A comparison of cross- and single-layer retransmission optimisations for 802.11

MRes Thesis

Macquarie University, Science and Engineering Faculty, Computing Department

Jennifer Gielis

Supervisors

Rajan Shankaran and Michael Hitchens

Thursday 31st May, 2018

Abstract

Retransmissions in the 802.11 wireless LAN protocol often contain data that was successfully conveyed in the initial message, but is ignored due to corruption during transmission thereby reducing network efficiency. By compartmentalising data, and consequently the corrupted portions of the frame, into distinct sub-fragments 802.11 protocol performance can be further improved. Cross-layer designs provide opportunities to assist in accomplishing this improvement that may not otherwise present themselves in normal architectures using isolated layers.

This thesis presents two new and novel schemes, one that is based on cross layer design and the other using the traditional single layer approach. Both schemes are based upon the same principle of compartmentalising corruption. Additionally, a comparison between the two schemes is provided in order to investigate the strengths and weaknesses of cross-layer design, while simultaneously presenting an 802.11 protocol with enhanced spectral efficiency.

Both schemes divide layer 2 frames into small fragments separated by frame check sequences in order to facilitate retransmission of data that was corrupted during transmission, while avoiding resending correctly received data. They differ in the meta-data mechanisms relied upon to manage this process, and the layer in which retransmission decisions are made, but are fundamentally similar; permitting a unique opportunity for the informative comparison between cross- and single- layer designs.

Moderate bandwidth increases were observed over standard 802.11n with both schemes at medium to high channel error rates, alongside situational decreases in latency. The cross-layer scheme performed slightly worse than the single-layer scheme, however it provided a significant opportunity for further optimisations that were not readily apparent in the single-layer scheme.

Statement of Originality

This work has not previously been submitted for a degree or diploma in any university. To the best of my knowledge and belief, the thesis contains no material previously published or written by another person except where due reference is made in the thesis itself.

Signed

Jennifer Gielis, Monday 28th May, 2018

Contents

Chapter 1

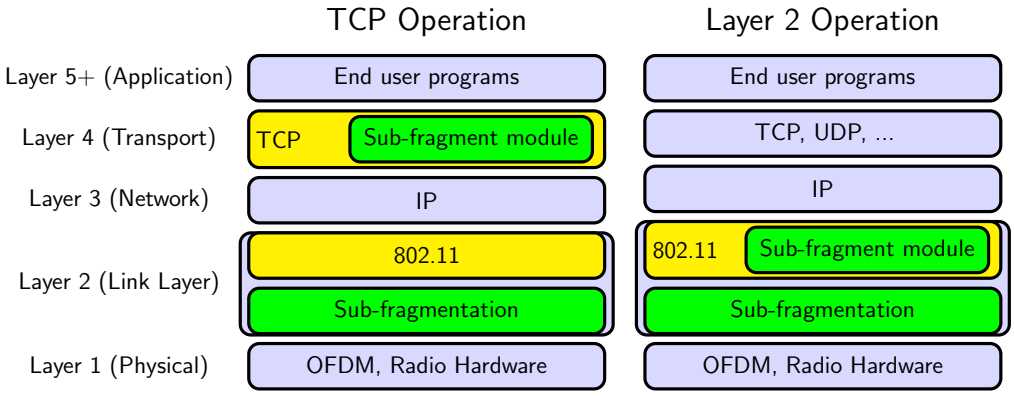
Introduction

Wireless data networks are an integral part of a large number of corporate and residential networks. In domestic usage they are rapidly supplanting wired networks as the access method of choice to the wider internet, as well as to local resources. The efficient usage of the electro-magnetic spectrum is consequently, and increasingly, an important issue given the fundamentally limited spectrum that is available in any given area.

The IEEE 802.11 working group has laid out an ever evolving set of local wireless networking standards over many years which are widely used, with each new revision of the standards quickly seeing rapid real world development and roll-out. Despite this, the protocols defined in these standards have some significant weaknesses under a number of conditions. The IEEE wireless 802.11n standard (2009 Revision) in particular added several key features with a view to increasing bandwidth and efficiency, such as Multiple-Input Multiple-Output (MIMO) radio arrangements and MAC Protocol Data Unit (MPDU) aggregation. Despite this, there is still left for further improvements in spectral efficiency. Chapter 2 aims to clarify some of these weaknesses and possible improvements.

802.11n is new enough to still be in wide usage and is old enough to have accessible simulations. It is fundamentally very similar to the latest released standard revision, 802.11ac/ad, which is an alteration of 802.11n by making previously optional features mandatory [1], along with a few new additions that do not fundamentally change the standard; for these reasons, 802.11n has been selected as the platform for this thesis. Looking further forward, 802.11ax may introduce the usage of Multi-User Multiple-Input Multiple-Output (MU-MIMO), which would have an impact on this research work as it is likely to significantly alter the way media access contention is performed.

Cross-Layer Optimisation (CLO) is a technique used for the purpose of improving the performance of traditionally isolated modules in a network stack. It has been broadly applied to wireless, including 802.11, networks in the past; usually allowing end-user



Note: Unmodified from standard in Blue, Modified in Yellow, New in Green

Figure 1.1: Modules in the sub-fragmentation schemes

applications [2], or network routing functions [3] to vary their behaviour based on channel conditions. The process of sharing information between non-adjacent network layers in the protocol stack is a fundamental violation of normal network design practices, and it does bear a cost in implementation complexity and potentially lowered compatibility [4].

802.11n network nodes are often [5][6][7] forced to retransmit data that has already been sent, either due to corruption or collision, which renders the data unreadable by the receiving node. Such retransmissions grow more common with degrading Signal-to-Noise Ratio (SNR)s. This data-corrupting interference is detected by the means of a single Frame Check Sequence (FCS) in the frame's trailer, the result of which is that any error large enough to be uncorrectable by Layer 1 (L1) Forward Error Correction (FEC) (such is included in Orthogonal Frequency-Division Multiplexing (OFDM)) will result in the entire frame being discarded.

The nature of FEC in 802.11 is such that it cannot compensate for burst errors large enough to invalidate an entire OFDM symbol. This research aims to improve the weaknesses of using this limited style of FEC that involves the use of a single FCS; specifically by allowing data that is part of a partially corrupted frame, but which is not itself corrupted, to be stored by the receiver - instead of re-sent - until such a time as the corrupted data can be resent and the full frame restored. This will be accomplished through a mechanism of sub-fragmentation, wherein any frame transmitted by the modified 802.11 network is logically split into multiple sections, each protected by their own dedicated FCS. Herein, this mechanism will be referred to as the sub-fragmentation scheme.

This research further aims to demonstrate the advantages, and disadvantages, of using CLO methodologies by specifying two modes of operation for the sub-fragmentation scheme; one of which is based on CLO and the other being a single layer scheme that includes enhancements to the standard Layer 2 (L2), Media Access Control (MAC),

functionality. Both operate on nearly identical principles to achieve the same goal. A graphical overview of the modules added to a TCP/IP stack to execute both modes of operation is provided in Figure 1.1. As depicted, 'Transmission Control Protocol (TCP) operation' is the cross-layer scheme, requiring modification of TCP and 802.11 protocol modules; 'Layer 2 Operation' has all necessary functionality contained within L2.

CLO approaches allow for different layers in a stack to have more information about each other's function than what is traditionally possible and this allows more efficient protocol operations; but frequently result in reduced compatibility and more complex implementations.

CLO methods, the basic architectures of which are illustrated in Figure 1.2 and [8], are design techniques that allow different layers in an inter-networking stack to communicate with each other in ways that differ from, and often contradict, the classic OSI and TCP/IP stack models.

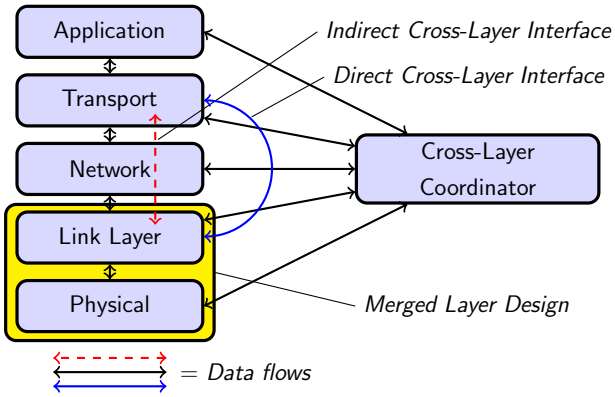


Figure 1.2: An overview of cross-layer optimization architectures

The sub-fragmentation TCP operation scheme uses an *indirect cross-layer interface*, meaning that data passes through intermediate layers but is not acted upon by them, being treated as payload. As will be shown in the following chapters, this permits minimising the side-effects of using CLO (specifically, reducing implementation complexity) while maintaining the benefits of sharing information across disparate layers.

This thesis is laid out as follows. Chapter 2 comprises a literature review and background information on CLO and 802.11 based techniques that have similar goals to this research, in order to provide a theoretical foundation to contextualise the following chapters. Chapter 3 deals with the model proposed in this thesis, in progressively greater detail, along with aspects of 802.11's design that are relevant. In Chapter 4 simulation results are provided to evaluate the proposed models in practical network environments, alongside an analysis that aims to explain the model's performance. Finally, concluding comments and direction for future works are discussed in Chapter 5.

Chapter 2

Background

As previously introduced, the underlying aim to this research is to improve spectral efficiency for devices participating in 802.11n+ networks through the use of CLO techniques and in the process provide a contrasting analysis of the benefits and drawbacks of CLO. The sub-fragmentation schemes this research focuses upon presenting are designed to meet these goals in environments where wireless signal reception is of a particularly poor and randomly varying quality. These schemes will have minor improvement to a slightly negative effect on networks operating in ideal conditions ($< 5\%$ chance of a single symbol error per 1500 byte frame).

802.11 currently handles decreasing signal quality quite well, until a certain threshold (which is dependant upon signal encoding) when OFDM's FEC starts to regularly fail. After this threshold, 802.11 consumes an ever increasing amount of airtime on retransmissions, costing spectral efficiency and power.

High spectral efficiency is of particular importance, as it improves the user experience for all users of wireless networking in the local area; not necessarily just those within the jurisdiction of one Basic Service Set (BSS).

Power savings are a secondary goal; there is a small likelihood that the sub-fragmentation schemes will increase overall device power consumption, though this is heavily implementation, and hardware, specific. This is a result of the reality that a surprisingly large fraction of 802.11 wireless node power consumption goes into decoding incoming traffic (which these schemes makes more complex), to the point that some devices have a higher receive power loading than the one it presents during transmission. Regardless, overall power consumption may be reduced by the lower total radio active time achieved with the aim of this research; fewer retransmissions of each payload byte.

As is to be covered in the following sections, this research is by no means the first to explore the use of CLO schemes to improve wireless performance, nor the first to consider enhancing TCP for the same effect. Despite this though, this research appears to be

investigating a new and original CLO method to achieve its aims, and as such represents a step in the understanding of CLO and 802.11 optimization methods.

The first section of this chapter presents alternative techniques proposed for use with 802.11 with similar goals to sub-fragmentation, as well as discussing how sub-fragmentation relates to them. The second section discusses CLO and some of the research that has been conducted using CLO techniques.

2.1 802.11

The IEEE 802.11 protocol suite is generally designed to maximise operational throughput as well as maintain a relatively 'fair' distribution of airtime. There are a number of decisions made during the design or implementation processes which have clearly selected mathematical maxima. Frame lengths are a good example of this, [9] shows that throughput is maximised when frame lengths are approximately 2300 octets; which lines up very closely with the often selected default fragmentation threshold of 802.11 protocols in real world implementations (although the 802.11 specification does not actually directly specify this figure [1]).

Application throughput, at maximum network load, is a result of the spectral efficiency of the network. This is because any condition that causes the network to operate inefficiently will also cause application data to be sent at a slower rate. The sub-fragmentation schemes aim to improve spectral efficiency through the reduction of unnecessary retransmissions; however there are other factors and methods which are instructive to review.

802.11 is designed to handle a number of real-world complications, many of which are covered within this chapter, along with some commentary regarding some of the techniques used in the management of these complications.

2.1.1 OFDM

OFDM is the current modern standard for many high speed data networks (e.g. 802.11 a, g+ and 4G mobile technologies), and it relies upon dividing the allocated bandwidth into many sub-channels, most of which are encoded as if it were a standalone channel. Since the symbol rate possible on a given channel is defined by the available bandwidth, each individual sub channel encodes symbols at a much lower rate than a single channel utilizing all available bandwidth. While this may not appear to be useful since the theoretical data rate is the same, in real world applications a low symbol rate reduces the occurrence of Inter-symbol Interference (ISI) caused by the signal travelling over multiple pathways, thus

reducing the SNR required [27]. OFDM also has the property that each sub-channel is orthogonal to its neighbours; this is, they will not interfere with each other - but frequency centres of individual sub-channels are only exactly as far apart as is required to ensure this outcome.

The primary disadvantage of OFDM is that the receiver needs to carefully synchronise all channel states with the current transmitter, for which purpose a number of sub carriers are dedicated to synchronization, these being known as Pilot Carriers [27], visualized in Fig. 3.3. [28] provides an overview of the synchronisation methods used in OFDM.

OFDM is almost always deployed with a FEC code, with the purpose of correcting any errors that individual sub-carriers accumulated during transmission. Traditionally these have taken the form of simple convolutional coding, which are easy to calculate and acceptably efficient. In recent years, the use of turbo codes and other advanced FEC codes have improved the efficiency of non 802.11 OFDM deployments.

Interleaving is an important part of ensuring that the FEC decoder is able to perform its function, even in situations that may normally cause problems, such as a situation wherein a large number of adjacent sub-channels fail simultaneously. The interleaving process instead assures that nearby sub-carriers do not hold data that is adjacent in the data stream.

2.1.2 Signal-to-Noise Ratio

Operational SNR has an inordinate influence in 802.11 link and application performance [10][11]. The interaction between power consumption and data rates means that in order to achieve the best power consumption per bit, the highest data rates are preferable; however Transmit (TX) power naturally reduces as data rates increase, which rapidly leads to data corruption. There is therefore a specific data rate which is the highest possible at an acceptable Symbol Error Rate (SER) for any given combination of medium conditions and radio hardware.

The sub-fragmentation schemes aim to increase the resiliency of 802.11 primarily by breaking the frame into individually verifiable chunks. While there is still a definable transmission rate at which the data corruption becomes unacceptable, it is hoped that an increase in available and effective data rate reduces overall power consumption for a given workload.

Dynamically determining the cause of a transmission failure can be difficult, and is important for maximising data rates over time. [12] demonstrates a technique designed to distinguish between packet loss due to low SNR corruption and frame collision. The sub-fragmentation schemes may also be able to provide additional context information to any rate adaptation schemes, in order to improve the performance in highly collision-prone

environments.

The number and location of corrupt sub-fragments in a frame have the potential to convey significant information about the operating state of the network to 802.11, and TCP in the CLO version. This information can permit more finely grained calculation of current channel SNR, and thus more accurate selection of data rate adaptation and TX power.

2.1.3 Power Consumption

The cost of data processing in terms of power consumption for each frame is important, as there are some changes proposed in the sub-fragmentation schemes that may result in a considerable increase in processing load; primarily this is in the form of FCS calculation and possibly the added processing requirements of TCP while it is considering additional cross-layer information. Unfortunately, it is difficult to breakdown the power consumption of existing devices finely enough to know exactly where non-radiative energy goes; and in the cases where it has been done, it is highly variable based upon the device under test.

Power consumption in TX mode is highly related to the radiated power, and the number of TX chains (i.e, MIMO configurations [13]). Interestingly, as data rates increase the effective TX power per bit reduces [14], the consequence of which is lower power consumption at higher data rates for a particular payload; however the downside of this is a lower SNR, the result of which is higher Bit Error Rate (BER)s and SERs.

As the lowest power consumption mode that the hardware can achieve is being in a sleep state, higher data rates also allow the completion of application-induced workloads more quickly; the sub-fragmentation schemes mostly reduce power consumption through this mechanism, completing transmissions faster or simply sending less data overall, thereby reducing the total amount of time spent in active states.

2.1.4 Medium Induced Data Corruption

Low SNRs result in more exposure to data corruption due to numerous environmental conditions; including random environmental noise and artificial signal interference. The work in [15] shows some data regarding channel usage in urban environments, for the purpose of dynamic frequency selection. Signal fading and reflections should not be significant factors for stationary nodes, as the rate adaptation scheme in use will set a SNR sufficient to avoid issues caused by these factors, though this cannot be said for mobile nodes.

Any environmental condition that causes data corruption can have several possible symptoms, the most obvious symptom (to the radio hardware) is an invalid symbol. At

higher data rates, modulation schemes with a greater number of valid symbol states are required, and so invalid symbols (as opposed to corrupt symbols) are less likely. In the event that a symbol is received incorrectly, but is not invalid, this is not immediately obvious to the radio hardware and must be corrected through FEC, or detected through the use of the FCS. 802.11 uses the IEEE 32-bit Cyclic Redundancy Check (CRC), which is capable of detecting medium-induced errors with extremely high probabilities [1][16].

Within [17] there is a study of the effects of noisy channels on ad-hoc 802.11 networks, with some emphasis on fragmentation and therefore the transmission of multiple fragments connected to the same payload. The maths employed for this study have relevance for managing the size of corruption failures domains; albeit in a different form than for sub-fragmentation.

The usage of CRCs in the sub-fragmentation schemes is therefore appropriate, for much the same reason that it was selected in 802.11. The computational cost is the main drawback, due to the number of times that it may be applied in a single frame.

2.1.5 Contention

One of the primary sources of interference is that which is generated by other nodes running on the same network. In order to minimise this influence, there are a multitude of ways to manage the cooperation between nodes, and just as many challenges introduced by the needs of users.

In the days of Continuous Wave (CW) morse code systems, operators could receive and transmit pseudo-simultaneously and so detecting if another station was trying to talk at the same time was similar to the principles used in wired systems; thus a transmission could be aborted quickly and contention could be managed. This worked because of the relatively large portions of silence in CW transmissions.

For most radio equipment, the problem is that the transmission from the local station overwhelms any remote signal to the point that they are not decodable. There have been numerous attempts to correct this over the years with various degrees of success, but none are currently deployed in common end-user hardware. [18] presents a recent attempt at dealing with this problem, and an overview of the challenges.

In most commonly deployed systems, there are effectively three main approaches to collision avoidance; frequency, time or code division. Frequency division is simple to implement on point-to-point links with special hardware; each station listens to one frequency and transmits on another allowing full duplex communications.

Time division can be categorised by the nature of the network involved; in networks with small node counts and fixed node locations, the stations can simply agree to a rotation of frequency ownership, with each one only transmitting in their allocated slot. More

dynamic networks are more complex in this scheme - each node listens to the medium for some time and transmits if not busy. The node is only aware of the success or failure of transmission by some external mechanism, such as through the use of acknowledgements from the recipient.

Time division schemes are complicated by light speed delays; which can be significant over long ranges. A distant node may already be transmitting, but can not be heard until the signal propagates. This may lead the receiver to mistakenly assume empty medium. More recently, the broad availability of GPS devices has permitted certain network nodes to have specific knowledge of their locations and thus precisely schedule transmissions within a TDMA scheme, maximising spectral efficiency. This approach has been used in 802.11, but usually in highly vendor-specific point to point networks.

Code division has been commonly used in some environments (i.e. Global System for Mobile Communications (GSM)) to alleviate the need for either more frequencies or complex inter-node timings. This works in a very similar way to Direct Sequence Spread Spectrum (DSSS) spread spectrum techniques, because it multiplies all node transmissions by an orthogonal chipping code that allows the receiver to recode multiple simultaneous signals. The downside to this method is that it cannot be easily de-centralised, and each node only receives a small portion of the available bandwidth - generally not compatible with modern internet usage patterns.

A recent development in large scale networks has been a version of OFDM-MA (also called MU-MIMO), which is similar to code division, but is in fact frequency based. Individual sub-carriers of OFDM channels are temporarily assigned to different network users and are essentially independent micro-channels during their assignment. OFDM uses different sub-carriers for alternate roles and so the exact distribution of sub-carrier ownership is a complex problem. [19] provides a (somewhat old) method for uplink synchronisation amongst distributed nodes.

The L2 only sub-fragmentation scheme affects contention by permitting back-off free retransmission opportunities.

2.1.6 Protocol Overheads

Data corruption may be unrecoverable by any FEC, and in this instance retransmissions become necessary. The 802.11 standard retransmission mechanism has some serious limitations, such as being unable to recognise if any data in a partially corrupted frame was received correctly and thus requiring potentially redundant frame retransmissions. There is also an issue with the amount of time necessary to initiate any corrective transmissions, due to various timing specifications in the standard [1]. These issues are discussed in greater detail in later chapters.

The issue of high MAC function costs on retransmission were recognised in [20], wherein a scheme involving back-off free retransmissions and an additional ACKnowledgement (ACK) packet to notify the sender (and implicitly, neighbouring nodes) of failed frames, was specified. This is in fact quite similar to the intermediate ACK specified in the L2 sub-fragmentation scheme.

[21] proposes a scheme of μACKs to deal with the inefficiencies of retransmissions. The scheme involves a secondary frequency range that the destination node transmits μACKS on, simultaneously whilst receiving the main data stream. The sender listens to these μACKs and records which data segments failed. This permits a highly granular knowledge of failures, in a very timely fashion. The main drawback of this scheme is that it is not compatible with existing hardware, which cannot typically transmit and receive simultaneously.

Similarly, the sub-fragmentation schemes address the frame corruption issue in 802.11 by adding the ability to detect the failure of any given sub-section of the message. The result is that the amount of data that must be retransmitted in the event of an environmental data corruption can be reduced, and can be performed in a back-off free manner.

2.1.7 Link Adaptation

One of the primary mechanisms that permits data networks to continuously maximise the data rate available to users is Link Adaptation; a process by which communicating nodes agree (implicitly or otherwise) to change some aspect of their transmission scheme. Many parameters can be dynamically changed - in 802.11 networks, the modulation and FEC coding rates will adjust themselves; most commonly this is done in response to a packet transmission failure, or some kind of Received Signal Strength Indicator (RSSI) parameter. The number of antennas currently in use, when MIMO is employed, can also be dynamic.

Maximising the data rate of a node at any given moment is a function of the data rate adaptation scheme, an overview of common variants of these schemes is provided in [10] and some more specific concerns are raised in [13]. Typically, adaptation schemes use either SNR of received packets, or packet loss rates in order to regulate their activity [10].

Link adaptation is an extensively studied area. [10] provides a thorough overview of some data rate adaptation techniques deployed a number of years ago; more recent works have typically focused upon ultra-high data rate, or large user count networks. Dynamically altering fragmentation parameters (fragment size, retransmission timers) is proposed in a number of works, as it is not included in the base specification [22][23][24]. [25] proposed an energy aware rate adaptation scheme, as does [13]. Somewhat related to

Link Adaptation are other channel condition estimation schemes, which use link health data for other purposes. As an example, the work in [26] proposed a technique to calculate wireless channel impairments in a distributed fashion.

Link adaptation is relevant to the sub-fragmentation schemes as, when used, it provides additional information around the cause of specific error types. For example, in 802.11, the partial corruption of a frame results in exactly the same behaviour, as observed by the sender, as the node moving out of range of the destination. Sub-fragmentation permits the sender to distinguish these conditions, and thus react more intelligently when applying data rate adaptation.

2.1.8 Forward Error Correction

Despite OFDM having FEC codes, they are ultimately quite limited and can only deal with small amount of interference. An overview of the maths behind OFDM FEC is provided in [29]. More extensive use of FEC codes is frequently employed in other areas (such as deep space communications); however there are serious computational consequences to using FEC codes at the high line rates of 802.11n+ solutions, and this may prove impractical, or may completely eliminate any power savings achieved through the reduction in extraneous data retransmissions.

[30] provides an interesting overview of an 802.11a OFDM system, both with and without FEC, as well as presenting the maths behind Viterbi hard and soft decision decoding for a range of modulations. Despite its age, the results are still viable for current OFDM systems; though the relatively low peak data rate of 802.11a should be kept in mind.

A scheme whereby the additional FCSs of the sub-fragmentation schemes could be dynamically swapped out with FEC codes, or even dropped entirely in low-noise environments, would seem to be the best compromise in order to avoid high overhead issues. [31] presents a MAC layer 802.11 FEC that is very similar in principle to L2 sub-fragmentation, as it divides the payload into separate data chunks that each have their own trailer for verification. The primary difference being that the scheme in [31] performs FEC, and has a more limited retransmission scheme.

2.2 Cross-Layer Optimisation

Cross-layer design has been researched in many different ways with a view to its to application in 802.11 networks, and is most often studied in the context of PHY and MAC layer interaction (such as frame lengths in [9]). [32] as well as [33] provide a slightly

dated, but useful overview of some cross-layer designs.

This research will focus upon an indirect cross-layer interface between the MAC and Transport layers (see Figure 1.2); meaning that additional data will be passed along with packet payload *through* any intermediate layers, rather than relying on direct out-of-stack information transfers. There are other ways to classify the type of CLO scheme that best describes the TCP sub-fragmentation scheme, such as the categories proposed in [8]. In line with this classification scheme, this research is non-manager, distributed.

The decision to avoid out-of-stack interfaces is intended to reduce the prerequisites of implementing the final scheme in real world scenarios.

By offloading the retransmission process onto TCP instead of the traditional MAC layer management, the usage of pre-existing functions on the network stack is maximised, while allowing additional MAC functionality without significant added implementation effort.

The next 802.11 revision (802.11ax) may have changes that affect the results of this research, however it seems unlikely as large parts of this work are dependant upon CLO techniques, which have been typically avoided in the base 802.11 standards. [4] points out some of the reasons that may underlay the choice to avoid CLO, as they bring a number of valid concerns regarding architectural cleanliness in the networking stack.

Despite concerns about CLO's possible negative side effects, it has great potential to mitigate the limitations present in the 802.11 standards. [33] covers a number of different architectures that highlight some of the benefits of CLO schemes. These architectures range from singular entities coordinating the actions of all layers simultaneously and directly within a single host, to other approaches that manage multiple hosts on specific layers, via either a distributed or centralised mechanism.

CLO has been explored by some previous works as a facilitator in improving 802.11 performance. [25] proposes a low level CLO scheme that would control the data rate of 802.11 networks on an energy-aware basis. Wireless sensor networks are an area where there has been extensive research into CLO techniques, simply due to the extreme limitations often imposed upon the hardware in such networks, resulting in a need for extreme optimisation; these networks are frequently not 802.11 based but serve as a good basis for determining what is possible with CLO.

A good example of work in this area is [34], which covers the usage of CLO to improve Quality of Service (QOS) in sensor networks; primarily by conceptually merging the QOS queues of different layers in order to ensure high priority packets proceed past lower priority traffic quickly. Typically, this behaviour is limited as those layers lower than where QOS is applied have queues of their own, and attempts to maximise data rate necessarily consume buffers to their fullest, forcing high priority packets to wait.

[35] presents a method of bit rate adaptation that is distributed between receiver and transmitter, and acts between the Physical (PHY) and MAC layers. In this, it is representative of many optimisations already present between these two layers due to their close coupling in 802.11. The merged layer nature of 802.11 itself is often overlooked but is noteworthy, as the speed and reliability already present in wireless data networks would not be possible without CLO that is already present. The presented mechanism provides active feedback to transmitters of their received BER, thus allowing the transmitter to adapt its behaviour in the most appropriate manner.

Closer to the research at hand, some previous works have dealt with (or proposed to deal with) the interaction of TCP and 802.11. [11] almost directly proposes the avenue (but not specific method) of investigation this thesis is approaching; that being a scheme where TCP and 802.11 cooperate in the management of retransmissions, and also provides some background into the problem this research is attempting to solve. Despite it's age, [36] provides a good selection of some of the techniques used to improve TCP's performance over 802.11 networks.

2.3 Background Summary

Within this chapter, we covered important background information about 802.11 and why it behaves as it does in several areas related to retransmission of failed frames. CLO, and its previous uses were also highlighted. Additionally, we covered a number of studies that propose similar goals or methods to the sub-fragmentation schemes, but have unrealistic pre-requisites or are not likely to provide a great enough benefit to justify the implementation costs.

Chapter 3

Model

Fundamentally, the sub-fragmentation schemes are designed to reduce the total number of bytes transmitted over the wireless medium. In order to accomplish this goal, the sub-fragmentation schemes exploit a feature of the 802.11 protocols where there is currently substantial inefficiency - retransmissions; the weaknesses of the existing model will be covered in following sections.

This chapter is organised into three sections. The first section aims to address the current state of the 802.11 architecture, and how it deals with the problem of transmission corruptions. The second section deals with high level design principles of the sub-fragmentation schemes, while placing an emphasis on key issues that generally tend to introduce complications in the design process. Finally, the third section examines details pertaining to the practical implementation of the model, while exploring the efficacy of the model alongside other systems.

3.1 802.11 Standard Architecture

Existing 802.11 architectures primarily attempt to limit the amount of wasted air time through the use of collision avoidance, data rate adaptation and fragmentation, along with L1 FEC.

Data rate adaptation helps by continuously altering the data rate of the transmitter, thereby ensuring that the SNR of the receiver is sufficiently high to keep the BER of the receiver low enough to have a high probability of reception without corruption. OFDM's low level FEC (Viterbi convolutional codes) and constellation point estimations help make this a reasonably reliable mechanism in static conditions.

The problem with data rate adaptation is that it has no direct knowledge of the wireless channel conditions between two nodes; instead, estimations are generated based on particular metrics, defined by the adaptation scheme in use. Frame loss (i.e. no ACK

packet received by the sender) and RSSI from receiver control packets are the two primary mechanisms used for channel condition estimations [10], though alternate schemes have been proposed - some even using CLO [35]. In either case, the reaction to highly unstable noise conditions can often be unacceptably slow, resulting in lost frames while adaptations take place.

802.11's default fragmentation scheme assists in a reduction of wasted airtime by limiting the size of each frame. It does this by breaking down MAC Service Data Unit (MSDU)s submitted from higher layers which are longer than a prescribed length, that is defined by the 802.11 implementer, into multiple MPDUs. In doing so, the MSDU is exposed to un-compensated changes in channel conditions for less time. A detailed discussion of frame length optimisation is presented in section 3.2.1.

The only aspect of 802.11 design that pro-actively heals errors, without retransmissions, is the L1 FEC mechanism (OFDM's for 802.11n). A convolutional code is used in all transmissions, with coding rates that vary based on the modulation scheme that is currently in use. Generally speaking, higher data rate schemes employ coding rates that are less redundant; presumably with the idea that any selection of higher data rates implies better channel conditions. This FEC mechanism does do a reasonable job of correcting medium errors, when used in combination with other techniques such as interleaving (spreading failures over different parts of the medium), however it still imposes limits on the amount of corruption that can be dealt with in a single symbol.

Constellation point estimation (see Figure 3.1 for an example constellation), which assigns incoming samples to the nearest neighbour sampling point (or matches received samples to expected points by some other estimation scheme) also serves a useful function in correcting low-to-medium scale errors [37], however large scale errors are likely to corrupt most or all bits represented by a particular symbol.

In the event of a transmission failure, the standard retransmission mechanism is very simple - attempt to resend the entire lost frame until either a successful ACK is received from the destination, or the hardware retry limit is reached (typically, this is is on the order of 4-10 retransmission attempts). The consequence of this behaviour is twofold; first, frame delivery is quite reliable and second, it is possible to incur an order of magnitude increase in the cost (delay and total spectrum time used) of transmitting any given frame. A short inter-frame timing value is employed to give priority usage of the medium during this process, however this mechanism can be improved and long blockages of other traffic on the network could be reduced.

Each frame transmitted in 802.11 networks contain two values to uniquely identify it; a sequence number, and a fragment number. These represent, respectively, an incrementing counter of the number of MSDUs sent between a pair of nodes, and the MPDU that

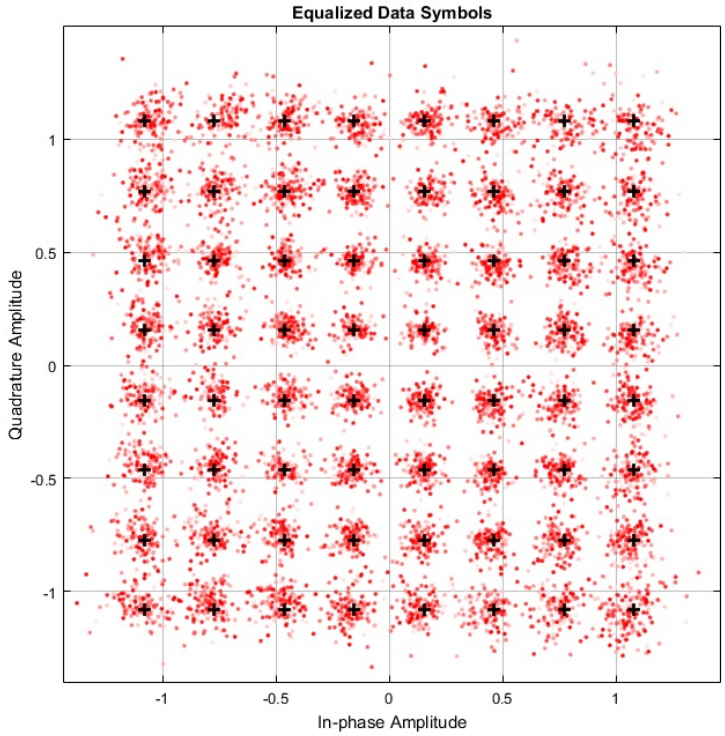


Figure 3.1: Simulated Quadrature Amplitude Modulation (QAM)-64 constellation with a 26dB SNR

comprises a fragment of the MSDU. These values are used in the retransmission process to avoid data duplication and identify the frames that need retransmission.

Despite the joint efforts of fragmentation, data rate adaptation and L1 FEC; 802.11 requires only a single symbol of the transmission to be corrupted before frames are discarded (a discussion of failure sizes is presented in following sections), the result of which is that a great deal of valid data may survive reception but be ignored. It is this property that makes the sub-fragmentation schemes promising in improving overall 802.11 throughput.

MPDU aggregation is a major feature of 802.11n+ that deserves special mention. This involves the formation of AMPDUs, which are comprised of a series of MPDUs within a single L1 frame. This mechanism is quite similar to the way that either sub-fragmentation scheme divides data into chunks separated by checksums, due to the fact that each MPDU retains its original FCS. The receiving node can individually read each MPDU and retransmit only those that have been corrupted. There are 2 factors that differentiate sub-fragmentation from MPDU aggregation. Firstly, MPDU aggregation operates on entire frames submitted from higher layers, resulting in typically quite large amounts of data per FCS; sub-fragmentation typically has data chunks around one order of magnitude lower. Secondly, each MPDU within an AMPDU typically contains MAC,

IP and transport layer headers; massively increasing the overhead per data chunk; by contrast, sub-fragmentation only adds the size of a single checksum.

3.2 High Level Sub-Fragmentation Design

In order to improve 802.11's behaviour, an approach of limiting the amount of data retransmitted was devised during this research. The existing approach to data validation checks the entire frame as one logical entity; this is the simplest design and is common in other networking protocols as well, but it can be limiting on data channels that are lossy or unpredictable. The loss of only a small sequence of bytes out of a payload of thousands has the same result as a total loss.

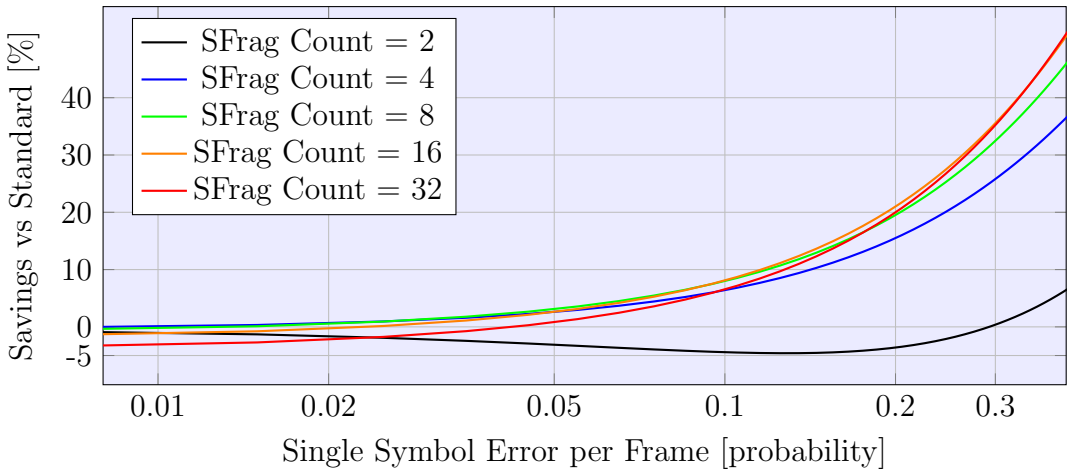
The basis of the sub-fragmentation schemes presented here is to add additional checksums designed to compartmentalise the side effects of minor corruptions. The exact number of divisions with which to separate a frame, and the optimal size of those chunks, is discussed in later sections (Figure 3.2 shows a prediction of the improvements possible at various sub-fragment per frame counts); however the most direct relationship is the one between the probability of an error breaching L1 FEC (and thus a frame loss in the standard) and the overhead of the checksums added by sub-fragmentation, from which one can calculate the reduction in retransmitted bytes.

In order to provide an overview of the benefits and drawbacks of CLO, two separate implementations of this sub-fragmentation scheme are presented in this thesis. The CLO scheme uses TCP's existing technologies, in combination with a low layer sub-fragmentation module (see Figure 1.1), to achieve partial data retransmissions. The other is a purely L2 solution, which has no cross-layer mechanism. Each has its own merits, which are indicative of the strengths of performing retransmission functions at the concerned layer of each approach.

The 802.11 working group most likely did not include a cross-layer solution as the standards they produce are intended to be independent from higher layers, however they likely did not employ something similar to L2 sub-fragmentation due to the complexity of the logic involved.

3.2.1 Frame Size and Corruption

Generally speaking, the higher the SNR (and thus the lower the BER at a given PHY rate) or the fewer stations on the network, the higher the achievable throughput. [9] provides a good overview of the mathematics of these relationships; although they calculate for DSSS and not OFDM. These calculations operate on a number of assumptions that are



Note: this applies to the L2 operation mode only, with a 1500 byte frame

Figure 3.2: Reduction in bytes sent for various sub-fragment counts, 2 byte checksums; without consideration of MAC function losses and limited to 2 retransmissions

altered by the presence of either sub-fragmentation scheme, namely, the assumption that the loss of a frame results in total data loss, and that any retransmission will resend every byte in the originally failing frame.

As error rates approach 0, and disregarding latencies, optimal frame sizes grow arbitrarily large; however, the MAC function will generally attempt to achieve a low enough BER to ensure that spectral efficiency is not significantly compromised by retransmissions, but not attempt to achieve an absolute minimum BER value, as that would typically require a prohibitively low PHY rate which would, in turn, limit bandwidth despite the large frame sizes made practical.

When using the sub-fragmentation schemes, and considering only losses due to corruption (other factors are covered in following sections), it is less important to consider the total length of the frame; instead it is better to focus upon the optimal length of each individual sub-fragment, and the total number of sub-fragments. This is due to the fact that an arbitrarily long frame will accumulate errors only with the loss of individual sub-fragments; in this way the frame is behaving much like a sequence of unrelated frames in a standard network, without the MAC function to separate them. The singular exception to this is corruption losses of the first sub-fragment, which contains the meta-data necessary to interpret all following data.

Both sub-fragmentation modes are compatible with AMPDU operation, and in the simulation created for this thesis AMPDUs are used in all results. It is entirely possible for the concept of MPDU aggregation to be discarded when using either sub-fragmentation scheme, alternatively employing very long stand alone MPDUs. With that being said, TCP sub-fragmentation is far more likely to benefit from such a configuration, as it has

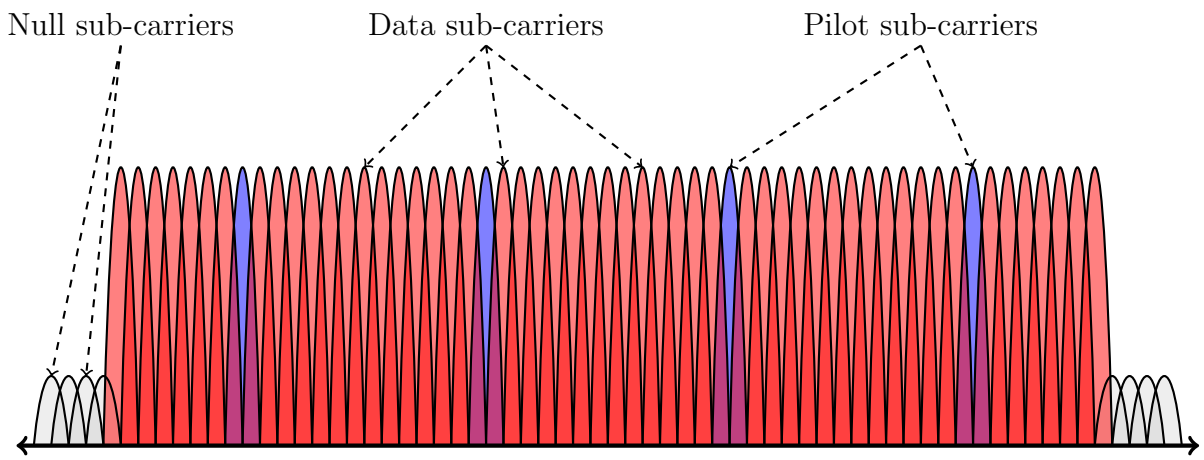


Figure 3.3: 802.11n OFDM sub carrier diagram

the fewest circumstances under which a retransmission of the entire physical frame is necessary.

3.2.2 Corruption Losses

To calculate the effect of corruption induced losses, the length of data corrupted by any specific error should be determined. L1 manages the FEC with convolutional codes applied on a per symbol basis, the payload size of these symbols varies from 26 bits (for Binary Phase-shift Keying (BPSK) with 52 data sub-carriers, see Figure 3.3) up to 2,320 bits (for QAM-64 with 116 sub-carriers and 4 spatial streams) when using OFDM. Any error smaller than these sizes may result in all the data of that block being corrupted, and so they can be thought of as the atomic units for errors presented to higher layers.

Each OFDM symbol is processed using the Viterbi algorithm, with coding rates varying between 1/2 and 5/6. Viterbi decoding has an upper bound on unrecoverable errors of:

$$P_f < 1 - (1 - P_e)^L \tag{3.1}$$

where P_e is a probability bound on per-bit errors, and L is the length of the frame in bits. Deriving P_e is beyond the scope of this work, however there is a particular bounding equation for each modulation scheme (see [38]), which is dependant upon SNR. The multiplicative increase in error probabilities in this equation is responsible for the first major component necessary in calculating optimal sub-fragment sizes.

Another factor that is pertinent is the size of the checksum that is used to determine if a particular sub-fragment is corrupt. In order to achieve maximal efficiency this should be a small value, but robust enough to detect errors reliably. CRC-16, or other similar 16 bit CRC codes, seem to be an appropriate choice to meet these needs, and it is assumed

through the rest of this thesis; although fully optimising this selection requires further analysis. This results in a 16 bit overhead for each sub-fragment.

In 802.11n, the OFDM Physical Protocol Data Unit (PPDU) header's length field is only 12-bits long which creates a hard limit of 4096 bytes of payload. Since the largest possible OFDM symbol is 2,320 bits (though it will be larger again in 802.11ac), and as it is preferable to avoid the overhead of having too many CRCs introduced, a minimum sub-fragment size on this order of magnitude seems to be a sane starting point; a 256 byte minimum would allow for up to 16 sub-fragments per frame. The benefits of greater numbers of sub-fragments per frame decrease due to the higher overheads (caused by the added FCS in each sub-fragment), see Figure 3.2 for a graph of the predicted returns from running sub-fragmentation.

If we assume 802.11n green-field mode with a single spatial stream, the OFDM PPDU has a per-frame overhead of 24 µs (other modes have different, usually longer, overheads) plus MAC header, however the overhead ascribable to any specific sub-fragment varies based on the total number of sub-fragments present in the frame.

The two sub-fragmentation schemes have slightly different equations to use when calculating the overhead per packet, for a given symbol error rate. This system of equations begins by estimating the total airtime used, on average, when using 802.11 for a single payload transmission, given an error probability E.

$$T_{802.11} = (T_{frame} + O_m)(1 - E) + (T_{frame} + O_m)E + \sum_{1}^{C_r}(T_{frame} + O_{mr})E^{C_r} \qquad (3.2)$$

where T_{frame} is the time needed to send the frame, O_m is the time cost of the MAC function for a normal frame, O_{mr} is the time cost of the MAC function when retransmitting (increased, due to MAC back off) and finally C_r is the maximum number of times that a retransmission will be attempted on failure.

MAC function costs are left quite generic within these equations, as they depend upon the co-ordination function in use on the network; usually Distributed Coordination Function (DCF), Point Coordination Function (PCF) or Hybrid Coordination Function (HCF) for 802.11n.

The time estimation equation for the L2 operation mode is quite similar, but has the added cost of the sub-fragment CRCs, and a reduction in the number of bytes sent on retransmission.

$$T_{SF-L2} = (T_{frame} + O_m + T_{crc}N_{sf})(1-E) + (T_{frame} + O_m + T_{crc}N_{sf})E + \sum_{1}^{C_r}((T_{frame} + O_{mr})(\frac{E}{N_{sf}})^{C_r} + (\frac{T_{frame}}{N_{sf}} + O_{mr} + T_{fl2m})(E - \frac{E}{N_{sf}})^{C_r}) \quad (3.3)$$

where T_{crc} is the time to send a sub-fragment CRC (overhead of sub-fragmentation), N_{sf} is the number of sub-fragments in the original frame, and T_{fl2m} is the transmission time of the meta-data for a single retransmitted fragment (including the length added to the ACK frame). Note that this equation does not correctly estimate for the case of two or more simultaneous sub-fragment failures within a frame, which makes it progressively less accurate (more optimistic) at high error rates, however it should be a reasonable approximation at low error rates as the probability of two errors occurring in the same frame is also proportionally lower.

The equation is structured as a combination of the probabilities of various outcomes. It is split into two top-level sections - the cost if no retransmission is needed (the section multiplied by $(1-E)$), and then the cost if it is necessary (the remaining sections). The last line is separated, respectively, into the costs for a first sub-fragment loss (and therefore total frame retransmission) and losses of any other fragment.

TCP average frame time estimation has a few more changes to consider. Firstly, the MAC function does not receive priority on it's retransmissions, thus bearing the full cost of a normal frame; additionally a second full frame has to be sent (in the opposite direction) for the TCP level ACK.

$$T_{SF-TCP} = (T_{frame} + T_{ack} + O_m + T_{crc}N_{sf})(1-E) + (T_{frame} + T_{ack} + O_m + T_{crc}N_{sf})E + \sum_{1}^{C_r}((T_{frame} + O_{mr})(\frac{E}{N_{sf}})^{C_r} + (\frac{T_{frame}}{N_{sf}} + T_{fack} + 2O_m + T_{ftm})(E - \frac{E}{N_{sf}})^{C_r}) \quad (3.4)$$

where T_{ack} is the time taken to send a TCP ACK frame, T_{ftm} is the added time to send sub-fragment TCP meta-data and T_{fack} is the time taken to send a TCP ACK frame with sub-fragment information.

As with the L2 calculation, this does not deal with multiple simultaneous segment failures; additionally, in TCP the cost of each MAC function used by retransmitted bytes is likely to be shared by other data in the TCP stream (assuming the connection is still actively generating new data), thereby effectively reducing the MAC overheads for retransmissions. This is important to consider, as without keeping these limits in mind

the TCP operation mode has a definite disadvantage. For this thesis, we are relying on the simulation to bear out these more complex results.

3.2.3 MAC Losses

Any given 802.11 implementation utilises one of a number functions to allow nodes to cooperate on the network without prior knowledge of each other; DCF, PCF or HCF, all of which use delays for each frame transmission in order to implicitly negotiate access, and priority of access, to the medium. This process is imperfect and can result in packet collisions when 2 stations unknowingly start transmitting at almost the same moment. Since 802.11 has a propagation delay limit of 2 µs, and even at high data rates this is a small interval compared to the total frame transmission time, it is not likely that substantial data recovery will result in the case of collisions.

As an example, consider 2 stations attempting to transmit at the same moment at 54Mbs. A transmission for a period of 2 µs at this speed permits only 108 bits before a collision must have occurred. In practice, this means that only partial header information can be sent before failure, resulting in total frame loss.

As such, the sub-fragmentation schemes do not affect the probability of any data being recovered from a collision; the results of [9] can be reused in calculating optimal frame sizes.

The algorithm used in [9] for this is as follows:

$$P_{collision} = \frac{1 - (1-r)^N - Nr(1-r)^{N-1}}{1 - (1-r)^N}$$

where $P_{collision}$ is the probability of a collision, N is the number of nodes that might attempt to transmit and r is the probability that a station in the set of N may choose to transmit in a given window. This equation is most relevant to the TCP operation mode, which requires more medium contentions in order to function than when compared to the L2 approach; it is included implicitly in the previous calculations. A good overview of many of the issues involved in MAC function can be found in [39].

In addition to collision losses, there are also the more fixed overheads that are related to MAC and PHY layer operations, with costs that are specific to the headers and time probabilities for how long it will take to successfully access the medium. These costs are largely unaffected by the usage of sub-fragmentation schemes, with possibly the most significant factor to change being the increase in the number of medium accesses in the TCP operation mode.

The equations in section 3.2.2 do rely upon these fixed overheads, and the two operation modes have different MAC mechanisms, so it remains important to discuss the specifics

of MAC overheads. Many of the points specific to each mode of operation are discussed in section 3.3.

3.2.4 Security

When 802.11i based security is deployed, optimal sub-fragment sizes are effected. Without security, the failure block size is equivalent to the physical symbol size; as discussed above this varies from 26 to 2,320 bits in 802.11n. Security encryption operates in 128 bit block sizes, and corruption anywhere within this block will cause the entire block to become unreadable. It will be rare for the highly variable length OFDM symbol edges to align with encryption blocks. The worst case scenario for these misalignments would result in a failure block 254 bits larger than the current OFDM symbol.

The easiest way around these issues is to select sub-fragment sizes that are equal in length to the minimum number of encryption blocks that can contain an OFDM symbol in the current configuration. Such a configuration performs well much of the time, but also makes it possible for a sub-fragment checksum to be lost along with the start of the next fragment, when the end of an encryption block does not line up with the underlying OFDM symbol boundary. Without padding, this will degrade the performance of the sub-fragmentation schemes. Adding padding is possible, but has a moderate probability of excessive overhead; though this is highly dependant on the OFDM symbol size. Extensive testing and calculation is needed to optimise this behaviour.

In order to ensure that there are no security issues introduced by the sub-fragmentation process, it will need to be implemented at a lower layer than encryption. Performing encryption on a payload that already has sub-fragmentation applied may have difficult to foresee consequences, as it would cause different sections of the payload to be predictably related to some other section. This decision does have an impact on the TCP operation mode of sub-fragmentation, as it is dependant upon reading and altering payload information. As a result, 802.11 is modified to handle TCP interaction 'above' the security module, whilst maintaining the 'lower' sub-fragmentation module to perform actual encoding and decoding work, and to pass meta-data regarding frame corruptions around the decryption stage when decoding.

3.3 Modes of Operation

The sub-fragment schemes chiefly operate at L2 of the networking stack, but also include some interaction with Layer 4 (L4), specifically TCP, in one of its operation modes. The cross-layer design opens up a number of possibilities to optimise system behaviour through

the cooperation of otherwise separated functions.

Both sub-fragment schemes involve the division of a MPDU into sub-fragments that are relatively small compared to the overall packet size; however, the MPDU remains a single logical entity and is not split over multiple physical layer transmissions. Each sub-fragment possesses its own check sequence, immediately following the data it verifies.

The following sections present two distinct implementations of the sub-fragment protocol; one that employs cross-layer design and one that does not. In the simulation runs done to validate the protocol, a TCP-interactive version is presented alongside a pure L2 solution. In either case the normal 802.11 retransmission mechanisms are largely circumvented, being replaced with either an alternate L2 scheme, or using TCP's implementation directly. The intention of employing two separate model versions is to demonstrate some of the strengths and weaknesses of CLO with two schemes that are very similar.

3.3.1 Common Properties

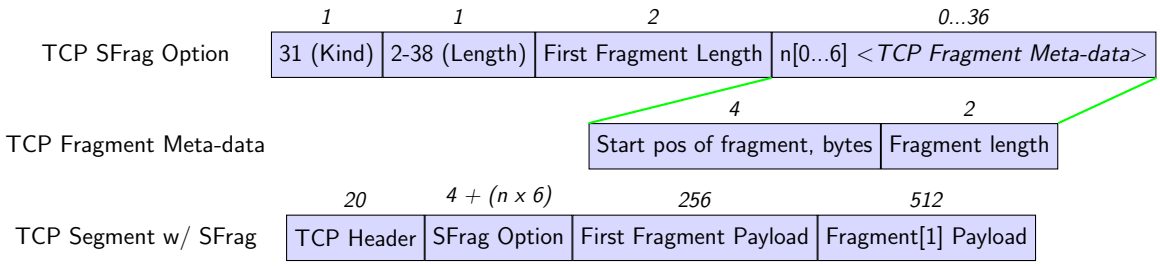
Some behaviours are shared between both modes of operation. For instance, although both schemes largely replace the standard 802.11 retransmission behaviour, they both fall back to standard full-frame retransmissions when the incoming frame is so corrupted that the headers cannot be read.

Both operation modes also include the same L2 module to actually perform the sub-fragmentation process. This module has similar processing requirements in either mode, although its behaviour is different between the two modes.

Perhaps the most important shared property is the one that allows two communicating sub-fragmentation modules to correctly interpret checksums from the other party. The final design involves a standard set of checksum patterns which are based upon the modulation rates and whether security encryption is present. This is to say that each Modulation and Coding Scheme (MCS) rate, with and without encryption, has a predefined size of sub-fragment and checksum to use.

The particular arrangement of sub-fragments defined for each modulation scheme is to be assigned on the basis of the assumed error rates that result in said scheme being selected by the data rate adaptation scheme. A simulation or experimentation could be used to optimise the arrangement.

An 802.11 network node will negotiate the availability of sub-fragmentation upon association with the network; during this process a version number will be specified which will implicitly specify the standard arrangement of sub-fragments to use. Ad-hoc networks present a complication to this design, as there is no association process to perform this negotiation. It would be relatively simple to add an additional control frame that could



Note: field lengths in octets

Figure 3.4: TCP SFrag Option Design

handle this role in association-free networks, however most uses of 802.11 are access-point based (even ad hoc topologies, where mobile nodes communicate with each other, usually rely on one node acting as the access point) and as a result the specification of such a frame is beyond the scope of this thesis. Note that an 802.11 header bit is set to indicate whether or not sub-fragmentation is enabled for a specific frame, separate from broad capability negotiations.

An alternate design that was considered involved the addition of a meta-data field at the start of each frame specifying the following sub-fragment layout. Due to the small number of sub-fragments that are useful in each frame, a single byte would have been enough to specify the number of sub-fragments present, as well as the size of checksums. Although the overhead is acceptably small, this scheme would likely make hardware implementations of the sub-fragmentation schemes more difficult (due to potentially inefficient combinations of MCS rate and sub-fragment sizes); and it would also allow stations to impose higher than necessary processing loads on nodes with which they communicate - downsides that are estimated to be more important than the limitations of having pre-assigned sub-fragment sizes.

3.3.2 TCP Operation

As each MPDU is received by an 802.11 node, each sub-fragment's check sequence is verified. If the sub-fragment that contains the TCP header has survived transmission without error then the L2 sub-fragmentation module inserts information about the validity of each sub-fragment into the TCP header in the SFrag option field (see Figure 3.4). If the TCP header does not survive transmission it is not possible to perform TCP sub-fragment operation (as it is not possible to forward the surviving data), and as a result normal 802.11 retransmission rules apply for any frame that looses the TCP header.

The TCP header's SFrag option field functions by maintaining a list of all good data in the current frame; this list being updated by any L2 sub-fragmentation modules

encountered after receiving a frame. In the event that a sub-fragment is found to be corrupt, two actions are taken:

- The data that was contained within the corrupt sub-fragment is removed from the payload.
- The SFrag option is updated such that it now contains an account of the data that remains in the payload. Typically, this will take the form of a fragment meta-data field being added.

This process leaves the TCP segment in a format that is no longer readable without the SFrag option, as the payload data is now non-contiguous; with pieces of data now adjacent that used to be separated by the corrupted payload. In order to read the payload of a segment using sub-fragmentation, the receiving TCP implementation must go through a particular process:

1. Check for the presence of the SFrag option in the TCP Header. If not found, sub-fragmentation is not in use.
2. Check the first fragment length field of the SFrag option. If it is the same length as the payload data, no corruption has occurred in transit and the payload can be decoded as standard TCP. If corruption has occurred, the number of data chunks within the payload can be determined - this will be between 1 and 7.
3. Extract and separate the data chunks. Each chunk will be within the payload in the same order as that of the fragment meta-data fields, and be of the specified length; thus the first chunk can be extracted by using the expected length from the meta-data and reading that many octets from the start of the payload. Subsequent chunks can be read using the preceding chunk's end as a start point.
4. Using the sequence number in the TCP header, add each chunk into the TCP data buffer at the appropriate position.

After this process, the receiving TCP implementation can determine if it is missing any data (as it would in standard TCP operation) and generate the appropriate Selective ACKnowledgement (SACK) to transmit back to the sender. The SACK mechanism is not affected by sub-fragment operations, though it is a requirement for efficient sub-fragmentation scheme operations. A pictorial example of this whole process, including the frame sent after the sender received the SACK, is included in Figure 3.5.

If the TCP SACK option is not supported by one or both endpoints, the receiving TCP module has no information on the chunks lost and therefore most of the benefits of sub-fragment operations are not possible.

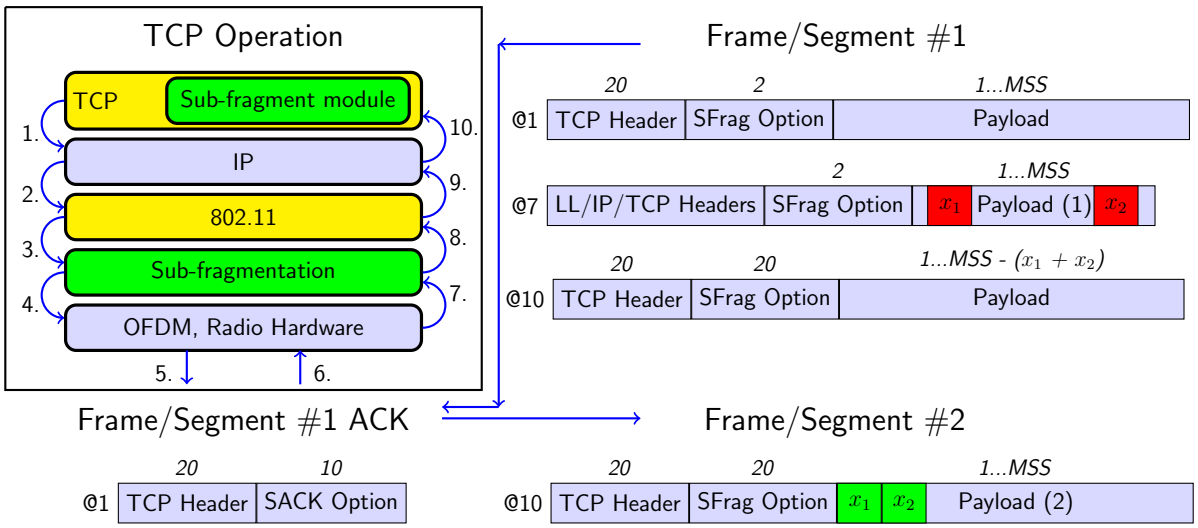


Figure 3.5: Example TCP operation flow

Unfortunately, TCP has a length limit of 320 bits (40 bytes) on the option field of its header; with a per fragment meta-data size of 6 octets, plus a base of 4 octets for the option itself and the first fragment, there is only room in the header for 7 blocks of payload meta-data. In the event that more than 7 non-contiguous data blocks need to be represented, the sub-fragmentation scheme will attempt to retain the maximum amount of valid payload by disposing of the smallest length block between two other blocks first.

The use of TCP sequence numbers within the fragment meta-data guarantees that the payload can be correctly reconciled by the receiver; it does however have the obvious downside of being quite large (4 bytes per meta-data fragment). The usage of a 2 byte offset from the TCP header's sequence number would seem valid as each fragment could still be uniquely and correctly identified, in addition to consuming fewer header option bytes; however it has the major limitation that it is no longer possible to send a single outbound segment that contains fragments separated (within the TCP stream) by more than 65,536 bytes. Considering the overheads involved in sending multiple frames (such as a new L2 frame), it was estimated that the chosen scheme would be more efficient, particularly as fragment loss rates increase; although future research may wish to explore this issue in greater detail.

Pruning the corrupt data out of the payload rather than retaining it ensures that the network's Maximum Transmission Unit (MTU) is not exceeded by the added meta-data (particularly important due to the complications introduced by fragmentation, covered in following sections), and also reduces unnecessary load on the downstream network nodes. It should be noted that it is possible for the packet size to increase in the pruning process, however, for this to happen, it requires minimally small (<4 bytes) sub-fragment payload sizes which are not likely to be selected in real world operation, as it is unlikely that such

a configuration would be beneficial in any real world operation.

The usage of TCP-header option fields for sub-fragment corruption information avoids any requirement to have special cross-layer communications interfaces, and ensures compatibility if the packet happens to be traversing any node that does not understand the sub-fragment protocol, the primary downside being that the option header consumes 4 extra bytes, even in perfect transmission conditions. The absence of the TCP SFrag option indicates to the L2 sub-fragmentation module that the scheme is disabled. A particular strength of the TCP option-based approach is that the TCP sub-fragment traffic can traverse multiple 802.11 networks on its route; each of which can contribute to the SFrag option independently, which is to say that L2 modules from multiple hosts can simultaneously cooperate with the receiver's TCP without any pre-coordination.

It would have been possible to specify alternate CLO architectures; for example, the L2 sub-fragmentation module could, instead of modifying a TCP header option, directly send the corruption information up to the TCP module via some side-channel. This has the benefit of removing the 6 fragment limit caused by the TCP header's option field length, but would cause a major limitation: the scheme would only function within a single node - leaving intermediate nodes unable to perform TCP sub-fragmentation, instead having to fall back to standard retransmissions.

There is a special case of behaviour with the SFrag option which requires further clarification. If the sender wishes to enable sub-fragmentation in the network path, they add the SFrag option to their TCP header. The only meta-data necessary to add is the payload length into the first fragment length field, representing the single unbroken data chunk in the segment at the time of sending. This behaviour reduces the overhead of the protocol slightly (4 fewer bytes per outbound segment), and more importantly impacts less on the MTU/Maximum Segment Size (MSS) available for payload.

It is anticipated that the TCP version of sub-fragmentation will have higher end-to-end latency when compared to L2 only sub-fragmentation, while increasing throughput. This is due to the fact that L2 only operations have a potentially much shorter delay between nodes that need to retransmit corrupted data, along with high MAC priority. This is not desirable for all traffic and due to the way that sub-fragmentation is enabled and disabled in the network path, it is possible for the sending node to dynamically prioritise latency over the throughput provided by TCP sub-fragmentation simply by including or excluding the SFrag option in its header. Interestingly, disabling TCP sub-fragmentation may trigger L2 only operation, should the node support both modes. This is likely optimal, as L2 only operation, as will be shown in the simulation chapter, typically reduces 802.11 delivery latency if retransmissions occur.

L2 only operation does not have a similar option to enable TCP sub-fragmentation;

which provides an example of CLO allowing more intelligent network behaviour.

Limitations

The chief limitation of the TCP operation mode is that the TCP header's option field is limited in length to 320 bits. The implications of which have been discussed.

The L2 sub-fragmentation module component of TCP operation also gains a requirement to parse and decode headers all the way up to and including the TCP header. This not only increases the complexity of implementation, but presents a potentially significant run-time load onto the receiving node, and any intermediate nodes participating.

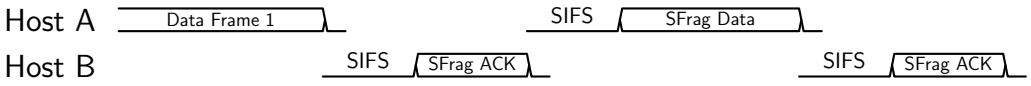
The need to decode and understand headers also makes the TCP scheme's operation quasi-incompatible with any higher layer fragmentation. The scheme still functions, however it only operates on the first fragment in the sequence (i.e. the one that contains the TCP header). This means that only a fraction of the payload is protected; however, in most real world scenarios, the TCP MSS is set at a level that avoids IP fragmentation; and transit networks that operate at much smaller MTU values (such as MPLS) will reassemble the packet in full before crossing back into an 802.11 network. Additionally, IPv6 disallows fragmentation; thus it is not expected that this limitation will cause an issue from a practical perspective.

Expanding upon the explanation of operation in the presence of IP fragmentation; because the first fragment length field lists the full payload length of all IP fragments, a sub-fragment meta-data field can be added to cover all of the payload that is not directly visible to the fragment with the TCP header. As subsequent fragments are not processed by the sub-fragmentation module, they will be subject to standard (or L2 only) retransmissions.

A further major issue with the TCP scheme is that transmission faults are corrected no faster than the round trip time of the total end-to-end connection, rather than the link layer delay. This may lead to noticeably increased end-user latency in long distance connections, when compared to L2 only operations. Ideally, the increase in available throughput will allow applications to intelligently optimise between normal operation and lower latency, or better throughput, by selectively enabling the TCP sub-fragment scheme as discussed previously.

An interesting way to mitigate latency related issues may arise from the fact that the first sub-fragment must always survive in order for any data to reach the destination TCP, and any such first-fragment will likely contain at least the first few bytes of the payload. The result of this is that the TCP connection will always advance slowly through the data stream immediately upon the reception of every segment (assuming earlier segments in the stream have already been successfully received).

(a) Normal transaction, with corruption in data frame



(b) ACK Loss

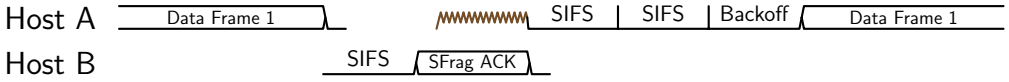
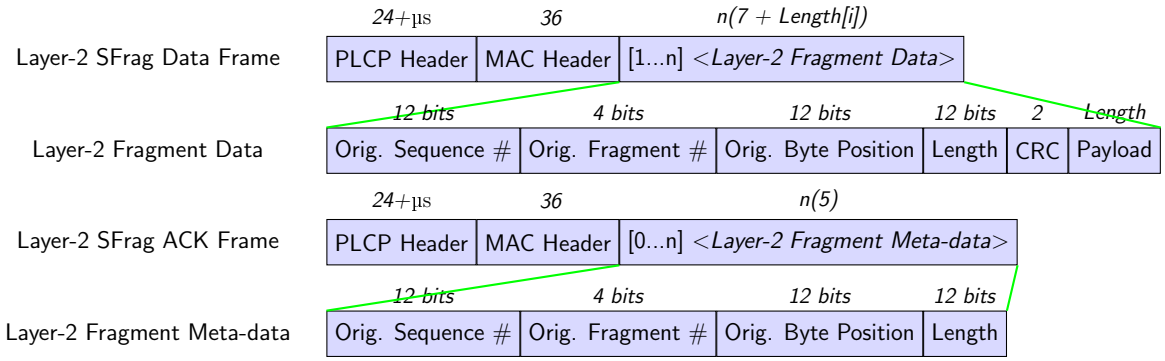


Figure 3.6: Example L2 SFrag retransmission timings



Note: field lengths in octets, n is the number of fragment meta-data fields

Figure 3.7: Layer-2 SFrag Design

3.3.3 Layer 2 Only Operation

The L2 sub-fragmentation scheme is a result of modified 802.11 operations, independent of all other layers. It allows 802.11 to limit the extent of both data corruptions, and the length of retransmitted frames.

Protocol Description

The L2 version decodes sub-fragment check sequences and, should any sub-fragment be corrupt, returns this information in a special control message (an intermediate SFrag ACK), in place of the standard L2 ACK (or lack thereof). The transmitter then re-sends, using Short Inter Frame Spacing (SIFS) priority (see Figure 3.6), the corrupted fragment(s) based upon this information. It does this by generating a new MPDU, using a different (to be assigned) protocol ID in the SubNetwork Access Protocol (SNAP) header and a payload that conveys the previously corrupted sub-fragments, along with meta-data (See Figure 3.8 for a high level visual representation of this process).

The meta-data within both SFrag ACK and data frames is composed of a number of fields: the original 802.11 sequence number, the 802.11 fragment number, a byte offset representing the start point of the data in the original frame and data chunk's length.

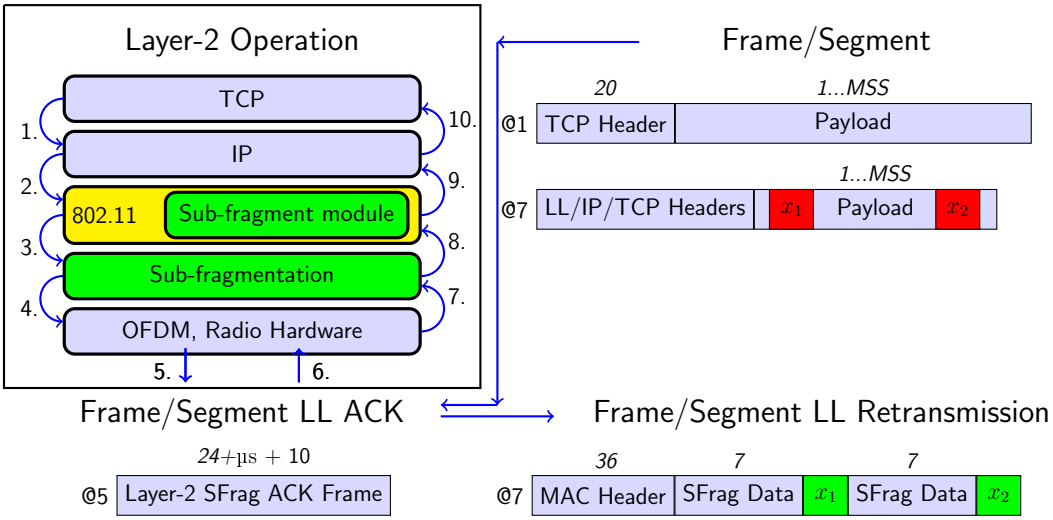


Figure 3.8: Example L2 operation flow

This set of data guarantees that each sub-fragment that is retransmitted can be reconciled and inserted into the previously received, partially corrupted, frame. Figure 3.7 shows the format that both the SFrag ACK and data frames use.

In poor channel conditions, it is possible that any given fragment may undergo a substantial number of repeats, and thus experience a high latency. In order to avoid this becoming a problem to higher layers, the receiver will stop reporting the corruption of sub-fragments on frames that have been awaiting reconstruction for a certain number of repetitions, and then discard the frame; much in the same way as standard 802.11 retransmissions.

L2 operation is not sensitive to the payload at all, permitting any protocol to be carried; this along with low latency are the two primary benefits to operation at this level over TCP. That being said, the first sub-fragment is still required to be received successfully, as it includes the MAC header, whose absence results in no L2 module attempting to read the frame at all.

The latency improvements over TCP operation are two-fold; firstly, the delay is only hop-to-hop instead of end-to-end, and more importantly, the 802.11 high-priority mechanism can be used - meaning that the SIFS can be employed after receiving the intermediate ACK from the receiver specifying which segments to resend.

L2 sub-fragmentation also makes a potentially substantial reduction in retransmission latency possible even when compared to standard behaviour. In 802.11, a back-off mechanism with an exponentially increasing contention window is used; creating a potentially quite large gap between a transmission failure and its repeat copy. This behaviour is sensible because a transmission failure does not result in an ACK from the receiver, and there are possible causes of a missing ACK besides a transmission failure (such as a node simply

not being within range of the transmitter) that would make using high-priority retransmission inefficient, should any other nodes wish to use the medium. Sub-fragmentation on the other hand, will generate an ACK when retransmitted data needs to be sent (with the exception of total frame loss), and reception of this intermediate ACK makes immediate high-priority retransmission reasonable, as it guarantees the presence of the target node within the transmitter's range.

When the receiving node sends information back to the sender about which sub-fragments were corrupted (as part of the intermediate SFrag ACK) those nodes that do not understand the intermediate ACK may decide that a contention event has occurred and initiate the appropriate back-off procedure. It is hoped that due to the use of SIFS this will largely be avoided, however it is difficult to predict the interpretation of the various implementations of 802.11. The Network Allocation Vector (NAV) is also affected by this issue, as nodes that do not understand sub-fragmentation will not be able to read the new NAV sent in the intermediate ACK. This is discussed further in the following section.

MSDU aggregation, along with block acknowledgement, (as introduced in 802.11n) provides the basis for a substantial improvement over standard performance when using L2 only sub-fragmentation. As an example, take the case where 10 MSDUs have been aggregated together in one outbound burst. If 10% of the total number of sub-fragments sent in this burst are corrupted, it is always possible to resend every byte of that lost 10% in a single retransmitted MSDU. Compared to standard performance, where a single MSDU retransmission is the best case scenario and retransmission of the entire burst is the worst case, substantial gains can be achieved.

Limitations

Unlike the TCP operation mode, it is not possible for a retransmitted frame to contain data from multiple MPDUs (with the exception of occasions where MPDU aggregation is employed). This is due to the fact that L2 retransmissions occur on a frame-by-frame basis in order to exploit the priority inter frame spacing that is available. Ultimately this increases the overhead per retransmission compared to TCP operation, though it is still less data to resend overall than in the original 802.11 standard.

802.11 uses the NAV mechanism to inform other nodes about the amount of time that the medium will be busy for the upcoming transmission. Unfortunately, this NAV includes an estimated time for the ACK frame. Since the sending node cannot know how much of the frame is corrupted ahead of time, it is no longer possible to know the length of the ACK frame ahead of time; the uncertainty is very small however, the ACK frame being only 7 bytes longer per corrupt fragment. The sending node could also choose to

increase the NAV value to account for the maximum possible length of the modified ACK, although this is simply another form of overhead and may not be necessary.

Since a new NAV value will be transmitted with the intermediate ACK, those hosts who understand sub-fragmentation can back-off until the next possible window. Unfortunately, this raises another issue for hosts that cannot understand the intermediate ACK. A possible solution is for the receiver to send a Clear to Send (CTS) message, indicating to other nodes that the sender has been cleared to transmit for the length of a new NAV; the obvious limitation being that the original sender did not send a corresponding Ready to Send (RTS) message, resulting in nodes hidden to the receiver not being notified correctly. As with other 802.11 transitional techniques, this could be a configurable option for network administrators until such a time that all nodes are known compliant.

3.4 Model Summary

This chapter has presented 2 separate possible implementations of the sub-fragmentation mechanism; however it has left out a number of details that would be required in a full specification. Primarily, what remains to be specified are the behaviours and exact format of data fields that would be used to negotiate sub-fragmentation on a 802.11 network segment, for both infrastructure and ad-hoc modes. While a design for these requirements has been constructed, it has been excluded and instead discussed at a high level for the sake of brevity; and as such detail lacks theoretical complexity.

In addition, the specification of methods for sub-fragmentation to interact with data rate adaptation schemes that are currently commonly deployed needs to be completed. Such a specification requires more research and testing, yet to be conducted. As a result, detailed discussion about such interactions has been excluded, however the performance of either sub-fragmentation scheme is highly dependant upon data rate adaptation scheme being designed in cooperation with sub-fragmentation.

The data exchange phases of operation have been covered for both sub-fragmentation schemes, and the implications of interaction with existing network infrastructure have been discussed, and a basic predictive model was built for testing during simulation. The possible limitations and barriers for real world deployment have also been discussed; and fortunately there are very few aspects of the design which are theoretical blockers to include in the 802.11 or TCP standards.

Chapter 4

Simulation

The development of the simulation of the sub-fragmentation schemes turned out to be quite challenging, as the original simulation platform would have had to be extensively modified to permit partial frame corruption in a correct manner. The result of this is a custom developed simulator platform for 802.11n and the sub-fragmentation schemes.

The process of simulator development and the choices made about which aspects of the 802.11 specification to include are covered in the first section. The results of the simulation and an analysis of the data are covered in the second section.

4.1 Development

4.1.1 OMNET

Development of the simulation originally started with the OMNET++ simulator and the INET framework. These are proven simulators with extensive coverage of 802.11 and numerous options for simulating physical environments.

Unfortunately, the code behind the OMNET model proved to be very difficult to modify in such a way as to be compatible with the sub-fragmentation schemes. In particular, INET's TCP model operates on the basis of frames held in memory and retransmitted in whole. Since the TCP based sub-fragmentation scheme adds the requirement to be able to handle retransmissions that are of arbitrary block sizes, unrelated to the frames that originally sent them, it would have been extremely time consuming to retro-engineer this feature into OMNET++ or INET.

This fundamental incompatibility is particularly important, as it is quite possible that many real-world TCP implementations depend upon a similar architecture. This implies that in many cases it may be unacceptably difficult to implement the TCP variant of the sub-fragmentation scheme.

4.1.2 Custom C# Simulator

After failing to produce an OMNET++ simulation in reasonable time, the decision was taken to start producing a new simulation platform from the ground up, primarily to avoid the complexities of INET interfaces in development.

The new physical environment model is one that does not implement Additive White Guassian Noise (AWGN) or path fading; instead it primarily deals with the propagation delays induced by the distance between nodes, and produces SERs based upon a user selectable parameter. This decision allowed for precise selection of SERs, easing the production of results based on corruption induced Packet Error Rate (PER), which is one of the two primary causes of retransmissions (along with MAC related collisions).

The downside of this limited environmental model is that there are potentially subtleties missed in the interaction between hosts and some types of interference that will not be accounted for in the final results. That being said, these parameters are typically highly variable when present, and so data is likely more generally applicable without them.

A specific example of the issues potentially caused by not accounting for environmental reflections is ISI. Due to the usage of 802.11n, and therefore OFDM, this effect is minimised but potentially still significant. In order to ensure that the influence of ISI is as minimal as possible, the long OFDM guard interval is employed in the simulation exclusively. A likely consequence of ISI would seem to be that corrupted symbols would occur in adjacent groupings, where the simulation corruptions do not. An additional consequence of this is that neither the 802.11 or sub-fragmentation mode results presented below are as fast as they could be (if short guard interval were allowed), however both are considered with the same environmental variables.

The 802.11 implementation of the model features an OFDM module that emulates the sending rates of various MCSs, as well as frame collision and corruption. Additionally, A-MPDU operation is fully supported, including retransmission of individual MPDUs within an A-MPDU. The back-off mechanism in use is purely DCF, not supporting any QOS operations. PCF operation, as used in infrastructure networks by the access point, may have resulted in a slightly reduced MAC function costs, but likely would have been a negligible benefit. MCS, and error rate, reduction on retransmission is supported.

The TCP implementation supports SACK, as it is required for effective TCP sub-fragment operation. Additionally, congestion and flow control based on TCP Reno, including slow-start and fast retransmit are implemented. Explicit congestion notification is not supported, along with urgent pointers.

Traffic Generation

The traffic generation process used for loading the simulated network had a few mechanisms built in to ensure that the simulator was performing correctly. Each simulated application generated outbound data by inserting an incrementing 4 octet counter 4 times in a row, and then inserting two sets of special values and a time stamp of the time that the particular frame was submitted to the transport protocol. This permitted the calculation of the total time between transmission and reception, as well as ensuring that the data received was correctly reassembled by any intermediate layers that fix transmission errors.

Each transport protocol had a limit on the amount of pending data that they would accept. When generating data, the application checks if these queues are full, and if not, generates data at an unlimited data rate to keep the queue full.

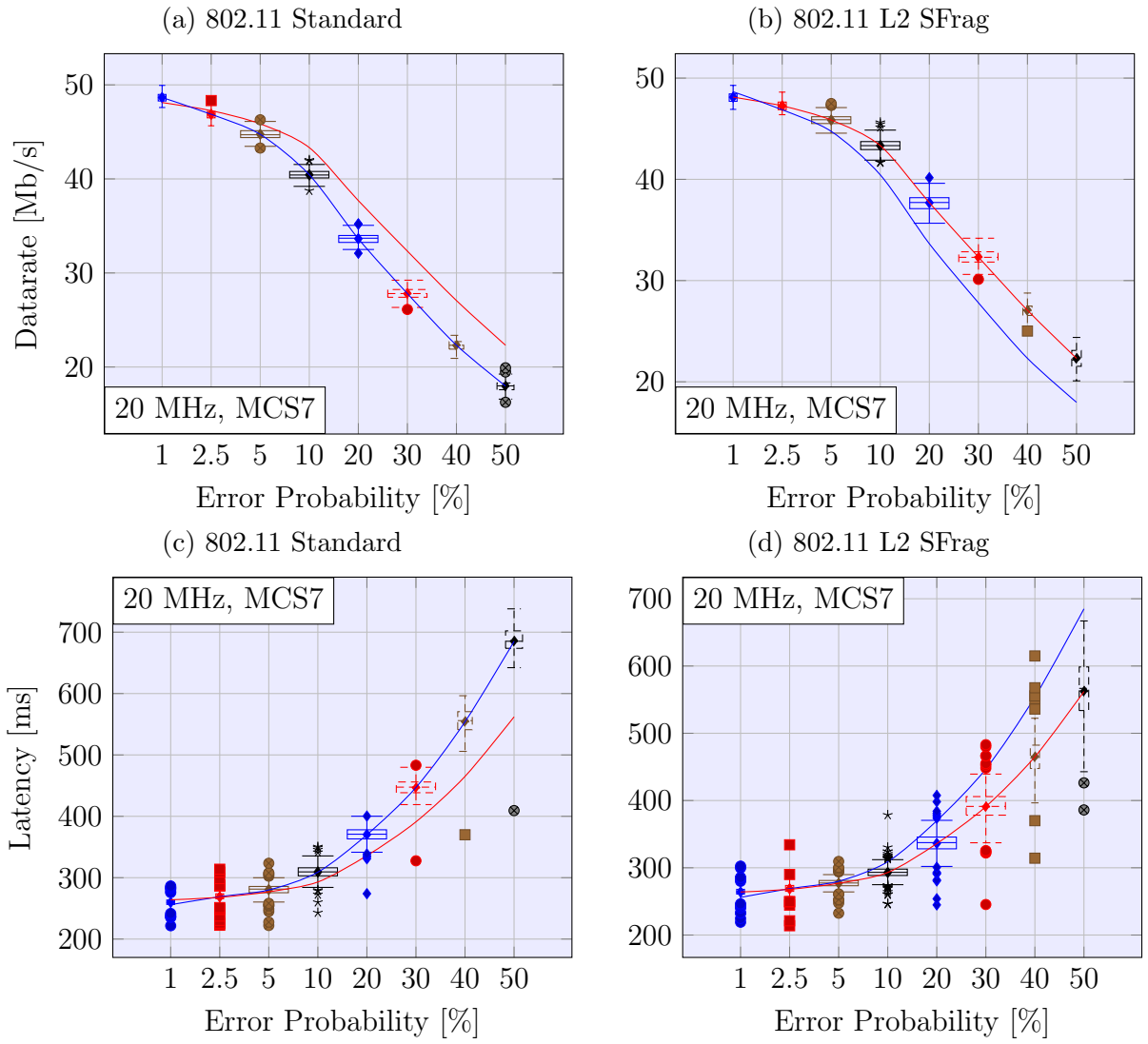
4.2 Results and Analysis

These results have been produced by an ultimately unverified simulator and may be inaccurate if contextualised against results from other implementations. This would be true to some extent of any simulator, due to the internal alterations made to 802.11 and TCP during sub-fragmentation implementation. The results must be interpreted by keeping this limitation in mind, however the results presented below should be at the least an indication of what might be possible with real world implementations.

Results were collected by running each scheme configuration at a variety of frame error probabilities (between 1% and 50%), using random frame sizes between 1000 and 1500 bytes (not including TCP control frames, which are simulated). Sub-fragmentation operations were run back-to-back with 802.11 standard operations. Typically, 5 hosts constantly sending and receiving traffic to and from a 6th host at random distances from each other comprised the basic network configuration. Most simulation runs were done with application data rates sufficient to saturate the network, however some runs were done with lower load levels. Saturating the network ensures that data rate is a good metric to measure the effectiveness of each mode of operation.

4.2.1 Layer 2 Sub-Fragmentation

For L2 operation, application traffic load was generated by a fixed length data queue which was continuously filled as data was removed and processed by 802.11. UDP was the transport protocol in use. This ensured that TCP did not influence L2 SFrag results with congestion and flow limiting mechanisms. The latency presented below is a function of the length of this queue, since it is measuring the time between submission to the outbound



Note: error rates are the chance of a 1000 byte frame having 1 symbol error

Figure 4.1: Data rates and latencies achieved with UDP on a saturated network, with 6 hosts and 128 byte sub-fragments, comparing L2 sub-fragmentation to 802.11 standard behaviour. Blue lines represent 802.11 and red lines represent L2 SFrag, with average-value lines copied to the alternate scheme's graph for easy comparison.

queue, and reception at the target node. It is particularly important to note that traffic was not checked to be in-order on reception.

The primary results of L2 sub-fragmentation operation are illustrated in Figure 4.1. It can be seen that an increase in bandwidth, and a decrease in end-to-end latency, were observed at all error probabilities exceeding approximately 2%. The increase in bandwidth from the cross-over point with 802.11 standard operation is approximately linear with respect to increasing error rates.

This is in contrast to the predictions of the algorithms presented in section 3.2.2, which had anticipated a pseudo-exponential increase in the gap between 802.11 and L2 sub-fragmentation. This is likely attributable to the fact that at high error rates, multiple corruptions per frame decrease the efficiency of sub-fragmentation.

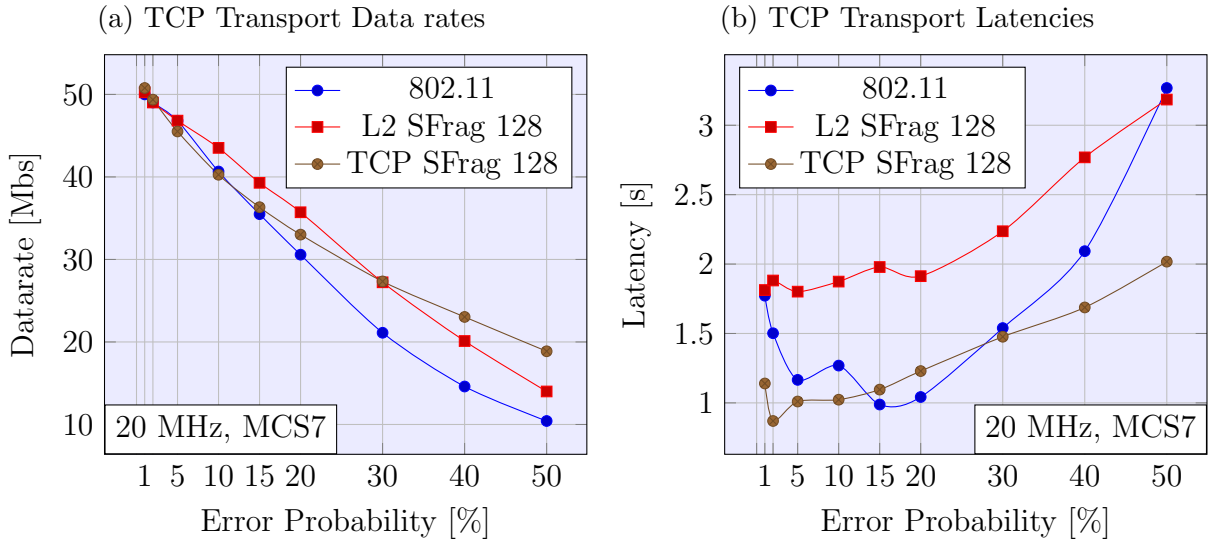
The lowered efficiency is caused by two factors; firstly, the total amount of data retransmitted more closely approaches 802.11 standard behaviour as the number of fragments requiring retransmission increase. Secondly, as error rates increase, the odds of corruption or loss of the intermediate ACK and Data frames increases proportionately, and if these are lost the entire frame is resent. The scheme still achieves better performance when compared to the 802.11 standard because the odds of having to send multiple full copies of the frame also increase with error rates.

The decrease in latency was greater than anticipated, and is most likely a result of the fact that no DCF back-off is required when retransmitting the lost fragments in a sub-fragmented frame. The increase in latency over standard operation at very low error rates is most likely attributable to the added bytes transmitted in checksums causing less application data to flow per time, thus increasing the service times of the fixed length data queue used to submit data to the network.

An interesting effect observed during these data runs is that there was a substantial reduction in the number of frame drops due to retransmission limits when sub-fragmentation was in operation. This is most likely caused by 2 factors. Firstly, sub-fragmentation does not participate in DCF contention for retransmissions and is therefore not affected by collisions; and secondly, the smaller amount of data sent on retransmission frames reduced the odds that retransmissions would be corrupted.

4.2.2 TCP Sub-Fragmentation

In contrast to the L2 only operation mode of sub-fragmentation, UDP could clearly not be used as the transport protocol. The 802.11 standard and L2 only data runs were therefore re-run for comparison purposes with TCP in use as the transport protocol. When comparing Figures 4.2 and 4.1, it can be seen that TCP, when compared to UDP, as the transport protocol generally achieves higher data rates at lower error probabilities



Note: error rates are the chance of a 1000 byte frame having 1 symbol error

Figure 4.2: Data rates achieved with TCP on a saturated network with 6 hosts and 128 byte sub-fragments, using varying retransmission schemes

and lower data rates at the higher error probabilities.

Figure 4.2(a) shows a comparison between the application data rates accomplished. Broadly speaking it can be seen that all three schemes perform very similarly at very low error probabilities; whereas L2 only sub-fragmentation tends to perform best of the three in moderate to high error rates, and finally TCP sub-fragmentation performs best at very high error rates.

The TCP cross-layer scheme's relatively weak data rate performance at realistic (5-20%) error rates is most likely attributable to the considerable overhead involved in both the SFrag and SACK options added within the TCP session's frames; however there is a substantial benefit to latency made possible by the rapid retransmission of corrupted fragments, as can be seen in Figure 4.2(b). It can be seen that 802.11 latencies in particular are quite unstable in this simulation, and L2 SFrag latencies are the worst of the 3 schemes, in contrast to performance when using UDP as the transport protocol.

Latency is possibly the most interesting result of this data set; it shows that when in-order data delivery to the application is guaranteed (by TCP in this case), the L2 SFrag scheme actually performs quite poorly. This is attributable to the way that AMPDU's are implemented in the simulator; L2 SFrag does not release any MPDU from inside an AMPDU to TCP until all of them are received without corruption. This behaviour ensures that the separators between MPDUs can also be checked by sub-fragmentation, but does increase latency when only one MPDU is corrupted in an AMPDU. An alternate method of implementation whereby the separators are only checked by their own checksums would likely improve this result considerably, on the basis that this is the technique used during

TCP sub-fragmentation.

An alternate explanation of the L2 SFrag's high latency is that the relatively low likelihood for frame loss is confusing TCP's congestion control mechanism, causing the outbound queue to grow large, and the latency along with it.

The highly variable result of 802.11 is harder to explain; most likely this is attributable to the fact that in 802.11 standard mode, the simulator forwards individual MPDUs from within an AMPDU as soon as its checksum is verified, regardless of whether other MPDUs from the AMPDU need to be retransmitted. This results in a highly variable delivery delay for any frame participating in an AMPDU which, when the network is saturated, includes most frames. At higher error probabilities this effect is lessened, as it becomes quite unlikely that all frames in an AMPDU survive un-corrupted.

TCP SFrag's low latency result is primarily attributable to the way that TCP generates SACK fields. When the TCP module is using SFrag, it can identify gaps in the incoming frames that are equal to only a few sub-fragments in size. When this occurs, it can know for certain that these fragments are not merely delayed in delivery (as would normally be the case in out-of-order data reception for TCP), and can immediately generate an ACK requesting the lost chunks. Interestingly, this also has the side effect of decreasing the data rate accomplished by TCP SFrag, since bandwidth is consumed in the ACK frames generated.

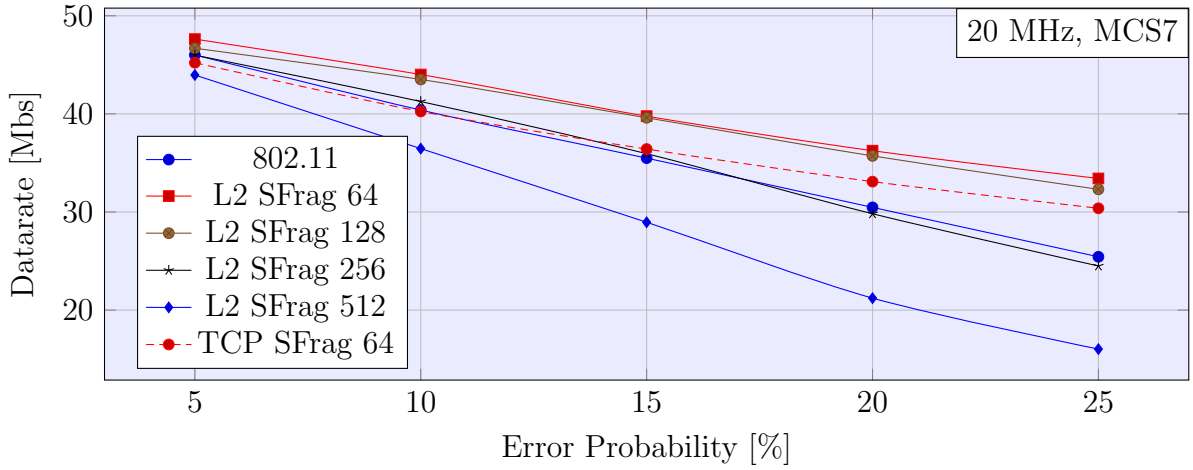
It is expected that TCP's performance, using either L2 only SFrag or with the cross-layer scheme, could be tuned to accomplish different effects, for example by delaying the generation of ACK frames on corrupt fragment reception, reducing the effective overhead per acknowledged corrupt sub-fragment. TCP tuning is a complex subject and there are a great many factors that interact with each other. No in-depth attempt was made at tuning the TCP implementation used in this simulation, with all the different schemes operating with the same logic for congestion and flow control.

Finally, the TCP sub-fragmentation scheme would likely operate substantially more efficiently if the size limit of the TCP header's option field were expanded. The 40 byte limit is a serious issue when both SACK and SFrag options are included in a segment. Given the reserved bits available within the TCP header, it should be possible to specify a higher TCP header size limit without substantial protocol modifications.

4.2.3 Comparison of Sub-Fragment chunk sizes

In order to understand the impact of sub-fragment block sizes, it is instructive to compare between the various schemes at variable block sizes. Figure 4.3 provides a graph of test runs with sub-fragment sizes between 64 to 512 octets, and error probabilities in a range of a 5% to 25% chance of a single symbol error within a 1000 octet frame.

(a) Data rates of different schemes at variable SFrag block sizes



(b) Data rates of different schemes at variable SFrag block sizes

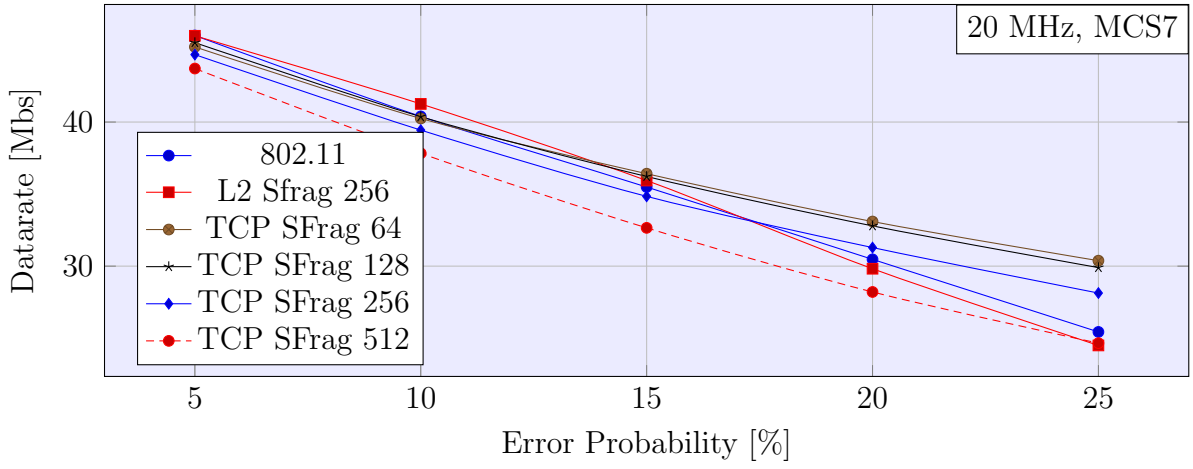


Figure 4.3: Data rates achieved with TCP on a saturated network with 6 hosts, using varying retransmission schemes

This translates to a per symbol error probability range of 0.0016 to 0.0081. The test runs behind this graph were simulated at MCS7 and with a 20MHz channel bandwidth resulting in an OFDM symbol size of 32.5 octets. The probability of a given sub-fragment having an error can be calculated quite simply with the following:

$$E_{SFrag} = \frac{L_{SFrag}}{L_{Symbol}} E_{Symbol}$$

Where E_{SFrag} is the probability of a sub-fragment being corrupted, and L_{SFrag} is the length of the sub-fragment payload. From this it can be seen that the error probability for sub-fragments increases with length, in exactly the same fashion as it does for frames when using 802.11 standard behaviour.

Since any corruption within a sub-fragment results in the entire sub-fragment being discarded, maximising the amount of data received without corruption or retransmission is best accomplished by matching the OFDM symbol size to SFrag lengths. Despite the overheads involved in retransmission mechanisms, and SFrag checksums, the results in Figure 4.3 strongly support this fact.

For both L2 only and TCP operation, the 64 byte SFrag payload size is unconditionally superior to any other operational mode, when only considering the accomplished application data rates. For TCP operation, the rule that smaller sub-fragment sizes result in greater performance also applies, to the point that any sub-fragmentation scheme should not consider operation with sub-fragment sizes larger than 128 bytes; though at very low error rates this may no longer hold true.

4.2.4 Model Accuracy

For L2 only operation, starting at a sub-fragment size of 6 bytes, and with an application payload size of 1460 bytes, each outbound frame contains around 23 sub-fragments. Equation 3.3 (and Figure 3.2, referred to throughout this section) predicted an approximately 4% performance improvement over standard at the 5% error rate with this many sub-fragments, and the simulation has produced a very slightly lower figure of 3.51%.

As the error rates climb, it was expected that the model's predictions would become progressively less accurate (more optimistic), primarily as the possibility of multiple corruptions per frame increased; however, this did not occur. At a probability of 25% for a single symbol error per frame and with around 20 sub-fragments, the model predicted an improvement of approximately 27%, whilst the simulation produced an improvement of 32.4%. Due to the fact that MAC function efficiency was not considered in the predictive equations, this area would seem to be the most likely area for the unexpected improvements, with significant gains seemingly observed; more than compensating for the losses expected

from the added meta-data of multiple fragments being corrupted.

By contrast, frames sent with very low sub-fragment counts dropped well below predicted performance. For instance at 3 sub-fragments per frame, and at a 10% single symbol error probability, the model had predicted performance to remain approximately the same as 802.11 standard. The simulation however produced a result that showed L2 sub-fragmentation being around 10% slower than standard.

This difference can most likely be explained by the fact that the model equations do not accurately account for the probability of an SFrag ACK, or SFrag data frame, being lost; and the fact that such a loss causes a total frame retransmission. Additionally, at large sub-fragment sizes, the MAC function necessarily consumes a smaller proportion of the total air-time of a transmission when compared with small sub-fragments. This second factor is implicitly accounted for within the MAC cost terms of the equations, but those fields were not given values during the original estimation process.

Improvements for TCP operation are harder to state categorically, as the calculations are more complex and lie to a large extent in the MAC function, which the simulator was designed to account for, but which the equations used in initial model projections could not include. Generally speaking, the TCP sub-fragmentation scheme, as simulated, does not perform as well as the L2 only scheme. Equation 3.4 might be seen to imply this due to the additional overhead terms it uses compared to Equation 3.3, although such a high level comparison is likely simplistic.

4.3 Comparison of Schemes

During the implementation of the simulation test bed for both sub-fragmentation schemes, the difficulties in implementing CLO techniques became quite apparent, in more than one aspect.

In order to implement the L2 only sub-fragmentation scheme, only the 802.11 protocol suite required modification and the coding process went quite smoothly, including testing, as the side effects of the changes being made had to just consider effects within a single module.

While implementing the CLO-based TCP sub-fragmentation scheme, both 802.11 and TCP needed to be modified. Setting aside that this particular scheme required an internal data structure that is atypical for a TCP implementation, the coding phase was considerably more complex than for L2 only, simply because there was more internal logic affected by the code changes.

On the other hand, the very complexity that made TCP sub-fragmentation difficult to implement also opened up a great many more choices for optimisation through tweak-

ing, or future protocol updates. It is this property that, given extra time for research, will most likely make the CLO based sub-fragmentation scheme perform better than L2 sub-fragmentation - especially under dynamic conditions. Giving TCP the opportunity to dynamically tune its own operating parameters, and selectively engage L2 sub-fragmentation could be very powerful and permit no drawback implementation (with the exception of processing overhead) of the sub-fragmentation schemes.

Without speculation as to future possibilities, it can be seen in the above results that CLO and non-CLO mechanisms can behave quite differently even when the underlying methodology of the two mechanisms is fundamentally similar. Comparisons between such similar schemes in this case has provided an insight into different designs used for the same purpose (i.e. reliable data delivery over a network segment, for TCP and 802.11).

4.4 Results Summary

Within this chapter we explored the results of simulations runs of both sub-fragmentation schemes. In the process some unexpected behaviour was noted; however the results are broadly in line with the model's predictions for L2 operations. Over a variety of transmission circumstances, a number of sub-fragmentation modes were found to not be useful; typically those with sub-fragment sizes in excess of 256 bytes, which were inferior to 802.11 standard at all realistic error rates. Sub-fragment sizes of less than 256 bytes however, frequently performed better than 802.11 standard, from very small percentage gains to nearly 100% performance increases in some cases.

Additionally, there was discussion about the design of the simulator itself, and of why some decisions were made regarding its implementation. A brief description of the simulator and what is implemented within it was also included.

We attempted to justify the behaviour of the TCP sub-fragmentation scheme (which had unfortunately poor performance at lower error rates), and provide insight into how future alterations might improve its performance further. This tied in to a discussion of the benefits and disadvantages of CLO.

Chapter 5

Conclusion

5.1 Future Works

An operating mode that is possible, but that was not implemented, is the cooperation of L2 and TCP sub-fragmentation schemes. Currently, TCP sub-fragmentation blocks any L2 retransmission that is not a total frame loss. It would be possible to employ the L2 sub-fragmentation's retransmission of individual corrupted blocks, but disable total frame retransmission in the event that an L2 intermediate ACK or SFrag data frame was lost. This would prevent all full frame retransmissions, and therefore the additional latency of the 802.11 back-off mechanism, whilst using TCP to ensure that all corrupt fragments are eventually delivered. Such a scheme would be worthy of further investigation.

The tuning of TCP to accommodate the new behaviour caused by using either of the two sub-fragmentation schemes would also be a viable area of future research. Queue lengths, retransmission timers, and the size of the TCP option field length are all areas that could substantially impact the efficiency of sub-fragmentation.

In a similar vein, tuning of 802.11 data rate adaptation based upon the new information made available by either sub-fragmentation scheme is an interesting area of future research. In general, there exists the potential for an improvement in both TCP and 802.11 performance if the cause of transmission failures can be more accurately determined.

5.2 Closing Remarks

While a number of methods similar to the 2 sub-fragmentation schemes have been proposed in previous works (see Chapter 2), they have generally produced relatively small benefits [20], have potentially prohibitive processing requirements [31], or have implementation requirements that are expensive [21].

With bandwidth improvements varying from negligible to as much as 50% , it can be

seen that the sub-fragmentation schemes, and CLO, have much potential to be exploited; however, it is also apparent that successfully achieving performance enhancement in the real world will depend upon correctly configuring the network to each circumstance. The results show that L2 sub-fragmentation is generally better at medium (5 to 20%) error rates, and TCP sub-fragmentation is faster above these rates. Neither scheme is inferior to 802.11n standard behaviour until very low error rates.b

Real world implementations of L2 sub-fragmentation are likely to not be prohibitively difficult to create and should work in a backwards compatible way with 802.11 nodes that do not understand it, assuming appropriate capability negotiations exist within a BSS. Additionally, neither sub-fragmentation scheme requires any special hardware as they employ only mathematical operations already in use in 802.11 hardware. L2 only operation should therefore be practical to include within the 802.11 standard.

TCP sub-fragmentation is more difficult in the real world, as it depends upon at least 4 separate nodes to support it in the communications path to be useful; those being the TCP source and destination nodes, as well as at least one intermediate set of 802.11 nodes. When combined with both the tuning required to accomplish significant gains over L2 only operation, and considering the changes required of internal data structures within existing TCP implementations, real world deployment of TCP sub-fragmentation will depend upon future research proving a high degree of further performance improvements made possible in CLO operation.

These two outcomes are good representations of the benefits and limitations of CLO; with the CLO having more potential for tuning and performance gains in the long term at the expense of implementation complexity, and the non-CLO scheme being more easily implementable but less versatile.

This thesis has contributed to the study of wireless data networks and CLO through the original specification and testing of the 2 sub-fragmentation schemes. For wireless networking, these schemes provide a further avenue of investigation for the improvement of spectral efficiency. For CLO, the comparison of the 2 different approaches to sub-fragmentation has provided further points of discussion for future research. This is made particularly useful due to the unique nature of 2 schemes that are directly comparable in function, despite the varying presence of CLO techniques. Finally, the basis for a possible future extension to 802.11 is laid out with sub-fragmentation; with further work and study, real world benefits could be derived from the results of this thesis in relatively short order.

Chapter 6

Bibliography

[1] IEEE. Ieee standard for information technology–telecommunications and information exchange between systems local and metropolitan area networks–specific requirements - part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications, 2016.

[2] Fethi Filali. Dynamic and efficient tuning of ieee 802.11 for multimedia applications. In *Personal, Indoor and Mobile Radio Communications, 2004. PIMRC 2004. 15th IEEE International Symposium on*, volume 2, pages 910–914. IEEE, 2004. ISBN 0780385233.

[3] D. P. Pezaros and L. Mathy. Explicit application-network cross-layer optimisation. In *Telecommunication Networking Workshop on QoS in Multiservice IP Networks, 2008. IT-NEWS 2008. 4th International*, pages 185–190, 2008. doi: 10.1109/ITNEWS.2008.4488151.

[4] V. Kawadia and P. R. Kumar. A cautionary perspective on cross-layer design. *IEEE Wireless Communications*, 12(1):3–11, 2005. ISSN 1536-1284. doi: 10.1109/MWC.2005.1404568.

[5] R. K. Sheshadri and D. Koutsonikolas. On packet loss rates in modern 802.11 networks. In *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, pages 1–9, 2017. doi: 10.1109/INFOCOM.2017.8057130.

[6] Konstantinos Pelechrinis, Theodoros Salonidis, Henrik Lundgren, and Nitin Vaidya. Experimental characterization of 802.11n link quality at high rates, 2010.

[7] M. R. Souryal, L. Klein-Berndt, L. E. Miller, and N. Moayeri. Link assessment in an indoor 802.11 network. In *IEEE Wireless Communications and Networking Conference, 2006. WCNC 2006.*, volume 3, pages 1402–1407, 2006. ISBN 1525-3511. doi: 10.1109/WCNC.2006.1696492.

[8] B. Fu, Y. Xiao, H. J. Deng, and H. Zeng. A survey of cross-layer designs in wireless networks. *IEEE Communications Surveys and Tutorials*, 16(1):110–126, 2014. ISSN 1553-877X. doi: 10.1109/SURV.2013.081313.00231.

[9] Zheng Feng and J. Nelson. Cross-layer adaptive design for the frame length of ieee 802.11 networks. In *2008 6th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks and Workshops*, pages 437–442, 2008. doi: 10.1109/WIOPT.2008.4586104.

[10] S. Biaz and S. Wu. Rate adaptation algorithms for ieee 802.11 networks: A survey and comparison. In *2008 IEEE Symposium on Computers and Communications*, pages 130–136, 2008. ISBN 1530-1346. doi: 10.1109/ISCC.2008.4625680.

[11] Fethi Filali. Link-layer fragmentation and retransmission impact on tcp performance in 802.11-based networks. In *7th IFIP International Conference on Mobile and Wireless Communications Networks*, 2005.

[12] S. Rayanchu, A. Mishra, D. Agrawal, S. Saha, and S. Banerjee. Diagnosing wireless packet losses in 802.11: Separating collision from weak signal. In *IEEE INFOCOM 2008 - The 27th Conference on Computer Communications*, page 1. ISBN 0743-166X. doi: 10.1109/INFOCOM.2008.124.

[13] H. S. Kim and B. Daneshrad. Energy-constrained link adaptation for mimo ofdm wireless communication systems. *IEEE Transactions on Wireless Communications*, 9 (9):2820–2832, 2010. ISSN 1536-1276. doi: 10.1109/TWC.2010.062910.090983.

[14] Qualcomm Technologies. Qca6234 integrated dual-band 2x2 802.11n + bluetooth 4.0 data sheet, 2016.

[15] S. Pagadarai and A. M. Wyglinski. A quantitative assessment of wireless spectrum measurements for dynamic spectrum access. In *2009 4th International Conference on Cognitive Radio Oriented Wireless Networks and Communications*, pages 1–5. ISBN 2166-5370. doi: 10.1109/CROWNCOM.2009.5189413.

[16] M. Paulitsch, J. Morris, B. Hall, K. Driscoll, E. Latronico, and P. Koopman. Coverage and the use of cyclic redundancy codes in ultra-dependable systems. In *2005 International Conference on Dependable Systems and Networks (DSN'05)*, pages 346–355. ISBN 1530-0889. doi: 10.1109/DSN.2005.31.

[17] Chesoong Kim, Seokjun Lee, Andrey Lyakhov, and Vladimir Vishnevsky. 802.11 ad hoc lans with realistic channels. *Journal of the Korean Institute of Industrial Engineers*, 33(3):381–392, 2007. ISSN 1225-0988.

[18] Yang-Seok Choi and Hooman Shirani-Mehr. Simultaneous transmission and reception: Algorithm, design and system level performance. *IEEE Transactions on Wireless Communications*, 12(12):5992–6010, 2013. doi: 10.1109/TWC.2013.101713.121152.

[19] H. Bolcskei. Blind high-resolution uplink synchronization of ofdm-based multiple access schemes. *2nd IEEE Workshop on Signal Processing Advances in Wireless Communications*, pages 166–169, 1999. doi: 10.1109/SPAWC.1999.783045.

[20] P. Mafole, M. Kissaka, and M. Aritsugi. Fragment retransmission scheme with enhanced collision avoidance for energy-efficient ieee 802.11 wlans. In *2016 Wireless Days (WD)*, pages 1–4. doi: 10.1109/WD.2016.7461475.

[21] Jiansong Zhang, Haichen Shen, Kun Tan, Ranveer Chandra, Yongguang Zhang, and Qian Zhang. Frame retransmissions considered harmful: improving spectrum efficiency using micro-acks. In *Proceedings of the 18th annual international conference on Mobile computing and networking*, pages 89–100. ACM. ISBN 1450311598.

[22] I. Djama, T. Ahmed, and D. Negru. Adaptive cross-layer fragmentation for qos-based wireless iptv services. In *2008 IEEE/ACS International Conference on Computer Systems and Applications*, pages 993–998. ISBN 2161-5322. doi: 10.1109/AICCSA.2008.4493666.

[23] Xi Yong, Wei Ji-Bo, Zhuang Zhao-Wen, and Kim Byung-Seo. Performance evaluation, improvement and channel adaptive strategy for ieee 802.11 fragmentation mechanism. In *11th IEEE Symposium on Computers and Communications (ISCC'06)*, pages 142–148. ISBN 1530-1346. doi: 10.1109/ISCC.2006.126.

[24] Sayantan Choudhury and Jerry D Gibson. Payload length and rate adaptation for multimedia communications in wireless lans. *IEEE Journal on Selected Areas in Communications*, 25(4), 2007. ISSN 0733-8716.

[25] Christian Senning, Georgios Karakonstantis, and Andreas Burg. Cross-layer energy-efficiency optimization of packet based wireless mimo communication systems. *Journal of Signal Processing Systems*, 85(1):129–142, 2016. ISSN 1939-8115. doi: 10.1007/s11265-015-1003-7. URL https://doi.org/10.1007/s11265-015-1003-7.

[26] R. Abdolee, B. Champagne, and A. H. Sayed. Diffusion adaptation over multi-agent networks with wireless link impairments. *IEEE Transactions on Mobile Computing*, 15(6):1362–1376, 2016. ISSN 1536-1233. doi: 10.1109/TMC.2015.2460251.

[27] National Instruments. Introduction to wireless lan measurements. *National Instruments Website*, 2014. URL http://download.ni.com/evaluation/rf/Introduction_to_WLAN_Testing.pdf.

[28] G. L. Stuber, J. R. Barry, S. W. McLaughlin, Ye Li, M. A. Ingram, and T. G. Pratt. Broadband mimo-ofdm wireless communications. *Proceedings of the IEEE*, 92(2):271–294, 2004. ISSN 0018-9219. doi: 10.1109/JPROC.2003.821912.

[29] Fernando H Gregorio. 802.11 a-ofdm phy coding and interleaving. *Helsinki University of Technology*, 4(8):1–6, 2006.

[30] Chi-han Kao. Performance of the ieee 802.11 a wireless lan standard over frequency-selective, slow, ricean fading channels. Report, NAVAL POSTGRADUATE SCHOOL MONTEREY CA, 2002.

[31] Sunghyun Choi, Youngkyu Choi, and Inkyu Lee. Ieee 802.11 mac-level fec scheme with retransmission combining. *IEEE Transactions on Wireless Communications*, 5(1):203–211, 2006. ISSN 1536-1276.

[32] S. Shakkottai, T. S. Rappaport, and P. C. Karlsson. Cross-layer design for wireless networks. *IEEE Communications Magazine*, 41(10):74–80, 2003. ISSN 0163-6804. doi: 10.1109/MCOM.2003.1235598.

[33] F. Foukalas, V. Gazis, and N. Alonistioti. Cross-layer design proposals for wireless mobile networks: A survey and taxonomy. *IEEE Communications Surveys and Tutorials*, 10(1):70–85, 2008. ISSN 1553-877X. doi: 10.1109/COMST.2008.4483671.

[34] M. Mohaghegh, C. Manford, and A. Sarrafzadeh. Cross-layer optimisation for quality of service support in wireless sensor networks. In *Communication Software and Networks (ICCSN), 2011 IEEE 3rd International Conference on*, pages 528–533, 2011. doi: 10.1109/ICCSN.2011.6014950.

[35] Mythili Vutukuru, Hari Balakrishnan, and Kyle Jamieson. Cross-layer wireless bit rate adaptation. In *Proceedings of the ACM SIGCOMM 2009 conference on Data communication*, pages 3–14, 1592571, 2009. ACM. doi: 10.1145/1592568.1592571.

[36] H. Balakrishnan, V. N. Padmanabhan, S. Seshan, and R. H. Katz. A comparison of mechanisms for improving tcp performance over wireless links. *IEEE/ACM Transactions on Networking*, 5(6):756–769, 1997. ISSN 1063-6692. doi: 10.1109/90.650137.

[37] A. Tarighat, R. Bagheri, and A. H. Sayed. Compensation schemes and performance analysis of iq imbalances in ofdm receivers. *IEEE Transactions on Signal Processing*, 53(8):3257–3268, 2005. ISSN 1053-587X. doi: 10.1109/TSP.2005.851156.

[38] Xiaomin Chen, Vijay Subramanian, and Douglas Leith. An upper bound on the packet error rate of 802.11 a/g viterbi soft decision decoding in the awgn channel. In *Wireless Days (WD), 2012 IFIP*, pages 1–4. IEEE, 2012. ISBN 1467344044.

[39] F. Cali, M. Conti, and E. Gregori. Dynamic tuning of the ieee 802.11 protocol to achieve a theoretical throughput limit. *IEEE/ACM Transactions on Networking*, 8 (6):785–799, 2000. ISSN 1063-6692. doi: 10.1109/90.893874.

[40] Ernst Bonek, Werner Weichselberger, Markus Herdin, and Huseyin Ozcelik. A geometry-based stochastic mimo channel model for 4g indoor broadband packet access. *XXVIIIth General Assembly of International Union of Radio Science (URSI)*, 28, 2005. URL https://pdfs.semanticscholar.org/8a99/eb1dd3566192a0d6e0dca6b8dd1b3c1215d0.pdf.