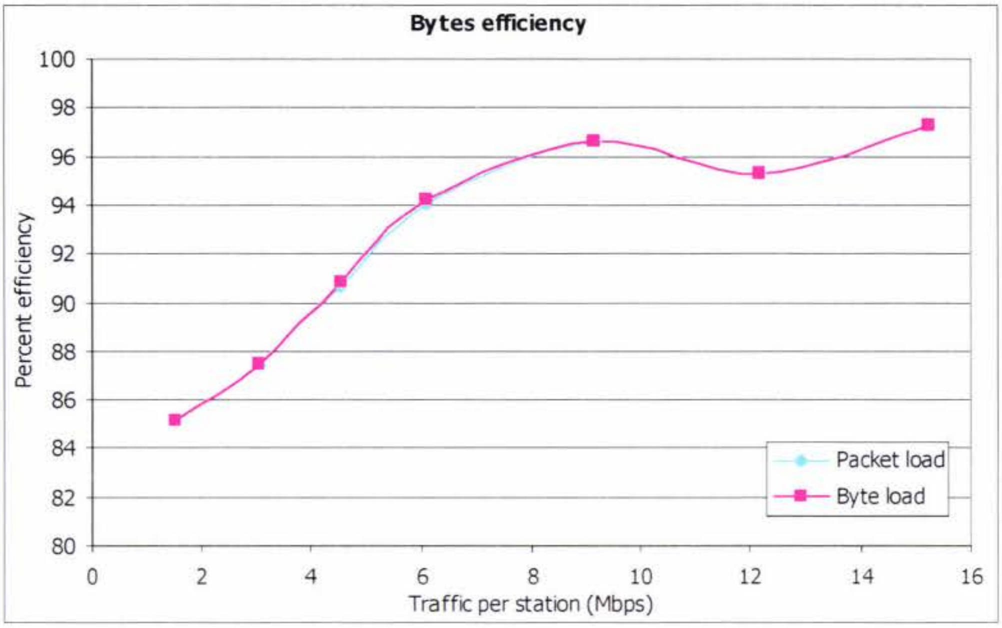


Figure 11.31: Efficiency in load based simulations with 6 stations

In terms of efficiency though, the two algorithms are almost identical (). This is an interesting result in the fact that the efficiency of the MAC sub-layer is not affected by the choice of algorithm. This result, plus the fact that the performance differences are minor, means that when aggregation is implemented using load based queueing control either a byte or packet basis may be used. The choice between them can be dictated by ease of implementation i.e. if it is easier to measure the packet load, use the packet based algorithm.

11.5 TEMPORARY QUEUEING PRIORITY

Two simulations were run for the temporary queueing priority (TQP). The first simply tested the initial idea of the TQP that gave overloaded queues temporary priority over the other queues to relieve the overload.

After this first simulation was carried out, the time based queueing priority was devised. Also, it was realised that the conditions tested in the first simulation were too general, and did not test a likely situation were TQP would be most useful. Therefore a second simulation was run with the addition of the time base priority for load based queueing, and different traffic load characteristics.

11.5.1 Load based TQP in the general case

In this set of simulations, the load based TQP outlined in 6.2.1 was implemented for all of the queueing control algorithms apart from the load based control, as this already used the queue load in its decisions.

From the medium traffic load queueing control simulations presented in 11.2.2, it was found that the results for a 4 station network gave the best overview of the performance of the queueing control algorithms. Therefore, the same group from this set of simulations will be used in this section.

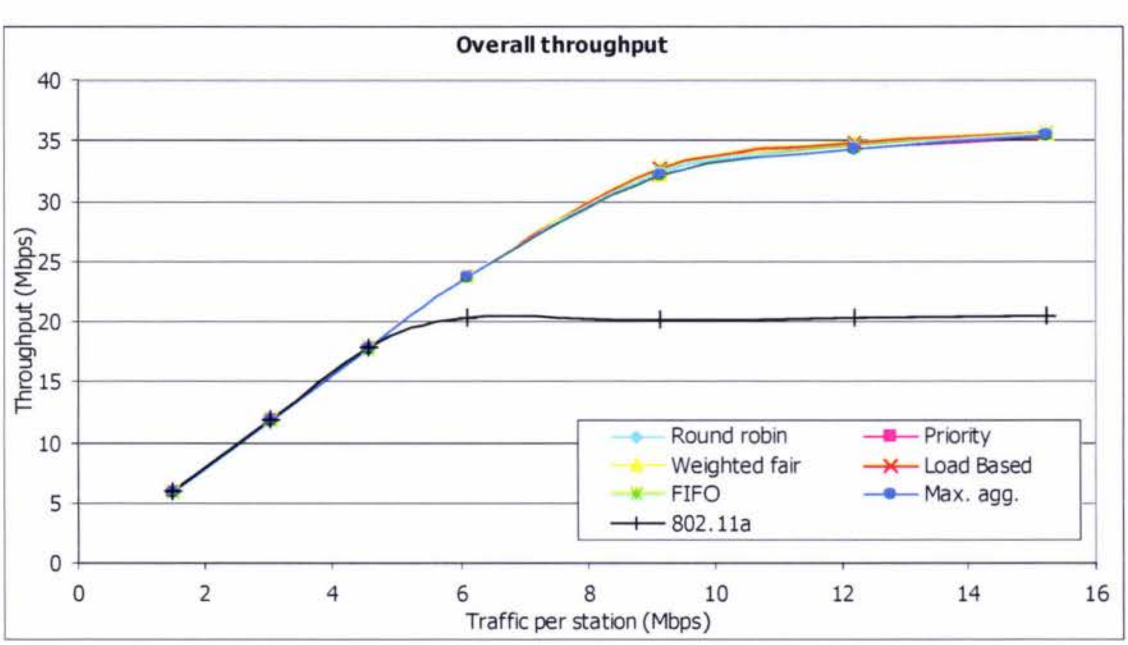


Figure 11.32: Overall throughput for first TQP simulations with 4 stations

Figure 11.32 shows the overall throughput results for the simulations. The load based queueing control and the existing 802.11a MAC sub-layer are included for comparison purposes, and are at the same level as that shown in the initial queueing control simulations. The other 5 queueing control algorithms however all have an increased throughput. In the case of the first-in and first-out and round robin algorithms, the increase is only slight, but the three remaining algorithms – priority, weighted fair and maximise aggregation – all have a significantly increased throughput. However, the load based algorithm still maintains the highest throughput, but by an even smaller margin.

The efficiency of the algorithms also improved for all queueing controls (Figure 11.33). At the higher traffic loads in the group of simulations, the efficiency was almost identical for all six queueing controls.

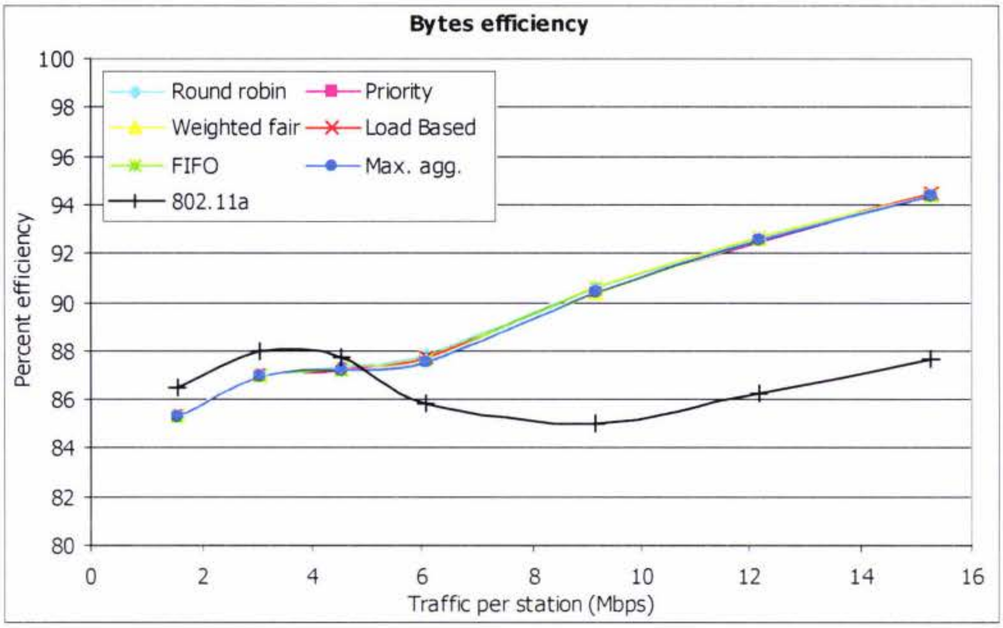


Figure 11.33: Efficiency for first TQP simulations with 4 stations

The packet delay measurements also showed that the queueing algorithms all gave very similar performance, as shown in Figure 11.34. However, the median packet delay statistics of the load based algorithm were worse than most of the other queueing controls in the simulations without temporary queueing control. Therefore, rather than TQP improving the performance of the algorithms, it actually changes there performance to be closer to the load based algorithm. This is due to the fact that the aggregation is congestion triggered. When the network is congested, the queues are more likely to fill up and trigger the TQP system which is essentially load based queueing, whereas at the lower levels, the performance differences between the algorithms are negligible.

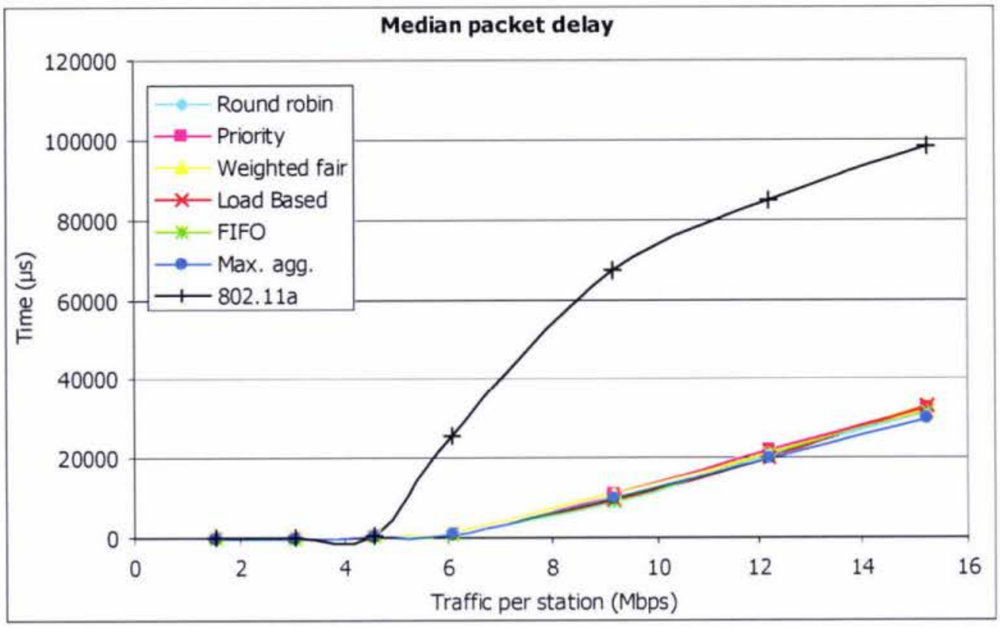


Figure 11.34: Median packet delay for first TQP simulations with 4 stations

The queue statistics, such has the maximum queue length shown in Figure 11.35 also bring the other queueing controls results closer to that of load based queueing. In comparison to the results of the simulations before the implementation of TQP (see Figure 11.21), the queueing statistics show the greatest improvement in the performance of the five queueing algorithms.

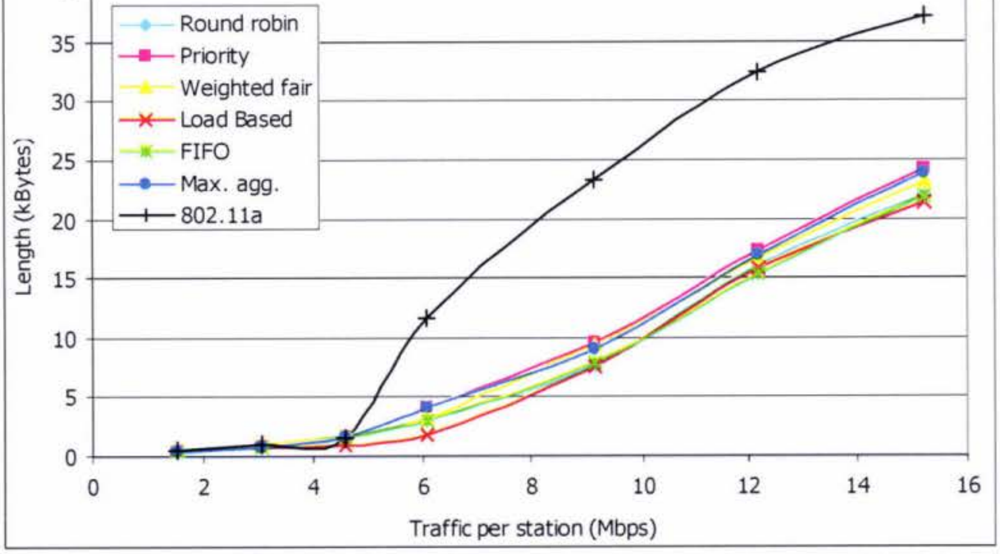


Figure 11.35: Maximum queue length for first TQP simulations with 4 stations

11.5.2 Time based TQP and specific traffic characteristics

In this second simulation of TQP one of the individual traffic loads was made to be double that of all the others. Also, the time based TQP system is implemented for the load based queueing control algorithm. This meant that one queue would be much more loaded than the others, and should therefore degrade the performance using the default queueing control algorithms, but should not degrade performance with TQP enabled.

The overall throughput for the simulations run without the TQP system with 4 stations is shown in Figure 11.36. Comparing this with the results shown for the original queueing control algorithm at the same traffic load and simulation settings (Figure 11.12), the results are very similar for the priority and maximise aggregation algorithms – better than expected for the priority algorithm as the heavier load was a given a low priority. The round robin and weighted fair algorithms remained relatively unchanged, but load based and first-in first-out increased in performance. The load based algorithm responds well to extra load, and the first-in first-out algorithm performed well as the packet arrival times over all queues was still evenly spread.

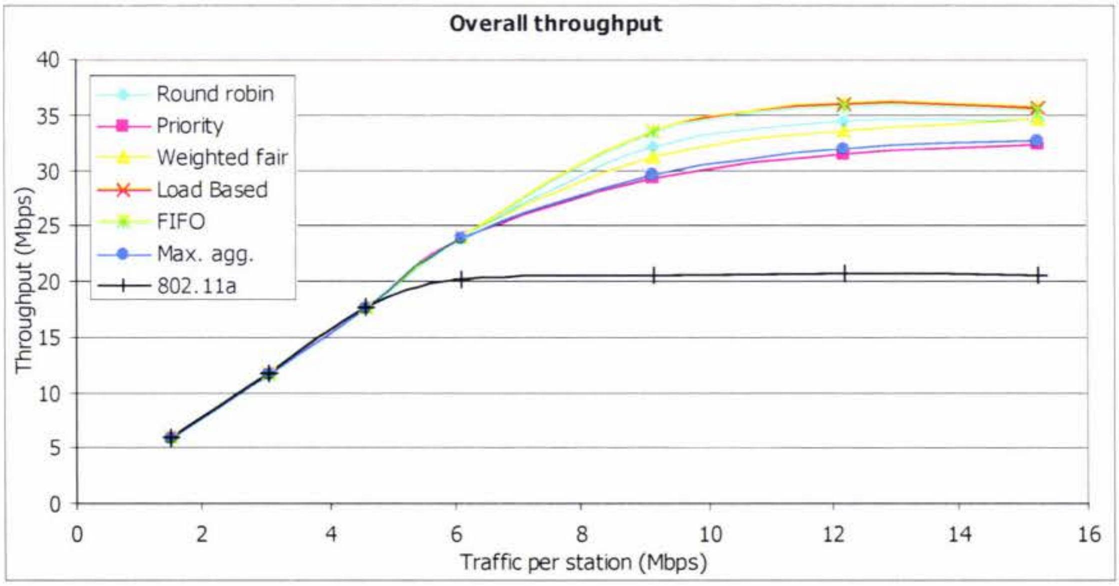


Figure 11.36: Overall throughput *without* TQP enabled, for special case traffic TQP simulations with 4 stations

Figure 11.37 shows the overall throughput for the same simulation, but with the TQP system now enabled. All of the queueing control algorithms achieve a higher throughput. The priority and maximise aggregation performance is now above 35 Mbps, an increase of about 3 Mbps. The other queueing algorithms all performed better as well, but the improvement is not as great as with priority and maximise aggregation algorithms. The order in performance remains unchanged, with load based queueing still performing best as its performance is boosted by the time based TQP.

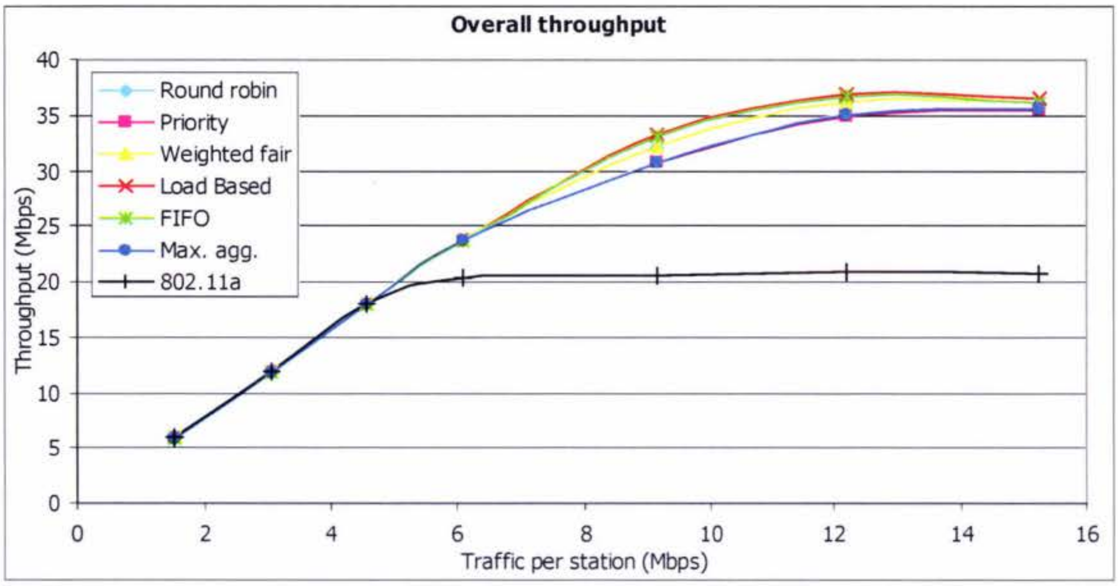


Figure 11.37: Overall throughput *with* TQP enabled, for special case traffic TQP simulations with 4 stations

The following four graphs (Figure 11.38 through Figure 11.41) show how the TQP system improves all of the algorithms average packet delay and average queue length statistics. The load based TQP significant reduces these metrics for all five of the algorithm it was used with. The time based TQP did not significantly change the average queue length for the load based algorithm, but did slightly improve the average packet delay.

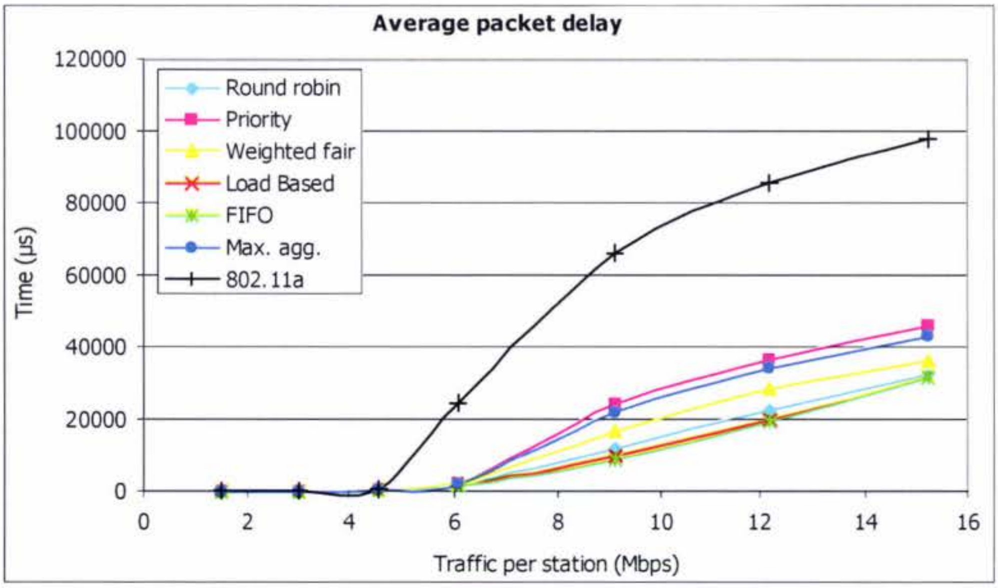


Figure 11.38: Average packet delay *without* TQP enabled, for special case traffic TQP simulations with 4 stations

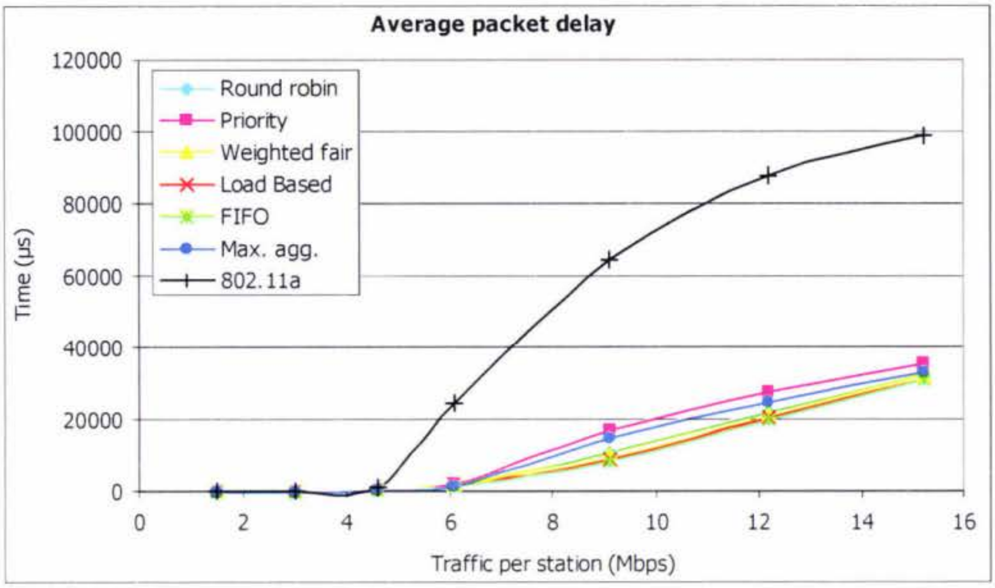


Figure 11.39: Average packet delay *with* TQP enabled, for special case traffic TQP simulations with 4 stations

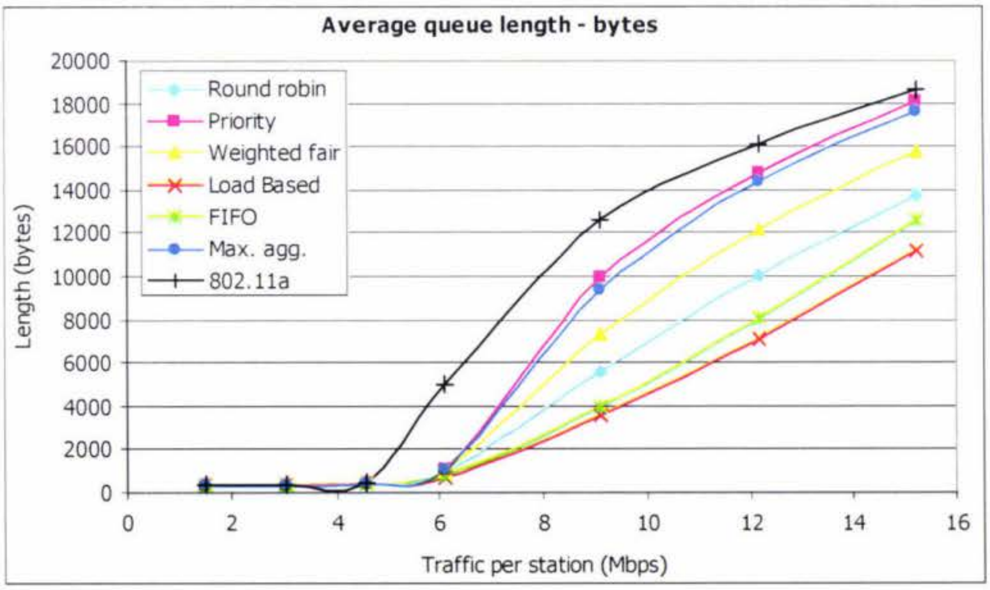


Figure 11.40: Average queue length *without* TQP enabled, for special case traffic TQP simulations with 4 stations

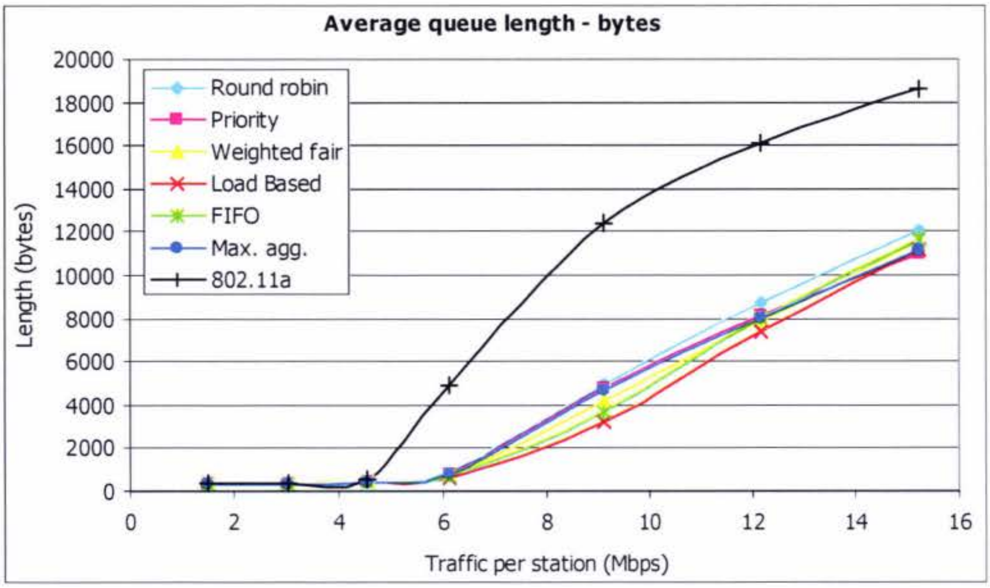


Figure 11.41: Average queue length *with* TQP enabled, for special case traffic TQP simulations with 4 stations

11.6 PAYLOAD SIZE BACKOFF

To test the payload size backoff (PSB) system described in 6.3, we implemented the error model describes in 10.3. This error model generated periods of interference and determined whether each transmission within these periods would fail or succeed. If the transmission failed, no acknowledgement packet was sent, and the station sending them would have to resend those packets.

The simulations were carried out with the same combined traffic loads as for the queueing control simulations, with the details shown in Table 11.2. The most representative results came from the groups of simulations with 2 and 4 stations.

The simulations with two stations gave the overall throughput shown in Figure 11.42. At the lower traffic loads, both with and without PSB, the overall throughput is unchanged in comparison with load based queueing without an error model (compare Figure 11.7). This is because the packets with errors would actually cause congestion which triggers the aggregation mechanism, and thus the lower throughput from the error period is offset by higher throughput in periods of no errors. The balance is enough to maintain the throughput for traffic loads of less than about 15 Mbps.

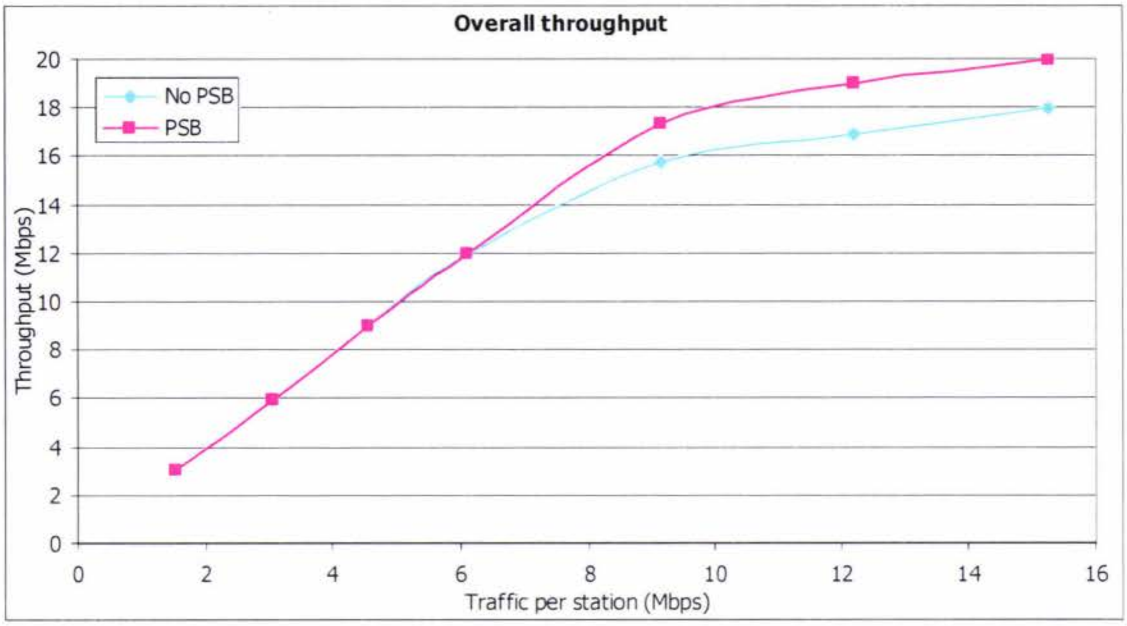


Figure 11.42: Overall throughput for PSB simulations with 2 stations

However, when the traffic load increases, the throughput drops of considerably. The load based queueing in Figure 11.7 was able to maintain about 30 Mbps of throughput at the highest traffic load in this group of simulations. With the error model enabled, the overall throughput has significantly dropped back to just 18 Mbps for the same load based queueing algorithm. The PSB enabled load based queueing achieves a higher result of 20 Mbps.

Figure 11.43 shows the percentage of transmissions that failed over the course of each simulation run. The rise in failures in the middle section is due to the higher amount of packets being sent within the error period. The percentage seems to level off at the highest traffic loads, which suggests that the number of packets sent within each pair of error and non-error periods has reached a limit.

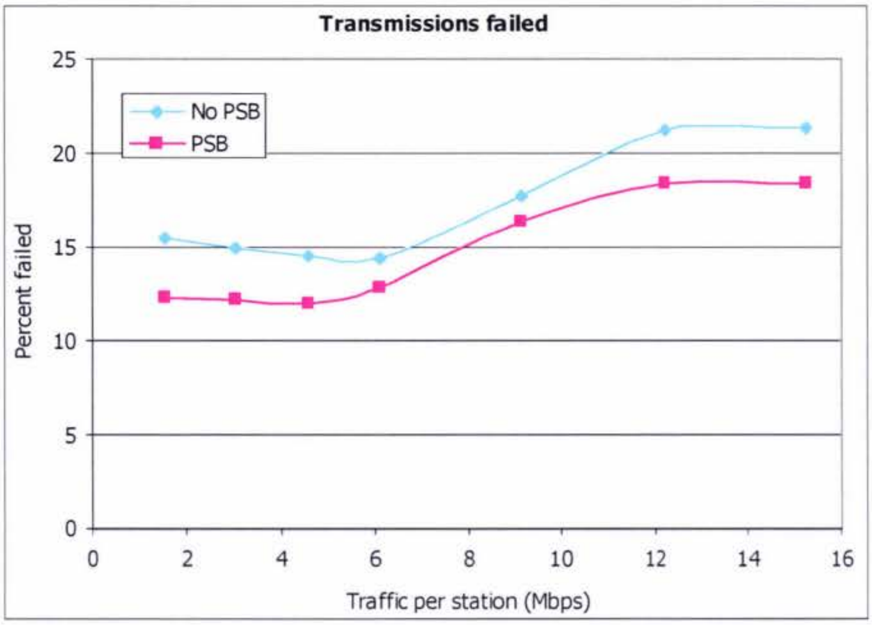


Figure 11.43: Transmissions failed in PSB simulations with 2 stations

The efficiency of the MAC sub-layer also drops considerably with the error model enabled than without the error model, as shown in Figure 11.44. The efficiency with no PSB enabled is consistently lower than when the PSB is enabled.

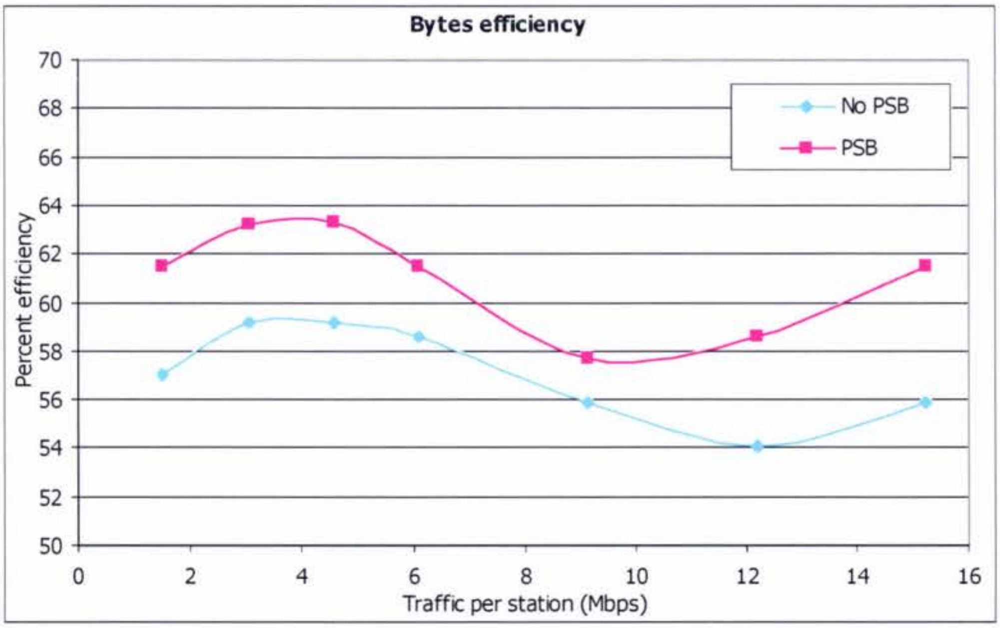


Figure 11.44: Efficiency in PSB simulations with 2 stations

With the 4 station simulations, we can clearly see that the overall throughput with the error model enabled is just under 20 Mbps with PSB enabled, and about 18 Mbps without any PSB system (Figure 11.45). This is significantly lower than the limit of about 36 Mbps seen for load based queueing without any transmission errors. This may be because of the error model was overly severe, which would mean a higher throughput in a realistic situation.

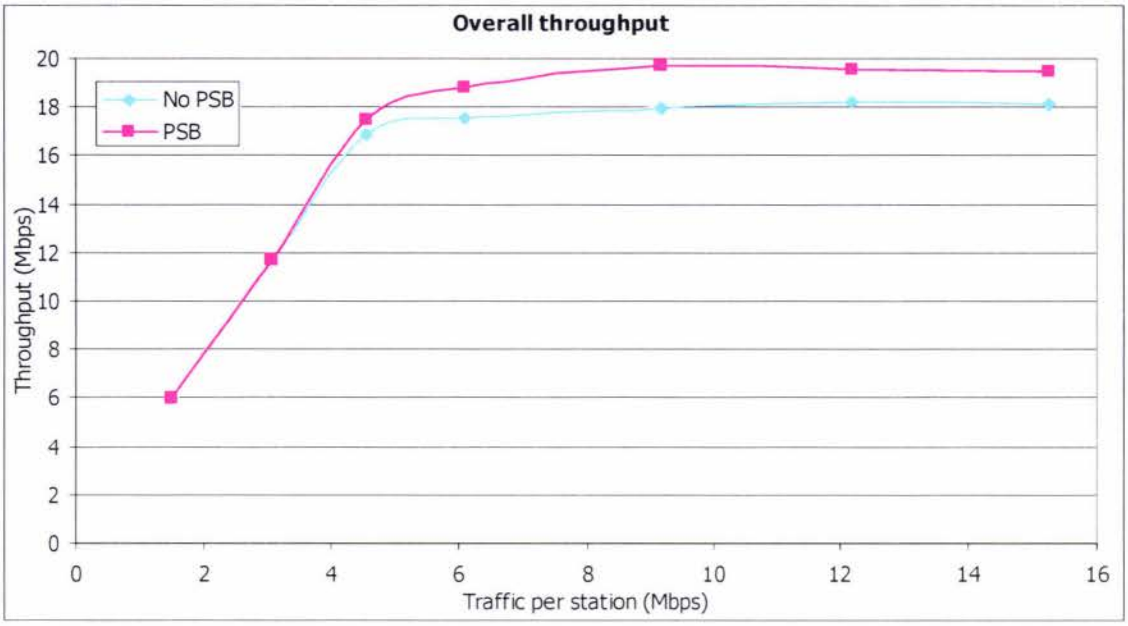


Figure 11.45: Overall throughput for PSB simulations with 4 stations

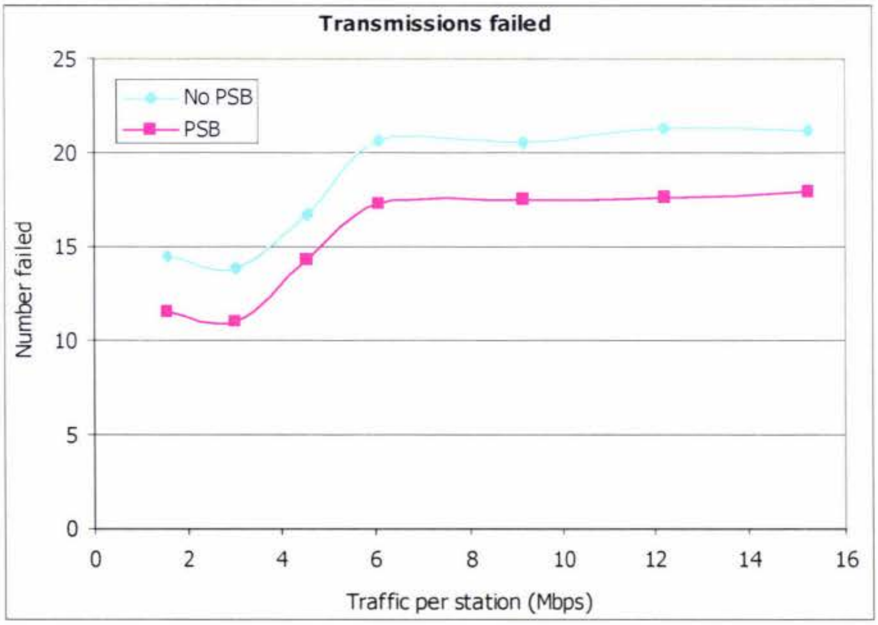


Figure 11.46: Transmissions failed in PSB simulations with 4 stations

Figure 11.46 shows the number of transmissions failed limiting in the same way as with the lower traffic loads. The throughput appears to limit at the same point, so the limit of transmissions failing is due to the fact that the number of packets sent over a given period of time becomes constant.

11.7 LARGER PAYLOAD SIZES

In the process of developing the future 802.11n standard (see 2.2.4), one of the possible changes to the MAC sub-layer is to alter the payload size. In relation to aggregation, this would mean more packets could be aggregated into each frame, which should see an increase in the throughput performance. A simulation was run with the existing usual default payload size of 1520 bytes (1500 byte packet plus 20 bytes logical link sub-layer header) as well as payload sizes of 2312 (maximum possible with the current form of 802.11) and 4096 (4 kB; one of the sizes possible for the 802.11n MAC sub-layer). Possible sizes in the final 802.11n standard will probably be different to those used here, and may be as much as 10 kB, but this simulation is interesting to show if aggregating more packets per payload will give higher throughput with current physical layer technology.

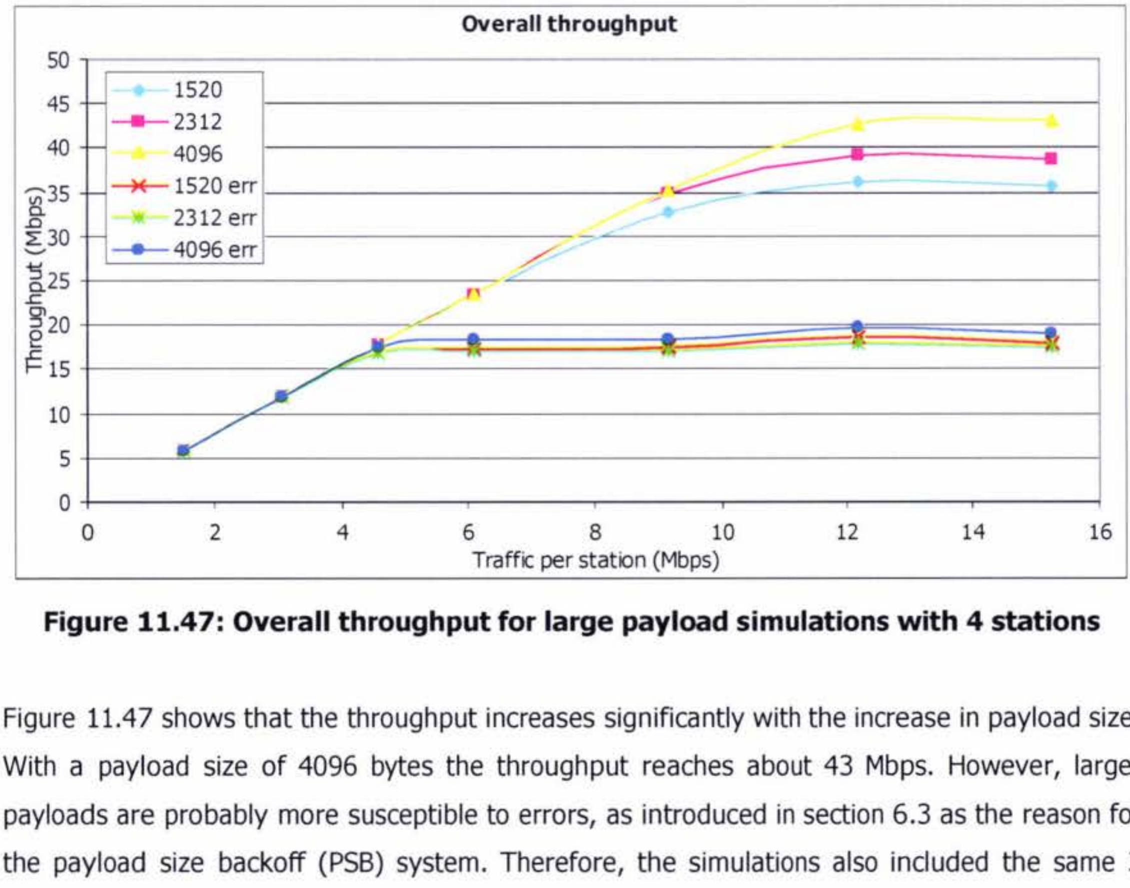


Figure 11.47: Overall throughput for large payload simulations with 4 stations

Figure 11.47 shows that the throughput increases significantly with the increase in payload size. With a payload size of 4096 bytes the throughput reaches about 43 Mbps. However, larger payloads are probably more susceptible to errors, as introduced in section 6.3 as the reason for the payload size backoff (PSB) system. Therefore, the simulations also included the same 3 payload sizes, but with the error model turned on and PSB enabled. Interestingly, the 2312 byte payload shows a decrease in the throughput, and the 4096 byte payload only shows a small increase. The similarity between the three different payload sizes is due to the fact that the amount sent in non-error periods balances out the amount of failures in the error periods – the larger payload allows more to be sent in non-error periods, but causes higher losses in periods of interference.

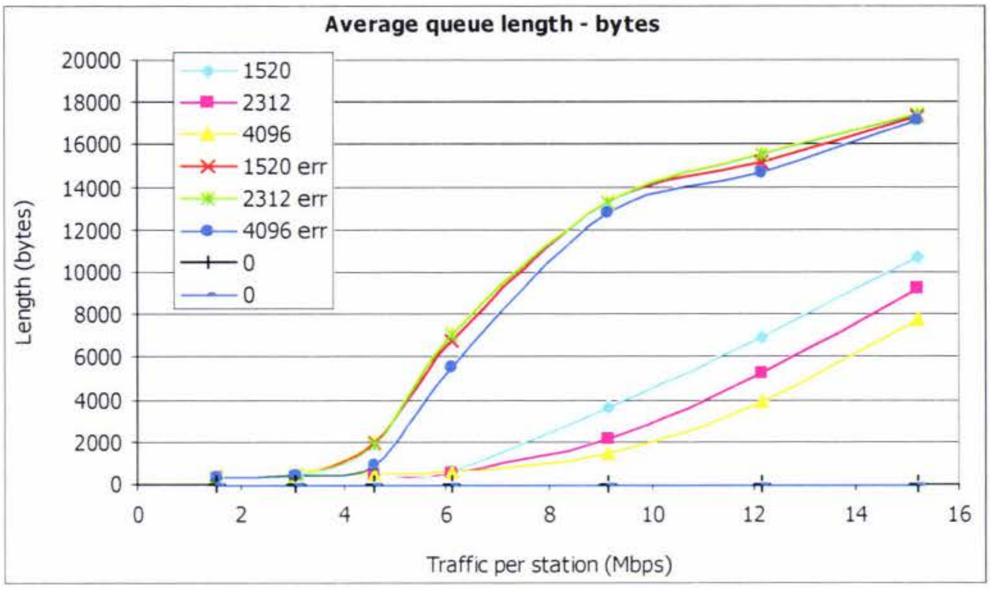


Figure 11.48: Average queue length for large payload simulations with 4 stations

Figure 11.48 shows the average queue length in bytes for these simulations. The average queue size noticeably drops with increasing payload size, as more data can be removed with every frame sent. However, with the error model included, the average queue size is very similar for the different payloads.

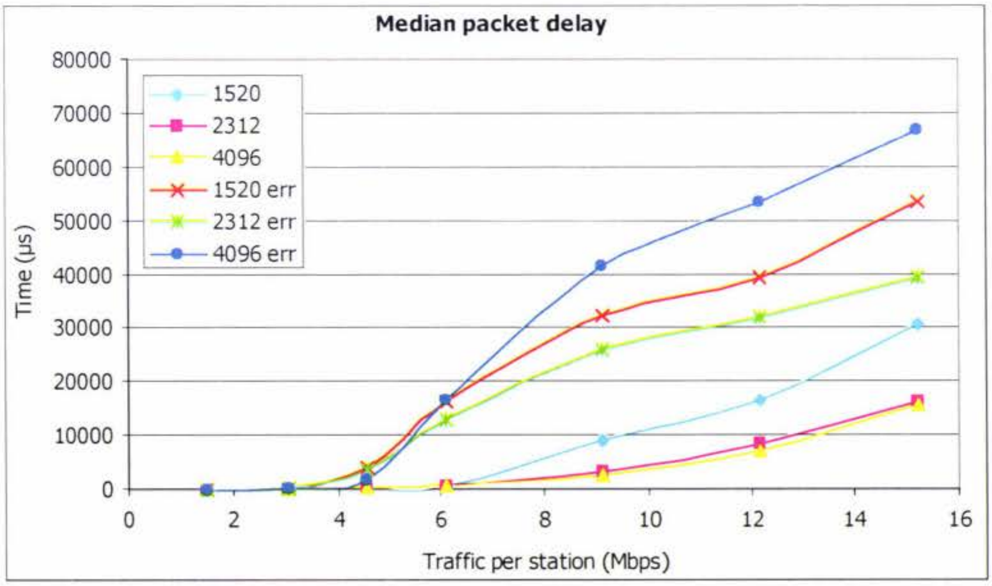


Figure 11.49: Median packet delay for large payload simulations with 4 stations

However, the median packet delay measurements are quite different (Figure 11.49). While the 2312 byte payload (without errors) significantly reduces the median packet delay compared to the 1520 byte payload, the 4096 byte payload does not reduce the medina delay very much further, unlike the improvements seen in the throughput and queue length. Also, the differences between the different payload sizes with the error model enabled may explain the similarity in the throughput. In terms of median packet delay, increasing the payload size to 4096 bytes significantly *increases* the median packet delay. So while an increased payload can deliver more data per frame, the delay on the packets within the frame may be greater, hence giving a lower throughput. The 2312 byte payload does decrease the median packet delay. This means that if the payload size is to be increased, the likelihood and level of errors must be considered, or else the throughput may be decreased instead of increased.

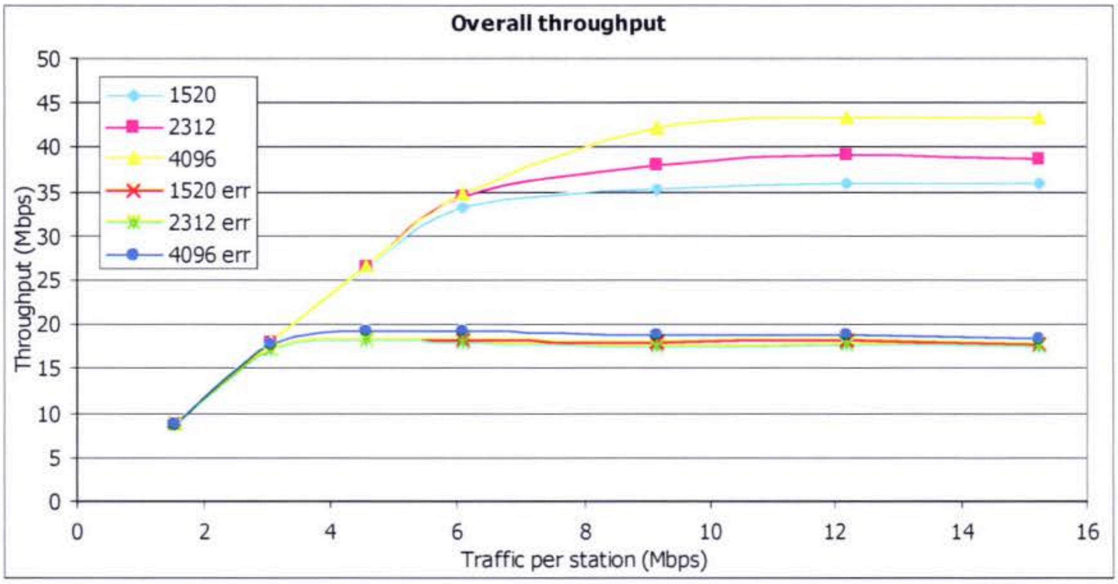


Figure 11.50: Overall throughput for large payload simulations with 6 stations

The throughput results for the 6 station simulation (Figure 11.50) shows that the levels reached in the highest combined traffic loads in the 4 station simulations are where the different payload sizes used reach their limit.

11.8 COMBINED ACKNOWLEDGEMENTS

The combined acknowledgement system is described in chapter 8. The simulations run were for the load based aggregation MAC sub-layer, and the existing 802.11 MAC sub-layer, with 802.11a used for the physical layer. The different settings for traffic loads per station and number of stations is the same as for the original queueing control simulations, with the combined traffic loads for each shown in Table 11.2.

Both the time based and count based combined acknowledgement systems were implemented. The time based acknowledgements sent an acknowledgement every 10 ms, and the count based acknowledgements sent every 10 packets.

11.8.1 Low traffic loads

At the lower traffic loads, the throughput (shown in Figure 11.51) does not change as all MAC sub-layers tested can meet the traffic demands. At the highest traffic levels, the existing 802.11a MAC sub-layer starts to limit at 20 Mbps as before. However, with the addition of just the combined acknowledgements to 802.11a significantly increases the throughput. Even at the highest level in this group, there is still headroom left in the aggregation MAC sub-layer's capabilities and no difference between the aggregation with or without combined acknowledgements.

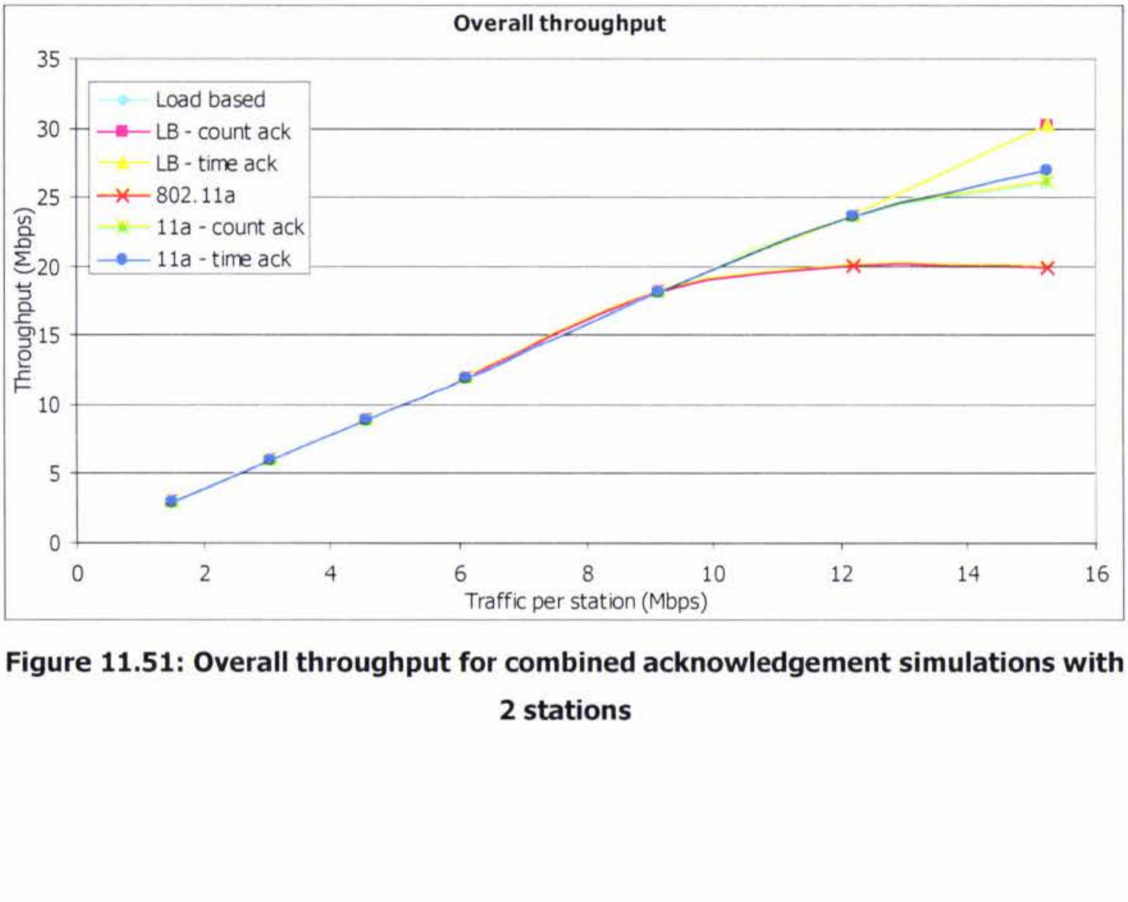


Figure 11.51: Overall throughput for combined acknowledgement simulations with 2 stations

11.8.2 Medium to high traffic loads

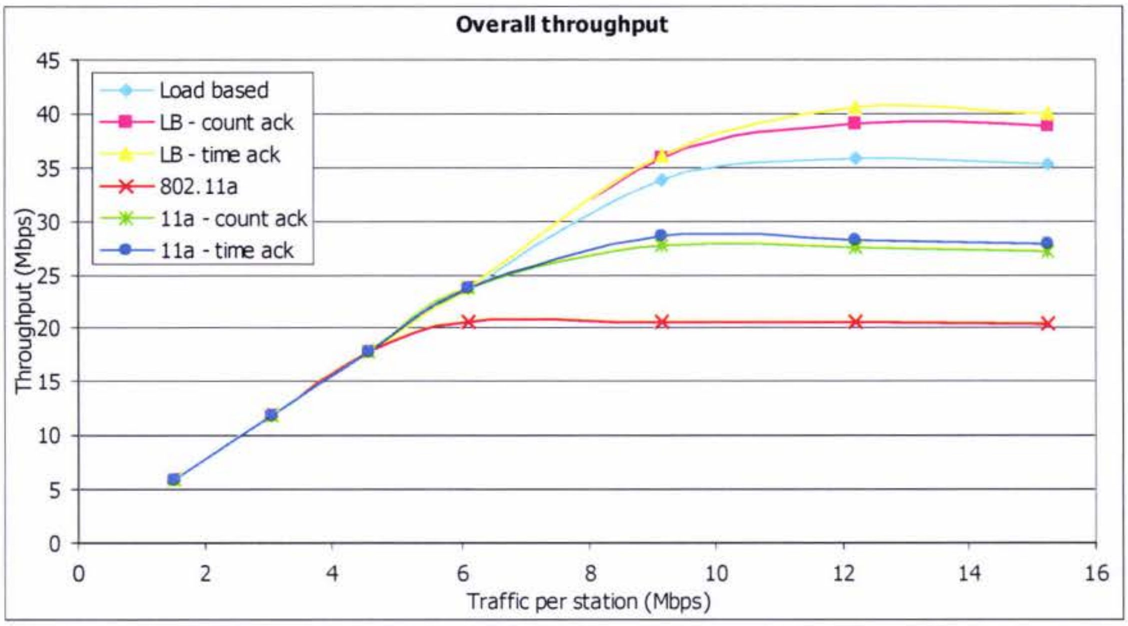


Figure 11.52: Overall throughput for combined acknowledgement simulations with 4 stations

For the group of simulations with 4 stations, the overall throughput is shown in Figure 11.52. The graph shows that the higher traffic loads in this group are enough to show the limit all MAC sub-layers tested. There are 4 main levels of performance ranging: the existing 802.11 MAC sub-layer at just over 20 Mbps; the existing MAC plus combined acknowledgements at about 28 Mbps; load based queueing aggregation MAC at just over 35 Mbps; and aggregation with combined acknowledgements giving the highest performance of about 40 Mbps. The combination of the aggregation and combined acknowledgements improvements has almost *doubled* the throughput compared with the existing 802.11 MAC sub-layer.

In terms of assessing the acknowledgement systems, the time based acknowledgement system has a higher throughput. However, this is probably because more than 10 frames can be sent every 10 milliseconds. In fact, looking back at the calculations for the maximum throughput, Table 3.5 gives the transaction time for a 1500 byte packet as 428 μs. This would mean at least 23 frames could be sent in 10 milliseconds. Therefore a better system would be to use a joint system of either 10 packets or 5 milliseconds, which although will not give as high a performance increase as the 10 milliseconds combined acknowledgements, will not cause as large delays due to errors and interference.

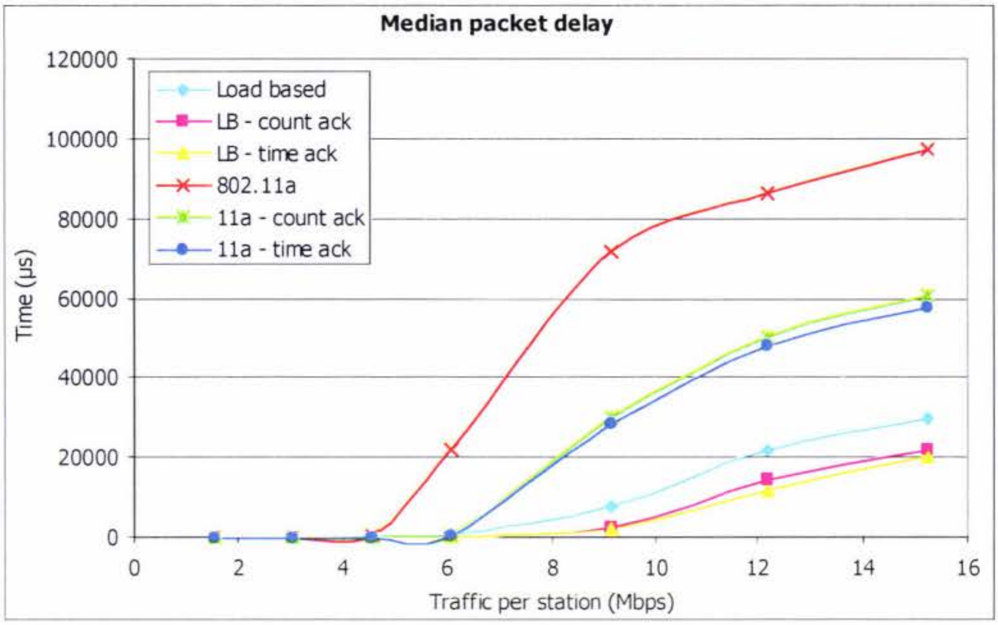


Figure 11.53: Median packet delay for combined acknowledgement simulations with 4 stations

Figure 11.53 shows the packet delays are also improved dramatically by adding combined acknowledgements. This is because rather than all of the stations waiting until the acknowledgement has been sent, once the data transmission has finished, they will then start contesting for the medium, thus removing a lot of delay. When a combined acknowledgement will be sent, it will after a SIFS, which will be before any other station transmits as there must be a DIFS gap period. Figure 11.54 shows that the same pattern is repeated in the queue metrics.

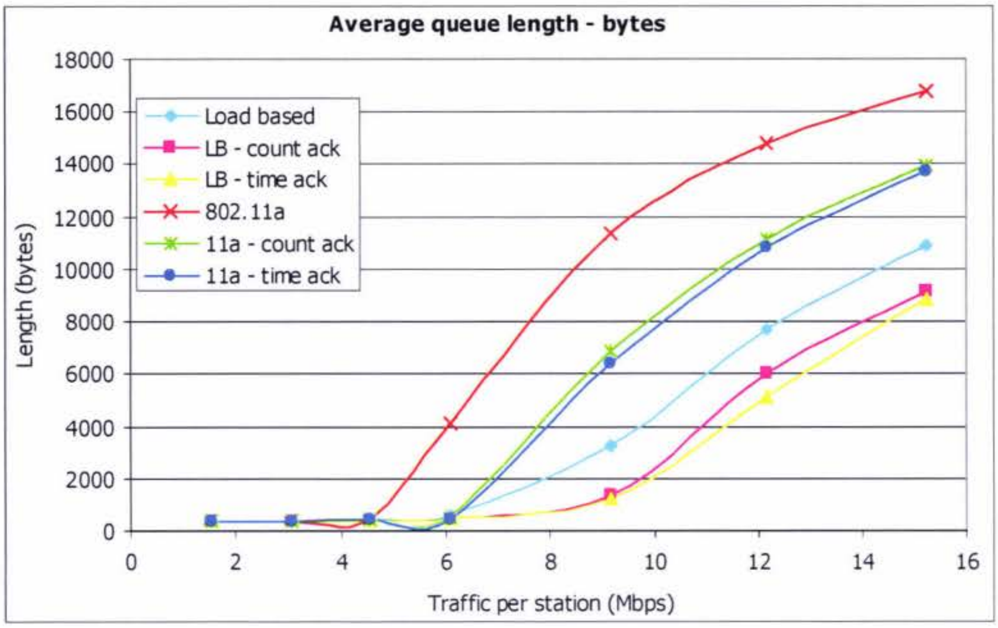


Figure 11.54: Average queue length for combined acknowledgement simulations with 4 stations

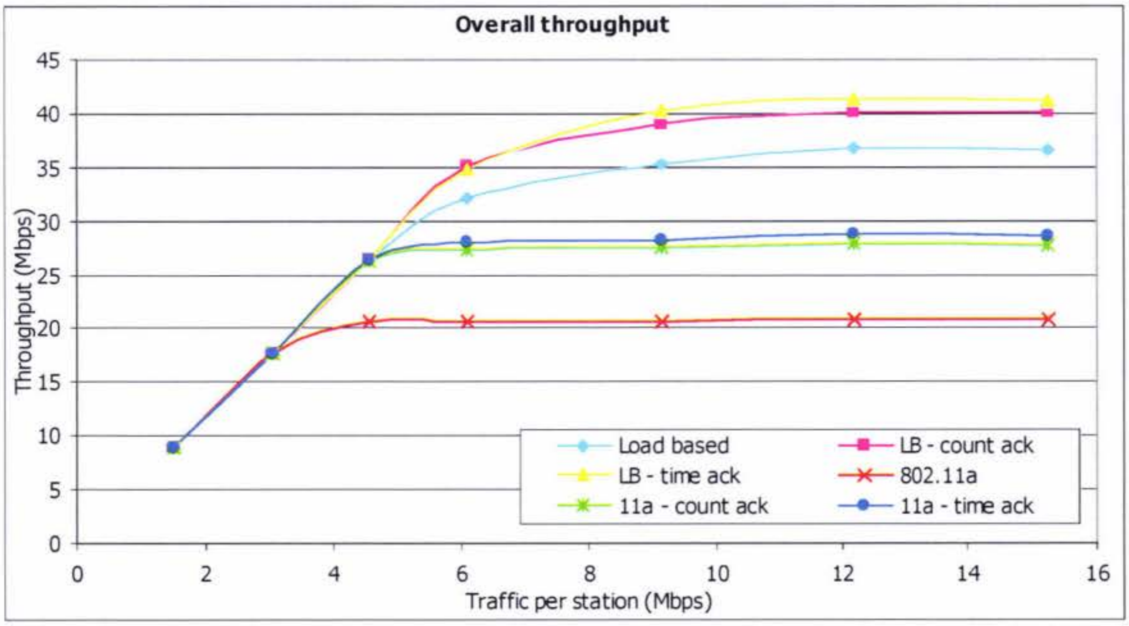


Figure 11.55: Overall throughput for combined acknowledgement simulations with 6 stations

Lastly, Figure 11.55 shows that the limits found in the 4 station simulations are indeed the limits, as even higher traffic loads do not give any significant increase in the overall throughput.

11.8.3 Summary

The combined acknowledgement system increases the throughput of the existing MAC sub-layer significantly. The existing MAC sub-layer has a maximum overall throughput of just over 20 Mbps with the packet distribution used in the simulations. The combined acknowledgement system, based on a combined acknowledgement for every 10 packets sent gives an increased throughput of over 27 Mbps.

Additionally, the combined acknowledgement system can be used in conjunction with the aggregation mechanism also proposed in this thesis. The cooperation of combined acknowledgements and packet aggregation using load based queueing (with look-ahead queueing) gave an overall throughput of just over 40 Mbps in these simulations. This represents an improvement of well over 90% over the existing 802.11 MAC sub-layer.

11.9 STORED FRAME HEADERS

The stored frame header system described in chapter 9 needs a control system for full operation where decisions have to be made on which headers to save at what time. Within the timescale of this thesis it was not possible to design this control system.

However, it was possible to test the algorithm for certain types of wireless network. For this simulation, we only used 4 stations, and assumed that 16 headers could be stored. In this way, all possible headers could be stored at each station. The overall throughput from this simulation is shown below in Figure 11.56.

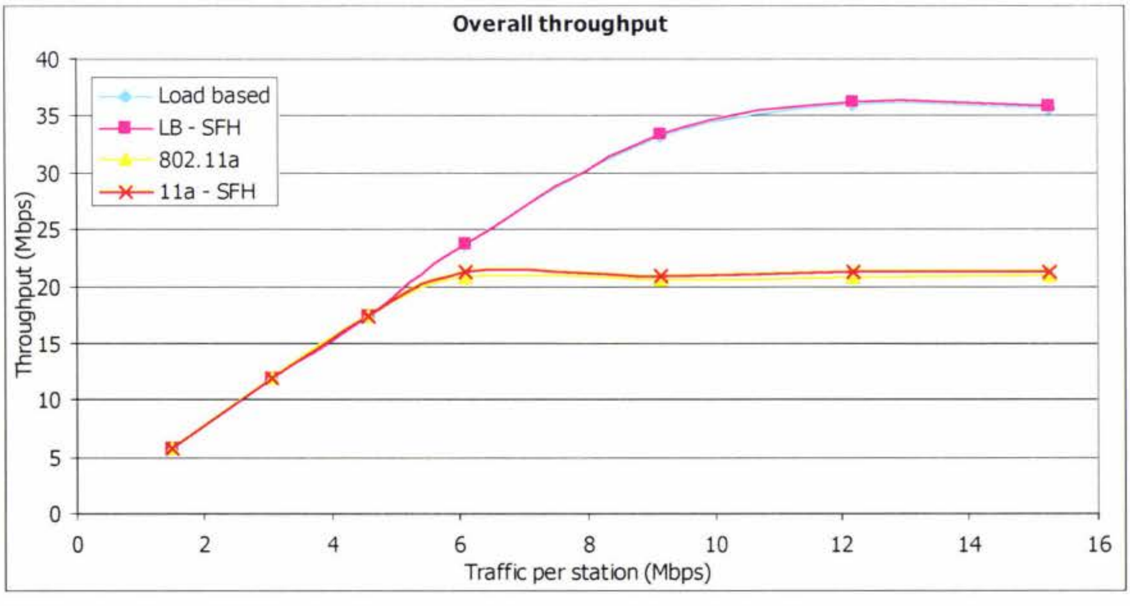


Figure 11.56: Overall throughput for stored frame header simulation with 4 stations

The stored frame header slightly increases the throughput marginally for both the existing and the aggregation MAC sub-layers. In the case of the existing MAC sub-layer, this improvement is about 1 Mbps for the higher traffic loads.

Figure 11.57 below shows that while the throughput increase is small, the increase ion the byte-wise efficiency is actually quite significant. This increase is due to the decrease of overhead data sent across the physical layer.

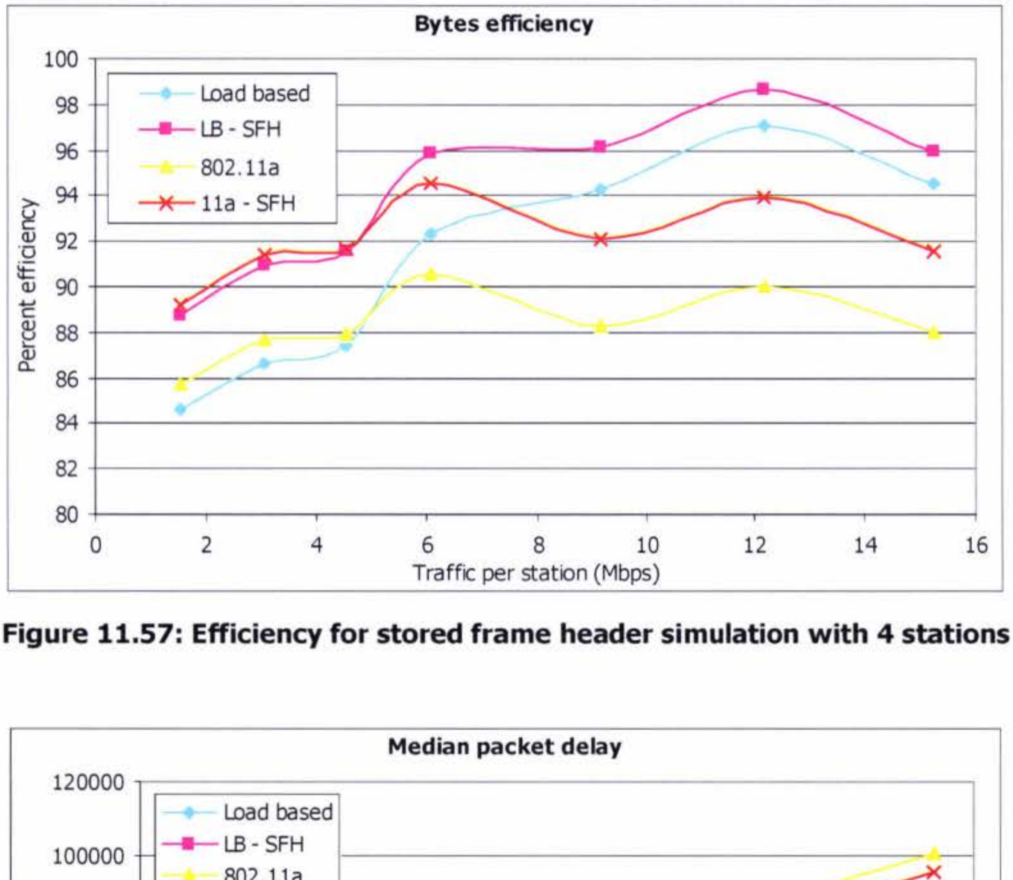


Figure 11.57: Efficiency for stored frame header simulation with 4 stations

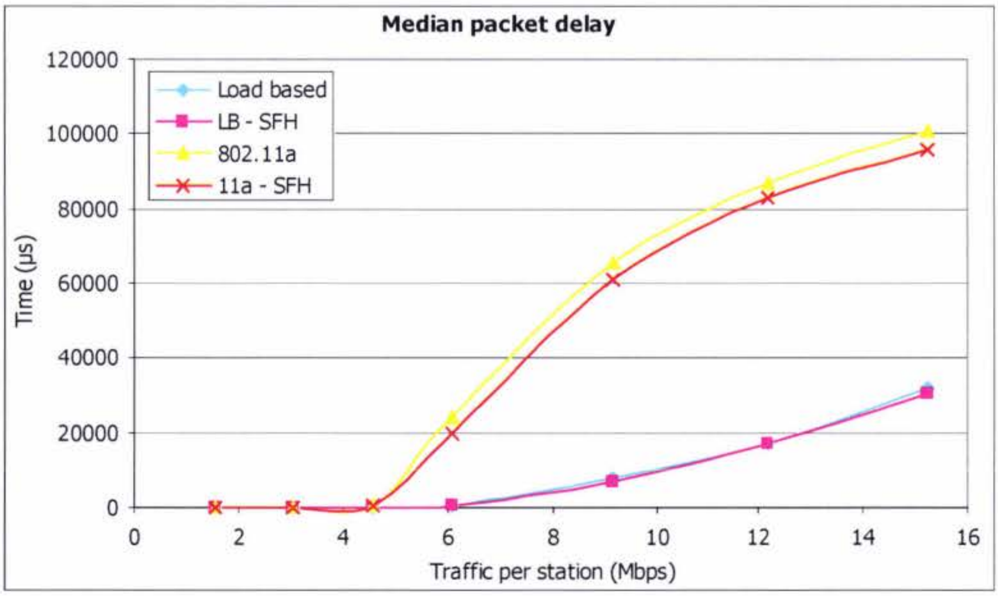


Figure 11.58: Median packet delay for stored frame header simulation for 4 stations

Figure 11.58 shows that a slight reduction in the median packet delay is achieved by the stored frame header system for the existing 802.11 MAC sub-layer. The aggregation MAC does not benefit in terms of packet delay from stored frame headers.

However, the stored frame headers scheme should give a substantial throughput increase if a slower transmission rate were in fact used by a particular 802.11 wireless network device.

12 CONCLUSIONS

The research presented by this thesis was initially undertaken with the aim of significantly increasing the throughput of wireless networks through the improvement of the 802.11 MAC sub-layer. So far in the history of 802.11 wireless networks, there has been no standardised improvement in terms of throughput for the 802.11 MAC sub-layer.

This objective was achieved by the addition to the MAC sub-layer of three separate improvements. Stored frame headers, only achieved a small increase in the throughput in the general case, but given the right circumstances it may offer much more. Also, for applications such as fixed backhaul links, this scheme is very easy to implement, and although probably achieving only a small throughput increase, the improvement is essentially free.

The major two improvements were packet aggregation and combined acknowledgements. Both of these ideas have been put forward as possibilities in the last few years, but only with limited availability or functionality. Therefore the main aim of this research was to develop these systems into more efficient and effective techniques that are obtainable for any form of 802.11 wireless networks.

Packet aggregation includes several packets into the payload of each data frame. This spreads the overheads of the 802.11 wireless network MAC sub-layer and physical layer over several packets, thus allowing a substantial increase in throughput. The aggregation is congestion triggered as first introduced by Jain et al [27], so in periods of low traffic, no extra delay or latency is incurred for the sake of "improving the number of packets aggregated". In times of congestion, aggregation offers a *much* larger throughput than any standard existing 802.11 system.

This research also looked extensively into the fine details of packet aggregation. This included finding the most efficient method of choosing the queues that packets were stored in before transmission, which was found to be a load based algorithm (using either byte load or packet load). This was followed by looking at how the queues were accessed by the aggregator when it placed packets within the frame payload. Also, a temporary priority system for the queues was developed to improve the efficiency of the queueing algorithm. This system reduced the susceptibility of the queueing algorithms to traffic fluctuations, and in certain cases will increase throughput.

Lastly, as frames with larger payloads are likely to be more susceptible to errors, a payload size backoff scheme was designed to reduce the maximum size of the payload in periods of interference. It was also found that increasing the maximum payload size beyond the typical size in use currently can give a marked increase in throughput, but with a certain level of interference will not increase throughput at all.

Packet aggregation, with the realistic packet size distribution used in the presented simulations, increased the throughput of 802.11a from just over 20 Mbps to over 35 Mbps. The throughput for 802.11b with the same packet distribution was increased from 4 Mbps to just over 7 Mbps.

Combined acknowledgements are an improvement on the block acknowledgement system described in the 802.11e standard. This system had its merits and successfully decreased the overheads of the 802.11 acknowledgement system, but was optional and only available in a certain mode of 802.11e. The thesis presented a system that extended 802.11e's system in a way that provided a significant improvement to the throughput of 802.11, and could be used in any operating mode of 802.11 networks.

The combined acknowledgement system was found to increase the throughput of 802.11a from 20 Mbps up to about 27 Mbps with a realistic number of packets acknowledgements per combined acknowledgement frame.

Perhaps most significantly though is that the packet aggregation and combined acknowledgements schemes can be combined. In simulations, this increased the throughput from just over 20 Mbps for 802.11a and the existing MAC sub-layer, up to 40 Mbps for 802.11 with aggregation and combined acknowledgements, as shown in Figure 12.1 below.

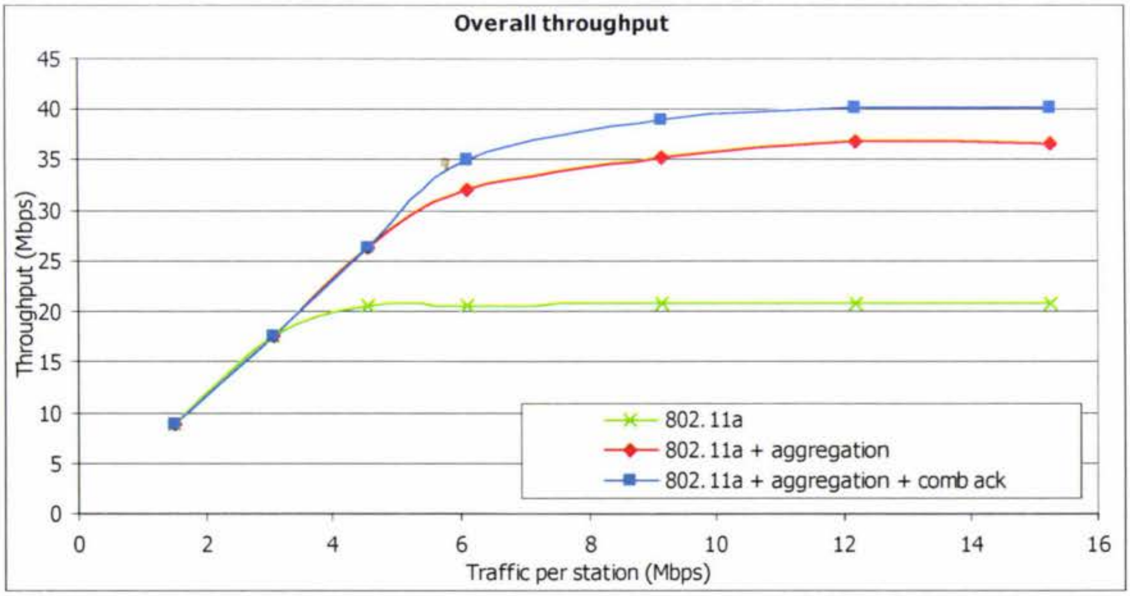


Figure 12.1: Throughput with aggregation and combined acknowledgements

13 FUTURE WORK

The main content of this final chapter outlines the issues discovered during the course of the research in regards to the implementation of an aggregation mechanism on a personal computer platform. These details are situated within the future work section as they do not apply directly to the direct findings of the main research, and are of great importance to anyone reading this thesis that wishes to use the findings contained within as part of ongoing research or actual implementation.

Also contained in this section are possible areas where aggregation could be extended or developed further.

13.1 IMPLEMENTATION OF AGGREGATION

The real test for aggregation as a performance improvement for 802.11 wireless networks is to implement in the real-world. This would first take the form of a proof-of-concept implementation to show that aggregation can be implemented using existing wireless networking hardware – irrespective of the specific physical layer. Then, ideally, it would be used to show that aggregation can significantly improve the throughput.

From the details presented in 7.2, the aggregation mechanism is positioned near the "top" of the MAC sub-layer. From a more technical viewpoint, aggregation operates upon the packets from the layer above it during the process of creating the frame. This is the first major operation within the MAC sub-layer for a packet being sent across a wireless network. Only after a frame has been constructed do procedures such as encryption and station coordination and synchronisation take place.

Typically, the operations within the MAC sub-layer specification are situated in both software and hardware. In general, the packets are processed within the software driver, as well as the management of the MAC sub-layer. Then the rest of the operations such as encryption, protocol control, joining/leaving the network and transmission/reception control are performed by firmware located on the wireless network card.

Therefore, aggregation will almost certainly be implemented within software, as part of the upper MAC sub-layer operations. This is certainly the case for implementation on a computer platform. Although the case may be different for solutions that use more embedded technology or applications where no direct user input is present, the research presented within this thesis is primarily targeted towards the user-focused, multimedia applications that are typically deployed on personal computers.

There are two possibilities for the implementation of the aggregation mechanism. One possibility is to create a driver that sits above the MAC sub-layer, instead of within the MAC sub-layer. This driver would work by queueing the packets dispatched to the MAC sub-layer from higher layers and aggregating packets into a single larger packet which is then passed to the MAC sub-layer. The MAC sub-layer would treat it as a single packet, without knowledge of the aggregation. Although this is outside the scope of the 802.11 standard, it should be able to operate using any existing wireless network device, independent of their operation.

The second possibility is to modify existing wireless network card drivers to add aggregation functionality. These drivers typically form a large part of the MAC sub-layer implementation, and therefore the aggregation mechanism can be placed in the correct position within the MAC sub-layer. However, it is likely that it would have to be re-engineered to move this implementation between different vendors wireless network devices.

13.1.1 Pre-MAC Sub-layer Aggregation

A pre-MAC sub-layer driver would be very portable between different wireless networking devices, but would sacrifice some of the functionality of aggregation, as direct access into the MAC sub-layer cannot easily be included into software located outside the network card driver.

The main advantage of using an additional driver is the portability to all wireless cards, as it will be portable across all the 802.11 physical layer types. Because most wireless card drivers are closed-source (all drivers under Windows environments are, and several under Unix/Linux (inc. Mac OS) are), a pre-MAC driver should be able to be implemented on any operating system. A Linux pre-MAC driver would be the easiest due to the entirely open source kernel – and well designed network 'hooks' giving access to packets (initially designed for firewalls). A Windows pre-MAC driver while still possible would be more difficult to implement due to the closed nature of the Windows network architecture.

An unusual advantage of the pre-MAC aggregation implementation is that several packets will be acknowledged at once. As the receiving MAC is unaware that a Data frame's payload may contain multiple packets, only one acknowledgement will be sent back to the transmitter. This is essentially a form of combined or block acknowledgement.

A pre-MAC driver will not be able to extend to deliver QoS differentiation. As the 802.11e QoS mechanisms are within the MAC sub-layer, a pre-MAC sub-layer driver will not have access to them, and would aggregate packets from different QoS categories.

A pre-MAC sub-layer driver will also need to have its own selective repeat scheme built into it. This would be used if an individual packet that was aggregated had an error, and needed to be resent. Although this in itself should not pose a problem, it practically adds another "layer" to the network stack. This may cause unforeseen overheads which will counter the effect of aggregation in reducing MAC sub-layer overhead percentage.

13.1.2 Aggregation within the MAC sub-layer

Aggregation within the MAC sub-layer (modifying an existing MAC driver) would require some effort to port to other drivers, and in some cases this may be impossible if the wireless card drivers are closed-source. However, it may allow the use of other MAC sub-layer services, and also the inclusion of algorithms that enhance the aggregation operation.

There is a steadily growing list of network cards that have open source Linux drivers – which would make the implementation of aggregation within the MAC sub-layer more likely. Several projects are in progress that have either achieved, or are on the way to, successful wireless networking with different wireless chipsets, e.g. those based on the Prism chipsets [30] and Atheros chipsets [31].

An almost complete list of wireless network cards with publicly available Linux drivers is updated regularly, as part of an open source project [32]. This project was started in 1996, and is funded by Hewlett Packard.

In conclusion, the argument for and against a pre-MAC sub-layer driver amounts to one of portability against advanced MAC sub-layer functionality. A pre-MAC sub-layer driver would be very portable, but would be able to access services built into the MAC sub-layer. Aggregation within the MAC sub-layer (modifying an existing MAC driver) would need extra work to port it for other devices, and in many cases impossible, as many wireless card drivers are closed-source. However, it would allow the use of MAC sub-layer services, and possibly the introduction of other enhancements that aid in the aggregation operation.

13.1.3 Implementation in Windows

An aggregation mechanism implemented as a pre-MAC under the Windows operating systems driver would have to interact with the Windows NDIS (Network Driver Interface System) system as an 'Intermediate driver' (IM). Figure 13.1 shows the locations of the miniports (or miniport drivers) where each different MAC sub-layer is implemented (e.g. Ethernet MAC, 802.11 MAC and so on). An IM driver sits immediately above the miniport drivers, with the ability to offer extra features for particular networking hardware.

The architecture NDIS system is actually well laid out for the inclusion of a pre-MAC driver. However, it appears that the interfaces that are used in NDIS are very complicated, and due to the closed-source nature of Windows, it is very difficult to find information about NDIS other than architecture details.

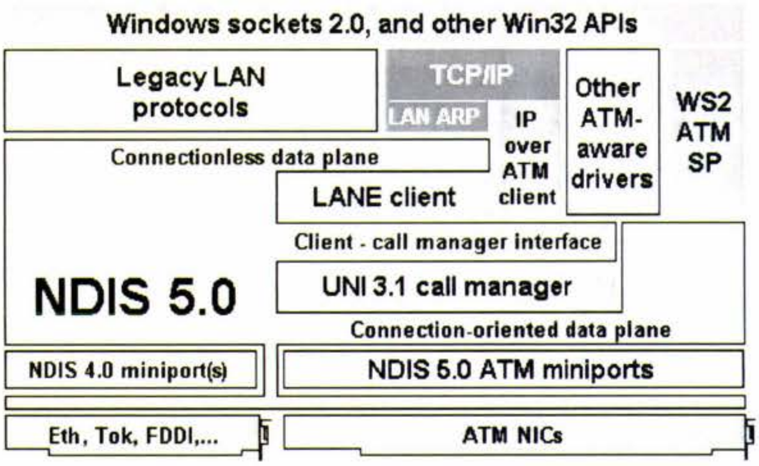


Figure 13.1: Windows NDIS v5 Architecture as from [33]

Also, the current version of NDIS (version 5) within Windows does not support native 802.11 frames as yet [34], although NDIS v6 will – (this is scheduled to be included in the Windows XP Service Pack 2 and near future updates Server 2003). To gain access to the 802.11 frames, one of the following must happen [34].

1) The actual vendor drivers have proprietary interfaces for the network card.

2) The vendor consents to supply information allowing the creation of a custom miniport driver.

Another problem with the current version of NDIS is that packets have 'ownership rights' – which means that if a program did not create a packet, it cannot modify the packet *unless* an

NDIS wrapper operation is used. This is impossible for a pre-MAC driver as because the current version of NDIS does not support native 802.11 frames, there are no native 802.11 NDIS operations.

Windows 2000, and later versions, do actually have two 'hooks' in the network stack. These are located within the IP layer. Both hooks ('filter-hook' and 'firewall-hook') only work by examining the packet and recommending an action to the IP layer – they cannot modify the packet. For example, 'filter-hook' can only look at the packet and tell the IP layer to forward it, drop it or pass it (where the IP layer can filter by itself).

13.1.4 Implementation in Linux

Under the Linux operating system kernel, a framework known as NetFilter handles packets that are sent and received by the kernel's network stack. It primarily consists of a number of 'hooks' within the network stack that can be registered to external functions or programs. These hooks are positioned at defined points in a packet path through the network stack. The positions are pre-routing, post-routing, local-in, local-out, and forward.

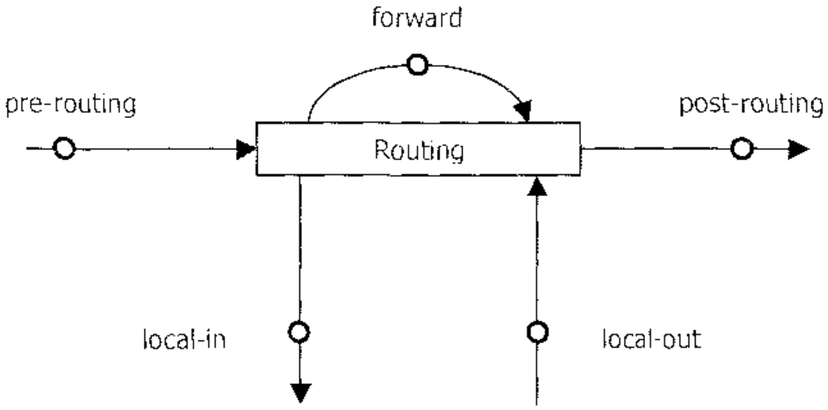


Figure 13.2: Location of NetFilter Hooks

Once registered, these programs are passed all of the packets that traverse the hook that they registered too. The programs have several options regarding these packets – they can simply block or forward the packets, steal the packets (whereby all responsibility of sending the packet then rests with the external program) or queue the packets.

Primarily, NetFilter is used to construct firewalls, do address and port translation and assist routing and quality of service decisions. However, due to its flexibility, it can also be used to add in third party extensions that perform a required task. These extensions have the ability to perform packet manipulation e.g. manipulate bits in the IP header to achieve higher quality of service.

This architecture is well suited to a pre-MAC aggregation driver. The aggregation program would register with the 'post-routing' hook for sending packets. This hook is situated at the last point in the Linux network stack before the MAC sub-layer – ideally suited for the aggregation driver. Likewise for received packets, the 'pre-routing' hook is immediately above the MAC sub-layer (in the case of received packets, the routing function determines whether the packet is destined for the local host, or if it needs to be forwarded on through the network). This is where the de-aggregation mechanism would be situated.

13.2 FURTHER EXTENDING AGGREGATION

13.2.1 Multiple Receiver Aggregation

A possible extension of the aggregation presented in this thesis is Multiple Receiver Aggregation (MRA), where packets destined for more than one receiver are aggregated into a single frame. The aggregation mechanism presented thus far has dealt with aggregating packets destined for a single station only.

MRA would require a totally different implementation to the 'single receiver' aggregation mechanism outlined previously. Firstly, the sub-header information for each packet aggregated into the payload would have to be much larger – and may even be the same as the full frame header. The throughput improvement offered by MRA would be higher than standard aggregation for lower network loads, but probably the same for higher network loads. Also, the need for complex queues and queueing control may be reduced.

Also, MRA would require a more complex MAC control mechanism. As there are several receivers, each medium reservation (RTS) will generate multiple responses (CTS), which will require a more complex controller, a more complex MAC-physical layer communication, and possibly changes to the overall medium access scheme to support multiple receivers.

Additionally, there is the possibility of the different receivers having varying capabilities – most importantly different transmission modulation and rate capability. This means that a single MRA frame may require rate changes within the single physical layer PDU used to send it. For this reason, MRA may be more effective and efficient as a physical layer feature, which could even be used in conjunction with a single receiver MAC sub-layer aggregation.

14 APPENDICES

As the material included in the appendices is extensive, it has been included on a CD-ROM, rather than as pages in this thesis. The list below gives an overview of what is contained on the CD-ROM.

Appendix A

MATLAB Simulation Code, including the following files:

byteVpkt.m	Byte load against packet load simulation
comeback.m	Combined acknowledgement simulation
iqm.m	IQM and look-ahead simulation
load11b.m	802.11b aggregation simulation
payload.m	Increased payload size simulation
pktsize.m	Packet generator
psb.m	Payload size backoff simulation
queueing.m	Queueing algorithm simulation
sfh.m	Stored frame header simulation
tqp.m	Temporary queueing priority
tqp2.m	Temporary queueing priority with special traffic characteristics
tqp2no.m	Special traffic, without temporary queueing priority

Appendix B

The results for all of the simulations carried out, in Excel format, including graphs. They are organised as folders with the same names as the files listed above for Appendix 1.

Appendix C

The author's publications, as listed in chapter 15 for ease of reference, in PDF format.

Appendix D

The 802.11 standards for ease of reference, in PDF format, including the following standards: 802.11; 802.11a; 802.11b; 802.11g.

15 AUTHORS PUBLICATIONS

M. P. Morrison, G. A. D. Punchihewa, L. De Silva and F. M. Al-Ali. "Novel Techniques for Improving the Throughput for the 802.11 MAC Sub-layer in Wireless Networks." *Proceedings of the Eleventh Electronics New Zealand Conference*, Palmerston North, New Zealand, pp. 196-201, November 2004.

M. P. Morrison, G. A. D. Punchihewa, L. De Silva, and F. M. Al-Ali. "Aggregating Packets to Improve 802.11 Wireless Network Performance". *Proceedings of the 3rd Workshop on the Internet, Telecommunications and Signal Processing*, Adelaide, South Australia, Australia, pp. 64-69, December 2004.

16 REFERENCES

1 Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. IEEE Std. 802.11, 1999.

2 Amendment to IEEE Std. 802.11, 1999 - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications – Amendment 1: High-speed Physical Layer in the 5 GHz Band. IEEE Std. 802.11a-1999.

3 Amendment to IEEE Std. 802.11, 1999 - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications – Amendment 2: Higher-speed Physical Layer Extension in the 2.4 GHz Band. IEEE Std. 802.11b-1999.

4 Amendment to IEEE Std. 802.11, 1999 - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications - Amendment 4: Further Higher Data Rate Extension in the 2.4 GHz Band. IEEE Std. 802.11g-2003.

5 J. Rosdahl. "Project Authorisation Request for High Throughput Study Group". IEEE Working Group Paper 802.11-02/0798r7, March 2003.

6 Amendment to IEEE Std. 802.11, 1999 - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Specification for Operation in Additional Regulatory Domains.

7 Amendment to IEEE Std. 802.11, 1999 - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Spectrum and Transmit Power Management Extensions in the 5GHz band in Europe.

8 Amendment to IEEE Std. 802.11, 1999 - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications - Amendment 7: 4.9 GHz–5 GHz Operation in Japan

9 Amendment to IEEE Std. 802.11, 1999 - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications - Amendment 6: Medium Access Control (MAC) Security Enhancements

10 M. Gast. "When Is 54 Not Equal to 54? A Look at 802.11a, b, and g Throughput". Available from www.oreillynet.com/pub/a/wireless/2003/08/08/wireless_throughput.html. 24 May 2004

11 M. Baker and D. Tang. "Analysis of a Local Area Wireless Network". *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, Boston, Massachusetts, USA, pp. 1-10, August 2000.

12 D. Kotz and K. Essein. "Analysis of a Campus-wide Wireless Network". *Proceedings of the 8th Annual International Conference on Mobile Computing and Networking*, Atlanta, Georgia, USA, pp. 107-118, September 2002.

13 D. Eckardt and P. Steenkiste. "Measurement and Analysis of the Error Characteristics of an In-Building Wireless Network". *Computer Communication Review*, Vol. 26 No. 4, pp. 243-254, October 1996.

14 B. J. Bennington and C.R. Bartel. "Wireless Andrew: Experience Building a High Speed, Campus-Wide Wireless Data Network". *Proceedings of the 3rd Annual ACM/IEEE International Conference on Mobile Computing and Networking*, Budapest, Hungary. pp. 55-65. August 1997.

15 P. Garg, R. Doshi, R. Greene, M. Baker, M. Malek and X. Cheng. "Using IEEE 802.11e MAC for QoS over Wireless". *Proceedings of the IEEE Performance, Computing and Communications Conference*, Phoenix, Arizona, USA, pp. 537-542, 2003.

16 A. Grilo and M. Nunes. "Performance Evaluation of IEEE 802.11e". *Personal, Indoor and Mobile Radio Communications*, pp. 511-517, 2002.

17 A. R. Dias, S. Rouquette-Leveil and S. Simeons. "Multiple Antenna OFDM Solutions for Enhanced PHY". IEEE Working Group Paper 802.11-04/0229r0, March 2004.

18 Y. Xiao and J. Rosdahl. "Throughput and Delay Limits of IEEE 802.11". *IEEE Communications Letters*, Vol. 6 No. 8, pp. 355-357, August 2002.

19 W. Y. Choi and S. K. Lee. "Maximizing MAC Throughputs by Dynamic RTS-CTS Threshold". IEEE Working Group Paper 802.11-04/0312r0, March 2004.

20 S. T. Sheu, T. Chen, J. Chen, and F. Ye. "The Impact of RTS Threshold on IEEE 802.11 MAC Protocol". *Proceedings of the Ninth International Conference on Parallel and Distributed Systems*, Taiwan, China, December 2002.

21 S. Yan, Y. Zhou and S. Wu. "An Adaptive RTS Threshold Adjust Algorithm based on Minimum Energy Consumption in IEEE 802.11 DCF". *Proceedings of International Conference on Communication Technology*, Beijing, China, pp. 1210-1214, April 2003.

22 M. Natkaniec and A. R. Pach. "An Analysis of the Influence of the Threshold Parameter on the IEEE 802.11 Network Performance". *Proceedings of the IEEE Wireless Communications and Networking Conference*, Chicago, Illinois, USA, pp. 819-823, September 2000.

23 P. Chatzimisios, A.C. Boucouvalas and V. Vitsas. "Packet Delay Analysis of IEEE 802.11 MAC Protocol". *Electronics Letters*, Vol. 39 No. 18, 4th September 2003.

24 K. Xu, M. Gerla and S. Bae. "How Effective is the IEEE 802.11 RTS/CTS Handshake in Ad-Hoc Networks". *Proceedings of the IEEE Global Telecommunications Conference*, Taiwan, China, pp.72-76, November 2002.

25 Z. Ye, D. Berger, P Sinha, S. Krishnamurthy, M. Faloutos and S. K. Tripathi. "Alleviating MAC Layer Self-Contention in Ad-Hoc Networks". *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking Posters*, San Diego, California, USA, September 2003.

26 D. Karl, S. Monahan and A. Whitman. "Integrity TurboCell – Enhancing the Capabilities of Standard 802.11". Integrity Data Systems. Available from www.integrity.com.au. 19 April 2004.

27 A. Jain, M. Gruteser, M. Neufeld and D. Grunwald. "Benefits of Packet Aggregation in Ad-Hoc Wireless Network". Technical Report CU-CS-960-03, Department of Computer Science, University of Colorado, Boulder, Colorado, USA. August 2003.

28 Atheros Communications, Inc. "White Paper: Super-G Maximizing Wireless Performance". Available from www.atheros.com. 19 April 2004.

29 R. Bruno and V. Carla. "Data Flow on Queue Machines". *Proceedings of 12th Int. IEEE Symposium on Computer Architecture*, Boston, Massachusetts, USA, pp. 342-351, June 1985.

30 Absolute Value Systems. "Absolute Value Systems, Inc. - linux-wlan Page". Available from www.linux-wlan.com/linux-wlan/. 4 Feb 2005.

31 "SourceForge.net Project Info - MADWIFI". Available at sourceforge.net/projects/madwifi/. 4 Feb 2005.

32 Jean Tourrilhes, Hewlett Packard. "Wireless LAN resources for Linux". Available from www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Wireless.html. 4 Feb 2005.

33 Microsoft Windows Hardware and Driver Central. "Network Driver Interface Specification - NDIS 5.0 Overview". Available from www.microsoft.com/whdc/archive/ndis5.mspx. 4 Feb 2005.

34 Ndis.com, Printing Communications Association Inc. "Can I use NDIS to send and receive native 802.11 frames?" Available from www.ndis.com/faq/QA08010302.htm. 13 Sep 2004.