Error Behaviour in Optical Networks

Laura Bryony James

Corpus Christi College

This dissertation is submitted for the degree of Doctor of Philosophy

30th September 2005

Department of Engineering, University of Cambridge

This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements.

Abstract

Optical fibre communications are now widely used in many applications, including local area computer networks. I postulate that many future optical LANs will be required to operate with limited optical power budgets for a variety of reasons, including increased system complexity and link speed, low cost components and minimal increases in transmit power. Some developers will wish to run links with reduced power budget margins, and the received data in these systems will be more susceptible to errors than has been the case previously.

The errors observed in optical systems are investigated using the particular case of Gigabit Ethernet on fibre as an example. Gigabit Ethernet is one of three popular optical local area interconnects which use 8B/10B line coding, along with Fibre Channel and Infiniband, and is widely deployed. This line encoding is also used by packet switched optical LANs currently under development. A probabilistic analysis follows the effects of a single channel error in a frame, through the line coding scheme and the MAC layer frame error detection mechanisms. Empirical data is used to enhance this original analysis, making it directly relevant to deployed systems.

Experiments using Gigabit Ethernet on fibre with reduced power levels at the receiver to simulate the effect of limited power margins are described. It is found that channel bit error rate and packet loss rate have only a weakly deterministic relationship, due to interactions between a number of non-uniform error characteristics at various network sub-layers. Some data payloads suffer from high bit error rates and low packet loss rates, compared to others with lower bit error rates and yet higher packet losses. Experiments using real Internet traffic contribute to the development of a novel model linking packet loss, the payload damage rate, and channel bit error rate. The observed error behaviours at various points in the physical and data link layers are detailed. These include data-dependent channel errors; this error *hot-spotting* is in contrast to the failure modes observed in a copper-based system. It is also found that both multiple channel errors within a single code-group, and multiple error instances within a frame, occur more frequently than might be expected. The overall effects of these error characteristics on the ability of cyclic redundancy checks (CRCs) to detect errors, and on the performance of higher layers in the network, is considered.

This dissertation contributes to the discussion of layer interactions, which may lead to unforeseen performance issues at higher levels of the network stack, and extends it by considering the physical and data link layers for a common form of optical link. The increased risk of errors in future optical networks, and my findings for 8B/10B encoded optical links, demonstrate the need for a cross-layer understanding of error characteristics in such systems. The development of these new networks should take error performance into account in light of the particular requirements of the application in question.

Acknowledgements

I am grateful for the support, encouragement and ideas provided by my supervisor Ian White, particularly whilst I was writing up. Many thanks also to Andrew Moore for untiring assistance, insightful advice and sage wisdom, and for eating my food so I didn't have to.

This work would not have been possible without many helpful discussions with Derek McAuley, Madeleine Glick, Richard Penty and numerous others in Intel Research, the Computer Lab and the Photonics Systems Group who spared the time to contribute their thoughts — thanks to you all.

In less technical ways, I received a great deal of support during my studies from mentors and former colleagues, who in various ways advised and encouraged me; you have my eternal gratitude. Markus Fromherz, Ursula Martin and both the ex-ORL/ex-AT&T and Menlo Studio crowds deserve special mentions.

Many thanks to Adrian Stephens, Olly Johnson, Richard Gibbens, Dick Plumb, Ian Wassell and David Hunter, for clarifying things I didn't understand and providing feedback on my work. I would like to acknowledge James Bulpin, for the development of the non-uniform error testbed of Section 5.4.1. Many thanks also to Adrian Wonfor, who kept me up to date with the intricacies of running Microsoft Windows Server, ensured that I had a working computer and file storage, and sometimes even helped me with my work.

I am extremely grateful to Guy Roberts, Michael Dales, Wenxin Tang, Tao Lin, Enrique Rodriguez de la Colina and Amyas Phillips for repeatedly having the patience to listen to my ramblings and still suggest ideas afterwards, for uncomplainingly proofreading, and for many a productive (or otherwise) tea break. Jeremy Sosabowski also helped me maintain a balance between work and tea, and the PhD experience would not have been the same without him.

The UK Engineering and Physical Sciences Research Council and Marconi Corporation supported my work financially through an Industrial CASE studentship.

Finally, most of all, I would like to thank my parents, and Paul, for valuable proofreading and invaluable support.

Publications

Parts of the following work have previously appeared in the following publications.

London Communications Symposium (LCS) 2003
Wavelength Striped Semi-synchronous Optical Local Area Networks
L B James, G F Roberts, M Glick, D McAuley, K A Williams, R V Penty and I H White

Passive and Active Measurement Workshop (PAM) 2004
Structured Errors in Optical Gigabit Ethernet
L B James, A W Moore, M Glick

London Communications Symposium (LCS) 2004
Beyond Gigabit Ethernet: Physical Layer Issues in Future Optical Networks
L B James, A W Moore, R Plumb, M Glick, A Wonfor, I H White, D McAuley and R V Penty

Optical Fiber Communications Conference (OFC) 2005
Packet error rate and bit error rate non deterministic relationship in optical network applications
L B James, A W Moore, A Wonfor, R Plumb, I H White and R V Penty

Poster presented at INFOCOM 2005
A Graphical Exploration of non-Uniform Errors
L B James, A W Moore, M Glick and A Wonfor

IEEE Communications Magazine, August 2005
Chasing errors through the network stack: A testbed for investigating errors in real traffic on optical networks
A W Moore, L B James, M Glick, A Wonfor, I H White, D McAuley and R V Penty

IEEE Journal of Lightwave Technology To appear October 2005
Optical Network Packet Error-Rate due to Physical Layer Coding
A W Moore, L B James, M Glick, A Wonfor, R Plumb and I H White

For my parents

Contents

List of Figures

Chapter 1

Introduction

The work described in this dissertation sits at the intersection between optical communications and computer networking. This chapter outlines some advances that have taken place in the field of optical data networking, particularly as it applies to short haul systems, and those used for computing applications. Some of the principles involved in computer networking are then described: the use of layer abstraction, and the basics of TCP/IP and Ethernet systems. An examination of the motivations behind this work, which contributes to the discussion surrounding the development of the next generation of optical computer interconnects, is also presented.

1.1 Optical Networking

As the demand for faster communication networks expands with the growth of computing systems, copper transmission systems can no longer meet the bandwidth requirements of many networks. Instead, for any system where mobility is not required, optical fibre networks can be used as they offer excellent bandwidth and flexibility.

Optical networking has seen huge growth in recent years, as fibre is used increasingly for data transfer as well as in the traditional areas of telecommunications. As the need for global communication grows, data begins to overtake voice traffic in volume, and it brings with it new requirements. At the network level, data usually requires packet rather than circuit switching, on very short timescales and often with more complex processing needed at the switch. Very low latency is desirable, especially in short haul systems; bursty data must be handled.

IP has become the core technology for data communications, and is ideal for many data transport requirements, although not perfect for all. It provides a simple address for every

host, and hides the differences in underlying transport networks, whether wired or wireless. As IP is commonly perceived as dominating communication systems worldwide, it is surprising to note that in 2001 the market size for IP routers at the core and edge of networks combined is much less than that for optical routers (SONET/SDH/WDM). These traditional optical networks are long haul, circuit switched and originally designed to carry voice signals [1].

Packet switching has always offered better use of the more scarce bandwidth at the edge than circuit switching. However, IP (often using the Ethernet data link layer) is now penetrating even into the classical optical networks of the core. In the opposite direction, edge systems themselves require faster connections, to each other and to the Internet, than ever before and optical links are often required to meet this need. All this brings new challenges to the world of optical communications, but innovative devices and technologies are being developed to handle them.

1.1.1 Optical Packet and Burst Switching

A multitude of projects have attempted to create a realistic vision of an optical system capable of carrying packet data. They range from large scale networks with thousands of routers [2], through metro networks and access and edge systems [3, 4, 5], to board or chip level buses in computers [6]. Some are passive broadcast networks rather than switched systems [7]. They may be synchronous or asynchronous, carrying variable or fixed-length packets. Some run data protocols such as IP across SDH/SONET or other optical systems [8]; Generalised Multi-protocol Label Switching (GMPLS, [9]) proposes a different method of transporting IP over an optical path, but these options may run on too coarsely grained timescales to be optimal for data traffic. Nonetheless, a great deal of work has gone into their study and implementation, including the integration of traffic control at the IP layer into the resource control part of the optical plane [9]. Label switching systems do offer a way of integrating legacy and new technologies, both interoperating between circuit and packet-switched networks, and with the protocols that run across them [10].

As well as traditional circuit switched schemes and the well understood packet switched networks also used in the wireline and wireless domains, another technology, Burst Switching, allows groups of packets to be collected and then forwarded as appropriate, and has been widely studied of late in the optical domain [11]. This can present problems for practical implementations as buffering is usually required [12]; there are also many issues relating to lack of knowledge about the packets when they arrive at a routing node (e.g., the frame amplitude and phase are unknown). Clock and data recovery have to be performed without any knowledge of these phases or receiver thresholds, and although a great deal of work has been done on improving technology for burst mode systems, it still presents significant challenges [13]. However, the advantages of being able to work with cruder switching devices and reduced

processing requirements are notable, as the data can stay optical throughout the network, whilst electronics handle resource allocation [14]. Processing of headers and signalling between protocol layers are two major issues in the comparison of circuit-, burst- and packet-switched systems [15, 16].

1.1.2 Optical Networks for Computing

Since fibre optic transmission is often the bandwidth leading-edge, optical devices (which are common in core networks) trickle down to be used in local area networks more quickly than technologies for copper transmission. Many advances made in technologies for long-haul networking can be adapted for these shorter range systems. In computer networking, the areas with the greatest need for the bandwidth offered by optical media are short haul interconnects within a system, and connections between servers.

All these systems attain best performance when working with low latency, and local area network engineers prefer to have well-used networks rather than leaving excess capacity, so good utilisation is important. Cheap components, which use minimal power, are desirable. Currently, links between servers are commonly switched data over fibre, as are in-building backbones and campus area networks. The latest developments in commercial optical data networking are fully switched in the electrical domain, with optics only used for point-to-point links from the endpoints to the switch. These systems use low cost components where possible, but the 10Gbps or faster switches remain expensive, as the electronics required at these speeds are new, complex and delicate.

All-optical packet switched techniques are particularly interesting, as these should provide the best networks for data, with good bandwidth, minimised latency and the flexibility to carry bursty traffic. Comparatively little work has been done to actually build optical packet-based systems which provide useful data transport, particularly away from the long haul realm [17, 18]. For instance, some proposed systems use excessively long packet sizes which would limit their usefulness in a data switching environment [19]. As transmissions speeds have increased, the bottleneck in optical computer networks is now in the switching and end systems, and development of these presents an exciting challenge. An example of an optical local area network prototype, utilising optical switching, is given in Chapter 2.

1.2 Computer Networking

This is a study of data networking and so some of the features of common networks today which will persist in future systems, and which affect the design and performance of networks, are considered here. In particular the use of *layering* is discussed, as this principle will recur

throughout this work.

1.2.1 Layer Abstraction

Many computing systems use layer abstraction to enable successful integration of components from varied manufacturers or system types. Each layer defines a set of services offered to the layer above, which does not need to know details of the implementation of these services or how layers further down operate. This abstraction removes the need for those working at one layer to fully understand the other levels. Specialists at those other layers are able to implement their own services there in the most appropriate way, providing they honour the appropriate definitions when connecting to the layers above and below.

In networking, each layer at one host appears to communicate directly with the equivalent layer on another, but no data is passed directly between them (with the exception of the physical layer itself). Instead, perhaps combined with some control information, it is passed to successively lower layers until the physical channel is reached. The data can then be sent across the medium (be it wireless, wires or optical fibres), up through the layers at another host, until the original layer is reached. Each layer, then, deals with its peer at the other side of the communications link using a relevant protocol. A layer interface defines what services the lower layer offers the upper layer. An entire layer may then be replaced with an alternative implementation, if required, providing the same services are still offered to the layer above.

A user surfing the web, for example, deals with an application at the top layer and has no need to know what systems will handle the communications. Layer abstraction makes this simple. A series of layers in the user's computer (a protocol stack, comprising software, hardware and firmware components) handle the requests for information from above, and serve up the supplied network data in the correct format. The actual channel could be a phone line, satellite link or short range wireless connection, but will most likely be made up of many stages each using a different medium; at each stage the signals will be processed into a suitable form for the next leg. At the remote host the reverse happens as the received message is decoded and passed up through the layers to the web server application which receives a straightforward request for a web page. This is then passed down the layers, which gradually package it up as a message suitable for transmission over the network. The user's web browser receives a web page, perhaps made up of text and pictures, which was transmitted on the wire, fibre or radio as a number of packets of encoded binary data which would be incomprehensible to the user or their application software. (This is illustrated in Figure 1.1.) The system works end to end, allowing web browsers and web servers to communicate regardless of the intermediate communications links, because of the abstraction permitted by layering.

Different layers in networking systems must perform various tasks, the complexity of which

illustrates why abstraction is helpful to system designers and implementers. Some examples are given here; they may apply to one or more layers. As a given layer may form part of a network and not just a simple point-to-point link, at least a destination address must travel with the data payload in a packet network such as IP; this will often be added to the message as a header. The route taken by the message to this destination must be determined; addresses may need to be converted if the packet moves through different network types. A subsystem may have size limits on the data blocks it can handle, meaning that messages may be aggregated or split into shorter frames. In this case information about how the message has been reformatted must travel with each frame, and the relevant layer at the destination must be able to reassemble the original data. One host may only be able to handle data at a much slower rate than another, so at some point in the protocol stack flow control will be required. If the data to be sent is private, it may be encrypted; this requires layers at both hosts to agree on an encryption system and perhaps exchange keys in some way, and to be able to process the data appropriately. No physical communications channel is perfect — errors may occur — and if error detection and/or correction mechanisms are to be used, both ends of the link must know what methods are being used. If some packets of a data exchange go missing altogether (in a system with a reliable transport layer, such as TCP), the hosts need to determine this and handle it, perhaps by requesting retransmissions.

So layering offers many benefits to network designers, users and manufacturers. There may be some differences between the protocols used in practice, and the reference model which allows a clear understanding of the system. The common ISO/OSI model, which provides a useful nomenclature, is detailed here, followed by an outline of the TCP/IP protocols and Gigabit Ethernet model which are relevant to subsequent chapters.

The ISO/OSI Reference Model

In the late 1970s, it was clear that there was a growing need to exchange data, and that systems which could only communicate with other equipment from the same manufacturer would not be suitable in the future. The International Organization for Standardization (ISO) set up a committee to work on an architecture which could help define a standard framework for linking different types of computer system. This work ultimately led to the well-known seven layer Open Systems Interconnection (OSI) model, which has become the reference for all shorter range networks (local and metro area networks, LANs/MANs) developed by the IEEE working groups in that field (IEEE 802). This model has since been used in the design of many LAN/MAN networks, such as Token Ring, all the Ethernet family, and more recently Wireless LANs [20, §1.4].

This reference model combines network application functions, with those of the network itself; seven layers were chosen as a balance between a desire for clear abstractions and a

lightweight architecture [21, 22].

The seven layers, with the basic functions each performs, are as follows.

Layer 7 - Application The user level applications which use the network, including handling password authentication, directory information, and so on.

Layer 6 - Presentation Data structures to be used, character sets, encryption.

Layer 5 - Session Negotiate who transmits when, and address mappings, for a session; synchronisation.

Layer 4 - Transport Split data if needed, make sure all the segments arrive at the destination, offer a specific link or network service type to the session layer.

Layer 3 - Network Route packets, control congestion and quality of service.

Layer 2 - Data Link Form frames of suitable sizes, control line rate and control access to a shared medium if necessary.

Layer 1 - Physical Mechanical connectors to the physical medium, electrical and optical specifications, timing and directionality issues.

This model covers the tasks a protocol stack might need to perform and provides a structure for designers to consider. In practice however one does not come across systems with seven distinct software or hardware blocks; protocols have been developed as needs arose. One obvious case is that of TCP/IP; the protocols existed before a reference model was retrofitted to them. This model is worthy of mention here, because it is so widely used and because it is referred to again later, although it does not relate well to other protocol stacks except TCP/IP itself.

TCP/IP Networking

First defined in 1974 [23], the TCP/IP reference model is named for the two protocols which comprise its implementation.

In this model there are 4 main layers. The **application** layer is at the top, and contains the high level protocols, which the user can see. The **transport** layer supports end to end conversations, similarly to the OSI transport layer. This usually carries one of two protocols — either TCP for a connection-oriented datastream service or UDP for simpler, connectionless frame transfer. TCP is reliable, fragmenting the information to be sent, handling flow control, and reassembling the data and ensuring it was all received at the far end. UDP offers no flow or sequence control, allowing users to supply their own control schemes, or just utilise

basic message passing. Beneath this is the **internet** layer, allowing packets to travel between heterogeneous networks; the frames are sent individually, may be routed independently and are not necessarily guaranteed to reach their destination. This is equivalent to the OSI network layer and the protocol at this layer is, of course, IP. The bottom two levels of the OSI model are not really described by the TCP/IP model; it is merely assumed that below the internet layer there is some layer which handles the carriage of IP packets.

An Example of Layering in a Real System

In the case of a TCP/IP system, running over Gigabit Ethernet, the OSI model may be outlined as in Figure 1.1.

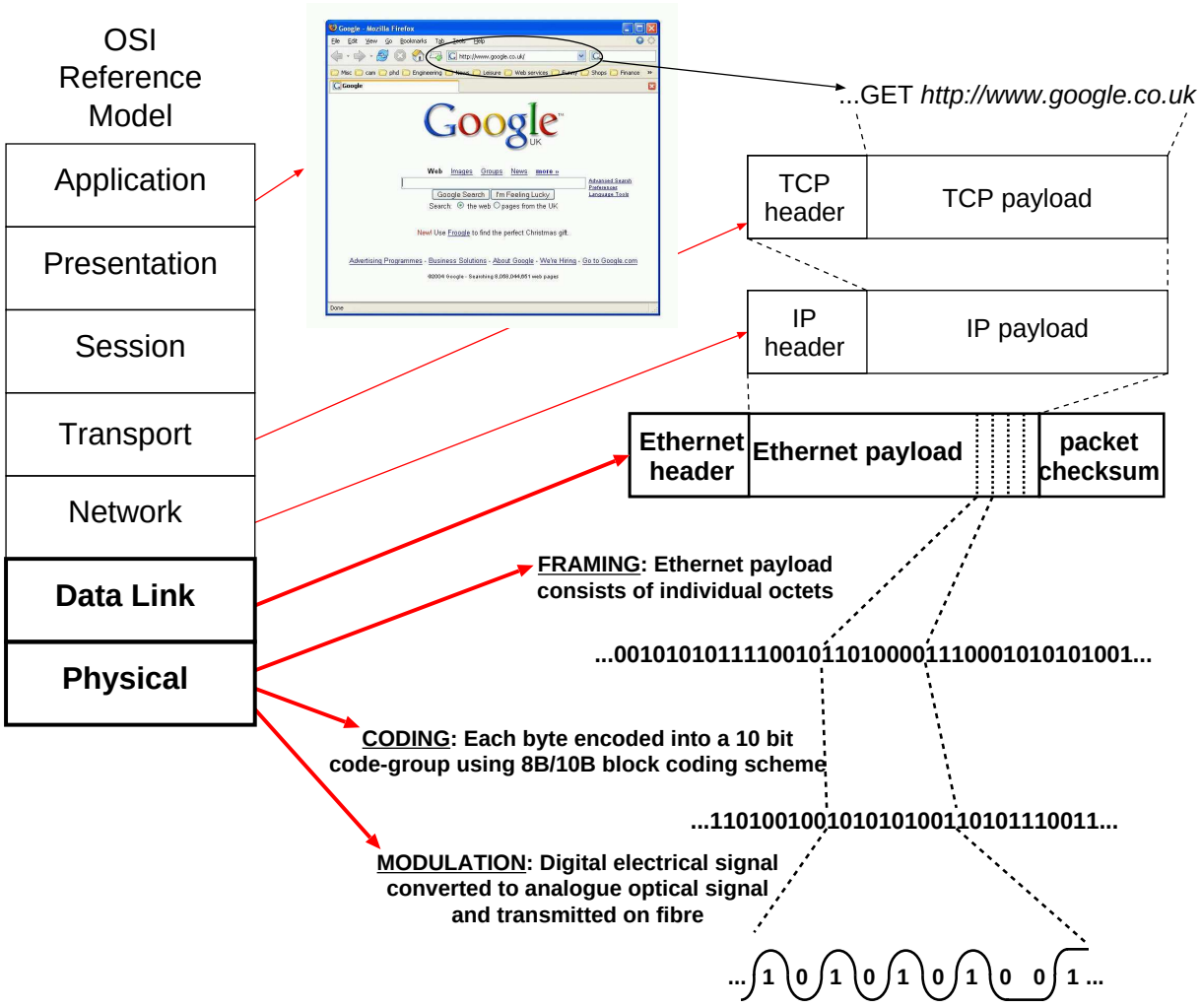


Figure 1.1: Layering system as it applies to a Gigabit Ethernet network with example application

Ethernet is one of the most common underlying systems in TCP/IP networks. It is of interest as some of our subsequent work focuses on Gigabit Ethernet, and also as an example of a system which has developed far beyond its original incarnation as network and application

demands have changed.

1.2.2 The Evolution of Ethernet

Ethernet has come a long way from its original implementation as a 3Mbps baseband transmission LAN running on a shared coaxial cable [24]. The original goals were for a simple network, which could be constructed and utilised cheaply, where nodes had fair, equal access to the network, and all implementations of the system could communicate with each other. From these early principles developed the CSMA/CD protocol (carrier-sense multiple access with collision detection) which is now considered to be classic Ethernet. In 1978, Ethernet was formalised with the IEEE 802.3 standard [20, §1.5.3]; an example would be 10BASE-5, which represents a network running at 10Mbps, with baseband signalling, capable of working over a 500m shared segment of coaxial cable. Later, Fast Ethernet increased the bit-rate to 100Mbps, could be carried on twisted pair or fibre only and required the use of a hub or switch rather than permitting segment sharing. In all these variants, the Media Access Control (MAC) protocol limits frame size to a suitable range, adds preamble and start of frame octets, uses 6 bytes each for source and destination addresses, and appends a checksum to each frame. The addressing scheme allows for multicast and broadcast transmissions, and for local and global addresses to be distinguished.

An "Ethernet network" today often refers only to the size of frames and the 48-bit address structure [25]; the use of shared media is rare. Increasingly common Ethernet-like implementations include many wireless standards; Ethernet also dominates in the wired LAN, in terms of both copper UTP (unshielded twisted pair) and fibre-optic cable, now almost invariably as a switched network. Using individual links to a central switch allows the LAN to handle heavier loads, and adds network flexibility. Ethernet variants are extensively deployed; it has been estimated that 90% of all IP traffic originates in Ethernet LANs [26].

In 1995 work began on an even faster Ethernet variant - Gigabit Ethernet [27]. This supports full- or half-duplex transmission between stations, hubs and switches; maximum link length is determined by signal strength rather than by collision risk as was the case for shared links. The original standard defined a set of 1000BASE-X flavours: SX, for 850nm wavelength light on multimode fibre, LX, using 1300nm light on single-mode or multimode fibre, and CX for short-haul transmissions on shielded copper twisted pair. The 1000BASE-T variant, which operates over standard UTP cables, was a later addition. These all use line coding schemes different to earlier Ethernets. Gigabit Ethernet is now widely deployed from LANs to campus backbones, between servers in machine rooms, and so on, and even in longer haul links on the metropolitan area scale [28].

Other Ethernet-based systems include Ethernet in the first mile access network [29], and

various passive optical networks [30, 31]. Protocol extensions and modifications are being developed to allow Ethernet to be used in larger, metro area networks [32] and to scale from backplanes [33] to enterprise systems consisting of millions of nodes [34].

From a network using coaxial cable over very limited range, Ethernet has grown: Ethernet systems can be switched, run on twisted pair cable or optical fibre, at speeds up to 10Gbps, and over distances never dreamt of by its original designers at the time. "Carrier Ethernet" promises to take over the market currently occupied by SONET, with a scalable, 10Gbps and over, wide area network offering, which will be cheaper than traditional telco solutions. Bob Metcalfe, inventor of the Ethernet, commented on this in April 2005: "I see Ethernet developing in four directions: up, down, over and across. Up in speed... down to the 8 billion processors shipping each year that are not yet networked. Ethernet is increasingly moving over wireless links — WiFi, WiMax, ZigBee and others... and now it's moving even further across the chasm between LANs and WANs. [35]" Ethernet has come a long way, but clearly has further to go.

The Gigabit Ethernet Reference Model

Classic Ethernet divided the OSI model data link layer into two: Logical Link Control (LLC), offering datagram or connection-oriented service, and the usual Media Access Control. To actually implement the physical layer in a well-defined manner, this too has been split into a number of sublayers, which vary depending on the variety of Ethernet in question.

For Gigabit Ethernet, the lowest two layers of the OSI model are as represented in Figure 1.2.

The sublayers making up the OSI data link layer are carefully defined in terms of primitives — requests, indications and so on. This work is not concerned with media access issues; the lower layers of the physical sublayer group are of more interest. The Reconciliation and GMII sublayers are the same regardless of the medium used; the lower layers are specific to the physical channel. The Physical Coding Sublayer (PCS) defines the line coding scheme used (discussed further in Section 4.1.1 for the case of 1000BASE-X) and also Auto Negotiation, which allows hosts at either end of a link to agree on an operational mode (e.g., half- or full-duplex). The Physical Medium Attachment sublayer (PMA) handles serialisation of the bits for transmission, and also performs clock recovery and deserialisation at the receiving end. The Physical Medium Dependent sublayer (PMD) performs digital to analogue conversion, turning bits into correctly formed electrical or optical pulses, and the Medium Dependent Interface specifies the connector to the actual fibre or copper link.

Although these functions are carefully specified in the model, actual implementations of Gigabit Ethernet may not have clearly distinguishable sublayers. Many layers of software, hardware and firmware will interact to create a working system, each dealing as best it can

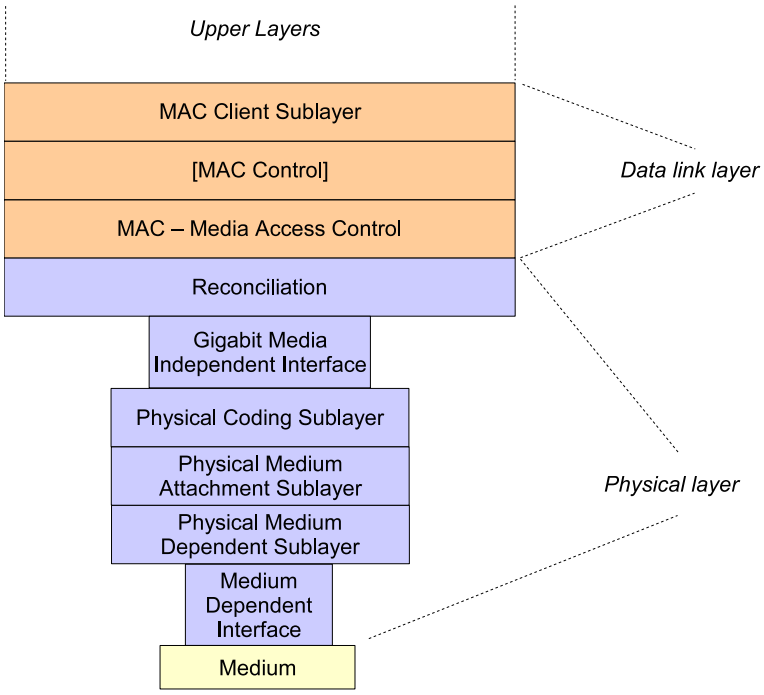


Figure 1.2: The Gigabit Ethernet Reference Model

with the information provided at its interfaces. Since an overall system is likely to be made up of components from different sources, working in ignorance of the decisions made elsewhere, it is possible for a full system to function adequately without necessarily meeting all the detailed requirements of the model.

As is often the case now, networking equipment may ship before the standards which will apply to it are fully defined. David Clark described "the apocalypse of the two elephants", where research activity on a networking principle and the investment which gets technology out into the field occur almost at the same time [20, §1.4.4]. In this case there is no clear period, after the research is sufficiently complete and before companies must start production, in which standards can be developed. Commercial pressures force companies to offer faster or more robust solutions to the market before compliance with a new standard can be assured. Most of this deployed equipment will interoperate with other manufacturers' pre-standard devices and with "official" equipment later, but may offer subtly different performance. These issues are returned to in Section 7.2.

10G and Beyond: The Future of Ethernet

There is still a demand for more bandwidth. Ethernet systems have proven flexibility and are a strong brand, even if current systems barely resemble the early networks. Ethernet works well with now-dominant TCP/IP networking and is simple to set up and maintain; in addition it has always had an easy speed upgrade path and has been relatively cheap. The standard

for 10Gbps Ethernet was fixed in 2002, and offers separate LAN and WAN physical layer implementations [36]. Despite initial pricing higher per Gbps than gigabit Ethernet, the speed increase and the desirability of the Ethernet brand have made 10Gigabit Ethernet a success [37]; 100Gbps will surely be next. How much it resembles previous Ethernet incarnations remains to be seen.

The switch is now a central part of all modern Ethernet systems; to date, these switches have relied on electronic frame processing. With data coming in from multiple high speed links, the electronics stage starts to be a bottleneck. Each frame arriving on either copper or fibre must be buffered — its headers read and perhaps rewritten — then redirected to the appropriate output port. If the data has come from an optical fibre, there are additional stages of optical-electrical conversion and electrical-optical conversion, requiring costly components. Memory access is likely to be parallel for speed, so deserialisation at the input, and serialisation at the output, will add latency. With links increasingly running over optical fibre for speed, an all-optical data path through the switch would be a valuable addition. Research into ultra fast optical LANs and interconnects has investigated optical switching; an example prototype system with an electronically-controlled optical datapath is described in Chapter 2.

1.3 Motivations

The previous sections have shown seen how optical communications systems are used for computer networking; however, they still remain electrically switched despite high point-to-point bit-rates. One next move is to introduce optical switching to the LAN to keep up with ever increasing link speeds. As higher data rates are required in the even shorter-haul realm of backplanes and chip to chip interconnects, optical technology will migrate there too. The area of short range, switched optical data networks is a very exciting one at present.

In many fields, interesting behaviour tends to occur at edges, the interstices between more stable and well-understood regions.This work is interested in two edges. One is the edge between the physical layer (a world of photons, noise, and so on) and the data networking space where the physical layer is simply a cable that must be connected. The other is the edge between total failure and good performance, which can be hard to test.

For a variety of reasons, the errors in optical networks which lead to partial performance loss but not complete system failure are particularly interesting. Given the current active development of the next generation of networks, this is a good time to reflect on these effects so that design compensation can be made if necessary.

In particular, one common characteristic of these future optical networks is that they are likely to operate with more limited power at the receiver than has been the case to date.

1.3.1 Future Optical Network Design

Networks increasingly contain longer runs of fibre (e.g., Ethernet in the "last mile" [29]) or large numbers of splitters (e.g., passive optical networks [30]), which reduce the available power at the receiver(s). Active optical devices are more likely to be used in the future, and have stringent power requirements (for instance, keeping power below the level where it would saturate the device). These are particularly required by the more complex switched optical systems, such as the packet network described in Chapter 2. In addition, a traditional complication is that it is not unknown for networks to be installed in breach of their specification: one example might be the use of an excessively long cable or fibre. If the link appears to work and the majority of packets are transmitted without problems, this is unlikely to be noticed. In such low power regimes, the receiver will not have the power levels guaranteed by the specification.

The effects of attenuation (in terms of the errors caused by low receiver power) are therefore particularly of interest. Limited optical power means that future networks are more likely to be working near or at the receiver threshold, and so will suffer from errors more than has been the case in the past. The understanding of how these errors will affect network performance is vital in the design and development of these systems.

1.3.2 The Power Problem: Power Versus Speed

If all other variables are held constant an increase in bit-rate will require a proportional increase in transmitter power. A certain number of photons per bit must be received to guarantee any given bit error rate (BER), even if no thermal noise is present. The arrival of photons at the receiver is a Poisson process. If 20 photons must arrive per bit to ensure a BER of 10^{-9} for a given receiver, doubling the bit rate means that the time in which the 20 photons must arrive is halved. The number of photons sent per unit time must thus be doubled — doubling the transmission power — to maintain the BER (Figure 1.3).

So future system operating at double the current rates will either require twice the power (a 3dB increase), be able to operate with 3dB less power for the given information rate (equivalent to the channel being noisier by that proportion), or a compromise between these two.

Inevitably, the demand for bandwidth continues to grow; as network speeds increase, the transmitter power should increase, but additional power is not always available. Regardless of the optical issues of transmitting at high powers (the risk of saturating some devices, such as amplifiers, and dispersion problems due to fibre non-linearities), many environments are already working at the limit of available power or power density. Machine rooms are often operating at the capacity to which they can be cooled; their electrical supplies are also fully utilised. New systems which can operate at reduced powers will be in demand.

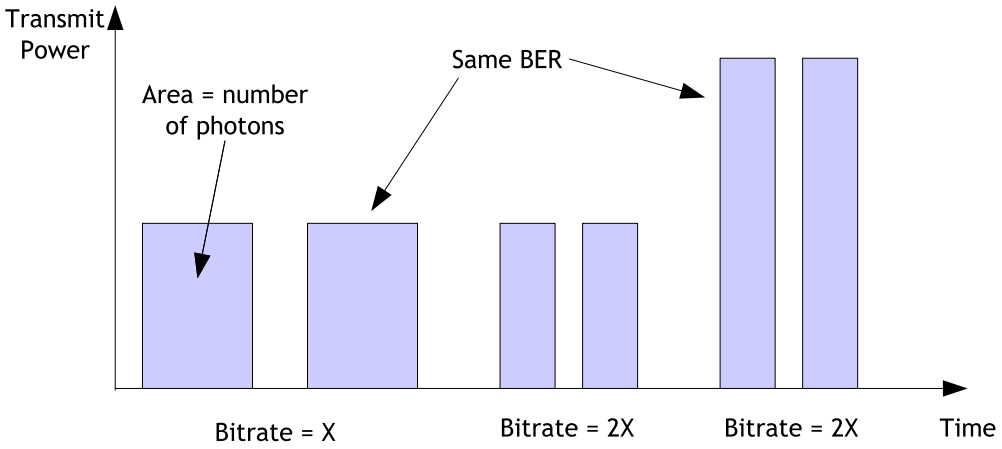


Figure 1.3: An illustration of bit-rate and transmission power trade-off

To reduce individual transmit powers, it may be possible to use a number of parallel links at lower individual bit-rates to obtain the desired overall bandwidth. Parallel fibres, or a multiple wavelength system (Section 2.1.2) could be used for this. In general, though, future links operating at high bit-rates are more likely to operate with receiver power less than the ideal, and the probability of bit errors at the receiver will be increased.

1.3.3 Cheap and Cheerful: The Commoditisation of Computer Networking

Computer networking products are extremely price sensitive. A difference of a few cents can be all it takes to push a single product to success in the marketplace. Computer interconnects are subject to price pressure even more than separate networking parts.

All forms of electronics have been optimised to reduce cost over the past decades, and now one can add circuit boards, memory and even analogue/digital interface components to a product with negligible cost implications. In optics, however, components are still comparatively costly, although of course prices come down as volume manufacturing is introduced for each new generation of products. A Gigabit Ethernet network interface card (NIC) is still far more expensive with an optical media connection than with copper. At the time of writing, a quick search on a popular UK computing hardware site confirmed that a 1000BASE-T Gigabit Ethernet NIC can be had for under £15, whereas the cheapest fibre variant is over £200.

With optics parts still relatively expensive, then, manufacturers will want to reduce the costs in this area as much as possible. Inferior optical components and build quality can be expected, provided performance criteria are met. Equipment which performs “robustly”, in that data gets from A to B at a reasonable speed regardless of inferior cabling, say, will sell better than a version which repeatedly disables the link whilst reporting “interface unplugged”

errors. This, interestingly, favours the sale of networking kit which will keep a link alive even when power levels drop below those defined by a precise interpretation of the specification.

Luckily, if optical link performance is expected to be below that required, electronics are now so cheap that extra electronic processing at transmitter, receiver or both could be added to compensate for poor physical layer performance. Extra error detection and/or correction systems would be an obvious choice. These must be present on all communicating hosts, as both transmitter and receiver need to be aware of the scheme in use. Manufacturers must therefore agree as to what systems will be implemented at the design stage; current research is required to determine what is necessary. An example would be the recent investigations into the use of digital signal processing (DSP) systems to provide electronic dispersion compensation for optical systems [38, 39].

1.3.4 Why Errors are Interesting

Nobody likes to talk about errors, preferring positive performance statements, and yet they exist in all communications systems — a perfect link exists only in theory. Specifications do define error handling behaviour, but on reading any standard or textbook one finds the emphasis is always on such things as how many bytes make up a frame, or how routing works. Link errors are often only mentioned in two places: once where the physical link bit error rate is noted, and another where the checksum at the MAC layer is described as used to detect errors.

People consider errors differently depending on their area of expertise. The manufacturer of an optical modulator will consider the channel bit error rate. An engineer programming a device which will boot over a network will check whether the link interface is up or down. The developer of a network application worries whether some of the packets will be routed to the wrong place or will arrive too late to be of any use. These concerns reflect the different causes of network problems, and the varied metrics which are used to assess performance. The use of layering abstraction further obfuscates the ways in which errors are handled through the protocol stack (further discussed in Section 7.2).

Comparing network interface cards, one will pick the brand offering extra bit-rate — "New turbo plus extensions offer twice the speed! (when connected to a Brand X access point)" — and not even think about the error rate. In most consumer products this is acceptable; even if the bit error rate is above the expected threshold for the network type in question, this will just lead to a dropped packet once in a while, and a TCP retransmit will hide that this ever occurred. But in heavily loaded server interconnects, where huge blocks of data must be moved rapidly and with minimal delay, errors are more of a problem. A retransmission will occupy a link and delay the transfer; an undetected error could corrupt a vitally important file. In

these systems even operating at low bit error rates, such as one error every 10^{12} bits, errors will occur comparatively frequently as multiple gigabit or faster links are used to capacity. (On a single gigabit Ethernet link, this rate translates to one error on average every 800s.)

It has been shown that next generation optical networks will operate with limited receiver power, and that errors are therefore more likely. Clearly there is the potential to design error detection schemes for these systems, but to do this one must understand the errors and failure modes which will occur in order to protect against them. An investigation into the effects of low power states in computer networks on optical fibre is therefore both timely and interesting.

1.4 Outline

This dissertation describes work into the relationship between channel bit error rate and packet loss for optical networks using 8B/10B line coding. It particularly focuses on Gigabit Ethernet on fibre, 1000BASE-X, as in terms of data framing and coding this resembles the SWIFT network prototype, the development of which originally motivated this work.

Chapter 2 begins with a short review of relevant research into optical packet switching systems. In Section 2.2, a novel packet switched optical local area network prototype, which provides context and motivation for the investigations in subsequent chapters, is described. In Chapter 3, line coding schemes, cyclic redundancy checks and checksums are discussed, and a review of situations where the use of layering abstractions has lead to problems is presented.

Chapter 4 introduces the main work of this dissertation, with a theoretical study of the effects of a single bit line error within an 8B/10B encoded Gigabit Ethernet frame. This analysis is enhanced with information about actual implementations of the 8B/10B decoder, making it directly relevant to deployed systems. An experimental consideration of error behaviour in Gigabit Ethernet is described in Chapter 5, with an investigation into the errors that arise as a result of low optical power margin in a link transporting uniform data payloads.

Chapter 6 considers the effects of the error characteristics observed experimentally on real network traffic, and potential causes of packet loss. This is then combined with an understanding of the 8B/10B coding scheme to develop a novel mapping from channel bit error rate to packet loss, for both uniform and Internet traffic. Finally, the work described in this dissertation is summarised in Chapter 7, with a discussion of its implications in terms of the use of layer abstraction. The impact of these findings on the design of the next generation of optical networks is considered.

Chapter 2

Development of a New Prototype Optical Network

This chapter describes a number of technologies which may be used in the development of the next generation of optical networks for computing applications. It begins with a short review of relevant research into optical packet switching systems, and the various ways in which multiple wavelengths can be used. A discussion about why optical networking is increasingly being investigated for use in local area networks and interconnects is also presented. The second section describes a novel packet switched optical local area network prototype, which provides context and motivation for the investigations in subsequent chapters. This network uses wavelength striping, a semisynchronous timing scheme, and centralised MAC scheduling making use of a control channel on a separate wavelength. Such complex networks may be more susceptible to data errors at the receiver. In this prototype system, data is encoded for the physical channel using the 8B/10B block code.

2.1 A Review of Some Optical Networking Technologies

This section documents research in the field of optical networking, concentrating on work relevant to short haul optical interconnects for computing applications, which are the focus of this dissertation. The complexity of such networks is illustrated, along with some of the issues which affect their implementation.

2.1.1 Optical Packet Switches

A switched network can be more flexible and useful than a passive one in most circumstances. The network bandwidth can be fully allocated to the hosts most in need of it at any given time. In addition, whereas a passive optical network (PON) allows all users to observe all downstream traffic (leading to potential security issues), switched networks when operating correctly do not. Packet switching is particularly desirable to support IP traffic, whereas optical switching has traditionally been circuit based. Various methods have been proposed for transporting IP over the traditional circuit-switched optical networks of the core. The two main models for this are overlay (where IP networks are carried on top of optical layers, which simply provide transport and run their own protocols) and peer (where IP and optical networks collaborate and run the same protocols for routing and so on). In either case, circuit switched optical networks usually give inferior bandwidth efficiency when IP is carried over them [3]. Other work on Internet traffic in the core network has suggested using optical circuits to directly support TCP flows; this exploits the advanced stage of research into optical circuit switching [40]. However, this work concentrates on optical packet switches and their use in the edges of the network (such as LANs) rather than the core.

A conventional IP router performs three tasks - maintaining *routing* information, determining where to *forward* each packet to, and *switching* the packet to the correct output. A full optical packet switch will accomplish all these, and there is a wide choice of internal structures and systems.

The structure of an optical packet switch consists of some combination of switching, buffering, and header translation stages (although buffering may be avoided with a full contention avoidance scheme throughout), and control logic to oversee all these stages. The switching of a packet from an input to the correct output may be accomplished by conventional space switching, or filtering by wavelength (possibly including wavelength conversions at input and/or output) [41]. The balance between these components depends on routing strategy, bit-rate and network usage patterns [42]. Variable length packets present greater challenges than fixed length ones, as all lengths must be catered for [43]. Other features of a switch might include packet delineation and synchronisation if needed, and wavelength or format conversion if required for transmission on the output side [44]. Regeneration up to the 3R standard (reshaping, retiming, and reamplification) of the signal may be required at input and/or output; this has been demonstrated with an all-optical system [45, 46].

Control of Data Transmission and Switching in Optical Networks

Traditional packet switching requires a header or label containing addressing information to be transmitted with the data in some way. In optical networks, this header may be sent

serially with the payload, either at the datarate or slower to ease processing, or multiplexed onto a subcarrier transmitted in parallel with the data, or on a separate wavelength. Serial transmission allows the header to arrive a little before the main packet, so there is time for the switch to read the header information (and replace or process it) before the payload arrives and requires switching to the correct destination. This may remove the need for optical buffering, especially if it is possible to add a guard band between the header and payload. Subcarrier multiplexing may be performed electrically or optically [47, 48], and the header may be sent at a lower bit-rate, reducing the cost of processing electronics; it may also be coded differently, perhaps to aid clock recovery. In parallel transmission schemes, care must be taken to ensure that the time taken to transmit the header remains less than or equal to the payload time so that the header does not overrun.

Optical label switching technologies have been much discussed and range from Multi-Protocol Label Switching (MPLS) to All-Optical Label Switching (AOLS), where routing and packet forwarding occur in the optical layer. Many technologies have been developed to allow optical labels to be removed and added, so that the processing of the contents of the header (address extraction, routing table lookup, etc) can be carried out electrically [49]. With some extremely simple packet and address formats, and careful optical manipulation, it may be possible to produce an automatically self-routing network, with the addressing bits of the packet header used to set a basic switch to the correct configuration [50]. In most circumstances however, the required header processing is too complicated for currently available all-optical logic processing systems, which would add cost and complexity to a system. Even if the data payload remains in optical form through the switch, some control infomation must be converted to electronic form to control the switch (and optionally to be processed and retransmitted) [51].

One packet switched network is that of the KEOPS project, in which IP packet switching and WDM are combined, so that the switching occurs in the optical domain, with forwarding and routing left to electronics [52]. The optical side uses a broadcast and select switch architecture, with multiple wavelengths used for internal switching, but not for external links or contention resolution [18]. Another optical packet network is the WASPNET project, which uses tunable wavelength converters together with arrayed waveguide gratings (AWGs) and shared delay lines in order to provide contention resolution at the inputs to a space switch [53]. Lu *et al.* [54] proposes a packet routed system with a synchronous, distributed control scheme to avoid contention without the need for optical buffering. Each packet consists of data on eight wavelengths, plus single-pulse WDM header information on further wavelengths. Semiconductor optical amplifiers are used for both switching and loss compensation; the end-to-end data path passes through many such stages, which may contribute to an increased overall error probability.

2.1.2 Multiple Wavelength Systems

Wavelength Division Multiplexing (WDM) enables the use of the installed fibre base for much higher bandwidths than originally planned, by allowing several information channels, coded onto different wavelengths, to be transmitted along one fibre. Commonly in WDM systems, the wavelengths carry independent data and may be routed to different destinations, resulting in a switching granularity of one wavelength. When combined with Time Division Multiplexing (TDM), WDM enables greater granularity of bandwidth allocation, as wavelength channels can be shared [55]. Also, *waveband switching* has been proposed, where groups of wavelengths (of equal or varying number) are switched together, providing more flexible allocation of bandwidth and perhaps more efficient switching systems, as the port count is minimised [56]. Optical Time Division Multiplexing and multiple fibres, such as ribbon fibre, also add ways of dividing bandwidth between users.

WDM systems may employ Coarse wavelength division multiplexing (CWDM), where a smaller number of individual wavelengths are spaced further apart and actual transmit wavelengths may fluctuate slightly (allowing less expensive uncooled transceivers to be used) [57]. An alternative is DWDM (Dense WDM), where many wavelengths are used and they are tightly packed onto well-defined specification "grids". An example is the ITU DWDM grid, which separates adjacent wavelengths by 100GHz, equivalent to approximately 0.8nm [58, 59].

Much work has been done to try to bring the bandwidth benefits of WDM to the LAN, even suggesting systems to run over multimode fibre (the legacy installed base) [60]. To date, a typical WDM LAN might have used a broadcast star architecture, with a passive coupler to connect all the endpoints, and tunable transmitters and receivers. An example might be the 1997 IBM project Rainbow-II aimed to create a fast (1 Gb/s) local area network between powerful computers, running real applications. This used fixed wavelength transmitters, tunable receivers, and a passive star coupler at the centre. To improve performance, the network nodes performed much protocol processing rather than the computers themselves, to ensure the end users could take advantage of the maximum available bandwidth. Although the network was fast, it was circuit switched which led to very poor performance for TCP/IP applications [61]. WDM rings have also been considered, whether based on a broadcast and select system as above, wavelength routed, or based on token passing [62]. Current work on short-haul WDM components promises low cost parts, including uncooled components, which are desirable for use in LAN applications [63, 64].

Wavelength-routed Networks

Wavelength-routed networks have been the subject of much discussion, whether the routing occurs throughout the network or merely within a switch fabric [52]. The components are

fairly simple and cheap; many different architectures can be used, and the network can be "transparent" - i.e., any type of modulated signal can be supported. This does however suggest that the signals are interpreted as analogue, and so advanced digital techniques cannot be used to enhance them [65].

It is interesting to note that many of these much-discussed networks might be hard to implement, as optical amplifiers change their gain in response to a change in the number of wavelengths passing through them, and most of these networks would require amplification somewhere. When a wavelength is dropped from a set, leaving a reduced number of channels at an EDFA, there is a large power transient in the existing wavelengths, leading to bit errors at the receiver [66]. It can take a very long time on a bit scale — perhaps up to milliseconds — to recover from this, which could greatly reduce the effective network efficiency. A similar effect occurs when semiconductor optical amplifiers are subject to power variation [67]. These effects would also give poor performance for a system working with packets of data, which are likely to be short compared to the error period each time a wavelength was added. Other schemes, using tunable lasers and filters to create a network where different colours can be used to transmit data from specific sources at certain times, would also be problematic in a packet data network. The tuning times for lasers and filters to a settled wavelength are many times longer than would be suitable for a packet timescale, presenting problems for wavelength-routed schemes [53]. Fast tunable components are costly, which may make them unsuitable for LANs with strict cost limitations. In either case, the bandwidth cannot be used very efficiently as certain wavelengths are reserved for specific routes and bursty traffic would thus be handled poorly.

Wavelength Striping

Also known as optical bus or bit parallel coding, wavelength striping was first proposed for use in LANs in 1988, to reduce the quantity of high speed circuitry required [68], and is an alternative to conventional WDM. Wavelength striping also reduces the maximum serial bit-rate required; this is a particular benefit when the electronics at transmitter and receiver are considered. Multiple transceivers at lower bit-rates may be simpler to implement and integrate into a system, and/or cheaper than a single faster unit (see Sections 5.2.2 and 7.3.2).

There are various ways of splitting information for transmission between wavelengths. One would be to split each data word into its component bits, and to code each bit into a pulse on a different wavelength, all pulses then being sent simultaneously. This is analogous to the transmission of parallel data on an conventional parallel electronic link, and could allow existing parallelism in the electrical domain to be exploited, reducing serialisation/deserialisation delays. Headers may be sent separately from payload, either on a single wavelength or with each divided between a number of wavelengths; the header information is then easy to fil-

ter out for examination [69]. High-speed processing systems have been proposed to handle wavelength-encoded data, including all-optical logic gates [70, 71, 72]. The additional dimension provided by wavelength coding can also be used to implement error correction schemes, which can be implemented efficiently in parallel form [73]. The behaviour of optical amplifiers in the presence of fluctuating power levels can be compensated for by coding across a number of wavelengths such that the overall power level is maintained [67, 74]. Further discussion of multiple wavelength encoding schemes can be found in Section 7.3.2.

In many applications of wavelength striping it is desirable to compensate for bit timing skew between wavelengths; various methods have been proposed for this depending on the type of delay (due to wavelength drift or fibre length variation, and so on) [75]. These may use training sequences to inform the receiver of the wavelength-specific delays, followed by tunable electronic delay lines to realign the data [76]. An alternative is to integrate the skew compensation system with the bit clock recovery [77]; a variant of this which may be suitable for long distance links, if less than one bit time of skew is experienced, is to sample the received channels using a high speed optical switch [78].

One obvious competing technology is the use of parallel transmission on multiple fibres, perhaps using ribbon fibre to reduced cabling complexity. This may permit cheaper transceivers to be used, as wavelength stability is not required; however, it does have disadvantages. It is difficult to manufacture ribbon fibre which can maintain accurate time alignment across fibres [79]. Ribbon fibre also does not scale well to many integrated fibres, as the multiple-fibre connectors become very complex and costly (this is especially true for single-mode fibre, for which coupling is more difficult than for multi-mode) [76].

2.1.3 Optical Local Area Networks

The need for very high speed and low latency data transfer in local area networks or systems interconnects is clear, as processing speeds grow and an ever-increasing mass of data must be stored, retrieved, processed and communicated. Processing speeds increase continuously; the cost of RAM drops, and new technologies to store huge volumes of data are invented. Optical links can provide the required bandwidth, and are increasingly feasible in the short haul realm, utilising components which offer low cost and low power. In some applications, other optical technologies (originally from the world of long-haul networking) such as packet switching and multiple wavelength systems, may also be considered.

Power and cooling are major issues for server farms and central offices. Conventional routers with copper-based backplanes must be supplied with many kilowatts of electricity and produce great amounts of heat which must be dissipated. Both the power needed to drive computing and switching equipment, and the cooling systems which are required to

support it, are costly [80]. Each Internet router speed upgrade consumes yet more power; network operators are limited in terms of the power they can supply and the heat that can be dissipated from a single rack, but multi-rack solutions using multiple stage switch fabrics may have unpredictable throughput [81]. An optical backplane system allows router components to be spread out, reducing the power density demand and easing cooling requirements [82]. However, simply using optical links between electrical switches only reduces the power density requirement — extra power is needed for each of the electro-optical conversion stages [83]. Optical switching is one solution to this; in addition, an optical switch dissipates less power (regardless of switching rate) than a high speed electronic switch [81].

Power is also a consideration for large computing systems of various kinds. The demand for data storage continues to grow, as individuals and corporations record vast amounts of information each day [84]. Storage area networks (SANs), where bulk files are stored, backed up and accessed by powerful servers, require uncongested, high capacity links over ranges of just a few metres. Optical solutions offer the required bandwidth together with reduced system noise; switched systems where data can remain in optical form end to end can provide further flexibility and benefits. Other applications which might benefit from the bandwidth offered by optical communications systems include multimedia processing (such as film production and distribution [85]), medical and scientific imaging, CAD-CAM systems and distributed processing. On a supercomputer scale, switched optical networks to connect multiple processors and shared memory have been proposed before (e.g. [86, 87]).

Within each computer, the backplane (connecting components together) can be thought of as a small scale local area network, rather than treating the computer as a collection of units fixed together with a bus, and with a separate link to the outside world. It might be more appropriate to connect the parts together — and to external systems — with a network, which brings new flexibility and adaptability to both conventional computers and other home or workplace equipment [88]. On a still smaller scale, it is now possible to construct silicon devices capable of producing continuous wave light, and of modulating light [89, 90]; optical chip-to-chip links may one day replace traditional parallel copper buses [91].

So these short haul systems are a growth area for optical networking, presenting a range of different challenges from conventional optical systems. Integration with electronic systems, together with small size and the potential for cheap, volume production are important for commercial success. In addition, low end-to-end network latency and limited power requirements are desirable.

2.1.4 Prior Work on Optical Networks Between Computers

The market for server connections using the high bandwidth available from fibre is large, although it is much harder to see when most desktops will require this level of connectivity, even with intensive applications such as streaming video. One project from NTT shows that this side of optical local area networking is still being considered, with a "Fibre to the Notebook" Cardbus interface, using a single fibre with different wavelengths for receive and transmit [92]. Despite only running at 100Mb/s, this illustrates that the low cost and low power parts (and Cardbus devices have very stringent power requirements) that are required for optical networking in the end user arena are already available.

Currently, commercially available computer networking solutions using optical fibre fall into three categories. They are either point-to-point links, broadcast systems (using passive signal splitters) or switched networks where the switching is performed in the electrical domain. This latter type requires data to be converted to electronic form for processing and switching, then conversion back to optical form, at each switching node [51]. Gigabit and 10 Gigabit Ethernet are both switched networks (whether over optical or electrical media), and run over fibre only from an endpoint to the switch, or switch to switch, before being converted back to the electrical domain. Other competing serial interconnect systems using fibre, such as FiberChannel, InfiniBand, and so on, all use electronic switching [93, 94]. These very high speed networks, such as SANs, can suffer from bottlenecks at the endpoints, where data must be marshalled into packets for transmission and vice versa. Various methods have been proposed to reduce this, such as user-level networking [95]. The approach taken to this issue in the SWIFT network is discussed in Section 2.2.1.

Local interconnects, especially within servers processing heavy data loads (such as encryption, streaming video, or graphics rendering), also need very high bandwidth, and the lowest possible latency. This is also true for machines which must handle communications data coming in over increasingly fast connections. Recently, local I/O standards have been moving towards switched serial systems, which can transport data over longer distances and at lower power than their parallel ancestors. One example, the PCI replacement PCI-Express (also known as 3GIO), supports transport over fibre optics as well as copper; a copper system might offer 16Gbytes/s, and a fibre system with greater line-rates would surpass this [96, 97]. The need for the bandwidth offered by optical systems is clearly present.

Most experimental optical packet networks that have been built use pseudo-random bit sequences as packets; the use of genuine data traffic from actual computers is extremely rare. The MONET project [98] does use PCs as network endpoints for an optical label switching testbed, but, in published work, only reports the use of a specially-written application to manually create packets one by one for transmission. It is assumed that the latency of the network is poor; certainly no traffic of a realistic intensity or pattern has been used for testing [99]. It

seems unlikely that this would be a useful network as described.

2.2 The SWIFT Network Prototype

Thus there are a number of computing network applications which can benefit from the bandwidth gain offered by optical links, and optical switching brings further advantages to some of these systems. A novel prototype network which has been constructed using some of these technologies is now presented. As well as detailing the motivations behind the project, I describe the design and development choices which were made in light of the previous work in the field (such as the technologies noted in the previous Section). The design decisions leading to the network architecture as implemented in this prototype are relevant here, as they provide the context for the investigations in subsequent chapters.

The SOAPS project, a collaboration between the University of Cambridge, Intel Research and others, aims to build a local area network with a switched end-to-end all-optical data path [100]. The design targets ultra-fast packet transfer between computers using a simple architecture, and aims to achieve high utilisation of the network with minimal latency. The project will fit into the Data Link and Physical layers of the ISO stack, in a similar manner to Ethernet, and will be capable of carrying conventional TCP/IP traffic. It will be possible to exploit the increasing availability and feature range of low cost optical devices in the design.

The prototype network built for this project is the *SWIFT* system, which carries real application data between three computers. The construction uses a modular design which is constructed from commercially or near-commercially available components, with the aim that ultimately these can be replaced by smaller-sized and integrated components. Establishing the architecture of this network was the early focus of my PhD work, and investigations during the course of my design and development of the early endpoint and MAC architectures motivated the work described later in this dissertation. The goals of the network, and aspects of the architecture described here, were developed in collaboration with others. This prototype also illustrates the complexity of such packet switched optical networks, which increases their susceptibility to errors.

Advantages of Designing an Optical LAN

In the previous Section a number of factors driving development of optical systems for use in local area computing applications were noted. Another reason for targeting short haul systems is to limit optical problems due to non-linearities, e.g., dispersion, and to reduce the transmission power requirements, so that amplification and dispersion compensation are not required. Regeneration and scaling issues, which are studied widely elsewhere, are not of

concern here [100].

We wish to keep the data in optical form end-to-end within the network, to eliminate unnecessary electrical/optical conversions. (These would require the use of additional transceivers, which in turn would require power and add cost to the network.) However, we recognise that the processing necessary to successfully route data through the network is not yet possible using optical components [28]. Even conventional electronic memory parts are pushed to their limits to process and buffer data at today's line rates of 40Gb/s and beyond [101]. For now, then, the network management tasks must be performed electronically (as in the POND testbed, for example [102]). Information from the hosts will still be required to control the switch, but to avoid using any electrical connections between the hosts and the switch, a separate optical channel will be used.

Optical buffering is currently restricted in practice to fibre delay lines, offering multiples of a fixed time delay, which might be useable for packet data in some circumstances but which add loss and noise [103]. It is therefore preferable not to buffer the optical data within the network; all data buffering must occur at the endpoints prior to conversion to optical form. LANs can offer good utilisation without needing large amounts of network buffering, and remove the need for inband processing of data (which may be required for larger scale systems). A single-hop LAN avoids the buffering issues of a multi-stage switch, and requires only edge buffering, which can be done electronically before the data is converted to optical form. Issues related to network boundaries, such as Quality of Service, security, trust and so on are also avoided, which is advantageous as dealing with these is computationally intensive and would have to be handled electronically (until significant advances are made in optical computing).

An Overview of the SWIFT Network

The SWIFT network connects three PCs, and carries TCP/IP traffic. It therefore specifies the physical and data link layers in the same way as Ethernet.

An overall system diagram is shown in Figure 2.1, which highlights design features which will be discussed in subsequent sections. Multiple, parallel wavelengths (wavelength striping) exploit the available fibre bandwidth and offer notably increased capacity over single copper line solutions. As no optical buffering is to be used within the network, a pre-transmission coordinated media access control system is used to avoid contention issues. This is based on a request-grant architecture, where hosts reserve bandwidth (in fixed length time-slot units) ahead of their requirements, using a dedicated control wavelength. The MAC scheduler is implemented in a central hub, co-located with the optical switch. Switching occurs at the boundary time between slots, and is performed using a semiconductor optical amplifier based switch.

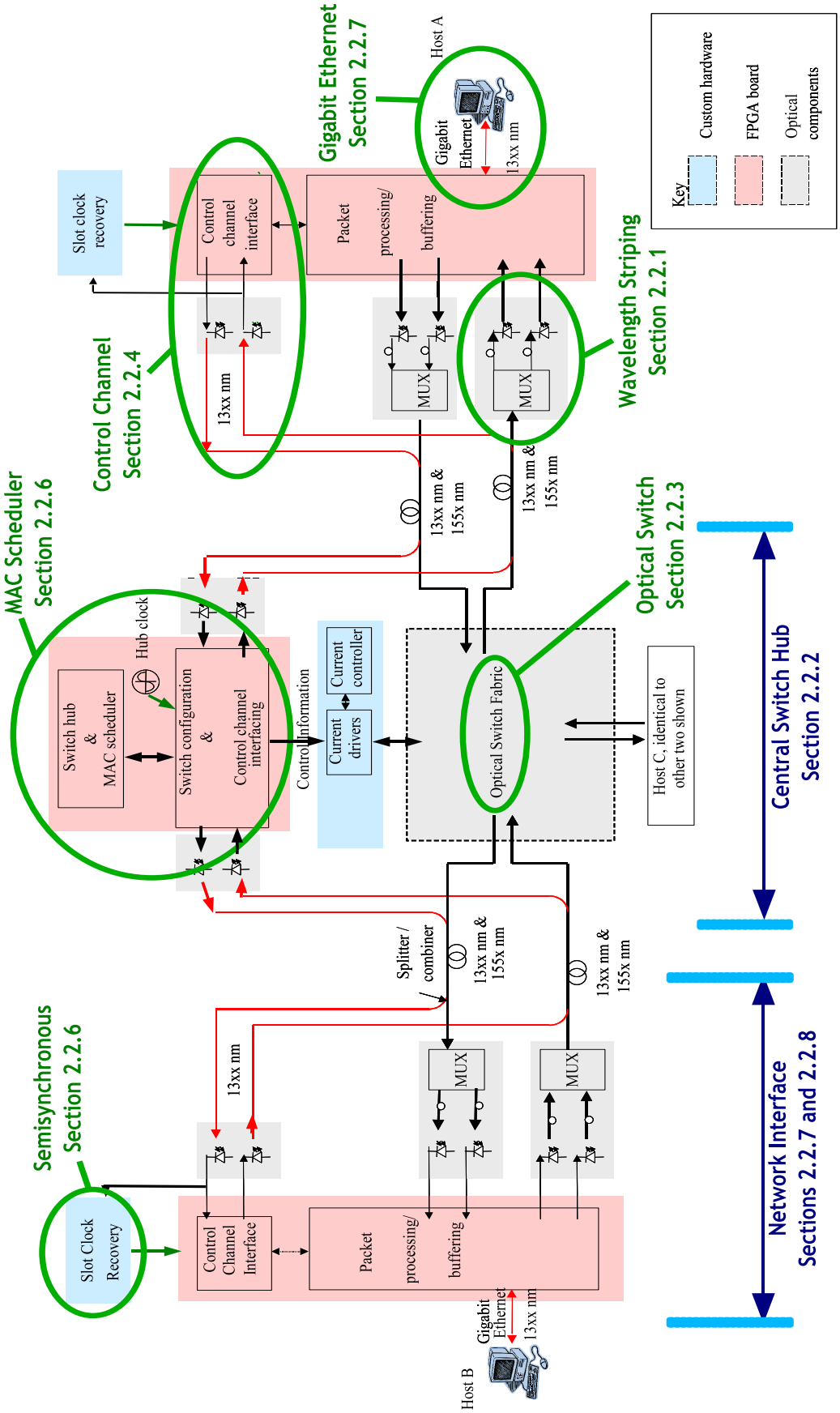


Figure 2.1: Overall architecture of the SWIFT demonstrator

2.2.1 Wavelength Striping

To maximise the use of the available fibre bandwidth, the SWIFT system stripes data across two wavelengths and is scalable to many more. We have selected a broadband switch, which can route all wavelengths simultaneously; the end-to-end data path is thus well utilised all the time.

Unlike most previous work on wavelength striping, the SWIFT network is focussed on the LAN, and so rather than concentrating on the overall bit error rate which would be relevant for long haul systems (e.g., [78]) we are more interested in minimising the timing errors which would greatly affect packet loss. Wavelength striping should offer reduced latency, by reducing the time spent in serialising/deserialising the data at either end of the network, and is thus ideally suited to a LAN where computer data will be supplied to the network interfaces in parallel form [104].

Another packet network has been proposed which uses multiple wavelengths in parallel to balance the load between channels; however, this is implemented using a single fast tunable laser, feeding into delay lines and a 2x2 crosspoint switch to shift the separate parts of the serially-generated packet into parallel [105]. This would be unduly complex and expensive to achieve, and also adds delay when serialising the wavelengths at the transmitter and deserialise at the receiver. The SWIFT demonstrator uses one laser and receiver per wavelength per node to attain this effect.

The data is striped across wavelengths in the 1550nm band which matches the switch passband and which can be separated or combined as required using AWGs or other devices. Wavelength stability can best be achieved by using cooled components to minimise thermal drift at the present time, and standard parts from the ITU DWDM grid are used. Uncooled lasers providing acceptable wavelength stability at comparatively low cost are under development and would be very suitable for this application [63, 64].

Striping permits the use of novel line coding schemes, which can bring benefits by easing clock recovery and compensating for power level changes in semiconductor optical devices (see Section 7.3.2).

2.2.2 Network Topology

The overall network topology is that of a star, chosen for high availability and layout convenience, spanning out from a central switching unit with bi-directional links to each endpoint. Figure 2.2 illustrates this layout, for an example case with a ring architecture for the switch in the central hub. This design gives considerable physical layout flexibility. The central connection could contain any form of all-optical switching system; only the control within this

switching system needs to know the internal topology of the switch, and the endpoints can treat the switch as a black box.

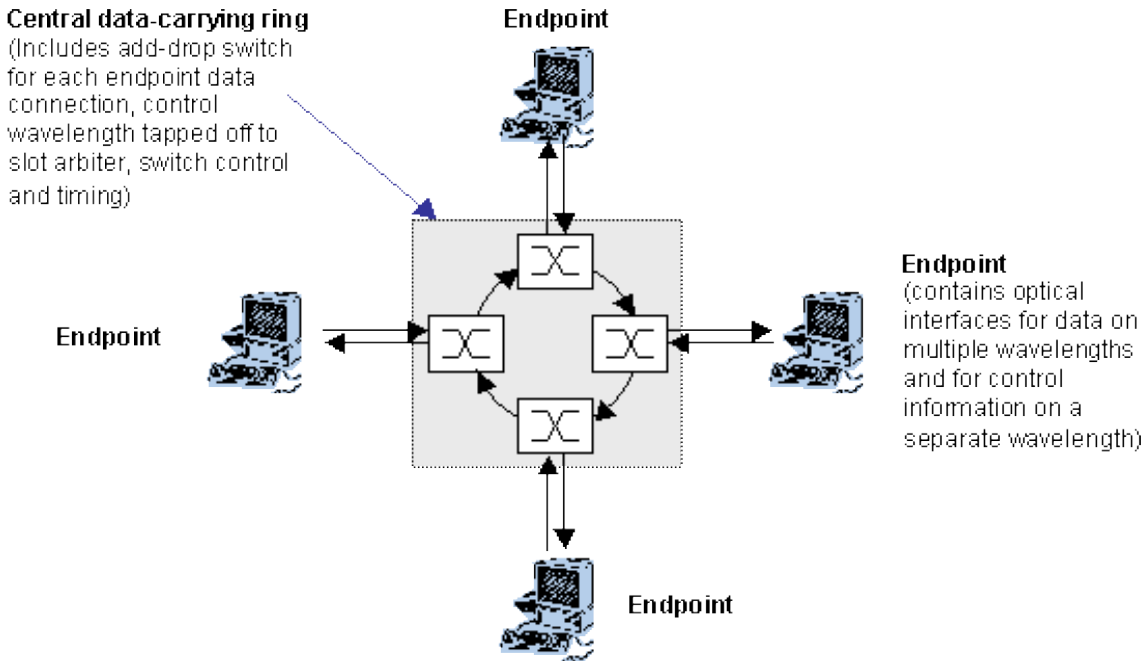


Figure 2.2: Overall network topology

2.2.3 Switch Design

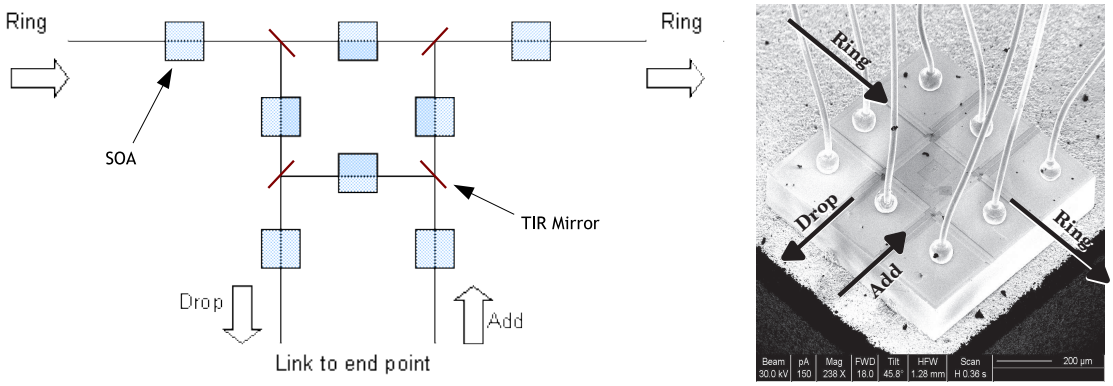
Initially, this system was to use an add-drop ring configuration, which is a well-understood form [106]. The pair of links from each endpoint would terminate in a 2×2 add-drop optical switch which connects to a unidirectional fibre ring (Figure 2.2).

The 2×2 add-drop switches are based upon semiconductor optical amplifiers (SOAs), capable of switching on nanosecond timescales and offering zero insertion loss with a good extinction ratio (Figure 2.3). They are thus suitable for packet switching at 1Gb/s data rates and above, where packets might be of the order of a microsecond long and so the switching time is a very small proportion of the packet time [107]. In addition, SOAs are electrically controlled, making them suitable for this network, which uses electronic systems for MAC scheduling. Even fast MEMS devices cannot be used here, as they do not get close to the maximum acceptable switching time of about 10ns [108].

The image of the switch in Figure 2.3(b) has the optical waveguides highlighted; the vertical connections are the wires used to electronically drive the SOAs.

All data wavelengths are switched simultaneously (as illustrated in Figure 2.5); SOAs are broadband, amplifying across a range of wavelengths, and can offer gain bandwidths of over 100nm [109]. For the first prototype, a line-rate of 1Gb/s per wavelength was used, and later

versions will probably move to 10Gb/s.



(a) Diagram of 2 × 2 switch (b) Image of 2 × 2 switch

Figure 2.3: 2 × 2 optical add-drop switch structure

However, packaging problems were encountered with the 2 × 2 SOA-based switches, and an alternative architecture was designed by Roberts *et al.* for the SWIFT prototype (Figure 2.4). This is a cross-bar style switch using the SOAs as gates (similar switch designs are proposed in Maeno *et al.* [110] and elsewhere). This is still based on SOAs; however, these are discrete devices rather than the integrated 2 × 2 chip [111]. Clearly this is not a scalable solution, but other integrated switches which would perform in a similar fashion should become available shortly.

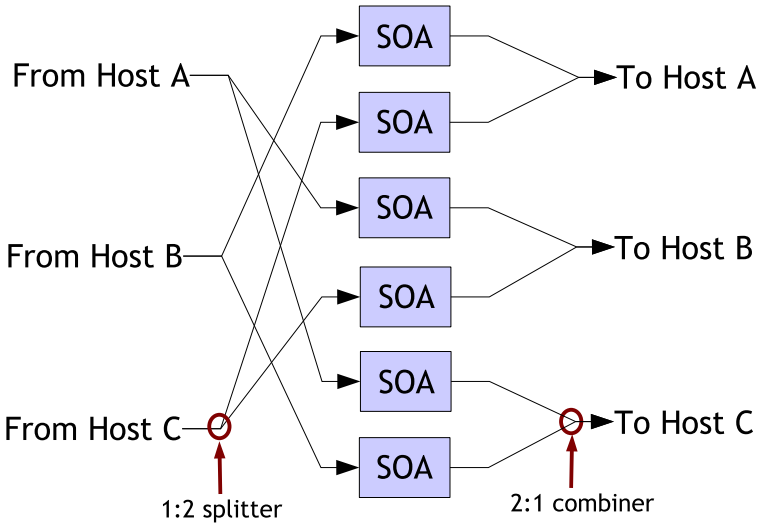


Figure 2.4: 3 × 3 cross-bar switch architecture using discrete SOAs

In this case, the set of SOAs, splitters and couplers together with control electronics makes up the central switch system.

2.2.4 Control Channel

Header replacement is not required in a single-hop LAN architecture, so optical label switching is not needed. However, some way of transmitting header information is required, as this is a switched network and the switch fabric needs to be configured to match the correct source-destination route for each packet. This network control information will be processed electronically, whereas the data will remain in the optical domain end to end, so it must be possible to filter out this information easily. It is simple to use a separate wavelength which will be shared between all the network entities for control, and which connects each endpoint to the central switch control unit.

If desired, this wavelength could be run at a lower datarate than the regular datapath wavelengths. This would keep the processing electronics cheap and should be adequate in terms of speed since there is unlikely to be as much control information as there will be data requiring transport (especially if multiple data wavelengths are used). Previously proposed networks sometimes run a distinct protocol, such as ATM, on the control wavelength [112]. Schemes with both wavelength striped data and striped control channels have also been proposed [113], but the gains of striping (additional bandwidth, reduced (de)serialisation delay) are not important for the SWIFT control channel. For simplicity of network design and testing, in the SWIFT prototype a control channel bit-rate of 1Gb/s is to be used, the same as on the individual data wavelengths. The protocol is a very simple one, designed specifically for this network [114].

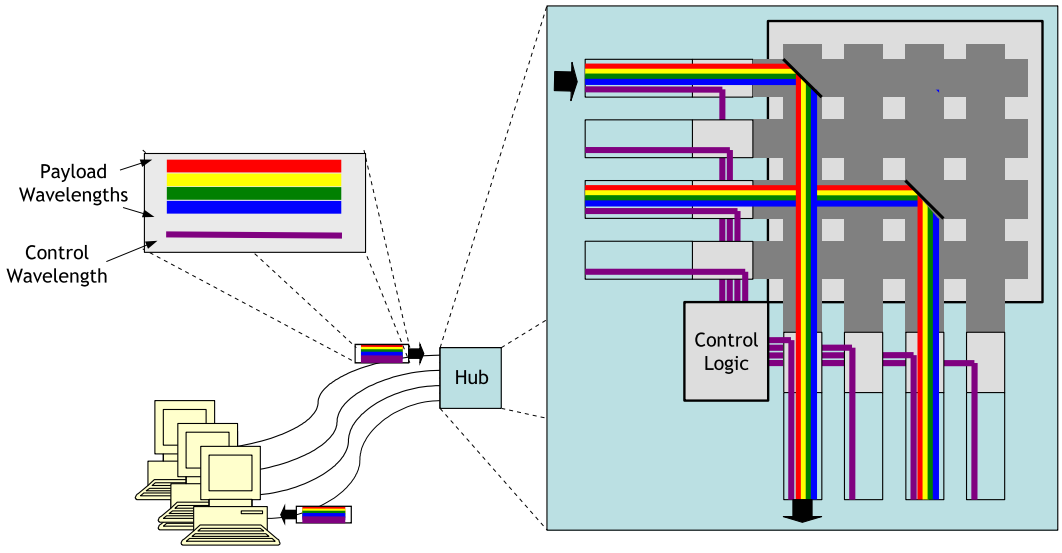


Figure 2.5: An illustration of the concepts of control channel architecture and broadband data switching

The control channel will run at 1310nm (easy to separate from the 1550nm data band) in the "star" fibres between the endpoints and the central switching system. The control wavelength will be separated from the data wavelengths before they enter the switch fabric, and routed to an interface which connects to the MAC scheduler and switch controller. Figure 2.5 shows

this configuration, for a non-blocking cross-bar switch architecture.

2.2.5 Semisynchronous Network Timing

Synchronous networks transmit a continuous signal across the system media. This assists with reception timing and enabling stations to lock their transmissions to the required clock frequency and phase. Asynchronous systems instead transmit data at the local clock rate, and receivers must reacquire the clock for sampling from the transmitted data for each received packet.

No suitable optical buffering is available to permit resynchronisation, so the SWIFT network is asynchronous at the bit level [104]. It is instructive to look at other examples of unbuffered networks, most of which are to be found in the wireless domain. Wireless LANs can use asynchronous schemes but longer range networks do not, as the jitter caused by radio propagation and transmitter movement means that the received signals do not arrive with sufficiently accurate timing for the recovered clock signal to remain valid.

Semisynchronous networks were developed to deal with this issue; these networks are asynchronous at the bit level, but use synchronous time slots for transmission. One example is the GSM mobile phone network, where 144 data bits are sent within a 156.25 bit time slot [115]. Optical semi-synchronous networks have also been proposed, and digital regeneration techniques may be used to deal with phase differences at the receiver [65, 116].

The SWIFT system uses a time slot scheme where the slot timing is synchronous, broadcast throughout the system using the control wavelength. The packets are transmitted one within each slot, at no particular time relative to the slot boundaries. This scheme has the additional benefit of an obvious time for switching to occur at the hub: on the slot boundary. The use of packets of data, sent within fixed time slots is similar to that of the KEOPS project, although without individual headers sent within the slot [52].

Timing Packet Transmission and Switching

Each packet is striped across multiple wavelengths, using an array of lasers, which is a much more elegant solution than using a tunable laser and fibre delay line to achieve the same effect [105]. The data is launched on all wavelengths as near to simultaneously as possible to reduce the skew time between wavelengths at arrival at the switch; however, some skew is to be expected due to dispersion in the fibre. In SOAs, as in EDFAs, changes in net optical power, averaged over all wavelengths, result in a per-wavelength gain change. This is not important on a packet time scale as it will only affect the preamble bits which are sent before the data on each wavelength (Figure 2.6). Wavelength coding can be employed to improve the power

balance across wavelengths; in this case, minimum skew is important to ensure the coding scheme works well [67].

During switching, for a period of perhaps 5ns, the switch outputs will be undetermined, and care should be taken that this does not occur when bits containing information are passing through the switch. Therefore the switching itself should occur in the guard band between packets, when no data is being sent.

On each wavelength, the data is preceded by a preamble (Figure 2.6), which will be used for clock recovery at the receiver. The data levels should be established by the time actual data bits arrive, such that errors within the data itself are minimised. The slope in Figure 2.6 represents group velocity dispersion in the fibre which causes some wavelengths to suffer slightly greater delays than others. Although this is shown here, as the SWIFT prototype is designed for short-haul applications the actual amount of dispersion should be negligible; other delays may be caused by environmental effects (e.g., temperature). The overall guard band period which is required to account for this skew still permits acceptable levels of utilisation [104].

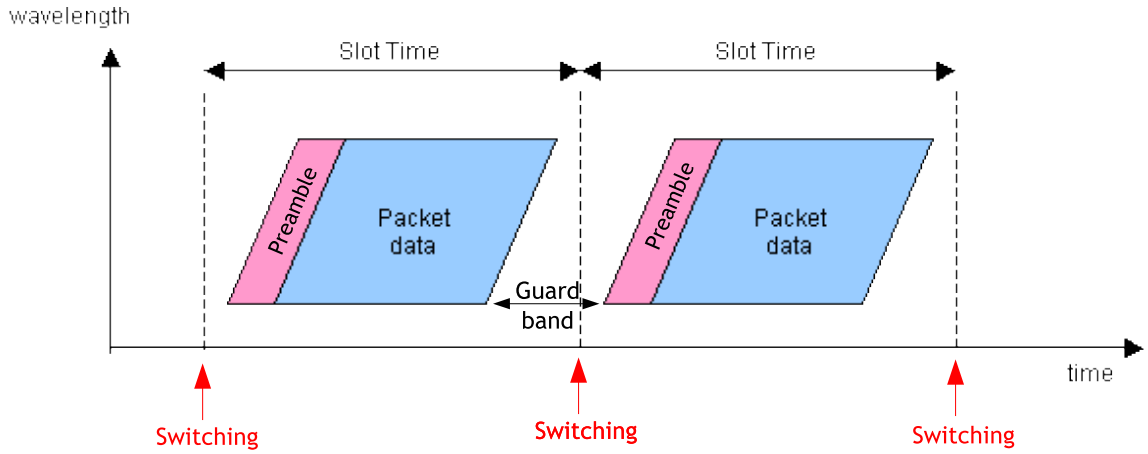


Figure 2.6: Slot and packet timing at switch or receiver

Burst Mode Clock Recovery

The asynchronous nature of this network at the bit level is challenging in terms of bit clock recovery, as the receiver must cope with recovering the clock for each packet, with a new phase and possibly slightly different frequency (from the independent clocks at each endpoint). It is desirable for the receiver to synchronise with the incoming data as rapidly as possible, so that the minimum length of preamble can be used, reducing the per-frame timing overhead. In addition, the receiver must "lose" the lock in a short time at the end of the packet, ready for the new clock to be acquired. This environment is called "burst mode" clock recovery (whether the data to be retrieved is sent in burst or packet format). Solutions do exist for burst mode clock recovery at gigabit rates; some examples may be found in hardware for Ethernet Passive

Optical Networks [117].

In the SWIFT prototype, the bit clock recovery circuits available on the network interface hardware (see Section 2.2.8) were used. These are phase-locked loops (PLLs) designed for point-to-point links, and give reduced performance in a switched system, but were already integrated.

Several other options, one or more of which could be used for the next version of the prototype, were also considered. Both different clocking schemes, and clock recovery schemes, which would work well at gigabit speeds and higher in the burst mode environment were contemplated. These are briefly detailed here; clock recovery is an issue which occurs again in subsequent work on error characteristics (Section 5.1.8).

Possible candidates for a clock recovery solution would be PLLs, an oversampling scheme, or some other more advanced digital coding technique. A PLL solution is likely to be difficult, as the acquisition time must be very short, perhaps 20 bits (20ns), and there will not be long for the loop to detune between packets. Assuming the system does not run at a very high bit-rate, serial oversampling may be possible as the sample rate may be reasonable; there are no repeaters in the system to cause cumulative error, and the packet lengths will not be so long as to contain significant frequency drift. An alternative might be a parallel oversampling scheme, where samples at the clock frequency are taken at a range of phase shifts. Each set of samples undergoes bit error rate (BER) testing, and the set with the lowest BER is taken to be sampled at the correct phase. If sufficient wavelengths are to be used in parallel, asynchronous coding (Section 7.3.1) could offer reduced preamble times, using only a single clock recovery unit.

There are also a variety of alternate clock solutions. A crystal company could have provided a batch of closely matched crystals, which could guarantee close to identical frequencies at each endpoint; then, each receiver would only have to compensate for a phase shift. We debated using a received clock for transmission, to try to hold the frequency the same throughout the network, but this may not have worked well with switched packets arriving at each node. An alternative might be to use a distributed clock throughout the system, perhaps at a reduced rate (e.g., a 10-bit symbol clock at 125MHz). The clock signal could be generated onto a copper connection and sent to each endpoint, or onto a spare “data” wavelength, or converted onto the control wavelength (see Section 2.2.4) and sent over fibre to create the appearance of an optical network not using synchronous endpoints. The endpoints could multiply up the clock if necessary for data transmission use, although this may add jitter.

2.2.6 MAC Scheduling and Switch Control

The optical switch system is co-located with the switch control electronics, the hub's control wavelength interfaces and the MAC scheduler, in the centre of the star (Figure 2.1). The control wavelength is filtered off the fibres coming in from the endpoints, so that data wavelengths are fed straight to the switch, and the control wavelength is sent directly to the hub interfaces without entering the switch fabric. Control wavelength broadcasts are spliced back in to the fibre links to the endpoints.

The control wavelength provides a channel for all endpoints to communicate to the central switch, and for broadcasts back out to the endpoints. The endpoints request data slots in advance on this control wavelength (c.f. [105]). When a slot (or slots) to a specific address is requested, the MAC scheduler located in the central switch block replies with a grant or refusal. If a slot has been allocated, the requesting endpoint is notified which slot it has been granted.

The control wavelength is also used to distribute the slot timing to the endpoints; nodes are then able to synchronise their own transmissions to this master time signal [118]. The slot timing also determines the switching times, and forms a part of the control information sent to the switch fabric. The network slot timing consists of both the actual start of each slot, and also global time information. This permits all endpoints, as well as the central system, to know which slot is currently in flight and to keep track of allocated slots in the future.

The MAC scheduler knows in advance the source and destination of the packet in each time slot, and is able to inform the switch control plane of the required configuration for each slot. The control system then turns on or off the appropriate SOAs within each switch at the start of each slot. The switch driver electronics may benefit from having this information a little in advance so that the switching time can be minimised.

The control channel and central MAC scheduler system together form the MAC layer for the SWIFT network. MAC development is a substantial component of the the project. Other work for similar networks proposes MACs ranging from Aloha and slotted-Aloha [119, §3.3], to more complex designs using multiple control wavelengths to increase utilisation of the network [120]. The MAC implemented at the time of writing employs a simple round-robin allocation system [111].

Advanced MAC Scheduling Options

Minimal latency is a particularly important feature in this network; bandwidth has been increasing rapidly for many years, but latency improvements have been limited in comparison, regardless of whether one considers memory or magnetic media access times, or microprocessor

performance. Prediction is an interesting way of reducing delays in a system [121]; a predictive bandwidth reservation system could be advantageous. Since all slot grants/refusals are broadcast, it will be possible for endpoints to detect network congestion and try to work around it. In addition, since the round trip time to request a slot and get a response may be long, endpoints may opt to reserve slots in advance of requiring them, so that as soon as they have data to transmit they are able to get some bandwidth. The delay between endpoints and the central system is analogous to satellite communications systems, where ground stations must wait a long time before receiving a reply from the satellite [104].

2.2.7 Gigabit Ethernet Technology Reuse

It had initially been hoped that much of the network could be built using Gigabit Ethernet components and/or network interface cards (NICs), as these are cheap, available, easy to use and run at a reasonable speed. However, it was unclear how this hardware would handle the optical switching setup, in terms of clock recovery, link setup, and limited transmission windows between switching. Standard Gigabit Ethernet is designed and tested only for use in a point-to-point configuration, where the bit clock and symbol clock need only be recovered from scratch when the equipment is first turned on. (Most Gigabit Ethernet Passive Optical Networks avoid this issue, as they only run Gigabit Ethernet on a "point-to-point" section of the network design, with specially adapted hardware used for the end nodes [30, 31].) Gigabit Ethernet's bi-directional setup signalling also presented a problem as the optical switch is not configured to provide a simultaneous return path. In addition, it would be necessary to alter the physical behaviour to transmit only within the slot times, and, if needed, to stop transmission during the switching period. The effort required to use Gigabit Ethernet components and work around any issues, given that no open hardware was readily available, led to the decision to create a custom physical layer and to build a NIC from scratch. This option also provides additional flexibility to modify subsystems as needed during development.

However, as many features as possible from Gigabit Ethernet and similar standards would be used, to enable the use of generic components where appropriate. The 8B/10B coding scheme adopted by Gigabit Ethernet is ideal as it is DC balanced, and the maximum run length of 5 zeroes or ones, and frequent bit transitions, improves clock recovery performance. The physical coding sublayer of Gigabit Ethernet and its benefits are described in more depth in Chapter 4.

2.2.8 Conclusions: The SWIFT Prototype

The SWIFT prototype has been constructed as shown in Figure 2.1. The effort required to implement PCI as the link from the NIC to the PC was felt to be too great. Instead the

prototype uses a standard Gigabit Ethernet link (1000BASE-LX) to communicate between the SWIFT optical interface cards and each PC. The control channel runs at 1310nm, with two data wavelengths in the 1550nm band; both operate at 1GHz. Xilinx Virtex-II FPGAs, with embedded microprocessors, are used to implement each of the endpoint SWIFT interfaces, as well as the central MAC scheduler system and control channel interfaces. Control of the discrete SOA cross-bar switch is performed by fast custom electronics. The overall system gives a round trip time of just 220-240μs, comparing favourably with Gigabit Ethernet between two hosts (a time of 150μs), particularly when the current round robin MAC scheme is allowed for. The project continues with advanced MAC protocol development, and testing with cheaper optical components, as well as speed improvements.

This prototype network, and the other packet switched optical testbeds described in Section 2.1, illustrate the complexity of such next generation systems. With an end-to-end optical data path passing through a large number of active devices, splitters and combiners, these systems are likely to be more susceptible to errors than has been the case for many optical links to date. High serial line-rates (if multiple channels are not used), and the desire for low implementation cost may also contribute to higher error probabilities.

Chapter 3

Context

A range of topics which are of interest for the study of the relationship between bit error rate and packet loss are reviewed here. First, line coding schemes, both scramblers and block codes, are discussed, and then work relating coding methods to error rate is outlined. Error detecting mechanisms, in particular Cyclic Redundancy Checks, or CRCs, are described, along with their uses in networking and issues relating to their use. Finally, this chapter reviews situations where the use of layer abstractions has lead to problems.

The first Section here reviews work on line coding and error schemes. Studies of errors or network performance tend to fall into two categories. One focuses on network layer issues (routing, packet fragmentation etc.). An example relevant to this work would be Finkler&Sidhu [122], which investigates the performance of Gigabit Ethernet operating in half-duplex mode, in terms of throughput, transmission time and so on, in light of the alterations made to the CSMA/CD protocol. This work is mainly interested in switched short haul networks, so detailed analysis of collision behaviour and routing decisions is not very relevant. The other type of study concentrates on low level error issues, and often discusses the performance of line coding systems in conjunction with this; these are described in Section 3.1.

In Section 3.2, the checksums and cyclic redundancy checks used in networking systems to detect errors are discussed. Finally Section 3.3 highlights a number of instances where problems have arisen from the use of layer abstraction.

3.1 Coding and Errors

Shannon [123] showed that for any noisy channel, there exists a code for which the probability of data error can be arbitrarily low. However, there are more criteria for code selection than

noise immunity - cost of implementation, acceptable redundancy levels, etc. Hamming took a more practical view of coding schemes, giving his name to a class of binary error-correcting codes [124]. Error control coding schemes such as these allow the bit error rate to be improved over the value for raw, unencoded bits.

The type and number of errors which are likely to occur depends on the channel and other systems involved in the communications link. Any error control method must be tailored to provide the best protection for a given network. Commonly, noise is modelled as an additive white Gaussian process, which is straightforward to analyse [125, §1.1], and usually represents some portion of the channel reasonably well. However, this model does not account for errors from other sources, such as interference which may be data dependent, transients which may last longer than a single bit, or systematic errors and so forth. If burst errors are likely, causing damage in more than one symbol, then special convolutional or cyclic error correcting codes may be selected [126, §14.5]. If a two-way link is present, then errors may not need to be corrected, but only detected, as a retransmission could be requested for the incorrect data; whether or not this is an appropriate solution is very application-specific.

However, this work is interested in optical networking systems where error correcting codes are not presently in use. Gigabit Ethernet, and the SWIFT prototype of Chapter 2, use a simple block coding scheme at the physical layer, with a per-frame 32 bit cyclic redundancy check (CRC) which detects some error patterns. CRCs are discussed in Section 3.2; firstly, work in the field of block codes and their error performance is considered.

Using a transmission code improves the resilience of a communications link, by ensuring the data stream has known characteristics that are well matched to the physical behaviour of the link. A coding scheme must ensure the successful recovery of transmitted bits; often this requires a minimum number of bit transitions to occur for successful clock and data recovery. A balance of 0s and 1s can also be important, as high speed transceivers can suffer from distorted pulses and baseline wander if the DC component of the signal builds up. In general, coding schemes bring these desirable properties to a data stream by adding redundancy. Redundancy can also be used to add error detecting or correcting properties (see Section 3.2). Line coding schemes also often provide control patterns or codes, perhaps including codes to permit the start and end of individual frames to be detected at the receiver; they may also supply special sequences for synchronisation, either at a bit or symbol level. If parallel data channels are used (in either the electrical or optical domain), channel timing alignment information may be required, and a dedicated parallel coding scheme may be used (Section 7.3.2). Numerous mechanisms exist to convert data to be communicated into a form suitable for transmission; two of the more common types used in optical networks are scramblers and block codes. The coding scheme used will affect the way in which bit errors on the line will propagate up the network stack. The Hamming distance, or minimum distance, of a code is the smallest number of bits which differ between any two codewords [127, §13].

Scramblers

Scrambling is where the transmitter, using a reversible function, modifies the input data in a known way. The receiver can reverse the function and recover the original data. Given the desire to maintain a balance of 0s and 1s along with sufficient transitions to maintain clock synchronization, a scrambler processes the input data, ensuring that there are suitable numbers of 0s and 1s for transmission. The operation of a scrambler may be considered as the multiplication of the input data with a random number; the receiver need only divide the incoming bits by the random number to recover the original data. Scramblers do have drawbacks, in that a malicious user may engineer input data that will cause a long stream of 0s or 1s, or a special control sequence, to be produced. Aside from attacks such as these, the scrambler has an inherent latency delay of the length of the random number, and can be complex to implement.

Block Codes

Block codes are another popular choice; they translate s bits of data into x bits for transmission, where x is greater than or equal to s. Such block codes include the 8th-bit parity check of RS-232 serial lines. They may be implemented as a look-up table, making them simpler and lower latency than the algorithmic scrambler operation. The redundancy added by using more bits for transmission than are required means that problematic codes, such as all `0`s, can be avoided. In links subject to large amounts of noise or distortion, more complex decision-based decoding methods can be used. One example might be "minimum distance decoding", where a received symbol is compared against the set of transmit symbols; the transmit symbol with the fewest bits in difference to the received value is selected.

3.1.1 Previous Work on Error Rates and Line Coding

A number of studies have been done into the information-layer bit error rate for systems where block codes are used. It is possible to relate the information-bit error rate to the probability of error in a symbol at the physical layer, in terms of the code weights and the decoding system, but this is complex and the exact relation is unknown for most coding schemes. Torrieri [128] derives a simple approximation for the information-bit error rate, for the case of a block coded system which uses minimum distance decoding. Systems like those this dissertation focuses on use simple block codes such as 8B/10B (Section 4.1), with a different decoding method — if the received codeword does not exactly match one of the valid words, it is not decoded. In addition, the minimum distance of this code is 1 — codewords may differ by as little as a single bit. The expressions derived in [128], then, which depend on minimum distances between code

words, are not relevant to this study. Similar considerations are made in Torreri [129], which relates information-bit, information-symbol and decoded-symbol error rates for linear block codes. [129] studies both erasing decoders (where symbols containing uncorrectable errors are erased from the stream) and reproducing decoders (where such symbols are left in), and shows that the information-symbol error rate and the decoded-symbol error rate are not necessarily equal for noncyclic codes. Again, this work is not relevant to this study as 8B/10B is not a linear block code and the all-zero codeword is not transmitted. An examination of word-, symbol- and bit-error rates for a variety of block error-correcting codes is made in Desset *et al.* [130]; initially this sounds promising, with a consideration of real channel issues and both bounded-distance and nearest-neighbour decoding. However, this also only applies to linear block codes.

One study which considers the effects of channel errors on a system including a block coding scheme and frame structure including a checksum is Jain [131]. This analyses the frame error rate and other network effects for Fiber Distributed Data Interface (FDDI), which uses a 4B/5B block code. Another is Fiorini *et al.* [132], which analyses the vulnerability to transmission errors of the data link protocol HDLC, in terms of its line code and CRC; the relationship between bit error rate and residual error rate (the rate of frames with undetected errors) is established for this case. These protocols are clearly different from Gigabit Ethernet, which is of interest because of the similarities of its coding scheme and frame structure to those of many other optical systems (see Section 4.1). A similar analysis for Gigabit Ethernet on fibre (1000BASE-X) is performed in Chapter 4.

The work done on 8B/10B block codes to date has not considered the relationship between actual line errors and frame error rate. Widmer&Franaszek [133], in which 8B/10B is originally proposed, makes some consideration of errors but the coding and decoding methods differ from those of the Gigabit Ethernet specification and actual implemented methods (Sections 4.1.2 and 4.1.3).

3.1.2 Bit Error Rate Measurements

Bit error rate (BER) is a useful way of assessing a link's performance; it is the expected number of errors in a given number of bits sent, or the average probability of a bit error. However, there are issues with depending on it as a sole metric which are sometimes forgotten.

Firstly, as for any statistical measure, one must observe a sufficiently large sample of bits to have any confidence in one's measurement; even at bit-rates of 1Gbps or higher, amassing a large enough number of bits takes a long time. Sometimes, the eye opening (see Section 5.1.5) is used to estimate the BER instead of waiting; this is only useful if a random distribution (i.e., a Normal or Gaussian distribution, say) of errors can be assumed. If other effects influ-

ence the occurrence of errors, such as crosstalk or pattern dependency, the assumption of a Normal distribution is incorrect, and so this is not a reliable measurement. In addition, many oscilloscopes can only display a comparatively small number of samples which is of limited use, depending on the sampling rate of the oscilloscope, data bit-rate and storage capacity of the oscilloscope. Rare events which may cause errors can therefore be missed. Bit error rate testers (BERTs) allow the actual errors to be examined as they sample at the line rate; however, they normally assume a flat response at the receiver as well as a Gaussian error model. The effects of equalisation or any frequency response distortion are not taken into account.

Although bit error rate is one method of assessing link performance, it does need to be used in conjunction with a range of other system measurements such as jitter, and tested with a range of data patterns.

3.2 Error Detection Methods

Many data network and storage systems use checksums or cyclic redundancy checks to attempt to detect and/or correct errors before the data is used in an application.

A checksum is usually an easy-to-calculate value appended to a block of data, which allows errors in the data to be detected. As a general rule, checksums are designed with ease of processing in mind, rather than for robust error detection. They are usually used in conjunction with the stronger error check provided by a cyclic redundancy check (or code). Cyclic redundancy checks, henceforth CRCs, are often described in terms of base-2 polynomial arithmetic; they have more powerful error detection and correction capabilities, but are usually more complex to generate.

3.2.1 CRC Operation

The block of data to be protected by a CRC can be considered as a polynomial over a Galois Field, GF(2) — each coefficient is zero or one [126, §6.5,7.3]. A generator polynomial $G(x)$, specific to the CRC in use, is used to divide the data polynomial; the remainder gives the value which will be sent along with the data word. This value is often referred to as the CRC or frame check sequence (FCS). At the receiver, the calculation is repeated; if the stored and received FCS values are not equal, then an error is declared to have occurred. An alternative method is to perform the division over the received, concatenated data block and FCS; if the result is zero, then, probably, no errors have occurred. This type of arithmetic is linear, which gives rise to a number of useful characteristics; an example is that the sum of any two valid codewords is also a valid codeword. This means that when examining the ability of the CRC to detect certain errors, only the codeword consisting of the pattern of errors need be considered,

and not the many potential data codewords which these errors could occur in (Section 4.2.6).

The weight of a polynomial is a measure of its usefulness, and depends on both the polynomial $G(x)$ and the length of the data block to be protected. The weight W_i is the number of error bit patterns consisting of i error bits which would be undetected by the CRC. Any CRC will detect any single bit error, so W_1 is always zero; in most cases only the first non-zero weight is of interest, as this indicates the minimum number of bit errors which may go undetected by the CRC. This non-zero value is the Hamming distance of the CRC.

An error burst is defined as the group of data bits starting with the first bit in error and continuing up to and including the last bit in error, and all the intermediate bits (errored or otherwise). To be undetectable, the pattern of the errors through the payload and CRC must give a valid CRC polynomial (due to the linearity of the polynomial arithmetic); this means that only error bursts of longer than 32 bits are undetectable for a 32 bit CRC. Combined with the Hamming distance, it is found that at least the Hamming distance number of bits in error, spread out over more than 32 bits, are needed for an error to go undetected by the CRC. Fujiwara *et al.* [134] states that any double burst error consisting of two error bursts each of length b or less is detectable if and only if any single burst error of length b or less can be corrected. [134] also shows that any double burst error of two bursts each of 9 bits or less within a codeword of less than 13000 bits is detectable.

3.2.2 The Use of CRCs and Checksums in Network Applications

IP uses a 16 bit one's complement checksum to protect the IP header fields; this provides basic protection, against corrupt memory in routers, for instance. UDP uses an optional 16 bit checksum to protect data and header, and TCP a compulsory checksum over data and header — both are the same 16 bit sum as used in IP [135].

CRCs are widely used in network and storage applications, as they are easy to implement in hardware or software; the mathematics of CRCs allow a variety of optimisations to give extremely fast FCS generation and validation [136].

Ethernet, and other related LANs, use the standard IEEE 802.3 CRC32 polynomial as a link layer frame check sequence, which was originally chosen by Hammond *et al.* [137]:

$$G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$$

Fujiwara *et al.* [134] details the error checking performance of this CRC for a variety of frame sizes up to the 12144 bit, 1500 byte maximum size code word used in the Ethernet standard. In particular this work highlights the weight distributions for various common frame sizes, and the probabilities of undetected error (assuming a binary symmetric channel with

simple bit error rate). Jain [131] extends this work to demonstrate that the 32 bit CRC as used in FDDI and Ethernet is sufficiently strong as to detect all 1, 2 and 3 bit errors for frames up to 8 KBytes in length. However, frames of greater length than this, or containing more than 3 errors, may have errors undetected.

3.2.3 Issues Relating to CRCs

The commonly-used IEEE 802.3 32 bit CRC has a weight of $\{W_2 = 0; W_3 = 0, W_4 = 223059, \ldots\}$ for a frame of standard Ethernet MTU length (1514 octets) [138]. This means that all 2 bit errors, and all 3 bit errors, will be detected by the CRC; however, there exist 223059 4 bit error patterns (out of all possible 4 bit patterns) which would go undetected. It is common for this to be interpreted as meaning that the probability of a 4 bit error, in a frame of 12144 bits, going undetected is given by:

$$\begin{aligned} P(\text{undetected 4 bit error}) &= 233059 / \left(\begin{array}{c} 12144 \\ 4 \end{array} \right) = 233059 \times \left(\begin{array}{c} 4! \times 12140! \\ 12144! \end{array} \right) \\ &= 2.46 \times 10^{-10} \end{aligned} \tag{3.1}$$

This, however, assumes that all 4-bit error combinations are equally likely to occur; this may not be the case if the error patterns at the protected layer are non-uniform. (In Chapter 5 it is shown that errors in Gigabit Ethernet on fibre do not occur uniformly, and in Chapter 6 it is found that the non-uniformity of real traffic may exacerbate this.)

For the case of a single bit channel error in Gigabit Ethernet,it will be shown that up to 4 bit errors may be generated at the link layer (Chapter 4), but these will be located in a single data octet, and thus detectable by the CRC. However, this *error amplification* increases the probability that an undetectable 4 or more bit error will occur.

The probability of an undetected 4 bit error is also often dismissed as insignificant, since any given number of bit errors, X, is a factor equal to the bit error rate less likely to occur than one error less, $X - 1$. Given the usual low to moderate bit error rates for network systems, the probability of successively higher numbers of bit errors rapidly becomes extremely unlikely. This is not necessarily a sound argument, though. As shown by Stone and Partridge, network packets containing errors far more often "pass" the link layer CRC than would be anticipated by these probabilities [139]. This is discussed in the context of related experimental results in Sections 5.1.6 and 5.1.7.

The selection of a CRC for an application is increasingly challenging, as many systems begin to support the use of much longer frames, whilst still carrying many short frames. A polynomial which offers excellent error protection at standard Ethernet packet sizes, say, as well as good properties for much longer frames such as 9000 byte jumbo frames, is required for

these applications. These are rare, as most CRC polynomials offer either good performance for short frames or for long frames, but not both. Koopman [138] identifies a class of polynomials which might be suited to this, but these could only be used in new systems, or run at an application level on legacy network hardware.

3.2.4 Statistics Relating to CRCs and Checksums in Real Network Data

The 16 bit checksums used in IP, TCP and UDP headers in a range of samples of real network data were examined. The network samples used were 24 hour traces from an ethernet access link between a large residential academic institution of 10,000 users to the internet.

In cases where checksum usage is disabled, the checksum fields consist of 0x00 octets. In all other cases, the checksums observed contain all possible octet values in approximately equal proportions - no particular values predominate.

For one trace, more than 1 in 65,000 UDP packets had the header checksum disabled. At one point both vendor recommendation and common practice was to disable UDP checksums, in the belief that the link level data checksums would be adequate protection for the data. This led to data corruptions for UDP-based network file system (NFS) and network name (DNS) services [139, 140, 141]. These results show that some equipment still operates without this additional error check. This is an example of a situation where problems arise after assumptions are made about other layers in a network.

3.3 System Design: Layering

Layering is a good way of going about network design, as discussed in Section 1.2.1. It permits architectural responsibilities to be clearly allocated, and offers the flexibility of a modular design, where a layer can easily be replaced by an alternative system. Abstract service primitives can be used to define the flow of communications information between layers, and the actual functions within each layer. These specify functionality but not implementation and permit developers to work on one small section of the overall stack without affecting others.

However, there can be problems with the way in which the layering principle is sometimes applied. Issues where layering has caused unexpected network behaviour have been noted before. These problems tend to arise where layer interfaces are poorly defined and misinterpreted, or misunderstood. For instance, Tennenhouse [142] observed the unfortunate effects of layered multiplexing on application jitter. Chakravorty *et al.* [143] provides a example of layer interactions between a packet network (GPRS) and transport protocols that rely on end-to-end

feedback loops (TCP). Crowcroft *et al.* [144] describes the behaviour of the remote procedure call (RPC) over TCP, affected by a data transfer problem between two layers of an implementation of the network stack. The difficulties of combining efficient implementation with layered abstraction design are also noted in Clark&Tennenhouse [145], where the possible benefits of integrated layer processing to reduce data manipulation requirements at the transport layer and above are discussed.

Another example of unfortunate layer interaction arose when SONET began to be used to carry packet data, such as ATM or IP. SONET uses a 7 bit scrambler for data payloads, to ensure sufficient line transitions for clock recovery purposes. This is adequate when SONET carried byte multiplexed payloads; however, when packet data is transported, user data fills a larger amount of the SONET frame, and with knowledge of the scrambler operation a malicious user could take control of the SONET envelope. By transmitting specific payloads, such a user could cause a long string of data without any transitions to be sent on the line, which may cause framing and synchronisation problems, and potentially a loss of signal indication; this type of attack could also be hard to trace [146]. To combat this, additional payload scrambling using an $x^{43}+1$ self-synchronous scrambler was implemented for ATM over SONET [147, 148].

Error behaviour and handling is one area which is particularly likely to cause problems; as has been shown, engineers working at different levels in the network stack have very different ways of measuring and representing error types and likelihoods. Expectations about error handling in other layers may easily turn out to be incorrect. Both the types and frequencies of errors that might be presented at a layer interface, and the handling that may have removed or detected other errors, need to be understood for suitable error detection and/or correction to be applied at the next layer. This topic is revisited in Section 7.2.

An example where UDP checksums were disabled, leading to higher layer data corruption, has already been noted. Even with checksums in use, unfortunate interactions can result. Stone *et al.* [149] describe how behaviour in an ATM network which is carrying IP network data, could lead to a high rate of errors which are not detected by the TCP checksum.

These examples illustrate that combining layers without a complete understanding of them can lead to unpredicted faults and unexpected error behaviour. In particular, differences in fundamental metrics such as the number and nature of errors at a given layer, or a misunderstanding of the underlying properties or needs of an overlaid system can cause problems.

Chapter 4

Error Behaviour of Gigabit Ethernet Using 8B/10B Coding

This chapter investigates the theoretical behaviour of Gigabit Ethernet when a single bit line error occurs in a frame. An outline of the line coding scheme used in Gigabit Ethernet, the 8B/10B block code, is presented, along with some observations about the ways in which actual implementations of this coding scheme differ from the specification. A common method of analysis for error behaviour in communications links is to consider the simplest form of channel error, and to examine how this affects the coding and protocols of the system. This analysis is performed for the case of Gigabit Ethernet on fibre (1000BASE-X), as it strongly resembles the physical and coding layers of the SWIFT prototype optical switched network. By considering both the specified and observed implementations of the 8B/10B decoder in this analysis, this work is made directly relevant to deployed systems. For this case, the simplest error of a flipped single bit within an encoded frame is examined, and it is found that this will always be detected by the coding scheme or by the frame validity checks (including a CRC) in the Ethernet MAC layer. However, if more than one channel error occurs within a frame, some longer frame sizes may not have all possible error patterns detected.

As shown in the previous chapter, communications systems use a coding scheme to convert the data to be communicated into a form which allows the channel to be utilised efficiently, detection of errors, and so forth. The choice of coding system depends heavily on the characteristics of the transmission medium. Wireless systems, where errors and loss are likely, use codes with strong error correcting capabilities. Wired systems traditionally do not usually employ such a mechanism, but must still use codes suitable for the line. High speed systems have more stringent requirements as the transmitting and receiving systems are likely to be working closer to physical limits. Gigabit Ethernet uses 8B/10B block coding, for both fibre

and one of the copper transmission modes (short-haul 1000BASE-CX) [27]. Infiniband and Fibre Channel, other popular optical local area interconnects, also use 8B/10B [93, 94]. In contrast, 10Gb/s Ethernet uses 64B/66B coding — this sounds like a block code, but in fact is a scrambling system with 2 added bits (signifying data or control frames) [36]. Gigabit Ethernet on medium haul UTP (1000BASE-T) uses a forward error correcting scheme with differential and bidirectional signalling on 4 separate wire pairs [150].

This chapter begins with a description of the 8B/10B block coding scheme, its advantages, and its application in 1000BASE-X Gigabit Ethernet. From the experimental work with 1000BASE-X equipment described in Chapter 5, it is deduced that the specification may not be strictly adhered to by manufacturers. Firstly it is shown that in a range of implementations the running disparity is not taken into account when decoding. It is also found that the system can operate and still receive frames in a state of reduced receiver power where errors occur, whereas the specification details a low level error indication for loss of signal should occur. In the second section, the theoretical analysis itself is presented.

4.1 Outline of 8B/10B Block Coding Scheme

This block code, which converts 8 bits of data into 10 bits for the physical layer, was originally developed from a combination of the 5B/6B and 3B/4B schemes at IBM [133]. It was used for Fibre Channel, and was then modified slightly for Gigabit Ethernet. These modifications apply to frame structure and the code-groups used to define this, and to initial configuration phases used when the network is first powered on.

The coding scheme is now frequently deployed, in systems as diverse as the 800Mbps extensions to the IEEE 1394 / Firewire standard [151], and PCI Express, where 8B/10B is the basis of the standard's coding scheme [97]. It is also the coding scheme in use in next generation optical computer networks, such as that described in Chapter 2. This work concentrates on the coding scheme as used in Gigabit Ethernet (in terms of the framing structure and coding specification); this can easily be generalised to other cases.

8B/10B coding has many benefits. It is a block code and simple to implement; in addition, as the data side is 8-bits wide it is particularly well-suited for use in byte-oriented end systems. It limits the run length of consecutive identical bits (of 5 bits, further limited to 4 bits in the data symbols) and offers a good transition density (3 to 8 transitions per 10 bit code) to assist clock recovery. The code is DC balanced, with a maximum running digital sum of 3, allowing use in AC-coupled transmission systems (used for high speed circuitry, such as optical interfaces). A DC balanced data stream also makes control of transmit power levels, receiver gain, and equalisation simpler. The power spectrum of the code is a good compromise between minimal DC content and bandwidth.

4.1.1 8B/10B Block Coding in Gigabit Ethernet

The 8B/10B codec defines encodings for both data octets and the control codes that are used to delimit frames and maintain the link. Individual codes or combinations of codes are defined to indicate the start of a frame, the end of a frame, the line Configuration setup, and other control functions. Also, *Idle* codes are transmitted when there is no data to be sent, to keep the transceiver optics and electronics active. The Physical Coding Sublayer of the Gigabit Ethernet specification [27, §36] defines how these various codes are used.

Individual ten bit *code-groups*, also called *symbols*, are constructed from the groups generated by 5B/6B and 3B/4B coding on the first five and last three bits of a data octet respectively. During this process the bits are re-ordered; the last 5 bits of the octet are encoded first, into the first 6 bits of the code, and then the first 3 bits of the octet are encoded to the final 4 transmitted bits. Some examples are given in Table 4.1, where the codes are displayed showing the 4 and 6 bit *sub-blocks* which highlight their construction from the 5B/6B and 3B/4B coding schemes. The running disparity (RD) is the sign of the running sum of the code bits, where a one is counted as 1 and a zero as -1. (The disparity of a code block may also be calculated; this is the difference between the number of `1`s and `0`s in the block, a positive disparity suggesting an excess of `1`s.) During an Idle sequence between packet transmissions, the running disparity is changed (if necessary) to -1 and then maintained at that value. Both control and data codes may change the running disparity or may preserve its existing value; examples of both types are shown in Table 4.1. The code-group used for the transmission of an octet depends upon the running disparity at the end of the previous code-group hence the two alternative codes given in the table. (Note that some octets use the same code-group for both disparity cases.)

A received code-group is compared against the set of valid code-groups for the current receiver running disparity at the start of the code, and decoded to the corresponding octet if it is found. If the received code is not found in that set, the specification states that the group is deemed invalid. In either case, the received code-group is used to calculate a new value for the running disparity. A code-group received containing errors may thus be decoded and considered valid. It is also possible for an earlier error to throw off the running disparity calculation causing a later code-group to be deemed invalid because the running disparity at the receiver is no longer correct.

The full set of valid code-groups, both control and data, can be illustrated graphically (Figure 4.1), where they are represented on a 1024x1024 space, which shows valid combinations of the current code-group (C_i) and the preceding one (C_{i-1}). The regions of valid and invalid code-groups are defined by the codec's use of 3B/4B and 5B/6B blocks.

Figure 4.1 highlights the selection of control groups which are unlikely to suffer from undetected errors (due to the larger distances between these and data codes, compared to that

Type	Octet	Current RD -	Current RD +	Note
data	0x00	100111 0100	011000 1011	preserves RD
data	0xf2	010011 0111	010011 0001	swaps RD
control	K27.7	110110 1000	001001 0111	preserves RD
control	K28.5	001111 1010	110000 0101	swaps RD; contains *comma*

Table 4.1: Examples of 8B/10B control and data code-groups

between data and other data codes). One notable feature is the *comma*, a 7 bit sequence including a 5-bit run of all 1s or 0s, which is found at a certain position within some 10-bit code-groups. The comma is present in some of the control code-groups to enable code-group/octet timing alignment, and also protects against misreading of these framing code-groups.

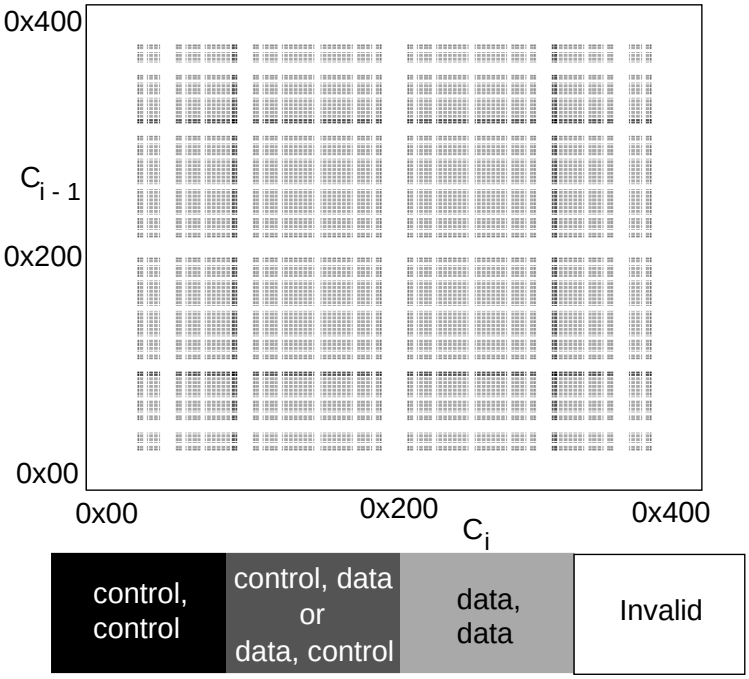


Figure 4.1: Valid 8B/10B code-groups represented on the full 10-bit codespace, showing the current code-group (C_i) and the preceding one (C_{i-1})

4.1.2 Differences to the Original 8B/10B Codec

The coding scheme as proposed by Widmer & Franaszek [133] in 1983 does differ slightly to that of the Gigabit Ethernet specification [27].

[133] states that the simplest undetectable error is a $1 \Rightarrow 0$ error and a $0 \Rightarrow 1$ in the same sub-block, as this will not change the disparity of the block. Any single-bit error will lead to a disparity change, which would be noticed by the receiver. This is true for the code as

proposed, which checks the running disparity after every sub-block. However, in the Gigabit Ethernet specification, the running disparity is re-calculated only at the end of each full 10-bit code-group. This means that single bit errors may be undetected by the disparity check.

On a whole packet basis, in the original scheme "in general... any error pattern in a packet for which the number of erroneous 1s is not equal to the number of erroneous 0s can be detected" [133]. As well as checking that the framing was correct, the coding system proposed has a large number of checks to ensure that errors are detected. Sub-blocks would be checked to ensure their disparity was always 0, 2 or -2. Other errors would be detected by the alternating disparity rule, requiring that the disparity of non-zero disparity blocks must alternate; this is not a check made in IEEE 802.3z. There is also a specific list of coding violations which should be checked for, in terms of bit sequences which do not fall into the coding alphabet. So the scheme of [133] contains a number of extra per-packet checks, which are not included in the Gigabit Ethernet physical coding sublayer specification [27, §36].

[133] notes that a single, isolated additive error leads to at most 5 errors in the decoded data stream. This paper then proposes a 16-bit CRC based on an analysis of error likelihoods, and demonstrates that this will give an overall undetected error probability of less than 1×10^{-5}. [133] also states that this "may be quite adequate for fiber optic transmission, which is expected to have very good noise properties." It is also pointed out that it is difficult to provide guaranteed error protection using error detection codes, as the distance properties of these codes do not apply to the incorrect detection of packet boundaries, for instance.

The original paper, [133], notes that of the available code-groups, K28.7 would make a particularly good comma, as no single error can generate a valid K28.7 code-group from any other and vice versa. This code-group is not used in Gigabit Ethernet. [133] also proposes that the symbols to indicate the start and end of a packet should each contain at least 1 non-zero disparity sub-block, to ensure that no disparity violations are carried forward to another packet; this is true for IEEE 802.3z.

4.1.3 Observations on Actual 8B/10B Decoders

In the experiments described in Chapter 5, transmitted and received octets are compared, and a great deal of data on decoded-layer octet errors was accrued. The packets analysed included those which would normally have been rejected at the MAC layer due to FCS failure (see Section 4.2.1), but otherwise the packets were the same as those which would be received by the IP layer in a normal host. Although the actual line representations of the data could not be observed, there are only two code-groups which could be used for each of the transmitted and received octets (corresponding to the two disparity values). The possible error patterns can therefore be deduced. According to the specification, there are two: one for the case of

negative running disparity code-groups for both transmitted and received octets, and one for the positive code-groups for both.

Some particularly frequently-occurring error patterns suggest the possibility that the decoder was not fully implementing the running disparity checking required by the specification. It seemed that the received code-groups might be being compared against both columns of the coding table, rather than just the one for the current running disparity value. The code-groups were then decoded regardless of running disparity, without an error (sufficient to stop frame reception) being indicated.

One example of this might be the octet 0xEA which was received as 0xFD over 22,000 times in a sample of nearly 4 million errored octets; this is the fifth most frequent error observed (for transmitted data consisting of random values), at more than 50 standard deviations over the mean frequency (59).

If the specified decoding scheme is used, this means that at least 5 bits must have been in error on the line. The two code-group transitions would be either `0101011110` ⇒ `1011100001` or `0101010001` ⇒ `0100011110`. Neither looks like a case of bit-clock loss, or slipped code-group boundary due to symbol clock loss; in any case this is an error in isolation, with correct code-groups around it. The alternative cross-disparity decoding of `0101011110` ⇒ `0100011110`, with only one physical layer bit error, is more likely to have occurred. A single bit error (in particular this type where a single `1` between two `0`s is lost) is the most likely to occur and remains consistent with an understanding of the system physical layer (Section 5.1.5).

Many other cases like this were observed, occurring at high enough frequencies to make it likely that the full specified disparity checking was not being undertaken. This behaviour was observed in a number of common implementations of Gigabit Ethernet.

It is notable that the two possible *strict* decoding transitions in the example given above both present a changed running disparity at the end of the code-groups. The transmitted group of `0101011110` leaves a positive RD, whereas `1011100001` gives a negative RD. Under the full specified decoding scheme, this disparity change would be detected as a subsequent invalid code-group. The presence of an invalid code-group, either through line error or disparity error, should cause a frame to be flagged as in error by raising the RX_ER flag (used by the PCS to indicate a receive error of various forms). One would expect most implementations not to pass such errored frames up to the higher levels of the stack. This is another example illustrating the presence of *relaxed* decoding. Further justification and examples are given in Section 5.1.6.

In the analysis which follows, two different decoding schemes are considered: the *strict* disparity checking of the Gigabit Ethernet specification, and the observed *relaxed* decoding.

4.1.4 Comments on Observed Implementations of Gigabit Ethernet Coding and Physical Layers

That the equipment can see these errored frames at all, reinforces the existence of implementations which do not fully adhere to the specification. According to the specification, when the receiver optical power dips below the relevant threshold, a loss of signal indication should cause packet reception to cease. At no point during the acquisition of these results was loss of light indicated; the equipment continued to function apparently normally. It should be noted that in a PC environment there is no mechanism for the user to be informed of such a physical layer problem, other than that the link is "down" (see Section 7.2). However some of these tests were performed using a network switch which also did not report any such error.

Two possible reasons for this spring to mind. One is that the layering abstraction allows a poor implementation of a standard at one level to discard signals which may appear unimportant. Higher level systems are thus unaware that an error occurred, and continue to see data packets arrive. A second reason might be that much Gigabit Ethernet equipment was designed and shipped before the standard (IEEE 802.3z) was finalised, and so may not be fully compliant with a, later-ratified, standard.

4.2 Analysis of the Effects of a Line Error in Gigabit Ethernet

This theoretical, probabilistic analysis is introduced with an overview of the framing scheme used at the MAC layer in Gigabit Ethernet, and how this connects to the physical layer framing of the 8B/10B coding scheme. The effects of a single bit channel error within an 8B/10B encoded Gigabit Ethernet frame are then followed through the decoding system and the physical and MAC layer framing validity checks. The analysis is performed for both the specified decoder, and the experimentally observed, *relaxed*, decoding scheme, and for regular MTU and jumbo frames. Expressions are derived for the probabilities of the different outcomes of the error (lost frame, error detected by the coding scheme, or error detected by the MAC layer). The effects of multiple 1 bit channel errors within a frame are also considered, with particular reference to the effects of these on the ability of the MAC layer frame check sequence to detect the error.

This section considers the case of a frame transmitted over Gigabit Ethernet on fibre (the optical forms of 1000BASE-X) although parts of this analysis might apply for some of the copper-based transmission systems. (In the case of 1000BASE-CX, differential signalling is used on a pair of co-axial wires in each direction; single-bit physical layer errors are thus not

likely to be the most common ones, although the coding scheme is the same as for the optical case considered here. For 1000BASE-T on unshielded twisted pair (UTP), the coding system is very different; the effects of error in this type of link are investigated in Section 5.4.) Clearly, the analyses for other optical systems using 8B/10B encoding are similar, depending on the frame size and structure used.

In the case of an optical fibre transmission of binary data using Non-Return-To-Zero modulation, the simplest line error is a single bit additive error, causing one bit to be misread (a zero as a one, or vice versa). The effects of this when it occurs within a frame are considered. This work does not examine the effects of errors other than during the transmission of a frame; errors may also occur during Idle sequences between frames, or during link auto-negotiation and configuration phases. These do not however directly affect data transmission and are likely to be recovered from before data transfer is attempted. It is assumed that frames are transmitted individually, rather than as part of a burst (using Gigabit Ethernet's carrier extension option [27, §36]).

In a 1000BASE-X network, the bit error rate is specified as 10^{-12}; therefore this analysis works on the basis that each bit is subject to a 1 in 10^{12} probability of error.

4.2.1 Gigabit Ethernet MAC Framing

A basic Ethernet MAC frame is considered here [152, §3.2]. The structure of this is outlined in Table 4.2.

Field	Size in Octets	Note
Preamble	7	Each octet is fixed, `10101010`
Start-of-Frame Delimiter	1	Fixed, `10101011`
Destination Address	6	
Source Address	6	
Length/Type	2	
MAC Client Data	46 - 1500	
Pad	< 46	Only used if < 46 octets of client data
Frame Check Sequence	4	

Table 4.2: Structure of an Ethernet MAC frame

Since this work is particularly interested in errors in the data part of the frame, frames containing high proportions of actual user data should be considered, and for this study an example frame containing 1500 data octets (a data payload equal in size to the Ethernet Maximum Transmission Unit, or MTU) is used.

Gigabit Ethernet also supports the use of jumbo frames, where between 1501 and 8982 data

octets can be carried. These are increasingly seeing use in deployed systems where there is a need to transport large blocks of data [138]. Therefore a second frame size, of a maximum length Jumbo frame, is also examined.

Clearly some of the octets in the frame have specified values which can be used in the error analysis which follows. Given no other information about the client data or application, it is assumed that both Address fields, the data itself and the Frame Check Sequence consist of random octet values (i.e., all octet values are equally likely and independently selected for each position). The FCS is a 32-bit CRC taken over the frame from Destination Address to the end of the actual or pad data inclusively (for further information on CRCs, see Section 3.2). This FCS uses the standard IEEE 802.3 CRC32 polynomial.

An empirical examination of CRC32 values calculated at the TCP and IP layers for real network frames (Section 3.2.4) shows that these CRCs contain an approximately equal distribution of all octet values. The assumption here is therefore reasonable. The Length/Type field will be defined by the number of data octets for the regular Ethernet frame; for the Jumbo packet this will be a special type value. However, since these are arbitrarily selected frame lengths, and this analysis should be kept as general as possible, the Length/Type field is also considered to have a random value.

The selection of random octets for addresses and data permits a reasonable analysis for the case of uniform channel errors. As will be found in Chapter 5, in fact the probability of error in this type of system depends heavily on the data transmitted; a worst-case analysis could have been performed (similar to that in [153], for instance), but generality and comparability to similar work is retained by using random data.

4.2.2 8B/10B Framing

This MAC frame is then encapsulated for transmission by the Gigabit Ethernet Physical Coding Sublayer. The first octet of the MAC Preamble is replaced by a Start_Of_Packet (`/S/`) code-group. Subsequent octets are replaced by their code-group equivalents as per the 8B/10B coding scheme, up until the end of the FCS. After the code-group representing the final octet of the FCS, the End_of_Packet code-group (`/E/`) is sent, followed by a Carrier_Extend group (`/R/`). If this Carrier_Extend group falls in an even code-group position, a second Carrier_Extend group is sent (to ensure that subsequent Idle code-groups are correctly aligned). After this, Idle code-groups are sent until there is another MAC frame available for transmission.

It is therefore possible to represent the two example frames which will be examined in terms of the code-groups that will be used to transmit them (Table 4.3). These consist of fixed start and end of frame sequences, with a block of either $L_R = 1518$ or $L_J = 9000$ code-groups of random data in the middle.

Field	Code-group	Number of code-groups
MAC Preamble	/S/ = K27.7	1
MAC Preamble	D10.5	6
MAC Start_of_Frame	D11.5	1
Destination Address	random	6
Source Address	random	6
Length/Type	random	2
MAC Client Data	random	1500 or 8982
Frame Check Sequence	random	4
	/T/ = K29.7	1
	/R/ = K23.7	1

Table 4.3: Structure of an Ethernet MAC frame in terms of 8B/10B code-groups

The total frame lengths in bits are therefore $F_R = 15280$ and $F_J = 90100$ for regular and jumbo Ethernet frames respectively.

4.2.3 Frame Validity Criteria

The decoder performs several validity checks as it decodes a received frame. The coding system itself defines several checks, such as the running disparity checks. In addition, once a Start_of_Packet has been received, if all subsequent code-groups are not valid data codes or a correct end sequence, a receive error will occur and the frame discarded [27, §36].

At the MAC layer, several checks are made before a received frame is deemed valid and passed to the next higher layer. The Destination Address is compared to the local address and valid broadcast addresses, the frame being discarded if it has reached the wrong host. If the frame is too long it may be truncated to a relevant length. It will also be truncated to the nearest octet boundary if it is not a multiple of 8 bits long. The FCS is checked, and if the received and calculated values do not match, the frame is declared invalid. The Length/Type field is then checked according to the local validity criteria (either it is a valid Length field, or a registered local type). If all these checks are passed, the MAC frame is disassembled into its component parts according to its type.

The full set of checks is summarised in Figure 4.2.

The behaviour if the MAC Preamble symbols or the MAC Start_Of_Frame symbol are received in error is not detailed here. Since these are usually simply stripped one may assume that errors would go unnoticed; however, a more thorough implementation might check that the correct octets were being removed, and if they were found to be in error might well discard the frame. In the analysis below, the latter behaviour is assumed. If errors in the MAC Preamble

are ignored, then the probabilities referring to their detection (P_{Ru} and P_{Ju} in Section 4.2.5 below) would become 0, and their original values can simply be subtracted from the overall probability of frame error.

4.2.4 Coding Layer Outcome of a 1-bit Physical Layer Error

Most of the frame clearly consists of data code-groups, and errors in these are considered in the next section. The goal of this work is to find out what happens to the code-groups used to define the frame, when a single bit error occurs in one of them.

The Start_Of_Packet code, `/S/`, is represented by the control symbol K27.7 (for which there are two possible code-groups, depending on the running disparity) and occurs once per frame. A single bit error in each possible bit position of the negative disparity code leads to 7 cases of invalid code-groups, one where the code-group is valid (for data octet 0xEB) but the disparity is wrong, and 2 instances of data code-groups (0xBB and 0x3B). For strict decoding, if an error occurs in the symbol, this gives a probability of 0.8 that the errored code-group will be deemed invalid, and 0.2 that a data code will be received instead. A similar analysis can be performed for the positive disparity code-group, and for the other special code-groups used in the frame. The outcomes are shown in Table 4.4. Where a framing symbol may be represented by 2 different code-groups depending on the initial disparity, the results assume equal probability of positive and negative disparity at the start. (An exception might be made for `/S/`, which always follows on from an Idle pattern (which sets and then maintains a negative disparity); so a negative disparity could perhaps be assumed. However, the `/S/` could occur between the two code-groups which define the usual Idle symbol `/I2/`, and where the disparity is positive.)

Framing code-group	Probability						
	Strict Decoding				Relaxed Decoding		
	Data	Control	Invalid	RD Error	Data	Control	Invalid
`/S/`	0.2	0	0.7	0.1	0.3	0	0.7
`/T/`	0.2	0	0.7	0.1	0.3	0	0.7
`/R/`	0.2	0	0.8	0	0.2	0	0.8
D10.5	0.9	0	0.1	0	0.9	0	0.1
D11.5	0.8	0.05	0.1	0.05	0.8	0.1	0.1

Table 4.4: Probabilities of outcomes of all possible 1-bit physical layer errors in framing code-groups

The outcome of a false control code only occurs once in Table 4.4; the control code generated by an error in the fourth bit of the D11.5 code-group is K28.2, which is not used in the Gigabit Ethernet specification. This would therefore lead to a receive error being indicated.

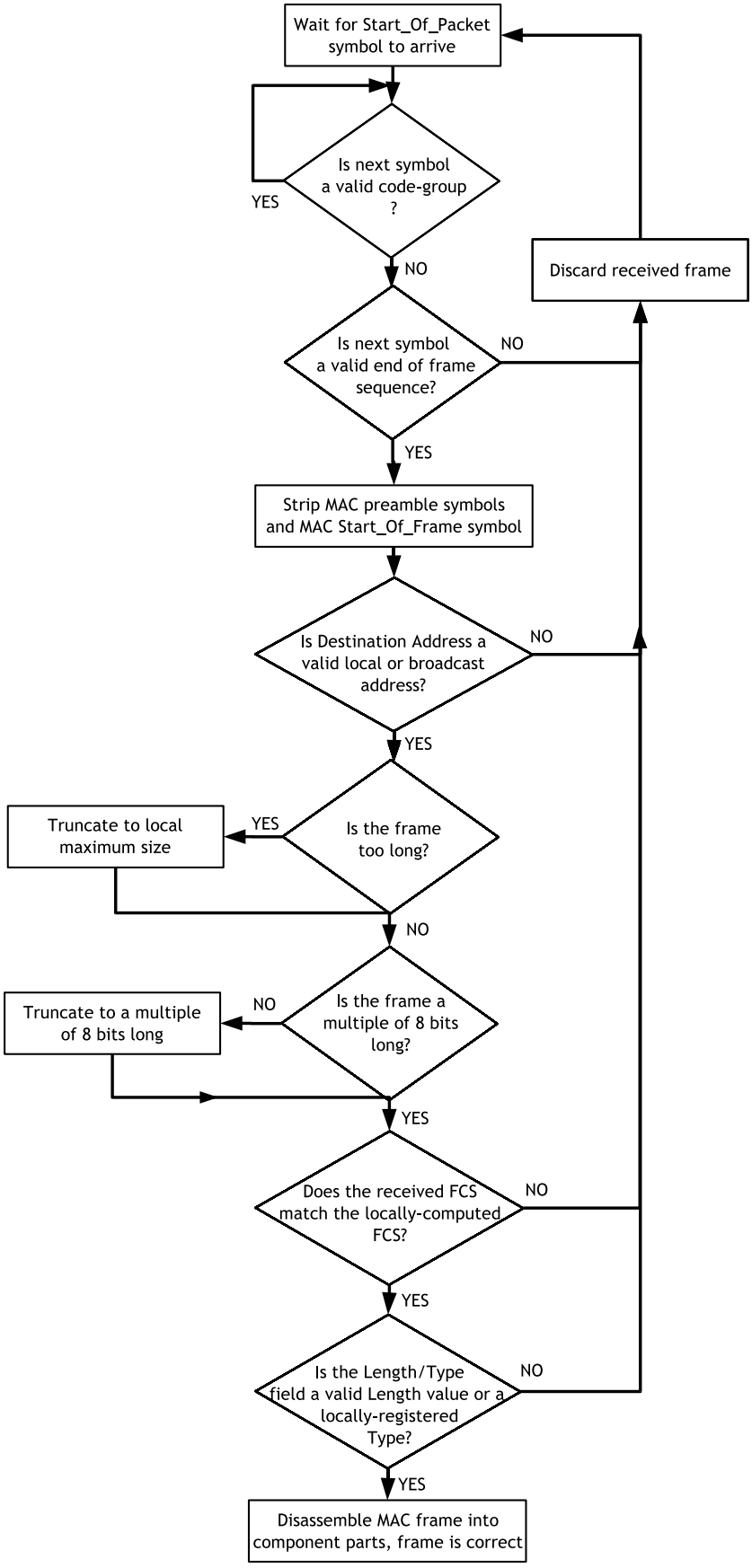


Figure 4.2: Frame validity check process at the PCS and MAC layers

If the Start_Of_Packet code-group is not received, the receiver will not enter a state where a frame will be read in, and so the frame will be lost. Any of the possible errors in the end of frame sequence (/T//R/), with either symbol being received in error as invalid or a data symbol, will lead to a receive error.

Again, it is known that any invalid code-group within this frame will cause a receive error; however, if the D10.5 and D11.5 code-groups (used for framing at the MAC layer) are received as other data codes rather than invalid codes, the overall frame will still be received correctly at the PCS layer. However, the Gigabit Ethernet MAC should detect the incorrect preamble sequence and reject the frame.

These special code-groups used for framing were selected for their resilience to errors. In particular, it was noted in the early Working Group meetings for IEEE 802.3z that the start of frame and end of frame codes should be robust to errors. One way that this is undertaken is that all these framing symbols have different code-groups for the two possible running disparities; this makes detection of error more likely, at least if strict decoding is used.

In the following Sections, particular error types are outlined, starting with errors which occur within the code-groups used to represent data, what happens when this type of error causes another data code-group to be received, and the special case of errors in the running disparity.

Errors in Data Code-groups

Line coding schemes, although they handle many of the physical layer constraints, can introduce problems. In the case of 8B/10B coding, a single bit error on the line can lead to multiple bit errors in the received data byte. For example, with one bit error the code-group D8.6 (current running disparity negative) becomes the code-group D7.6 (also negative disparity); these decode to give bytes with 4 bits of difference. In addition, the running disparity after the code-group may be miscalculated, potentially leading to future errors. There are other similar examples in the specification [27].

A 1-bit error in each of the possible positions (bit 0 to 9) for each valid data code-group is considered. The outcome may be another data code-group, a control ("K") code-group, or an invalid code-group. In the case of *strict* decoding, a running disparity error is possible (which would be handled as an invalid code-group); this is defined as when a received code-group is valid but occurs in the wrong disparity column of the codebook. These potential outcomes are shown in Table 4.5 for both decoding schemes. The total number of outcomes is 5120 in each case (corresponding to 2 possible transmitted code-groups for each of 256 octets, one for each disparity case, times 10 possible bit error positions in each code-group).

Resulting code-group	Frequency	
	Strict Decoding	Relaxed Decoding
Data	1812	3304
Control	40	102
Invalid	1714	1714
Disparity Error	1554	n/a

Table 4.5: Code-group outcomes of all possible 1-bit physical layer errors in data code-groups

Data code-groups received in error as other data code-groups

Given the protection provided by the comma system, and the distance between the control code-groups, the most interesting error case is that where a data code-group is received in error as another data code-group.

Table 4.6 shows the results of these 1-bit errors in terms of the number of data layer bit errors, once the received code-group is decoded.

Number of data bits in error	Frequency	
	Strict Decoding	Relaxed Decoding
1	692	1096
2	592	1084
3	336	680
4	192	444
5+	0	0

Table 4.6: Outcomes of all possible 1-bit physical layer errors in data code-groups leading to other data code-groups

An *effective bit error rate*, representing the mean or expected number of data bit errors resulting from a single line error, can be calculated. In the strict case this is 2.02 bits; for relaxed decoding it is 2.14 (counting over all data code-group to data code-group cases).

This of course does not account for errors resulting in control or invalid code-groups, the overall effect of which will probably be a lost frame. The effect of single-bit line errors leading to a greater number of data errors is henceforth referred to as *error amplification*. This could affect applications using 8B/10B without higher level error checking, or where the protection of, say, a CRC is weaker due to the error amplification not being allowed for in the initial design decisions.

Running Disparity Errors

It was noted above that an error in a single code-group may change the running disparity at the end of the code-group to be different to that at the transmitter. In the case of the *strict* decoding scheme, this may cause subsequent code-groups to be decoded incorrectly. This is only of interest if the code-group itself has been decoded in the correct running disparity column (i.e., is not invalid).

An analysis of the 1812 error cases where a data code-group results from a single bit error showed that all cause an incorrect running disparity at the end of that errored code-group. This raises the question of what happens to the following code-group — not itself received in error — when the running disparity is incorrect.

All the control codes have different code-groups for the two disparity cases, and so an incorrect disparity at the start will invariably lead to a receive error; if the disparity error generated by the bit error is carried through to the end of the packet, it will be detected there. Some data octets are represented by the same code-group regardless of disparity, so these will be decoded correctly into their corresponding octets. There are 72 of these. All the octets for which there are two code-group representations, one for each disparity, will cause a disparity error on decoding.

There remains the question of what happens to the disparity in the cases where correct decoding of the code-group subsequent to the errored code-group occurs. It could remain in the wrong state, carrying the running disparity error on to the next code-group, or be corrected. This is simple to check; all code-groups either flip the running disparity or keep it the same. In the case of the 72 octets represented by the same code-group regardless of starting disparity, the code-groups all keep the disparity the same, meaning the error is carried over to the next code-group.

This initial undetected bit error might eventually cause a code-group to be invalid due to disparity, or the disparity error might reach the end of the frame, or the error might not have any further effect.

If it is assumed that errors are equally likely to occur in any octet of the frame, the original bit error could have occurred anywhere in the data portion of the frame. It has been shown that the next octet must have been one of the 72 with only one code-group representation. Subsequent code-groups will either also be from these 72 (being decoded correctly and sustaining the disparity error) or not (in which case they will be observed to have the incorrect disparity, and will be deemed invalid). The probability of each subsequent code-group maintaining the disparity error is thus 72/256. If the error is sustained until the end of the frame (the /T/ code-group) then it will be detected, and the frame will be invalid. So, it transpires that, regardless of position in frame and succeeding octets, the disparity error will eventually

cause an invalid code-group to be detected in the frame in which the original error occurred.

In the strict decoding case, all data code-groups suffering a single bit error will either be identified as erroneous themselves, or will generate a running disparity error which will be detected during the same frame. The whole frame will therefore be deemed invalid.

4.2.5 Frame Error Rate at the Physical Coding Sublayer

The potential outcomes after a 1-bit error occurs in the physical layer during transmission of each of the sections of an encoded Gigabit Ethernet frame have been identified. This type of channel error is believed to be the most likely form.

Overall, there are two possible eventual outcomes. The frame may be detected as having one or more invalid code-groups, and will therefore be deemed in error. Or, the frame will be received but with undetected errors in the non-framing parts of the frame. The latter is only an option for the relaxed decoding scheme, as when using the strict decoding method the running disparity will inevitably detect any such errors (providing only one bit is in error per frame).

Frame Error Rate at the PCS, Using the *Strict* Decoding Scheme

For the regular length Gigabit Ethernet frame, then, at the specified line bit error rate of 10^{-12}, a number of probabilities which define the likelihoods of various outcomes, depending on where the error occurs in the frame and what data is being sent, can be determined. These outcomes are:

P_{Xf} Error detected by the decoding system at the PCS

P_{Xu} Error unnoticed by the decoding system and PCS but detected at the MAC layer by the MAC framing validity checks

P_{Xm} Frame is not noticed by the receiver at all

where X is either R, representing the Ethernet MTU frame case, or J, representing the jumbo frame case.

The sum of all the individual probabilities of these outcomes gives the overall probability of a line error occurring in a Gigabit Ethernet frame, i.e., the bit error probability multiplied by the number of bits in the frame. The two frame sizes, an Ethernet MTU and a jumbo frame, are considered separately.

The probability of a frame being detected in error at the coding layer using the strict decoding scheme is given by:

$$\begin{aligned}
P_{Rf} &= P(\text{1-bit error in any position, in any code-group of the frame except /S/} \\
&\quad \textit{or} \text{ the MAC preamble } \textit{or} \text{ the MAC start of frame)} \\
&\quad + P(\text{1-bit error in MAC preamble or start of frame, causing non-data symbol}) \\
&= \{P(\text{bit error occurring in the frame}) \\
&\quad \times P(\text{error in frame being in one of these positions})\} \\
&\quad + P(\text{1-bit error in MAC specified symbols D10.5 or D11.5, causing non-data symbol}) \\
&= \left\{(10^{-12} \times F_R) \times (\text{number of applicable code-groups / total number of code-groups})\right\} \\
&\quad + \left\{ P(\text{bit error}) \times (\text{number of bits}) \right. \\
&\quad \left. \times \sum_{relevantsymbols} (\text{symbols present}) \times P_{symb}(\text{result not being a data code-group}) \right\} \\
&= \left\{(10^{-12} \times 15280) \times ((1528-8)/1528)\right\} + \left\{10^{-12} \times 10 \times ((5 \times 0.1) + (1 \times 0.2))\right\} \\
&= 1.5207 \times 10^{-8}
\end{aligned} \tag{4.1}$$

The probability of the MAC detecting that the frame is in error, but no error being detected by the coding scheme is

$$\begin{aligned}
P_{Ru} &= P(\text{1-bit error in any position of the 5 MAC preamble symbols or} \\
&\quad \text{the 1 MAC Start of Frame symbol, other than those leading to an RD error} \\
&\quad \text{or an invalid or control code-group)} \\
&= P(\text{bit error}) \times (\text{number of bits}) \times \\
&\quad \sum_{relevantsymbols} (\text{symbols present}) \times P_{symb}(\text{result being a data code-group})) \\
&= 10^{-12} \times 10 \times \{(5 \times 0.9) + (1 \times 0.8)\} \\
&= 5.3000 \times 10^{-11}
\end{aligned} \tag{4.2}$$

For the jumbo sized frame, the equivalent probabilities are

$$\begin{aligned}
P_{Jf} &= \left\{(10^{-12} \times 90100) \times ((9010-8)/9010)\right\} + \left\{10^{-12} \times 10 \times \{(5 \times 0.1) + (1 \times 0.2)\}\right\} \\
&= 9.0007 \times 10^{-8}
\end{aligned} \tag{4.3}$$

for detected errors and

$$P_{Ju} = 10^{-12} \times 10 \times \{(5 \times 0.9) + (1 \times 0.8)\} = 5.3000 \times 10^{-11} \tag{4.4}$$

for undetected errors.

In both cases, the probability of the frame going entirely unnoticed by the receiver is simply

$$\begin{aligned} P_{Rm} &= P_{Jm} = P(\text{bit error}) \times P(\text{error in /S/ such that it is not received}) \\ &= 10^{-12} \times P(\text{error in /S/ leading to an RD error,} \\ &\quad \text{an invalid or other control code, or a data code}) \\ &= 10^{-12} \times 10 \\ &= 1 \times 10^{-11} \end{aligned} \tag{4.5}$$

Clearly, all these values are proportional to the physical layer error rate and so equivalents for other bit error rates can easily be obtained.

Frame Error Rate at the PCS, Using the *Relaxed* decoding scheme

Here, similar probabilities as those given above for the *strict* decoding scheme are derived. There is an additional possible outcome using the *relaxed* scheme — an error may go undetected by the coding scheme and MAC framing validity checks, leaving a decoded data block containing one or more errors. This will then be checked by the MAC layer FCS, where the errors will be detected (see Section 4.2.6). This is represented by the probabilities P_{Rd} and P_{Jd} for Ethernet MTU and jumbo frames respectively.

The probability of an error being detected in a frame by the coding scheme is given by

$$\begin{aligned} P_{Rf} &= P(\text{1-bit error in any code-group of the frame except /S/} \\ &\quad \text{where a control or invalid code-group is generated}) \\ &= P(\text{1-bit error in any of the } L_R \text{ data symbols or in /T/} \\ &\quad \text{or in /R/ or in MAC specified symbols D10.5 or D11.5,} \\ &\quad \text{where a control or invalid code-group is generated}) \\ &= P(\text{bit error occurring in the frame}) \\ &\quad \times P(\text{error in frame being in one of the relevant symbols}) \\ &\quad \times P(\text{the bit in error causing a control or invalid code-group}) \\ &= 10^{-12} \times 10 \end{aligned}$$

$$\begin{aligned} & \times \sum_{relevantsymbols} \{(\text{symbols present}) \times P(\text{generation of control or invalid code})\} \\ = \; & 10^{-11} \times \{(L_R \times ((102 + 1714)/5120)) + (1 \times 1.0) + (1 \times 1.0) + (5 \times 0.1) + (1 \times 0.2)\} \\ = \; & 5.4112 \times 10^{-9} \end{aligned} \tag{4.6}$$

The probability of the MAC framing being in error, but no error being detected by the coding scheme is

$$\begin{aligned} P_{Ru} = \; & P(\text{1-bit error in any position of the 5 MAC preamble symbols} \\ & \text{or the 1 MAC Start of Frame symbol, other than those} \\ & \text{leading to an invalid or control code-group}) \\ = \; & P(\text{bit error}) \times (\text{number of bits}) \\ & \times \sum_{relevantsymbols} \{(\text{symbols present}) \times P_{symb}(\text{result being a data code-group})\} \\ = \; & 10^{-12} \times 10 \times \{(5 \times 0.9) + (1 \times 0.8)\} \\ = \; & 5.3000 \times 10^{-11} \end{aligned} \tag{4.7}$$

For the jumbo sized frame, the equivalent probabilities are

$$\begin{aligned} P_{Jf} = \; & 10^{-11} \times \{(L_J \times ((102 + 1714)/5120)) + (1 \times 1.0) + (1 \times 1.0) + (5 \times 0.1) + (1 \times 0.2)\} \\ = \; & 3.1949 \times 10^{-8} \end{aligned} \tag{4.8}$$

for detected errors and

$$P_{Ju} = 10^{-12} \times 10 \times \{(5 \times 0.9) + (1 \times 0.8)\} = 5.3000 \times 10^{-11} \tag{4.9}$$

for undetected errors.

The probability of the frame going entirely unnoticed by the receiver is the same as for the strict decoding case:

$$P_{Rm} = P_{Jm} = 1.0000 \times 10^{-11} \tag{4.10}$$

The probability of an undetected error leading to a decoded data block containing errors, which will be detected by the MAC FCS, can be calculated as follows:

$$\begin{aligned} P_{Rd} = \; & P(\text{error in a data symbol generating another data code-group}) \\ = \; & P(\text{error occurring in a data symbol}) \times P(\text{error causing another data symbol}) \end{aligned}$$

$$\begin{aligned} &= (10^{-12} \times L_R \times 10) \times (3304/5120) \\ &= 9.7958 \times 10^{-9} \end{aligned} \tag{4.11}$$

Similarly, for jumbo frames,

$$P_{Jd} = (10^{-12} \times L_J \times 10) \times (3304/5120) = 5.8078 \times 10^{-8} \tag{4.12}$$

It has been determined above that given an undetected data error, there is an expected number of data bits in error of 2.14 bits per frame; the actual number of data bit errors may be 1, 2, 3 or 4 (see Table 4.6).

Summary of Probabilities of Outcomes of a 1-bit Channel Error in a Gigabit Ethernet Frame

These results are summarised in Table 4.7. As before, P_f is the probability of the coding scheme detecting an error, P_u is that of the MAC framing validity checks detecting an error, P_m is that of the frame being unseen by the receiver, and P_d is the probability of the decoded data block containing errors (which should be detected by the FCS). These probabilities, for each decoding scheme and frame size, add up to the overall probability of a frame being damaged for a bit error rate of 10^{-12} (P_{err}).

	Strict decoding scheme		Relaxed decoding scheme	
Probability	Ethernet MTU	Jumbo frame	Ethernet MTU	Jumbo frame
P_f	1.5207×10^{-8}	9.0007×10^{-8}	5.4112×10^{-9}	3.1949×10^{-8}
P_u	5.3000×10^{-11}	5.3000×10^{-11}	5.3000×10^{-11}	5.3000×10^{-11}
P_m	1.0000×10^{-11}	1.0000×10^{-11}	1.0000×10^{-11}	1.0000×10^{-11}
P_d	n/a	n/a	9.7958×10^{-9}	5.8078×10^{-8}
Overall P_{err}	1.5280×10^{-8}	9.0100×10^{-8}	1.5280×10^{-8}	9.0100×10^{-8}

Table 4.7: Probabilities of various outcomes for a 1-bit channel error in a Gigabit Ethernet frame at a line error rate of 10^{-12}

As expected, the overall probability of a frame being received in error increases as the frame size increases. Using the *relaxed* coding scheme does not alter the overall probability of a single channel error affecting a frame, but does increase the probability that the error will go undetected by the coding scheme, leaving a decoded data block with errors.

This work has considered detected and undetected errors at the Physical Coding Sublayer. Most errors will clearly be detected — either by the coding system itself, or the framing checks. However, some 1-bit errors may go undetected at this layer if the relaxed decoding scheme is

used. The next section describes work to check that these errors will always be detected by the MAC layer FCS.

4.2.6 MAC Layer Frame Check Sequence

Given the amplification of a single physical layer error into multiple data layer errors, and the possibility of an error going undetected by the coding scheme when the *relaxed* decoding method is used[1], whether the FCS at the MAC layer of Gigabit Ethernet will suffice to detect the error is now examined.

The Hamming distance for various frame sizes and the CRC32 polynomial are given in Table 4.8. The thresholds between Hamming distances are defined in terms of the number of bits per frame, and the maximum frame length in octets shown is therefore rounded down.

Maximum frame size in Octets	Maximum frame size in bits	Hamming distance
11454	91639	3
375	3006	4
37	300	5
25	203	6

Table 4.8: Hamming distances of the CRC32 polynomial attainable by various frame sizes

The data layer "codeword" lengths (including the CRCs) of the two frame sizes considered are 12144 and 72144 bits respectively. For both these frame sizes, the CRC32 polynomial gives a minimum Hamming distance of 4, and so all 1, 2 and 3 data bit errors will be detected. However, there are some 4 bit error patterns which may not be detected. For the 32 bit CRC to fail to detect an error pattern, the bits in error must form an *error burst* of more than 32 bits in length. While the decoding process can lead to 4 data bits in error, this is within one code-group and so cannot give a data burst of more than 32 bits in length. Any single 1-bit physical layer error in a regular or jumbo Gigabit Ethernet frame will therefore be detected by the CRC.

Multiple Physical Layer Events

We would like to know how the CRC will handle cases of more than one physical layer error, where the events occur sufficiently far apart as to generate error bits spread out by at least 32 bits. A brute force computer search was undertaken to investigate this.

[1]Sections 4.1.3 and 5.1.1 document observations of such decoding behaviour being in common use.

An undetected error could cause many application-layer problems, but clearly is proportionately less likely to occur than a single bit error, when noise events are assumed to follow a Poisson distribution. However, this does not take into account channel errors which might occur due to causes other than random noise. The degradation of the received clock signal, either in terms of a symbol (or code-group) clock or the bit clock, could cause a number of octets to be received incorrectly. This could potentially spread errors out over more than 32 bits. Clock errors are discussed further in Section 5.1.8.

Firstly, the error patterns generated at the data layer by all 1-bit physical layer errors which cause a data code-group to be received as another valid data code-group were determined. In the case of the strict decoding scheme, 32 8-bit error patterns exist; these form a subset of the total of 33 possibilities for the relaxed decoding scheme. These are given in Table 4.9. Although it was shown earlier in Section 4.2 that the strict decoding scheme would detect any single bit physical layer error in a frame, and so the detection ability of the FCS is irrelevant in this case, the data layer error patterns are shown here for comparison.

The frequencies given correspond to the frequency of this error pattern in the full range of 5120 possible outcomes of a 1-bit physical later error.

Given the linearity of the FCS polynomial, it is known that the result of exclusive-or'ing any two valid codewords together is another valid codeword. So rather than considering the effect of errors in actual or simulated data, it is possible to work out whether the FCS will detect these errors by calculating the validity of a codeword equivalent to a frame consisting of only the error patterns.

A search was completed for frames containing 2 errored octets, each generated by a single 1-bit physical layer error, in all possible positions. Each errored octet was represented in turn by one of the error patterns listed in Table 4.9, with the remaining octets in the frame set to 0x00. Data blocks of 1518 and 9000 octets in length, including the FCS, are considered. If the CRC32 calculated over the whole of these blocks is 0, then the data block has a valid FCS and the error patterns in it would not be detected.

For both the maximum size regular Ethernet frame and the maximum size jumbo frame, all cases of two 1-bit physical layer errors are detected by the CRC (regardless of decoding method). In the case of the Ethernet MTU frame, this is as expected from Fujiwara *et al.* [134], but this only provides information for double error bursts in frames up to 13000 bits in length (Section 3.2.1).

In the case of three 1-bit physical layer errors, both the jumbo and regular size frames were found to have some error patterns which would not be detected, for both strict and relaxed decoding cases. The searches for these cases are not complete as they are computationally time-consuming, despite being optimised for speed [136]; this was a reasonable step as the

Error Pattern	Error Bits	Frequency	
		Strict Decoding	Relaxed Decoding
0x01	0 0 0 0 0 0 0 0 1	24	48
0x02	0 0 0 0 0 0 0 1 0	40	80
0x05	0 0 0 0 0 0 1 0 1	16	32
0x06	0 0 0 0 0 0 1 1 0	16	32
0x07	0 0 0 0 0 0 1 1 1	24	80
0x08	0 0 0 0 0 1 0 0 0	24	80
0x09	0 0 0 0 0 1 0 0 1	24	48
0x0A	0 0 0 0 0 1 0 1 0	24	48
0x0B	0 0 0 0 0 1 0 1 1	40	80
0x0D	0 0 0 0 0 1 1 0 1	24	48
0x0E	0 0 0 0 0 1 1 1 0	40	80
0x0F	0 0 0 0 0 1 1 1 1	64	100
0x10	0 0 0 1 0 0 0 0 0	64	100
0x11	0 0 0 1 0 0 0 0 1	40	80
0x12	0 0 0 1 0 0 0 1 0	24	48
0x14	0 0 0 1 0 1 0 0 0	40	80
0x15	0 0 0 1 0 1 0 1 0	24	48
0x16	0 0 0 1 0 1 1 0 0	24	48
0x17	0 0 0 1 0 1 1 1 1	24	48
0x18	0 0 0 1 1 0 0 0 0	24	48
0x19	0 0 0 1 1 0 0 0 1	16	32
0x1A	0 0 0 1 1 0 1 0 0	16	32
0x1B	0 0 0 1 1 0 1 1 1	40	80
0x1D	0 0 0 1 1 1 0 1 1	40	80
0x1E	0 0 0 1 1 1 1 1 0	24	48
0x20	0 0 1 0 0 0 0 0 0	192	192
0x40	0 1 0 0 0 0 0 0 0	192	192
0x60	0 1 1 0 0 0 0 0 0	0	72
0x80	1 0 0 0 0 0 0 0 0	116	192
0xA0	1 0 1 0 0 0 0 0 0	192	192
0xC0	1 1 0 0 0 0 0 0 0	192	192
0xE0	1 1 1 0 0 0 0 0 0	128	128

Table 4.9: Data-layer error patterns generated by a 1-bit physical layer error

point of this work was to show the existence of non-detected error cases, not to fully describe them. Up to the point at which the searches were stopped, a total of 74 error patterns which would not be detected by the CRC had been found for the regular frame case. The search program uses reverse numbering, where position 0 is the last octet/code-group of the FCS to be transmitted; position is then described in terms of octets counted backwards through the frame. The search considers errors in positions i, $i + x_1$, and $i + x_2$, starting with $i = 0$, $x_1 = 1$ and $x_2 = 2$, then incrementing x_2 within the range of valid frame positions, then repeating for incremented values of x_1 and i in turn. The search was stopped at $i = 47$ for the regular frame size. Eight sets of three error patterns and positions had been found for the jumbo frame case before the search was stopped; only a very small amount of the search space had been covered in this case, with the search still considering cases where $i = 0$, $x_1 = 115$.

An example for the jumbo frame would be the error pattern 0xA0 at octet position 0, 0xA0 at position 2, and 0x08 at position 1621. The probability of this event can be derived. The probability of error pattern 0xA0 being generated by a line error is 192/5120; the probability of 0x08 is 80/5120. The theoretical probability of this particular sequence of error patterns occurring and the resulting errors not being detected by the CRC is given by:

$$\begin{aligned} P &= \prod_{\textit{3 error patterns}} \{P(\text{1-bit error in the frame}) \\ &\quad \times P(\text{relevant error pattern resulting from the 1-bit error}) \\ &\quad \times P(\text{error in the relevant octet position})\} \\ &= \prod_{\textit{3 error patterns}} \left\{(10^{-12} \times 9000)\right. \\ &\quad \left. \times\ P(\text{error pattern resulting from the 1-bit error}) \times (1/9000)\right\} \\ &= 10^{-36} \times 80/5120 \times 192/5120 \times 192/5120 \\ &= 2.2 \times 10^{-41} \end{aligned} \tag{4.13}$$

It is assumed that the other error pattern cases will occur with similar probabilities. So, for the jumbo frame case, the theoretical probability of three 1-bit physical layer errors occurring which will cause an error pattern which the CRC will not detect is at least $8 \times 2.2 \times 10^{-41}$.

4.2.7 Conclusions

This analysis has examined the potential errors which may arise from a single bit line error in Gigabit Ethernet on fibre which is operating at the specified bit error rate.

As will be shown in the experimental work of Chapter 5, some systems do not use the full specified 8B/10B decoding system but instead use *relaxed* decoding which does not check the

running disparity at the receiver. The work described in this chapter analysed the effects of a line error for both the standard, *strict* 8B/10B decoding system and this variant, and found that the coding layer would detect all possible 1-bit channel errors for the *strict* case. The *relaxed* decoder would not detect some errors, meaning that a data block containing a number of errors could be passed up to the MAC layer. It was shown that in all cases of a single channel error, the MAC layer FCS would detect these errors.

Clearly the error detection mechanism in Gigabit Ethernet is very strong for 1-bit line errors, and there is no risk of this type of error causing incorrect data to be presented to an application. However, even a detected error will lead to a dropped packet. In applications and systems using a reliable transport layer such as TCP, this will require the frame source to retransmit, once it is aware that the destination did not receive the frame. This will increase application latency and decrease network availability. In a Gigabit Ethernet system, the bit error rate at the channel is specified as 10^{-12}; even so, a line error may occur on average every 800s on a single link.

It was also observed that the 8B/10B coding scheme produces *error amplification* - a single bit line error can lead to up to 4 bits of error in link layer data. The 32-bit link layer CRC is well-known to detect all 3-bit errors but not all cases of 4-bit errors; the probability of this threshold being exceeded is higher than might be expected, due to this amplification. With longer frames, such as the jumbo frame, the likelihood of multiple error events within a frame is increased, and therefore the likelihood of an error pattern which cannot be detected by the CRC also increases.

It can be seen that the probability of an error pattern which cannot be detected by the CRC is extremely low for a channel bit error rate of 10^{-12}, since at least 3 independent channel errors must occur (for either frame size); regardless of the actual error patterns produced, this gives a maximum probability of the order of 10^{-36}. The next chapter investigates whether single-bit line errors are the most commonly occurring type in real links (which we assumed for the theoretical study of this chapter). It is also discovered that multiple error events within a single frame occur rather more frequently than the bit error rate would suggest. A study of the likelihoods of the MAC layer CRC detecting these multiple errors, for the case of the actual observed link layer error patterns, is described in Section 5.1.7.

Chapter 5

Observed Error Characteristics in Gigabit Ethernet

To complement the theoretical study of error behaviour in the previous chapter, an experimental investigation of the errors that arise as a result of low optical power margin in a real Gigabit Ethernet link transporting uniform data payloads is presented. It is observed that channel bit error rate and packet loss are both data-dependent, and that some payloads may suffer from relatively high BER and low packet loss, compared to others with lower BER and higher packet loss rates. It is also found that longer frames are more susceptible to errors than their length alone would suggest, and that the probability of an octet being received in error varies widely depending on the octet value. The physical reasons for this error hot-spotting *are explored, and some of the potentially unexpected implications of this effect are discussed. It is observed that errors occur more frequently than might be expected, both in terms of the number of channel bit errors and the number of separate error incidents within a frame. An examination of the effects of the block coding scheme on real channel errors leads into an analysis of the ability of the CRC to detect the resulting link layer error patterns. The use of a* data whitener*, to attain the uniform error probability (regardless of the transmitted data) that might be assumed, is investigated. In sharp comparison to these results for an optical 1000BASE-X link, equal error probabilities are observed for all data a copper-based network, 1000BASE-T.*

It has been shown that the line coding scheme affects the way in which a 1-bit line error might propagate through the lower levels of the network stack (Chapter 4). This work is also interested in the errors which actually occur in real networks at the physical layer, and how they affect errors at the data link layer and above.

In Section 5.1, the effects of low power at the receiver in a system is examined, as it is postulated that this state is likely to be increasingly common in the future (Section 1.3). Some of the implications of these findings are discussed in Section 5.2, and a method of eliminating some of the non-uniform error effects by using a data-whitener is presented in Section 5.3. Section 5.4 describes a testbed permitting the partial failure of a system due to non-uniform errors to be measured, and compares the error behaviour of Gigabit Ethernet on copper UTP (1000BASE-T) with that for fibre (1000BASE-X). Finally, the results of this experimental work are summarised in Section 5.5.

5.1 The Effects of Optical Attenuation

The optical signal in a Gigabit Ethernet link is attenuated to simulate the effects of reduced optical power budget margin; it is postulated that this type of encoded data under these conditions will be increasingly common in future optical networks. The ways in which physical layer errors propagate through the network stack will be the same regardless of their cause, be it low receiver power margin, patterning effects from the use of semiconductor optical amplifiers, or otherwise. Although by limiting the receiver power these experiments are going beyond the operating mode described in the standard, it should be noted that the link continues to function in this state and frames are still received.

5.1.1 Experimental Method

An extensive set of received packet data at low power margin levels was acquired and analysed, in terms of the errors observed, using Gigabit Ethernet on fibre (1000BASE-X [27]) equipment.

A combination of hardware and software was used to obtain the results given here; some software enabled transmission and processing of data during the network runs, other software was used for later analysis of the received traces. This section describes the overall test system, and then the software aspects, the optical physical layer, and finally the frame types used in the experiments.

Overall Test Setup

In the main test environment an optical attenuator was placed in one direction of a Gigabit Ethernet link. A traffic generator fed a Fast Ethernet link to an Ethernet switch, and a Gigabit Ethernet link was connected between this switch and a traffic sink and tester (Figure 5.1). The variable optical attenuator and, optionally, an optical isolator were placed in the fibre in the direction from the switch to the sink. It was believed that there may be interference due to

reflection and in some tests used an isolator to counteract this. However, the results with the isolator are much the same as those obtained without it (see Section 5.1.5).

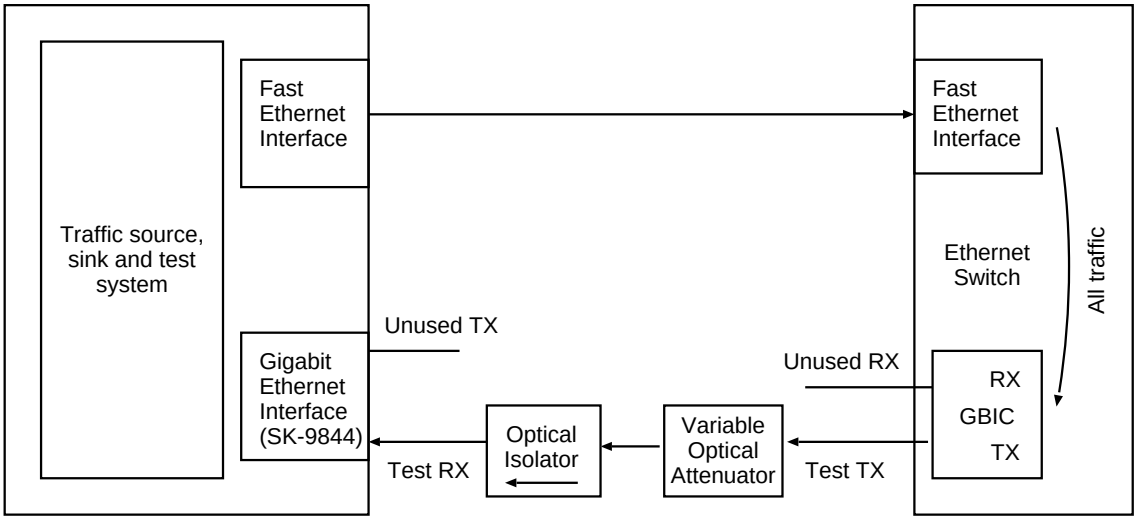


Figure 5.1: Main test environment for analysis of errors in gigabit fibre links

Interface and Analysis Software

A packet capture and measurement system is implemented within the traffic sink using an enhanced driver for the SysKonnect SK-9844 network interface card. Among a number of additional features, the modified driver allows application processes to receive error-containing frames that would normally be discarded, due to a failed frame check sequence at the Ethernet MAC layer.

As well as purpose-built analysis code in the receiving system, a special-purpose traffic generator and comparator is used. Pre-constructed test data in tcpdump-format is transmitted from one or more traffic generators using an adapted version of *tcpfire* [154], called *tcpfirediff*. *Tcpfire* simply transmits a frame or frames repeatedly; using *tcpfirediff*, transmitted frames are compared to their received versions and if they differ, both original and errored frames are stored for later analysis. This real-time software is illustrated in Figure 5.2.

The received traces, consisting of pairs of transmitted frames and their corresponding received versions which arrived in error, can be analysed. The analysis software logs various characteristics of the data, such as error position within the frame, and the octet value which is found in error and the data sequence it appeared in.

Optical Layer Setup

The desire to replicate as many of the the physical layer characteristics of the SWIFT prototype network(Chapter 2) as possible drove the selection of Gigabit Ethernet transceivers at

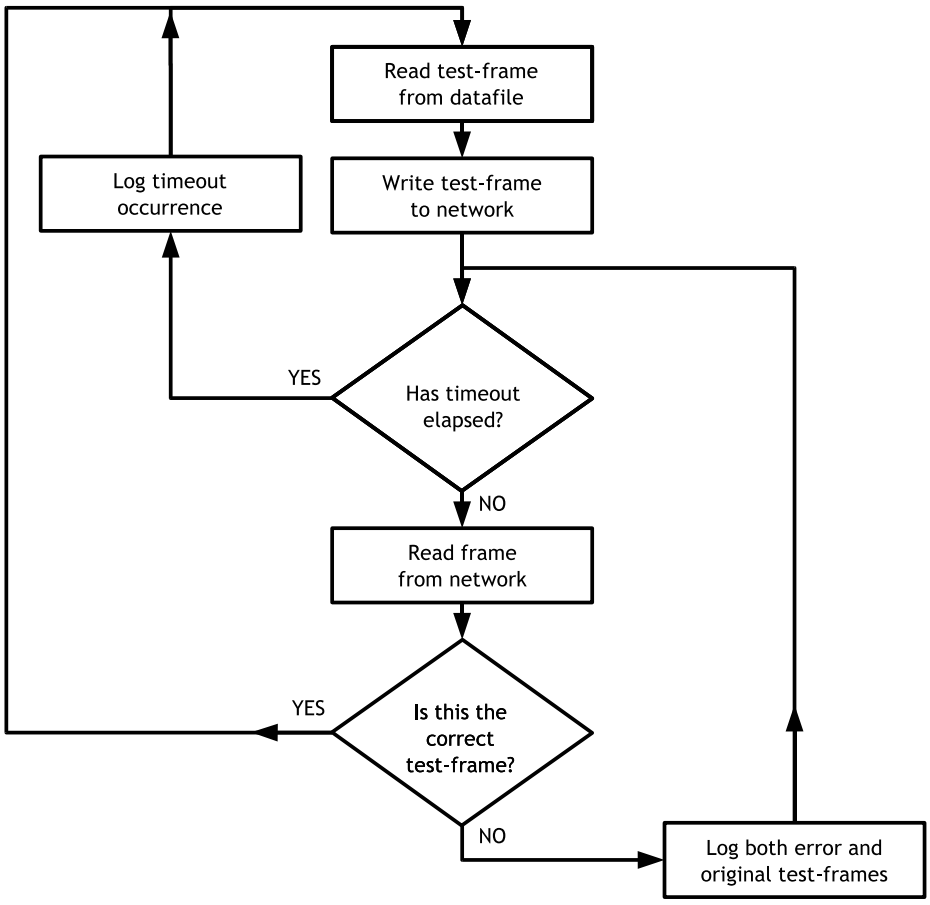


Figure 5.2: Flowchart of real-time fibre link test software, *tcpfirediff*

a wavelength of 1550nm. These are 1000BASE-ZX transceivers, designed for long haul fibre transmission; ZX is a Cisco proprietary extension to the official IEEE standard and, because of its rarity, it is not expected to be ratified in the foreseeable future by the IEEE [155]. This situation means that only the more limited specifications from the suppliers are available to us — no standards conformance is guaranteed for the behaviour of transmitters and receivers at this wavelength. Following discussions with a number of suppliers we were able to satisfy ourselves that these transceivers were sufficiently compliant and uniform in behaviour. This work generally used 1000BASE-ZX transceivers as transmitters, and 1000BASE-LX equipment (1300nm on single-mode fibre) for receiving.

Most of the results (notably those of error *hot-spotting*, Section 5.1.4, and *relaxed* decoding, Section 4.1.3) were confirmed by shorter tests, where other transceivers were used. A range of 1000BASE-LX transceivers were used, and also some special DWDM transceivers, which transmit at a closely specified wavelength (see Section 2.1.2). However, a certain range of attenuation values could only be realised for testing using the ZX transceivers. In many of the experiments, the available attenuators had high insertion loss and so that when connected the received power was too low for the receiver to maintain the link, even with no additional attenuation. However, the use of the high transmit power ZX transceivers overcame this; any model of optical Gigabit Ethernet transceiver could be used as the receiver. (The broadband wavelength properties of the photodiodes used as receivers in each case enable the treatment of 1550nm and 1310nm wavelengths equally.) In the cases of transmissions from other transceivers, it was possible to test a limited set of attenuations by loosening the optical connectors.

A range of receiver powers was used, but the results are similar for all powers. It is arguable that the powers are outside the IEEE 802.3z specification; however, during this testing the links were observed to operate normally — the hardware did not indicate insufficient receiver power (*signal_detect*). If the optical power level was reduced too far, the Ethernet link would go down, but during the tests where all the results mentioned here were obtained, this did not occur. See Sections 4.1.4 and 7.2 for further discussion of these issues.

Traffic Types

The traffic used for the majority of these tests consisted of frames with a distribution of sizes equal to that of a real network traffic sample (see Section 6.1). These frames were filled with uniform data: octets generated by a pseudo-random number generator. A subset of these experiments were also performed using the real traffic sample itself.

Level of Frame Damage Considered

In the results given below, frames containing large numbers of octets in error are not considered. These are usually frames where the signal has been briefly lost, or where the pairing software has compared against an extraneous packet not related to the test; this occurs rarely. The threshold selected was to consider only frames with fewer than 6 errors; frames with more errors than this made up only 0.96% of all damaged frames. Manual inspection of these cases showed that only approximately 1 in 20 of them were caused by effects other than unrelated frames interfering with the test. Most errors occured independently within the frames; consecutive errored octets comprised only 0.14% of the total error cases in the frames considered. These errors are not considered in the subsequent analysis.

The errors examined, then, occur in isolation within frames, separated by undamaged octets.

5.1.2 Comparing Channel Bit Error Rate and Packet Loss

The SWIFT network prototype outlined in Chapter 2 was to be undertaken as a collaboration between engineers with very different backgrounds, from computer networking to laser physics. These communities use very different methods to assess the performance of the systems they design and build, and it was important that they could understand what each others' metrics meant in their own terms. (This is just one of the challenging aspects of interdisciplinary projects, where groups with different methods, assumptions, priorities and terminology must work together.) To assist this work, it was decided to investigate how packet loss (used by the computer networking community) and bit error rate (BER, used in optical system assessment) were related. It might be expected that this relationship would be straightforward; assuming that errors follow a Poisson distribution, the probability of a frame x bits in length being received in error on a channel with a bit error rate of p would be:

$$P(\text{Errored frame}) = x \times p \tag{5.1}$$

if x is large and p is small. However, this is not found to be the case.

These experiments used two separate testbeds, the first being the main Gigabit Ethernet setup described in Section 5.1.1, and the second being a standard bit error rate test kit (BERT). In both cases, specially constructed frames were used. The *structured data* consists of a 256 octet long incrementing sequence, 0x00 to 0xFF, repeated to fill a 1500 octet frame. The *low error testframe* (perhaps a misnomer) is a 1500 length frame containing repeated 0xCC octets, whilst the *high error* frame consists of repeated 0x34 octets. These were sent using *tcpfire* over the Gigabit Ethernet link, where the packet loss rate was measured using *ifconfig*. For use with

the BERT, the frames were line encoded using 8B/10B as they would be in the computer link, and transmitted using a 1550nm DFB laser at 1.25Gb/s to an Agilent Lightwave receiver unit. The BERT counts the number of bits received in error and the total number of bits received, for repeated transmissions of the given bit sequence.

The results (originally reported in James *et al.* [156]) are shown in Figure 5.3. The receiver power scales are different in Figures 5.3(a) and 5.3(b), due to the different experimental setups used. In particular, the sensitivities of the receivers in the computer network interface cards, and the Agilent Lightwave unit, are not the same; this is because the equipment being compared is a computer networking card engineered for use in the long haul realm, and a standard lab bench optical-electrical conversion box.

It can be seen that bit error rate and packet error rate were not as obviously related as might be assumed. A frame which was found to be subject to a lower bit error rate than another would not necessarily show a lower packet loss rate. This effect is discussed further in Sections 6.3 and 6.4. In later parts of this chapter, it is found that channel errors are pattern-dependent, and that some patterns tend to be subject to multiple bits of channel error, rather than the single bit which might be assumed. These effects combine to create the data-dependent BER and packet loss rates observed here.

The next section presents results for error characteristics in pseudo-random data payloads. This work focuses on damage in payloads partly because the potential for undetected errors here is of interest, but also because it will be shown in Section 6.2.4 that the majority of errored frames are dropped due to errors within the payload which are detected by the FCS.

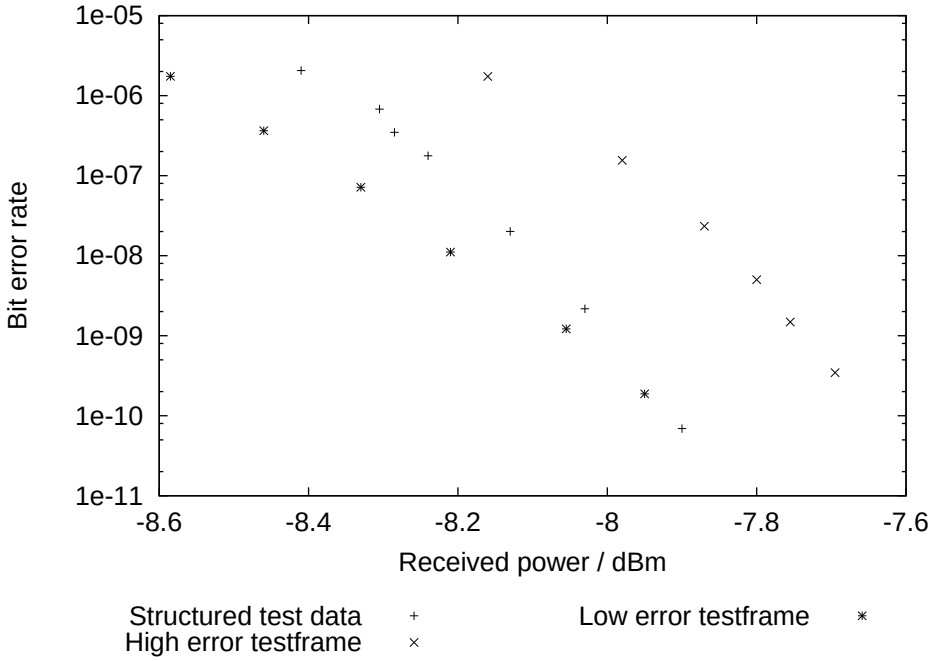
5.1.3 The Positions of Errors in Uniform Data Frames

These experiments generated many samples of errored frames containing uniform, random data. In terms of the positions of errors within these frames, it is instructive to consider frames of different lengths separately, rather than averaging over frames of all lengths. The frame lengths of 1492 octets and 46 octets are interesting, as they represent common frame lengths in real network data, making up 35% and 11% of the traffic sample respectively (see Section 6.1).

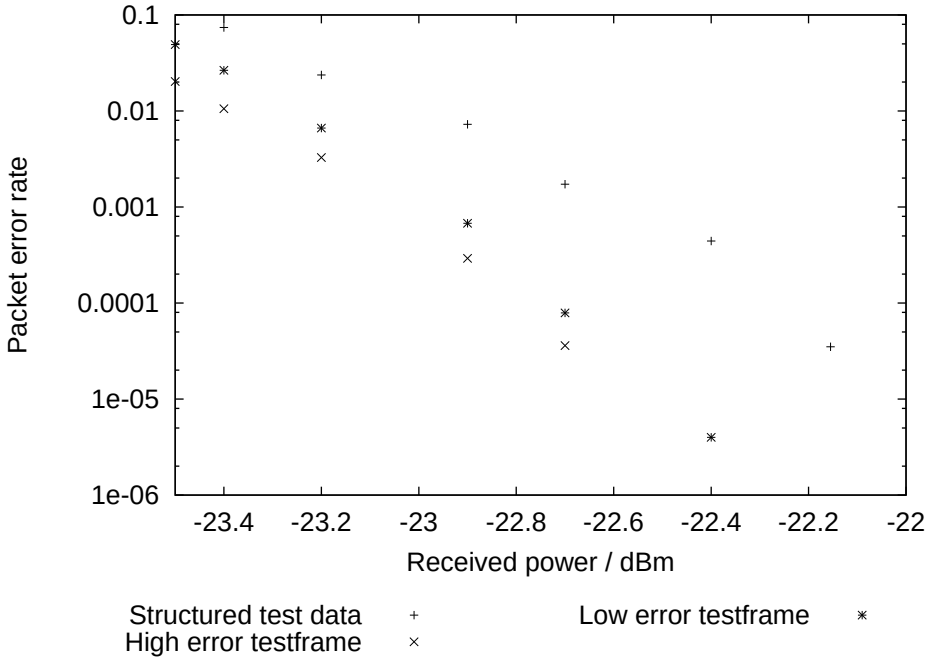
In the case of frames consisting of 46 octets of uniform data, the positions of errors are shown in Figure 5.4. The errors are approximately evenly distributed throughout the frame, with no evident correlation between error probability and position.

For frames of length 1492 octets, the error probability appears to increase steadily as the frame progresses, after an initial spike (Figure 5.5). The profile across the first 46 octets is similar to that for the 46 octet frames.

It is conjectured that these early octets may suffer from increased probability of damage



(a) Bit error rate versus received power



(b) Packet error rate versus received power

Figure 5.3: Contrasting packet-error and bit-error rates versus received power

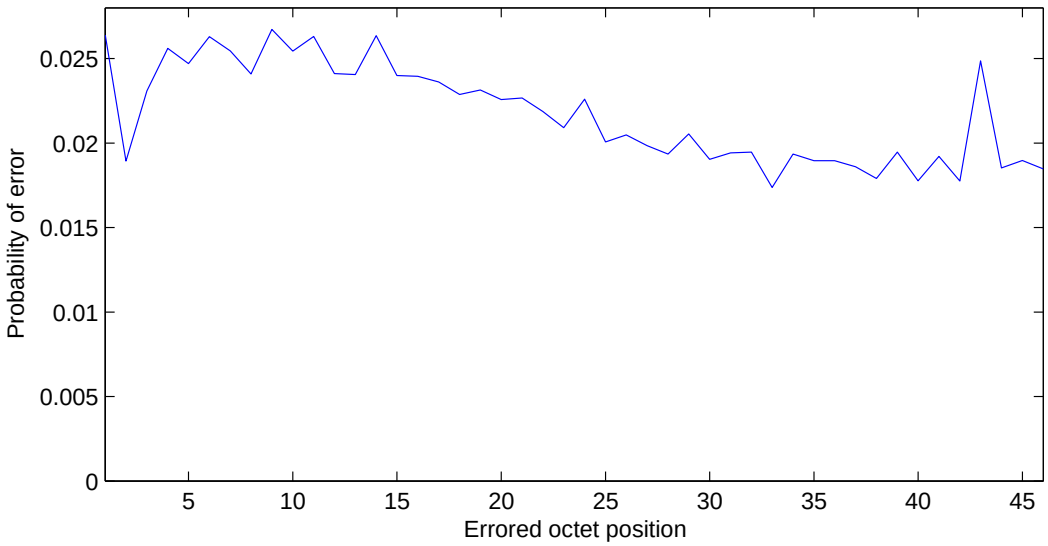


Figure 5.4: Error positions for frames of 46 octets in length containing uniform data

because the receiver electronics are still adapting to the arrival of a new data burst (power balancing in the analogue to digital converters (ADCs) and clock recovery systems, for instance). The gradual increase of error probability throughout the frame is most likely to be due to the growing time period since a reliable indicator of symbol clock (the 8B/10B *comma*, see Section 4.1.1). Issues related to clock failure-induced errors are also discussed in Section 5.1.8.

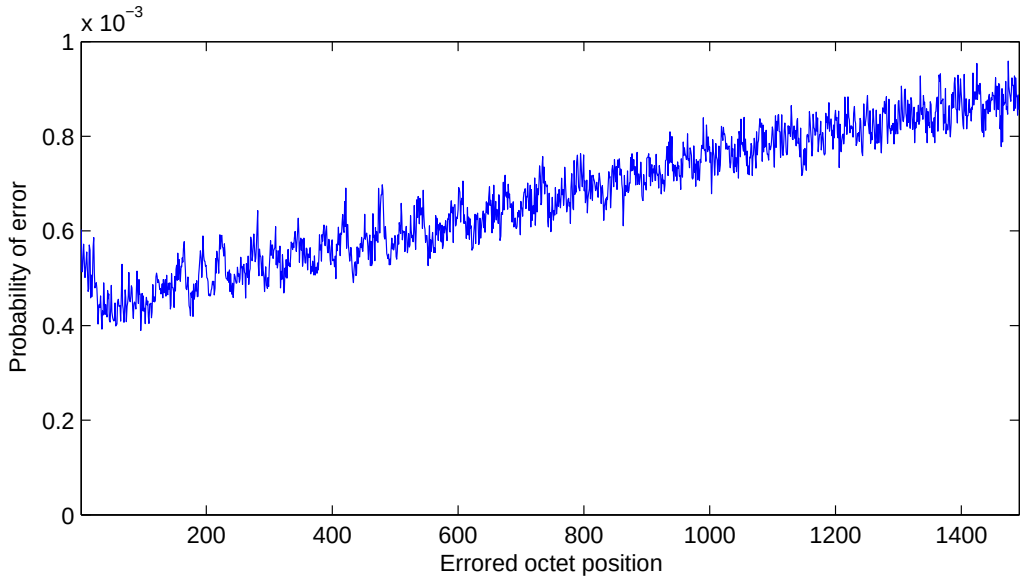


Figure 5.5: Error positions for frames of 1492 octets in length containing uniform data

This increasing probability of error through longer frames is confirmed when the rates at which frames of varying length are received in error is considered. All frames received with one or more errors are examined, and only those with frame lengths for which a reasonable number of errored frames were received (greater than 1000) are selected. To remove the effect of that part of increased error probability which is simply due to the frame length, the number

of errored frames for each frame length is divided by the number of octets in the frame. The error frequencies are then normalised by dividing by the number of times frames with those lengths are found in the traffic sample. The values are scaled and shown in Figure 5.6, which highlights the increased probability of damage in longer frames. Although no data is available here for jumbo Ethernet frames (9000 octet payload), it is reasonable to assume that these will be subject to a proportionately greater risk of error.

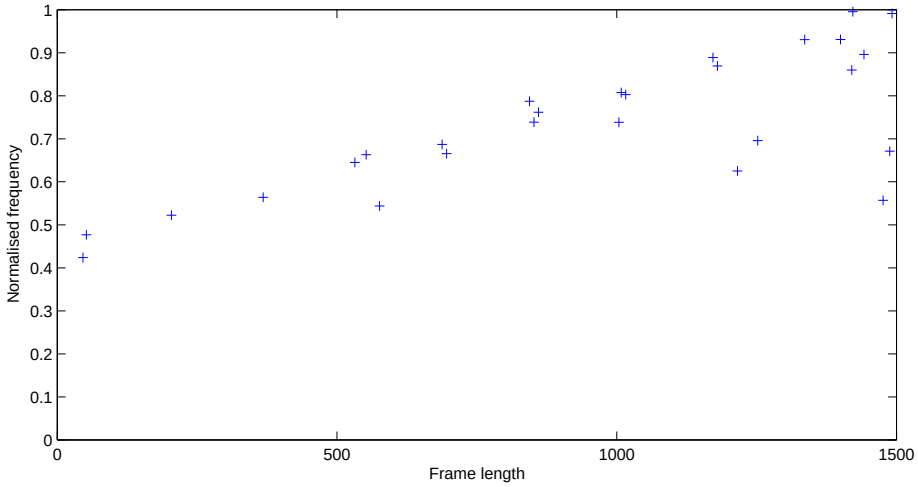


Figure 5.6: Normalised error frequencies for frames of various lengths

5.1.4 Correlation of Error Probability with Transmitted Data

From the results showing data-dependent packet loss (Section 5.1.2), it is anticipated that the probability of a block of data being received in error to depend in part upon the data values transmitted. The received data was analysed in terms of error probability per octet, as this is the basic data unit for 8B/10B encoded networks.

Data Octets Transmitted and Their Error Frequencies

A histogram of error frequency against octet value for the pseudo-random transmitted data case is given in Figure 5.7, where some octets can clearly be seen to be received in error far more than others.

The octets which are subject to a particularly high probability of damage at the receiver are called *hot-spots*. When these occur in network data, the probability of them being received in error is far higher than other, non-*hot-spot* octets.

The probability of errors in data is therefore not a uniform distribution as might be assumed; instead it is strongly data dependent. If some packets contained many of these *hot-spot* octets,

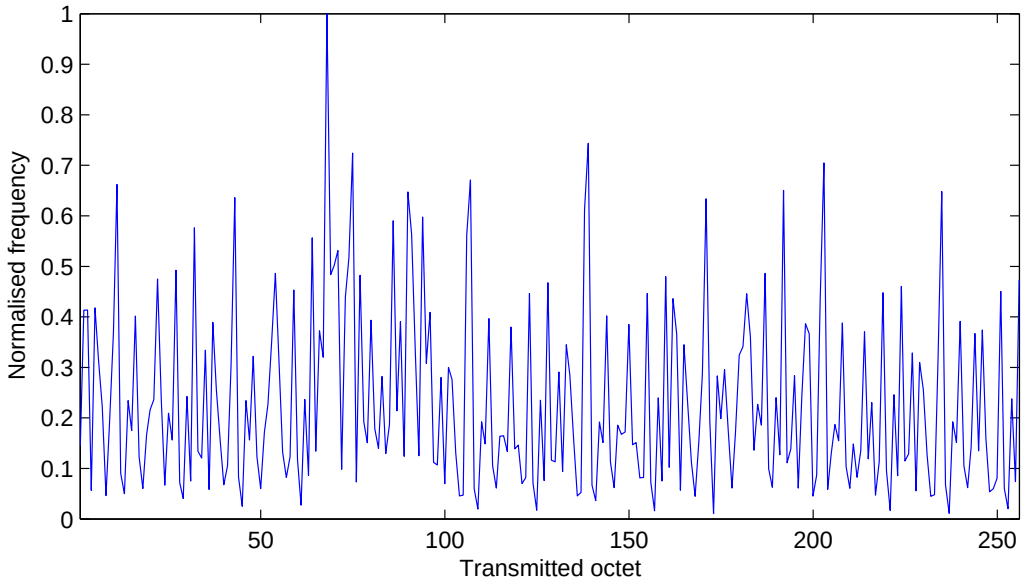


Figure 5.7: Error frequency versus transmitted octet values for uniform data

the likelihood of those frames being dropped would be much higher than that for frames containing octets with lower error probabilities. The reasons for this are discussed in Section 5.1.5, and further implications of error *hot-spotting* are given in Section 5.2.

Correlation of Error Probability With The Transmitted Data Sequence

Given that it is known that the coding scheme uses running disparity to maintain the line DC balance (Chapter 4), and that this involves the maintenance of state at the receiver, it would be particularly interesting to know whether errors are correlated with the data patterns preceding the observed errored octet.

The error frequencies for pseudo-random data were plotted again, this time for pairs consisting of transmitted octet which was received in error (X_i), and the preceding octet X_{i-1} (Figure 5.8). The error frequencies are shown on a logarithmic scale, with low error frequencies being blue, higher ones yellow, then red at the top of the scale.

Vertical lines on Figure 5.8 indicate octets which are found to be frequently in error, regardless of the preceding octet value - these are single octet *hot-spots*. Horizontal lines show octets which affect the probability of error in the octets following them; this indicates the presence of error correlation with the value of the previously transmitted octet. Intersections between these lines, and single "points" indicate two-octet sequences where the second octet has an extremely high probability of being received in error. These two-octet sequences have far higher relative error probabilities than a single octet by itself. (An individual "bad" octet may have an error probability up to 93 times greater than a "good" octet; when the previous

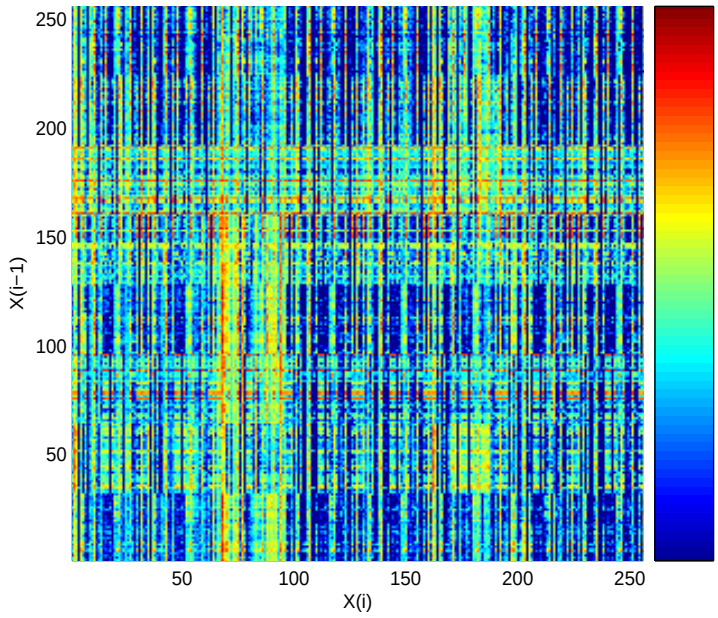


Figure 5.8: Error frequency versus octet values, for X_i and X_{i-1}, with colour scale from blue (low frequency) to red (high frequency)

octet in the sequence is also considered, a "bad" pair may be over 700 times more likely to be errored than a "good" one.)

Figures 5.9 and 5.10 show similar plots, but for pairs of octets going further back into the transmission history, beyond the octet transmitted as X_i, which was received in error. Figure 5.9 shows X_{i-1} versus X_{i-2} (the previous octet, and the one before that), and Figure 5.10 X_{i-2} and X_{i-3}. It can be seen that the correlation does not extend further back than the X_{i-1} octet (horizontal artifacts are due to the printing process).

In fact, as the work in Sections 4.1.3 and 4.2.4 showed, the running disparity is unlikely to be the cause of this correlation, as regardless of the decoding scheme a running disparity error will not cause a data code-group to data code-group type of error. In the next section, it will be found that pattern dependency is the most likely cause.

5.1.5 Why Data Dependent Errors Are Observed

It is interesting to observe the line codings used to represent the most frequently errored octets under the 8B/10B scheme. Examining the top ten most errored octets (regardless of preceding octet), it is found that the following ten octets give the highest error probabilities (independent of the preceding octet value): 0x43, 0x8A, 0x4A, 0xCA, 0x6A, 0x0A, 0x6F, 0xEA, 0x59, 0x2A. It can be seen that many of these in A, and the reverse-ordering of the 8B/10B scheme means that this causes the first 5 bits of the code-group to be `01010`. The high error octets not

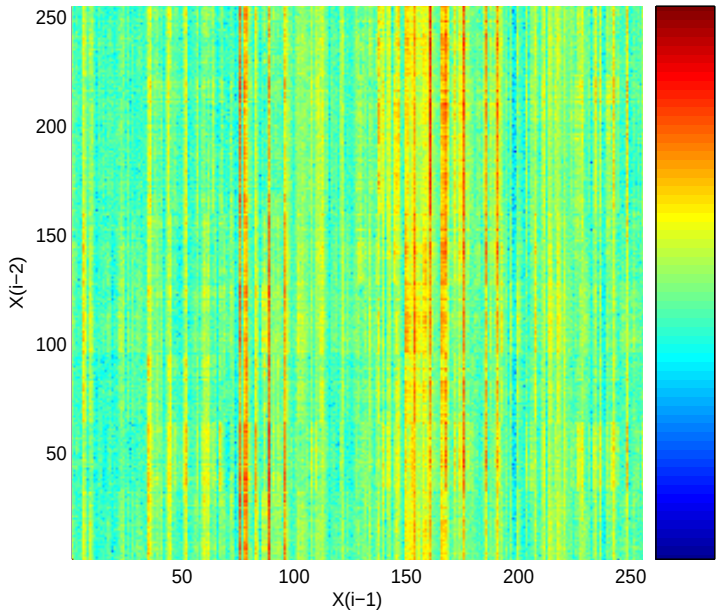


Figure 5.9: Error frequency versus octet values, for X_{i-1} and X_{i-2}, with colour scale from blue (low frequency) to red (high frequency)

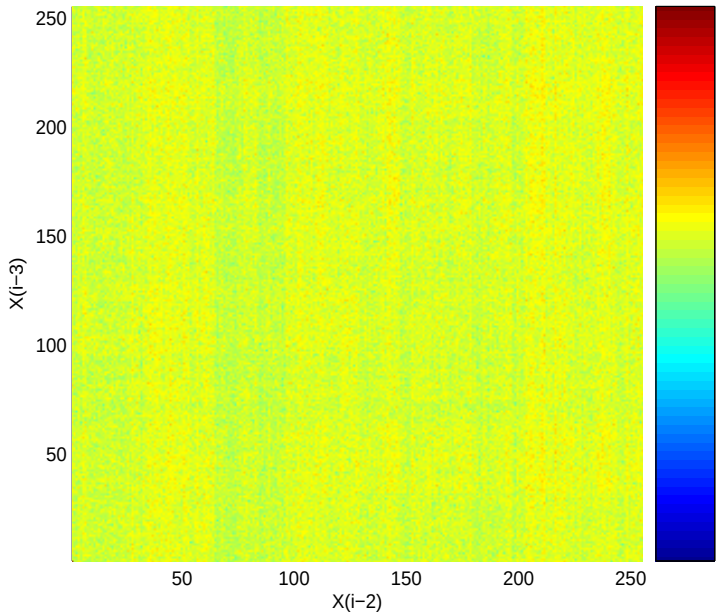
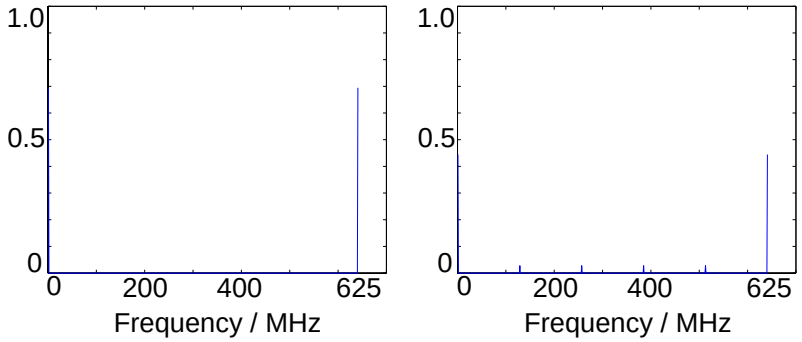


Figure 5.10: Error frequency versus octet values, for X_{i-2} and X_{i-3}, with colour scale from blue (low frequency) to red (high frequency)

beginning with this sequence contain at least 4 alternating bits. This pattern can also be expressed as a high frequency component at half the line rate (625MHz for Gigabit Ethernet). The Fourier Transforms (FTs) of sequences consisting of repeated code-groups for these octets illustrate this clearly (some examples are shown in Figure 5.11).



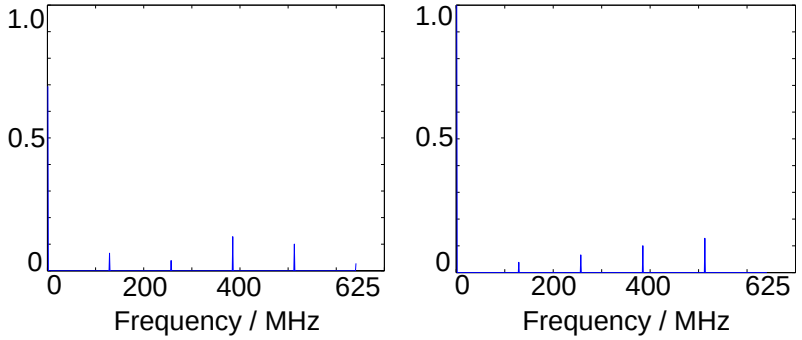
(a) FT of code-group for high error octet 0x4A

(b) FT of code-group for high error octet 0x0A

Figure 5.11: Fourier Transforms of code-groups for frequently errored octets

The ten octets giving the lowest error probabilities (independent of previous octet), are 0xAD, 0xED, 0x9D, 0xDD, 0x7D, 0x6D, 0xFD, 0x2D, 0x3D and 0x8D, and here the concluding D causes the code-groups to begin `0011`.

There is no notable peak corresponding to 625MHz in the FTs of the code-groups of these low error octets (examples in Figure 5.12).



(a) FT of code-group for low error octet 0xAD

(b) FT of code-group for low error octet 0x9D

Figure 5.12: Fourier Transforms of code-groups for infrequently errored octets

Possible pairs of octets, where the second transmitted octet is the one tested for error (X_i, with the preceding octet X_{i-1}), were also considered. The pairs of octets leading to the greatest error probability in the second octet give much higher error probabilities than any individual octet. The noted high error octets (eg. 0x8A) do occur in the top ten high error octet pairs and normally follow an octet giving a code-group ending in `10101` or `0101`, such as 0x58, which serves to further emphasise that frequency component.

The 8B/10B codec defines both data and control encodings, and these can be represented on a 10-bit by 10-bit space, illustrating code-groups C_i and C_{i-1} which correspond to octets X_i and X_{i-1}. Figure 5.13 displays the octet errors found in a real Internet data sample (Section 6.1) on this codespace, with darker regions showing code-groups with higher probabilities of error. These error probabilities have been normalised against the octet population in the real traffic sample. It can be seen that areas of high or low error tend to be clustered and that the clusters correspond to certain features of the code-groups. Two clusters have been ringed, those that are indicated as $C_i = 0011\ldots$ represent those codes with a low-error suffix. In contrast the ringed values indicated as $C_i = 010101\ldots$ indicates the error-prone symbols with a suffix of 0xA. Some code-groups and code-group sequences can be seen to suffer markedly higher probabilities of error than others; in some cases this is emphasised by the relative frequencies with which the corresponding octets occur in Internet traffic.

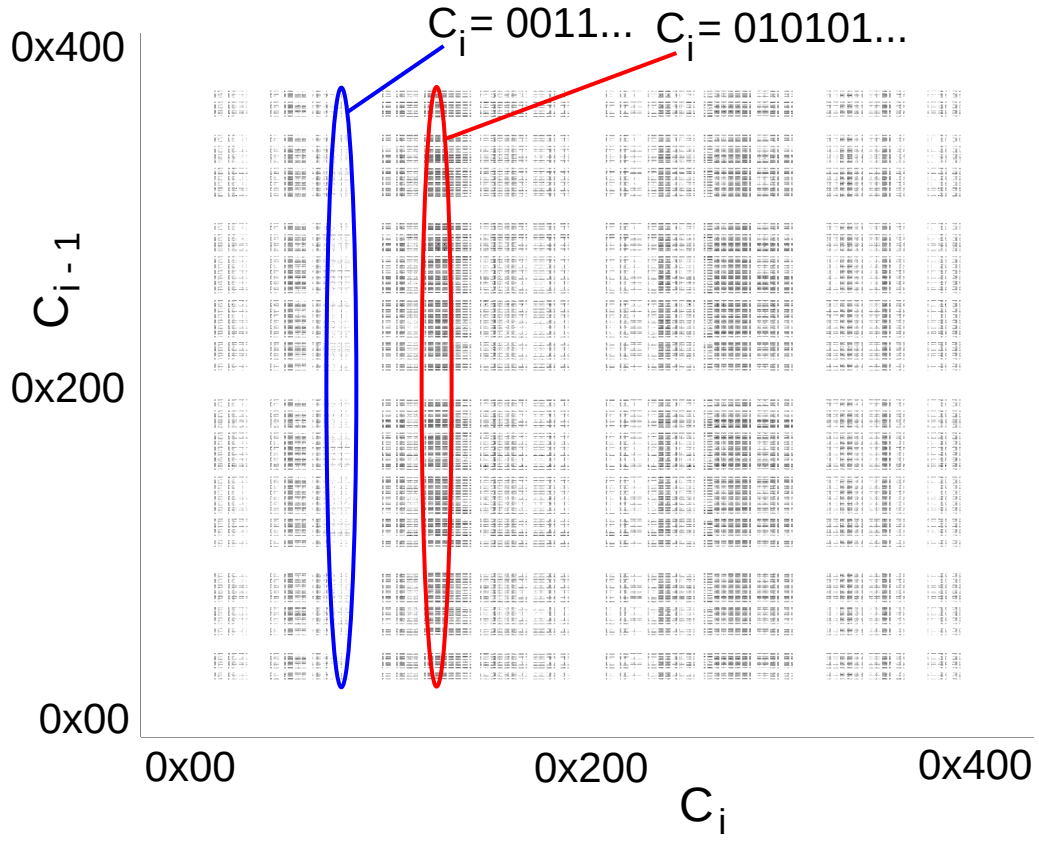


Figure 5.13: Errors in real network data as a function of code-groups, shown in terms of C_{i-1}, C_i pairs

Physical Causes of Frequency-dependent Errors

The bit frequency dependent error patterning observed is primarily due to effects in the electrical/optical interfaces. It is well known that in a directly modulated electrical or photonic

signal, it is possible that bandwidth limitations can cause *single ones* to achieve slightly less amplitude than a run of multiple ones.

An eye diagram is commonly used to assess the physical layer of a communications system, where the effects of noise, distortion and jitter can be seen in a single diagram [157]. Received analogue waveforms (before any decision circuitry) for each bit of a pseudo-random sequence are superimposed to create an "eye". The central open part of the eye represents the margin between `1`s and `0`s, where both the height (representing amplitude margin) and the width (representing timing margin) are significant.

In normal operation, the slight eye closure due to bandwidth limitations reducing the amplitude of single `1` bits has no effect on the error rate of the received signal. Figure 5.14 illustrates this effect in an operating Gigabit Ethernet link. Despite this eye closure, error-free operation (defined as that with a BER of 10^{-9} or less) is achieved at a received power significantly above the receiver sensitivity. However, as the received power is reduced toward the sensitivity of the optical receiver it is the *single ones*, e.g., `010101` which produce errors first, as these are of lower amplitude than the *multiple ones*, e.g., `110011`.

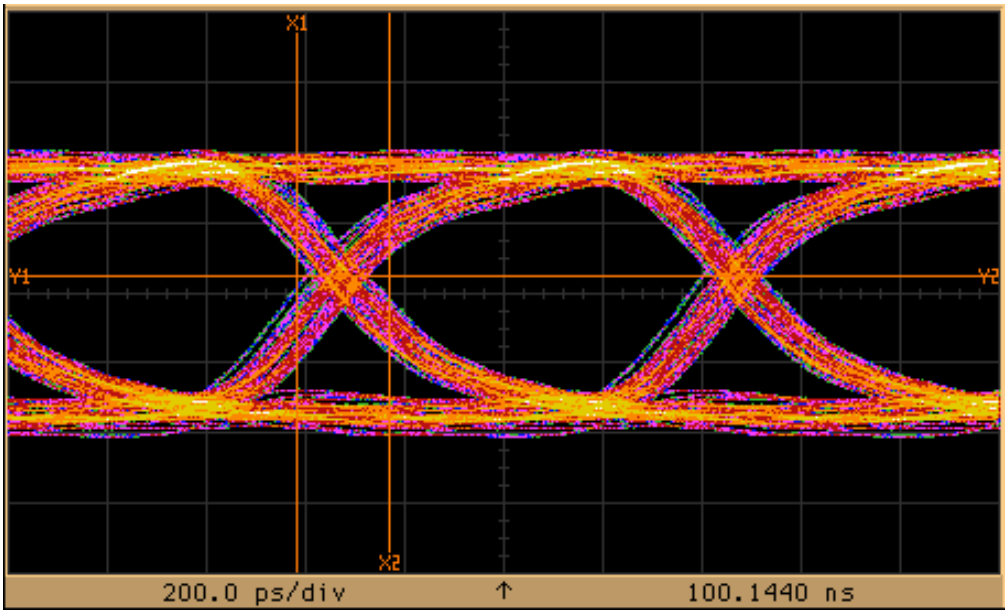


Figure 5.14: Example eye diagram for an optical Gigabit Ethernet link

In addition, optical packaging requirements and other circuits and systems (printed circuit board tracks, wires, etc.) in the electrical domain will exacerbate this effect. These broadband limitation effects will be much more significant at the increased modulation rates required for 10 Gbps Ethernet, and indeed for all faster optical networks.

It is worth noting that the code-groups with multiple transitions, which have been observed as subject to increased error probability, are also beneficial to the network. This is because

clock and data recovery systems at the receiver require these frequent transitions to recover and maintain an accurate bit clock.

The Effects of Reflection

Similar tests to those detailed above were performed using uniform data with an isolator in the fibre link, to examine whether reflections were causing some, or all, of the errors.

The results for this setup are much the same as for the case without the isolator; errors are observed in similar numbers for similar receiver power values, and the set of octets most likely to suffer damage remains much the same. The "top ten" octets with the highest probability of error are now 0x43, 0x5D, 0x4A, 0xCA, 0xA1, 0x8A, 0x0A, 0x2A, 0x01 and 0xEA. The new entries in this list all have code-groups with a substantial alternating bit sequence. Given the potential causes of increased error probability in these sequences, the minimal performance change when reflections are removed is not surprising.

Most computer networks using fibre (Gigabit Ethernet, FibreChannel etc.) will not include an isolator in the system. They are designed and built for minimum cost, and use comparatively simple optical subsystems at the ends of each fibre link. Chapter 2 showed that future optical network designs are much more complex; however, isolators do not usually play a role in their design. In prototype systems, isolators may be included to reduce reflection problems caused by imperfect connections; in deployed systems where components are not being added, removed or moved around, isolators are less likely to be required. These results are reassuring on this front - isolators do not significantly change the system performance.

5.1.6 Physical Layer Errors

Based on the frequency components of the code-groups, it is possible to predict the octets most likely to be received in error. However, the actual errors are also of interest: both in terms of the line errors occurring, and the data layer errors which affect the MAC layer.

In the experiments using Gigabit Ethernet, received frames including those where the data fails the FCS check were observed. However, it was only possible to observe the data at the decoded stage, not the actual line representations.

Deductions can be made about what line representations were used and therefore what line errors might have occurred. Comparisons like this were used earlier (Section 4.1.3) to deduce the existence of systems where the disparity is not checked at the receiver. In the case of the *strict* decoding scheme, there are two possible representations for each octet, depending on running disparity, and two possible line errors between any pair of transmitted and received

octet values. For *relaxed* decoding, the analysis is more complex, from the same two line codes for each octet up to 4 different line errors may have occurred.

Clearly, given a specified bit error rate such as 10^{-12}, one does not expect many bit errors to occur; and when they do occur, intuitively one might anticipate a single bit error to be the most common one. (Indeed, the analysis presented in Chapter 4 takes the traditional form, with single bit errors assumed to be those most likely to occur.) Although no measurement of actual line bit error rate was made for these experiments, the packets under examination were observed where the probability of a packet being in error was less than 0.1% in all samples (and often much less than this), and where each packet contained fewer than 6 octets in error. Errors are therefore comparatively rare. This work will now determine if only single bit channel errors are generated in this environment, or whether multiple bit errors per code-group are occurring.

In the previous chapter, the equipment with which experimental results were obtained was shown to use the relaxed 8B/10B decoding scheme. This relaxed decoding method is therefore used to deduce the physical layer errors which occurred here. However, it is reasonable to assume that similar line errors would occur in both relaxed and strict coding systems. The outcomes of these line errors, if they were received by a strict interpretation of the decoder instead, would be different. The previous Chapter showed that all 1-bit errors in data are detected by the strict decoding method; it is possible that some 2 bit errors may go undetected, since a pair of errors in the same code-group could go undetected by the disparity check.

Firstly, it is determined whether single bit or multiple bit errors are observed at the physical layer. For the observed octet transitions in pseudo-random data, the code-groups that can be used to represent the transmitted and received octets are examined, and the minimum number of physical layer bit errors which can have caused the octet errors noted for each ase.

Data documenting the number of physical layer bit errors per damaged code-group is presented in Table 5.1. The *occurrences* value indicates the number of transmitted/received octet pairs (out of a possible $256^2 - 256 = 65280$) where the minimum number of bit errors is as shown. The *frequency* column indicates the number of times the error transitions with this number of minimum bit errors were observed in the pseudo-random data sample.

Examining the data in Table 5.1 it can be seen that multiple bit errors are less common than single bit ones, but by no means rare. Out of the total sample or errored code-groups, 87% contain single bit errors, and 5.4% 2-bit errors; nearly 2% of octets in error have more than half their physical line representation received in error. (Remember that these results are based on deductions of the *minimum* number of bit errors which could have generated the observed octet error.)

When the 10-bit error patterns themselves are considered, it is encouraging to note that the 1-bit errors do dominate as expected. The ten most frequently occurring patterns are the

Bit Errors	Occurrences	Frequency/%
1	2480	87
2	7864	5.4
3	16124	4.5
4	18908	0.78
5	12280	0.46
6	4762	0.10
7	1800	1.5
8	978	0.14
9	12	0.00
10	72	0.00

Table 5.1: Minimum numbers of physical layer bit errors required to cause observed error transitions in pseudo-random data

ten possible single bit error positions. There are no other notable pattern features.

A Note on the Probability of Multiple Channel Bit Errors Per Code-group

Chapter 4 showed that a single physical layer bit error gives rise to 1 to 4 bits of data layer error, with an expected number of data layer errors of approximately 2 bits if uniform data was transmitted (Section 4.2.4). If the data is not uniform, then this value may be slightly different.

However, this assumes that the normal error type is a single bit channel error, and it was seen in the previous section that multiple bit errors also happen. Many interpretations of the bit error rate assume that it can be interpreted as the probability of any given bit being received in error. Multiple bit errors should then be much less common than single bit errors; given a bit error rate of 10^{-9}, say, a single error within a 10-bit code-group should occur with a probability of:

$$P(\text{1 bit error in a symbol}) = 10^{-9} \times (1 - 10^{-9})^9 \times 10 = 9.9999 \times 10^{-9} \approx 1 \times 10^{-8} \qquad (5.2)$$

However, 2 bit errors should be proportionately less probable:

$$P(\text{2 bit errors in a symbol}) = (10^{-9})^2 \times (1 - 10^{-9})^8 \times^{10} C_2 = 4.50 \times 10^{-17} \qquad (5.3)$$

and more than two errors should be even less so.

The ratio between these probabilities is of interest; 87% of octets were observed to have suffered single bit errors at the physical layer, and 5.4% with 2-bit errors. This ratio does

not match up well with the theory which assumes all bits are equally likely to be received wrongly, and which predicts 2-bit errors to be many orders of magnitude less likely, regardless of actual bit error rate. It is possible that these multiple bit errors are not due to noise, but synchronisation loss (see Section 5.1.8).

However, the bit error rate can also be interpreted as an expected value, rather than a probability [158]. Thus the expected number of bits in error in a sample of 10^9 bits with a BER of 10^{-9} is 1. Within the 10^9 bit sample, more bits than this, or none at all, may be found in error. Indeed single-bit errors may not be the most likely outcome at all; instead, an alternative can be considered. If, on average, the main error type is a 2-bit sequence received incorrectly every 2×10^9 bits, and other types of error including single bit errors never occur, the bit error rate is still 10^{-9}. Sections 6.3 and 6.4 revisit this potentially unexpected characteristic.

This is an interesting complement to the work of Stone and Partridge, who demonstrated that packets are far more often corrupted than simple bit error rates would suggest [139].

5.1.7 Data Link Layer Errors

The MAC layer of Gigabit Ethernet, and many other systems, uses a 32-bit cyclic redundancy check (CRC) to ensure the integrity of the data carried in a frame. CRCs were described in depth in Section 3.2, where it was shown that in their design and specification for an application, errors are usually assumed to be uniform and independent. This chapter has demonstrated that channel errors are not uniform — the error probability is strongly dependent on the data transmitted — and that the line coding scheme used distorts the error patterns further. Also recall that Section 4.2.4 noted that even single bit errors on the physical layer will often translate into multiple bit errors (up to 4 per octet) following decoding. It is anticipated that real channel errors, which sometimes affect more than one bit per code-group, will lead to a greater number of data layer bit errors.

A survey of the data layer bit errors found per damaged octet in the sample of errored frames of uniform data is shown in Table 5.2. A number of cases where all 8 bits in the octet were in error were found, but formed too small a proportion of the overall sample to be shown here.

It is found that the mean number of bit errors per octet at the data layer is 2.44. This is slightly higher than the effective BER worked out for the case of single bit channel errors in Chapter 4, because of the instances of multiple bit channel errors in the real system. Clearly, only 2 error events (2 damaged code-groups) on the line might be required to obtain the 4 bits which could exceed the CRC32's capacity for error detection.

Data layer bit errors	Frequency
1	27.6%
2	29.0%
3	19.5%
4	20.9%
5	2.2%
6	0.3%
7	0.5%
8	0.0%

Table 5.2: Number of data layer bit errors per octet for pseudo-random transmitted data

The Effect of Data Link Layer Errors on the Frame Check Sequence

While it is known that the 32 bit checksum used in Ethernet will detect all 1, 2 and 3 bit data errors for payloads no larger than the 1500 byte MTU from the specification [27], some 4 bit data-errors are not detected. (However we have shown that all 4 bit errors induced by a single bit channel error can be detected, in Section 4.2.6.)

This section examines how effective the 32 bit CRC is at detecting errors when multiple data layer bit errors occur; Section 4.2.6 described a similar investigation for the errors arising from a 1 bit channel error only.

For the case of a standard 32 bit CRC used to protect a 1514 octet block of data (equivalent to a frame containing a MTU-sized payload), a computer search was performed to detect whether any data layer errors would go undetected. It is known that an error burst of at least 32 bits is required for this, so no single octet in error will go undetected by the CRC, but two independent octets in error might. A search across all possible data layer error patterns, for two octets in error within a 1518 octet data block (including the 4 FCS octets themselves) revealed a number of error patterns which would not be detected (Table 5.3). These are defined in terms of the octet positions within the frame, and the error patterns that must be present at each position. The octet position is the offset of the octet in question counted backwards from the last octet of the data and checksum block, which is counted as zero. These error patterns could occur in any data and are merely bit patterns which could be added to a correct data frame to generate a frame with an undetectable error.

These are error patterns at the data layer. It would be interesting to know how often they occur in the experimental results, i.e., whether they are the outcomes of common physical layer errors.

Only 5 of these error patterns occur more than 100 times in a sample of 4 million link layer octet errors (most occur with frequencies in single figures). These are the patterns 0x1C

Octet Position 1	Error Pattern 1	Octet Position 2	Error Pattern 2
12	0x9d	453	0xba
16	0x72	369	0xee
37	0x1d	758	0x1f
45	0xd0	1262	0xfa
67	0x35	347	0xf0
100	0x98	1142	0x49
157	0x67	1470	0x2
199	0x64	807	0xcd
313	0x30	1013	0xf4
339	0x16	1298	0x91
447	0x70	502	0x3d
485	0xa3	1377	0xce
531	0xb9	1131	0x93
643	0xf4	1441	0x59
1008	0x58	1020	0x1c
1027	0x55	1335	0x97
1075	0xb3	1306	0xcd
1142	0xf2	1212	0xeb
1173	0xb9	1228	0x2d

Table 5.3: Two-octet error patterns in Ethernet MTU-sized frames, which cannot be detected by the 32-bit CRC

(2653 times), 0x1F (30120), 0x16 (51548), 0x02 (114924) and 0x1D (299294). This distribution reinforces the observed non-uniformity of error behaviour at the physical and data link layers.

From this the proportion of frames where the CRC is defeated by two octets in error, given that two code-groups are damaged within the 1518 octet data frame, can be calculated. Ignoring the cases where both the error patterns are infrequently occurring in the random data sample, there are four possible error patterns to consider. The proportion of 1518 octet frames where the CRC is defeated by two octets in error is given by:

$$\begin{aligned}
\text{Proportion} &= \sum_{\text{Error pattern pairs}} P(\text{Error pattern 1}) \times P(\text{Error pattern 2}) \\
&= \sum_{\text{Error pattern pairs}} \left\{ \frac{\text{freq of pattern}}{\text{total freq}} \right\} \times \{P(\text{required positions})\} \\
&= \left\{ \frac{1}{1518} \right\}^2 \times \left\{ (\frac{2653}{3894491} \times \frac{13}{3894491}) + (\frac{299294}{3894491} \times \frac{30120}{3894491}) \right. \\
&\quad \left. + (\frac{51548}{3894491} \times \frac{8}{3894491}) + (\frac{114924}{3894491} \times \frac{8}{3894491}) \right\} \\
&= 2.58 \times 10^{-10}
\end{aligned} \tag{5.4}$$

So 2.58×10^{-10} of 1518 data-octet frames containing two errored octets will not have their errors detected by the CRC.

For the case of jumbo size Ethernet frames, multiple instances of 2 octet errors were found to be undetectable by the CRC. This search is computationally time consuming and was terminated before all possible cases had been examined; as in the previous chapter, the goal of this work was to show the existence of undetectable error patterns rather than to fully enumerate them. At this stage, 48 cases where the CRC would not detect the error pattern had been found. An estimate of the total number of such cases, based upon the distribution of the results obtained thus far, suggests that around 500 cases might in fact exist.

Multiple Error Events Per Frame

It is easy to dismiss the possibility of a CRC-defeating error pattern as being extremely unlikely, as it may be expected that most frames will only suffer a single error event. This is certainly the result expected from a model which assumes each additional error event to be less likely by a factor of the error rate. However, although a single error in a frame is by far the most common type of error (making up 88% of damaged frames observed), multiple errors per frame do occur. Even ignoring the small number of consecutive errored octets, and the observed frames with more than 6 errors (some of which it is conjectured may have been due to causes

other than simple octet damage, and some of which were caused by non-test frames reaching the analysis software), nearly 10% of the errored frames had 2 octet errors, and 2% had between 3 and 5 errors.

For an MTU sized frame, assuming each octet error event is independent and occurs with a probability equal to some error rate ER, the probability of two octets being in error in one frame is given by:

$$P \approx \left(\begin{array}{c} 1518 \\ 2 \end{array} \right) \times ER \times ER = 1151403 \times ER^2 \tag{5.5}$$

From this, many orders of magnitude difference in the frequencies at which one and two errors per frame occur would be expected. In practice, multiple errors are much more common; this may be due to effects other than random bit errors, such as clock loss (see Section 5.1.8).

The equivalent expression for jumbo frames is:

$$P \approx \left(\begin{array}{c} 9000 \\ 2 \end{array} \right) \times ER \times ER = 40495500 \times ER^2 \tag{5.6}$$

The use of longer frames, such as the jumbo frame considered here, therefore also increases the likelihood of multiple errors occurring per frame, in this case by a factor of 35. This is in addition to the increased risk of longer frames being damaged, observed in Section 5.1.3.

5.1.8 Synchronisation Errors

It has been shown that multiple channel bit errors may occur within a code-group. In addition, multiple error events may occur within a frame — these are not in consecutive octets and so appear to be caused independently. It is conjectured that some of these may be due to synchronisation effects, rather than "simple" noise affecting individual bits. It is also known that long frames become subject to increasing error probability as time elapses from the last reliable indication of symbol clock (the *comma* in the Idle sequence between frames, Section 4.1.1). Jitter in the symbol clock, which is used to align the deserialisation system to the location of the 10-bit code-groups in the data stream, could cause the sampling of the bitstream to occur at a time shifted slightly from the desirable centre of the eye (Section 5.1.5).

Clock and signal jitter become more problematic for high speed serial links, requiring careful design and construction of electronic systems. This is likely to worsen at higher line-rates, for coding schemes with poorer clock recovery characteristics than 8B/10B, such as 64B/66B, and for packet switched systems which require burst mode clock recovery (see Sections 2.2.5 and 7.3.2).

5.2 Network Performance Implications of Error *Hot-spotting*

A number of potentially unexpected error characteristics in a real optical system, from error probabilities dependent on position within a frame and the data sent, to the effects of a block coding scheme on error propagation through network layers, have been noted. This section reflects on their implications.

5.2.1 Error Detection Issues

These observations about the non-uniformity of error patterns, caused by both the physical characteristics of the channel and the coding scheme, may impact the performance expected from CRCs. It was shown earlier how non-uniformity may distort the statistics which quantify the ability of the CRC to detect certain error patterns (Section 3.2.3).

It is believed that the forms of errors observed here will, in general, be detected by the CRC in use at the MAC layer; however, any CRC failure in the case of a TCP system will still lead to a frame re-transmission. This adds to the overall system delay, reduces available bandwidth and is an undesirable outcome, particularly in heavily-utilised networks.

The potential fraction of errors which would go undetected by the CRC is very small, but the ever-growing amount of data stored and transmitted worldwide means that that proportion still represents an increasingly large number of frames. The use of longer frames in faster networks and systems, particularly those which may be subject to higher error rates than has been the case to date, also suggests that the 32-bit CRC may no longer be entirely adequate for the error protection task in hand.

5.2.2 Impact on Higher Network Layers

So far this work has considered the effects of errors in uniform data; in real network traffic, some octets occur far more often than others. This could worsen the *hot-spotting* effect, if these common octets and octet-sequences coincide with the *hot-spots* for a given system. In addition to increasing the chances of a frame being discarded due to its data contents, the occurrence of *hot-spotting* also has implications for higher level network protocols. In addition to effects on user-level data payloads, this error concentrating effect could cause a significant level of loss due to network and transport layer header contents. In one hypothetical case, if a user was on a machine with an IP address that consisted of several high-error-rate octets their data would be at a proportionally higher risk of being received in error. This is discussed in more detail in Section 6.1.3.

A Note on Equipment Issues

It could be argued that the results detailed here are specific to the equipment used. These experiments were repeated using a number of different NICs and physical layer interface modules (GBICs) from various manufacturers, supporting standard Gigabit Ethernet as well as the ZX extension. The overall effect of error *hot-spotting* was shown in all cases. The *relaxed* decoding scheme was also observed on multiple hardware platforms.

Receivers will all have different tolerances, and will be prone to certain types of errors more than others. Some may suffer from poor `1/0` thresholding or equalisation. Others might have a poor high frequency response, as observed in this work; this is likely to be a particularly common type of pattern-dependency. As systems increase in speed, particularly where serial transmission towards 10Gbps and beyond is involved, it is extremely hard to design and build suitable electronic hardware [28]. A case in point here is the equipment used for the SWIFT prototype network interfaces (Section 2.2.8): despite being supplied by a firm experienced in device and circuit manufacture, the boards suffered from a number of faults making them incapable of working at the specified frequency range. Not only was the wrong type of chip holder used for the core FPGA, but the board tracks were poorly routed and the power lines insufficiently noise-resistant. Particular instances of non-random, pattern-dependent error have been noted before. Clark Gaylord [159] cites a specific section of a file which could not be transported in one direction of a certain FDDI ring, when other data could be sent without problems; this was possibly due to pattern-dependent reflection and was corrected by cleaning the fibre.

So each setup may have its own set of channel patterns which are particularly prone to corruption, leading to *hot-spotting* at different octet values. However, the alternating bit sequence on the line as described above is likely to be a commonly error-prone pattern due to the frequency response issues in the optics and electronics.

5.3 Whitening to Achieve Uniformity of Error

The non-uniformity of error observed in this chapter is interesting, and not necessarily expected by those working at higher network layers.

Experiments were performed to investigate the use of a data whitener and establish whether this would restore the uniformity of errors such that all data octets would suffer from equal probability of damage. This may or may not improve the probability of the damage being detected at the data layer; however, it is conjectured that almost all errored octets will be detected by the coding layer or the FCS at the MAC layer, so the actual damage is not of much interest. The aim is to offer protection to packets which consist of data which would

normally suffer abnormally high probabilities of error, such that their error rate would be similar to other packets. This could be viewed as equalising error probabilities across all users, or applications, or network segments.

5.3.1 Method

An alternative to block codes such as 8B/10B and introduced in Section 3.1, scrambling also provides a process of encoding digital "1"s and "0"s onto a line in such a way that provides an adequate number of transitions, and a given balance of "1"s and "0"s. A number of communications standards use scramblers; one example is SONET, which uses a 7-bit scrambler by default or a higher-grade, 44-bit, scrambler for data payloads [160]. Another example is the 10 Gbps Ethernet standard, 10GBASE-LR, which uses a 64B/66B encoding system [36].

Additionally, the use of scramblers to pre-process data before line encoding is common. This usage is termed *data whitening*. The IEEE 802.15.4 spread-spectrum wireless personal area network [161] specifies a whitener to suppress variations in the power spectral density. A further example is the 800 Mbps Firewire/IEEE 1394b specification which uses a data-whitener to normalise data and improve the performance of the 8B/10B codec used in that system.

An implementation of the 64B/66B scrambler from the 10 Gbps Ethernet standard was used to *whiten* the frames from the sample of representative network data, the *day-trace*, described in more detail in Section 6.1. A similar test could have been performed using the pseudo-random data frames described above, but the network data provides a more useful demonstration.

This real internet data is non-uniform, concentrated on certain octet values. Clearly this may serve to exacerbate the non-uniform error behaviour noted above, as some of the octet sequences most subject to error may also occur in the most frequently transmitted *day-trace* regions. By whitening the data before transmission, the octets transmitted will be spread over the entire available octet space, such that the 8B/10B codebook is fully and evenly utilised, and high error probability code-groups are sent no more often than low error probability ones. This also means that when a high error code-group or code-group sequence is received in error, it is not always the same transmitted data pattern which is damaged. So, all data octets are equally likely to suffer from errors; the mathematical justification behind statements of robustness for checksums and CRCs, based upon equal error probability throughout the frame, are now reasonable.

The scrambler is run continuously, rather than restarting for each new frame, and is implemented as a *shim*-layer between the data link and network layers. This only whitens the data of the Ethernet payloads, not the packet headers or the CRC (which is calculated at the data link layer itself).

5.3.2 Results

It was found that the whitened version of the real network data trace contains all possible octet pairs at frequencies similar to the uniform data frames, so the varied characteristics of the network data have been successfully evened out by the scrambler.

When the octet errors in the attenuated, 8B/10B-encoded system for these new, whitened frames, were examined, it was found that they follow a similar pattern to that for the pseudo-random frames. Notably these results display patterned errors (*hot-spotting*) in the scrambled data, but following de-scrambling no measurable correlation is present between payload contents and the octets in error. The whitener has therefore successfully improved the uniformity of the data errors with respect to the actual transmitted data.

5.3.3 Conclusions

The whitening scheme removes the non-uniformity of the data errors due both to concentrations of transmitted data at certain octet values and the *hot-spotting*. The overall loss level is unchanged (as this is due to the coding scheme and physical devices used). While not specifically useful at reducing the level of loss, the use of a scrambler has removed the occurrence of *hot-spotting* within the payload data. While the error-prone octets still exist, encoding with the scrambler removes any biases in the input data. By removing the *hot-spotting*, the data-dependent errors, the underlying uniformity of error has also been restored.

This demonstrates that the addition of a payload whitening scheme can restore the underlying assumption of uniform errors at the physical layer, and therefore it is anticipated that higher-layer functionality will not suffer. Since networks must often continue to work with legacy layers which cannot be changed or redesigned, the ability to work around their characteristics through the use of *shim* layers, such as the scrambler illustrated here, becomes increasingly necessary. However, this scrambler works above the stage of addition of CRC and header, and for these to be protected from hot-spotting a scrambler would have to be implemented below this, but above the channel codec.

5.3.4 Effects of Using a Scrambler

Scramblers do have disadvantages, aside from complexity of implementation.

Systems using short scramblers in particular may be vulnerable to attack from malicious users. One example might be the potential attack on SONET networks carrying unscrambled packet data, where certain data payloads, once scrambled for transmission, could be used to gain control over the SONET line (Section 3.3).

As stated earlier, one reason for requiring a given density of "1"s is a requirement for timing recovery or network synchronization. However, other factors such as automatic-line-build-out (ALBO), equalization, and power usage are affected by the number of "1"s present.

The whitening effect of some other systems, for instance virtual private networks or VPNs (using encryption systems such as IPSec), will also reduce the data-specific effects of the non-uniform error behaviour. This might eliminate the need for an extra scrambler layer in some applications or systems.

5.4 A Comparison With Copper Physical Layer Networks

Given the increasing use of Gigabit Ethernet, and in particular the unshielded twisted-pair (UTP) form, it was decided to compare and contrast the fibre work described above with the behaviour of Ethernet on this non-optical medium when subject to a lower-power regime.

This popular variety of Gigabit Ethernet on copper, 1000BASE-T, uses 4 pairs of a UTP cable, each running at 250Mbps, with simultaneous bi-directional signalling on each pair. It was a later addition to the original Gigabit Ethernet standard, 1000BASE-X (covering both optical links, and short haul copper, see Section 5.4.2). The IEEE 802.3ab standard for 1000BASE-T supports the use of Category-5 or better UTP cables up to 100m in length, and provides an easy upgrade path from earlier UTP-based systems [150], [152, §15.1].

The results suggest that the line errors on this medium are of very different types to those observed on fibre. Whereas in the fibre case, individual bit errors are observed, on UTP the most commonly observed error was entire loss of signal or clocking, such that all octets in a frame after the first octet in error are lost. This section describes the tests which were used to compare error behaviour in copper and fibre links; these also confirmed the error *hot-spotting* observed in the original work on optical links.

5.4.1 A Testbed to Examine Network Link Performance for Partial Failure

The testbed described here was developed to investigate partial failures of links; these would be similar to those noted in the optical case, where errors occur in the link but it stays operational. These data dependent errors might thus not be observed in a deployed link. If a link entirely fails, no data will get through regardless of value, whereas in a partial failure state some data will be transmitted without problems and other data may suffer from a high error probability.

This testbed therefore attempts to detect partial link failure by identifying non-uniform loss in the channel.

It is assumed, as before, that the FCS of the MAC layer in Gigabit Ethernet is sufficiently strong as to detect all errors in the setup. It is also believed that the loss rates will be dominated by the bulk payload contents; by using frames containing 1500 octets of the same value, the error probability of this octet will dominate over the small proportion of header octets present.

The testbed transmitter uses a uniformly distributed random function to select a payload to send, from a set of pre-defined patterns. By randomly selecting each payload content, it is possible to separate out content dependent loss, from other losses due to network behaviour. In this case, all the payload frames are of the same size (a 1500 byte payload) and each payload is filled with a repeated single octet. In the experiments here, it is anticipated that all content-dependent loss will be due to line effects; clearly in different circumstances, this testbed could detect other sources of pattern dependent error may be present (such as a bad router).

In addition the transmitter periodically sends a control message so that the receiver can track how many packets of which payload have been sent out. The receiver counts the number of packets that have been received for each pattern (ignoring control frames) and calculates a probability of loss for each payload in the population. This work is interested in measuring loss due to errors, rather than general network issues which will not be content specific, so the difference in loss between different payloads is examined.

In the copper (UTP) case, an over-long cable was used to induce errors; it is conjectured that the use of excessively long, beyond specification cables in real installations is not uncommon. The use of legacy Category-5 cables which may not all meet the stringent requirements of the 1000BASE-T standard is also likely to cause poor link performance. (The Gigabit Ethernet Alliance has suggested that up to 10% of pre-existing Category-5 cables would not meet two required parameters for 1000BASE-T [152, §15.1.2].) For 1000BASE-ZX on fibre, a variable optical attenuator was used to reduce the receiver power. In both cases, a range of links was used, both within and outside of specification.

5.4.2 Results and Discussion

Figure 5.15 is a histogram of overall error rates versus payload octet value for the fibre case using 1000BASE-ZX; Figure 5.16 gives the results for copper, 1000BASE-T.

Figure 5.15 can be seen to be very similar in form to Figure 5.7, where the probability of error for octets in pseudo-random data is shown.

The UTP case is a clear example of a link which does not exhibit content-dependent errors. Whereas the optical error *hot-spotting* is due to data-dependent channel errors, combining

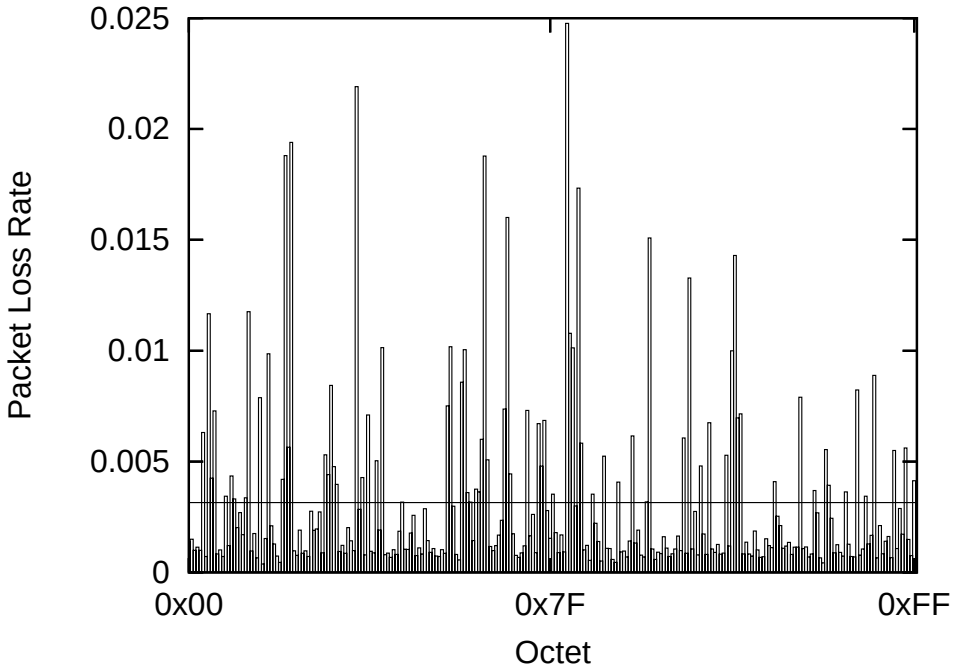


Figure 5.15: Packet loss rate for frames consisting of repeated single octet values, in 1000BASE-ZX

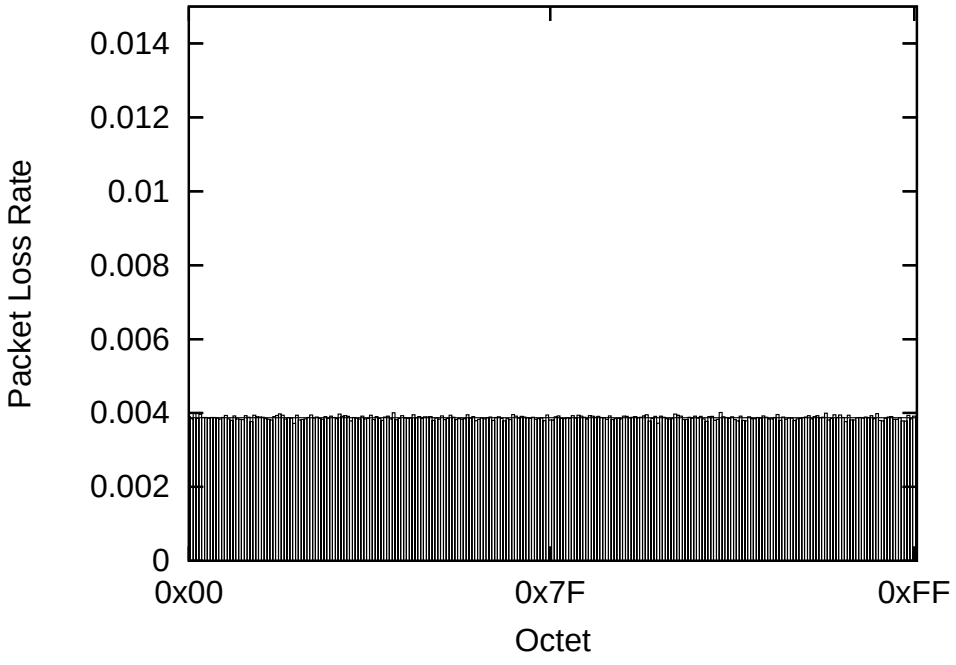


Figure 5.16: Packet loss rate for frames consisting of repeated single octet values, in 1000BASE-T

with non-uniform error amplification at the coding layer, the copper medium provides no variation in error rate due to payload contents. This is not unexpected. The physical layer of 1000BASE-T is very different to that used in the optical systems of 1000BASE-X. Rather than 8B/10B encoding onto a single channel, four-dimensional 8-state Trellis forward error correction (FEC) is applied to each octet; each of the 4 pairs of bits then comprising a byte for transmission is sent on a different copper pair, using 5-level pulse amplitude modulation (PAM) signalling. Each link runs at 125Mbaud, and is fully bidirectional, with symbols being sent in both directions simultaneously. The FEC is used to compensate for a poor signal to noise ratio at the receiver. The data is therefore effectively whitened before transmission, and data-dependent loss is eliminated.

However, in any case, the failure mode appears to be quite different, exhibiting clock synchronisation loss, rather than occasional bit errors within an otherwise correct data stream. Again, referring to the specification shows that this is indeed a likely form of failure, as the clock recovery system for a link such as this where so many mechanisms are required to enable data transmission at all, is necessarily complex. Master/Slave loop timing is used; one end transmits using its local clock, and the other using a clock signal recovered from the incoming data stream [152, §15.1.1], [150]. Whichever end is using a local clock for transmission, it is easy to see how the receive clock timing could go wrong. Clock desynchronisation would commonly appear to the user as a corrupted sequence of one or more octets, which would usually result in a dropped packet. This means that a 1000BASE-T system operating outside the specified receiver power range (such as that for over-length or out-of-specification cables) causes a detectable level of loss, but not the content-specific error patterning observed in optical systems.

A Note About Gigabit Ethernet on Another Copper Medium: 1000BASE-CX

1000BASE-CX (Gigabit Ethernet on short haul shielded co-axial copper cable) uses one pair of wires in each direction at 1.25Gbps, and is more comparable to the fibre case, as they both fall into the 1000BASE-X section of the specification and use 8B/10B coding [27]. This has not been tested; however, it is likely that high frequency components will be most likely to be received in error, as hardware inevitably performs less well at high speeds. The actual errors observed may be different however. The symmetry introduced by the use of differential signalling means that the case of a `1` being misread as a `0` would be no more likely to occur than a `0` read as a `1`.

5.5 Summary of Experimental Observations of Errors in Gigabit Ethernet

In this chapter, a range of error characteristics of Gigabit Ethernet on fibre were documented; the experiments simulated the low receiver power margins which it is postulated will be present in future optical networks.

It was found that channel bit error rate and packet loss were both data dependent and only weakly deterministically related; this effect will be explained in Section 6.4. When uniform data is transmitted over an 8B/10B encoded optical link, longer frames were found to suffer from an even higher probability of damage than their length alone would suggest; the probability of error increases throughout the transmission of such frames. More 10 bit code-groups suffer from multiple channel bit errors than might be anticipated. In addition, frames are more likely to suffer from multiple error events than an assumption of independent error probabilities would suggest. Once these multiple line errors have been amplified by the 8B/10B decoding process, the data link layer error patterns are such that as few as two errored code-groups in a jumbo frame may cause the CRC to fail to detect the error pattern.

The probability of an octet being received in error depends on the characteristics of the code-group used to represent it on the line — code-groups containing strong high frequency components are many times more likely to be received in error. This effect is called *error hot-spotting*. The use of a data-whitener was found to eliminate this, whilst not improving the underlying loss level. Gigabit Ethernet on copper (1000BASE-T) did not exhibit hot-spotting as a different form of line coding is used.

These error characteristics are anticipated to affect any 8B/10B encoded optical link which is operated with limited power budget margins.

Chapter 6

Measuring Errors Through the Network Stack

This chapter considers the effects of the observed error characteristics on real network traffic, and develops a novel mapping from channel BER to packet loss for both uniform and Internet traffic. The first section reflects upon some characteristics of Internet traffic in terms of payload and header octet distributions, and the issues which may arise when 8B/10B coding causes error hot-spotting. A sample trace of such traffic, the day-trace, is used experimentally to determine the relative levels of packet loss and payload damage (which can only be detected by the Ethernet FCS) for a range of reduced optical power margins. This empirical data, together with an understanding of Gigabit Ethernet line coding and previous results for errors in pseudo-random traffic, is then used to develop a mapping connecting packet loss rate, channel bit error rate and data link layer error rate for both Internet and uniform traffic. This mapping considers "average" behaviour; a more striking illustration of the data-dependent and weakly deterministic relationship between bit error rate and packet loss rate is given in the last Section, which returns to the original result of Section 5.1.2 and shows how this may have arisen in the light of the error behaviour in optical Gigabit Ethernet.

In the previous chapter, a range of error behaviours in Gigabit Ethernet on fibre in a state of reduced optical power margin at the receiver was observed. The bit error rate and packet loss rate relationship was found to be only weakly deterministic; specific payloads can have relatively high BERs and low packet loss rates compared to others with lower BERs but higher packet losses. Errors were more likely occur in longer frame sizes, due to increasing error probability through the frame. Error *hot-spots* are octets which are subject to much higher error probabilities than others (due to the code-groups used to represent them on the line).

Here, these error characteristics are discussed in light of real network traffic. The first Section describes some characteristics of Internet traffic, and possible interactions between it and error hot-spots. Section 6.2 details the use of a real network trace in the reduced optical power Gigabit Ethernet testbed in order to establish packet loss rates and causes. It is found that most lost packets are due to errors in the payloads, which are detected by the Ethernet frame check sequence (CRC32). These results contribute to the development of a mapping connecting channel bit error rate and packet loss, for both uniform and Internet traffic, in Section 6.3. This mapping shows that on average the packet loss rate is slightly less than the channel bit error rate; however, this apparently straightforward relationship hides the intermediate sub-layer stages of error hot-spotting and amplification previously observed. Without the smoothing effect of taking the average case, these non-uniformities can lead to much less predictable behaviour. This is illustrated in Section 6.4, which examines potential reasons behind the data-dependent and weakly deterministic relationship between bit error rate and packet loss rate.

6.1 Error *Hot-spotting* and Real Network Traffic

In the experimental work described in Chapter 5, most of the traffic used was pseudo-random, uniform octets in a range of frame sizes. The frame size profile was matched to that of a real network trace, the *day-trace*, which is introduced here. This real network trace is a sample taken from an Ethernet link between a large research institution and the Internet over the course of two working days [162].

6.1.1 Frame Sizes

This Ethernet link carries frames up to the standard MTU of 1500 bytes. The distribution of frame sizes is shown in Figure 6.1. This highlights the standard Internet traffic pattern of the most common frame sizes being acknowledgements 40 bytes in length, ones also carrying a 512 byte payload, and data frames of the MTU size.

In links where bulk data is carried over short distances (SANs, or distributed processing systems, for instance) frames of the MTU length or greater would be expected to dominate. It is known that in many fast Ethernet links jumbo frames are increasingly used, although they are not seen here [138].

Section 5.1.3 showed that longer frames are more susceptible to error, both in terms of the higher probability of one or more errors occurring in the longer transmit time, and the increasing risk of error during the transmission of a long frame. As the protection offered by a CRC decreases for longer frames, the risk of undetected errors also grows.

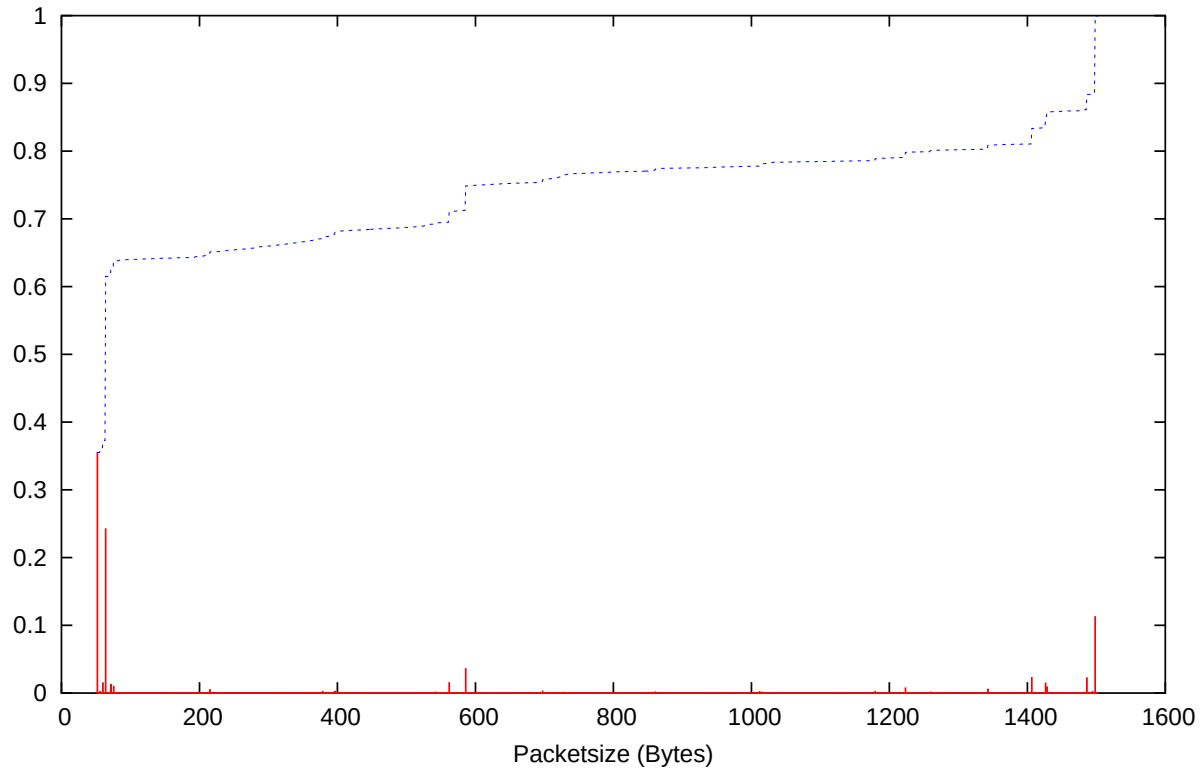


Figure 6.1: Probability distribution of frame sizes in the *day-trace* (cumulative distribution in blue, histogram in red)

6.1.2 Data in Real Network Traffic

The actual data payloads of network traffic do not contain random data; some octets are sent much more frequently than others (Figure 6.2).

It has been shown that error probability in a given octet X_i has a correlation with the "previous" octet, X_{i-1}, as well as the "current" octet, X_i (Section 5.1.4). In Figure 6.3, the correlation between two octets transmitted consecutively in the *day-trace* is shown, with X_{i-1} on the y-axis, and X_i on the x-axis. Dark blue indicates rarely occurring combinations, pale blue and yellow more common ones, and red the most common. The region of ASCII data can be clearly seen, as well as other individual peaks due to common octet pair sequences; a `0` followed by another `0` is the most common.

Effects of Error Non-uniformity on Real Payloads

If one or more of the octets most prone to error occurs frequently in transmitted data, then the frames containing them will suffer from disproportionately high rates of packet loss. In the sample of Internet data, none of the most frequently occurring octets was also an error *hot-spot* from the uniform data experiments. While this is fortunate for the users of this data set, it does not imply that other common network data sets would not have octets in common with

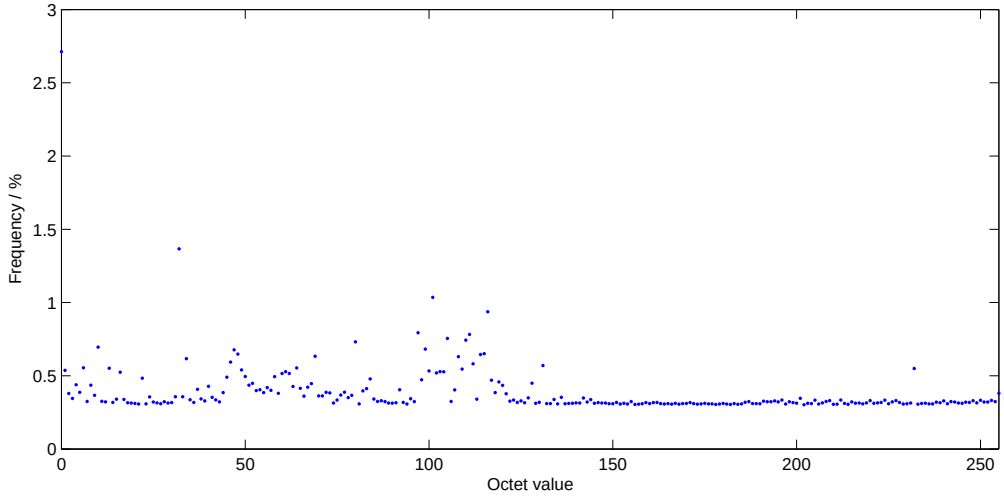


Figure 6.2: Octet occurrence in the *day-trace*

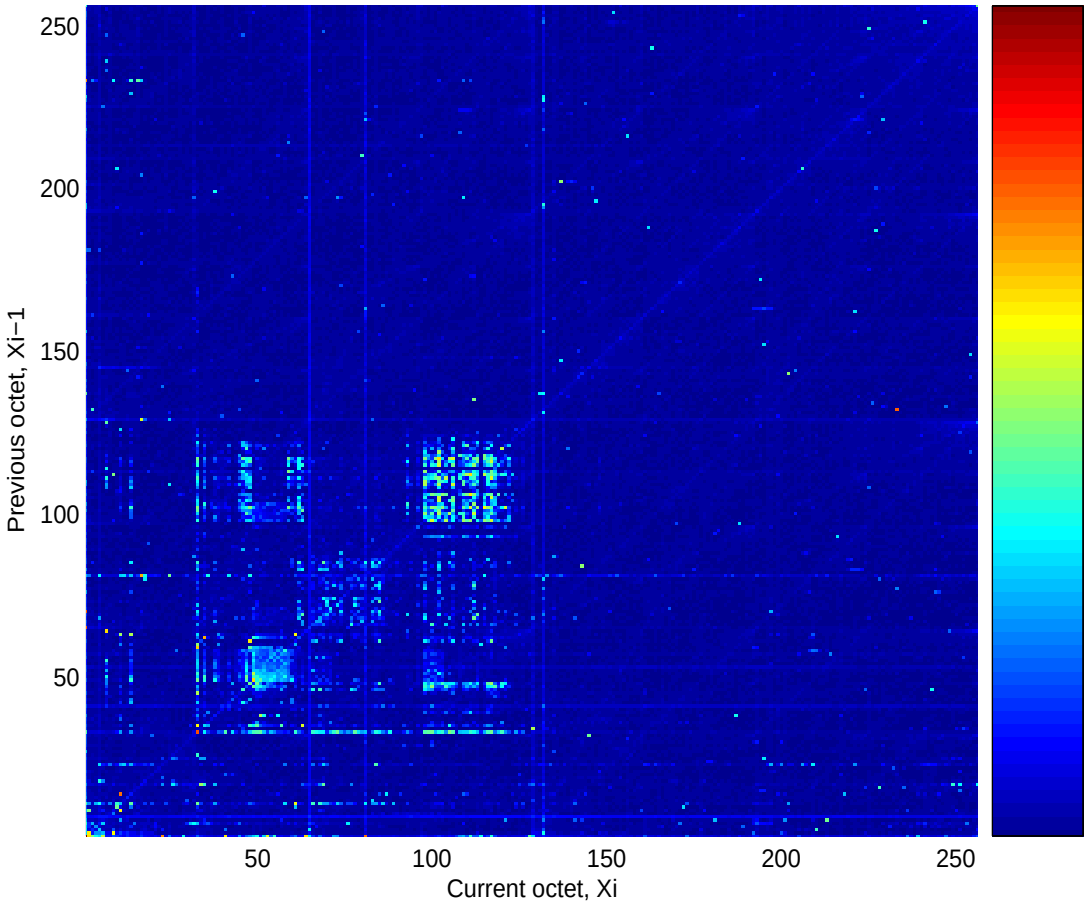


Figure 6.3: Octet occurrence in the *day-trace* in terms of X_i,X_{i-1} correlation, with colour scale from blue (low frequency) to red (high frequency)

the *hot-spots*. It is also possible to compare the hot-spots with the IP payloads of the data set used below in Section 6.1.3. The octet distribution for these is shown in Figure 6.4 and is fairly similar to the *day-trace*. Here, it is found that octet 0x0A occurs 0.65% of the time, the 17th most common octet, and this is also the 6th most common hot-spot. This however is not of great significance. It will contribute, to some extent, to the overall effect of error non-uniformity (along with the other 255 octets, their frequencies in the data set and their error probabilities).

The main point here is that *some* payloads, those for a specific application perhaps, will consist entirely of "lucky", low error probability octets, and that others will contain one or more "unlucky", high risk octets. The contrast between the error probabilities of these individual frames could, in a situation where errors are occurring at moderate frequency, lead to difficult to diagnose differential network performance. This type of varying frame error rate could also be caused by the TCP/IP headers which form part of the Ethernet payload (Section 6.1.3).

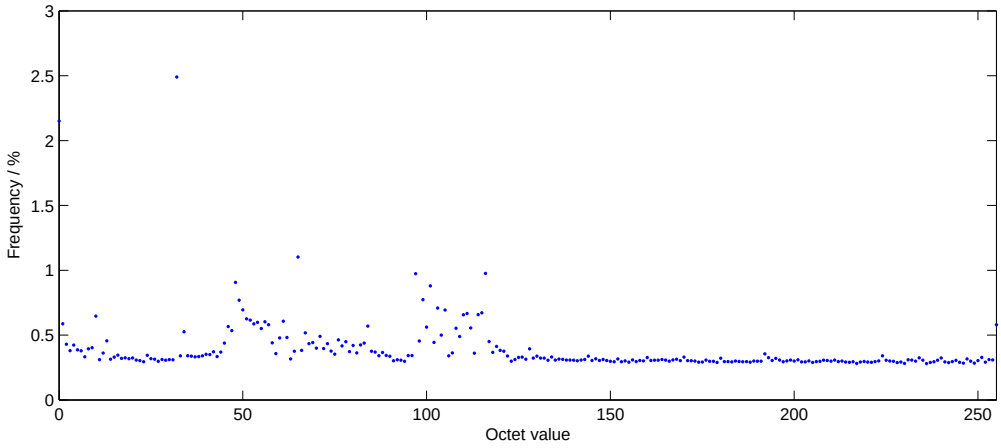


Figure 6.4: Octet occurrence in the large Ethernet traces

A much smaller risk is that the non-uniform data of real traffic could contribute to a higher than expected probability of the CRC not detecting errors, by highlighting hot-spots further. In the previous chapter, it was shown that in a jumbo frame only two octets need to be received in error for an undetectable error pattern to occur. For this to happen the two damaged octets must each give a relevant data link layer error pattern and be in one of the relevant frame positions. If a frequently-sent hot-spot octet is commonly damaged in such a way as to create one of these data link layer octet error patterns, the probability of a CRC-defeating error pattern in the overall frame is increased.

6.1.3 TCP, IP and UDP Headers

This section considers the potential for *hot-spots* to coincide with octets which are commonly found in higher level network header fields. The network traces used for this work were a

range of 24 hour traces taken from two Ethernet access links, one between a large residential university of 10,000 users and the internet, and the other from a research institution of 1,000 users to the internet. The octet profiles for the TCP and UDP headers for these traces are shown in Figures 6.5 and 6.6. For IP traffic, the octet frequencies for all fields except source and destination addresses are shown in Figure 6.7. All frequencies are normalised such that for each case, the sum of the frequencies across all octets is equal to 1.

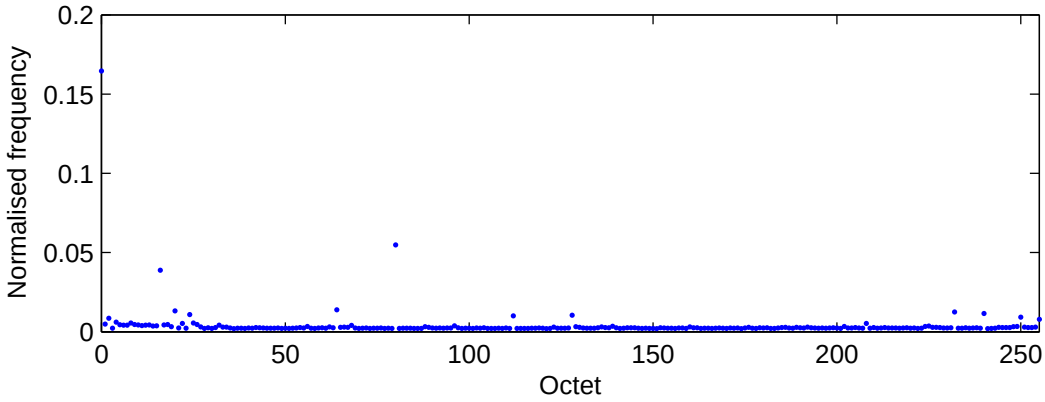


Figure 6.5: Octet occurrence in the TCP headers of real network traces

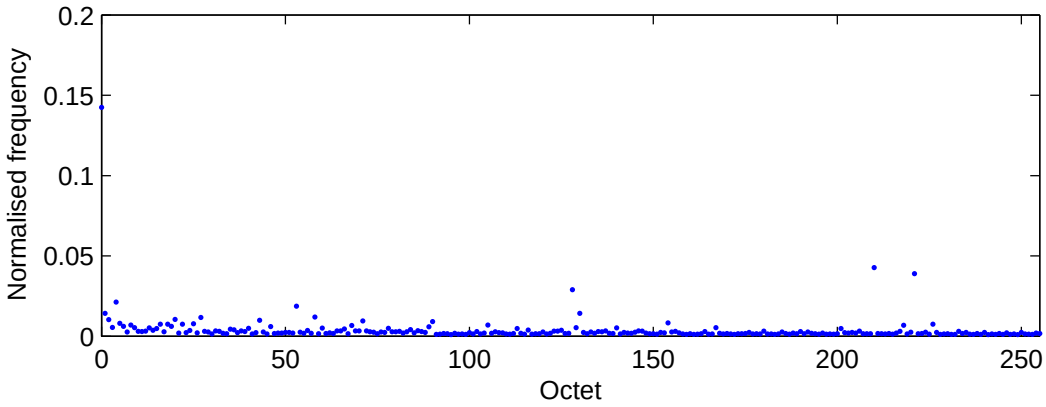


Figure 6.6: Octet occurrence in the UDP headers of real network traces

Clearly some octets and octet sequences occur many times more frequently than others in these headers; these are defined by the most common frame types, lengths, ports and so on. The most frequent octets in the sample of TCP, UDP and IP headers share no commonalities with the set of hot-spot octets in uniform data. This is not to say that some types of frames which do not dominate the overall traces might not have hot-spot octets in their headers. It is also helpful that errors are more likely to occur at the end of a frame (Section 5.1.3), not during the header fields at the start, for long frames at least.

Regardless of other header fields, a user with an IP address which contained a number of high-error-probability octets would be at a disproportionately high risk of having their frames dropped, compared to users without such octets in their addresses.

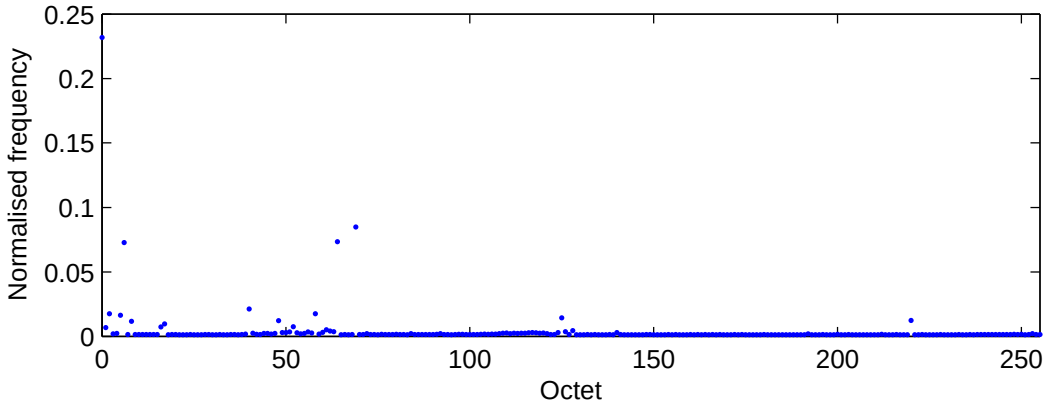


Figure 6.7: Octet occurrence in the IP headers of real network traces, not including address fields

In the cases where errors do occur, the checksum protecting the header fields should detect the damage; even if this happens, though, the packet would be dropped, reducing network performance. The checksum used by TCP, UDP and IP headers is a 16 bit, 1's complement sum (although it has been noted that this is not always used in the case of UDP [139, 140, 141]). This provides detection of error bursts up to 15 bits in length (although not byte reordering), which should be adequate for most cases of link-derived errors in the header fields [163, 164]. At least 2 octets would need to be damaged for this checksum to fail to detect the error, and in the short lengths of header fields (normally 20 octets for each of IP and TCP, 8 for UDP) the probability of multiple errors will be low.

6.1.4 Summary

In the particular network traffic traces examined here, no particular cases have been found where traffic non-uniformity would exacerbate the error hot-spotting. However, the most problematic outcome of hot-spots would be groups of frames belonging to specific users, applications or networks, which would suffer proportionately higher loss rates than others. This differential performance problem could be hard to detect and diagnose in many systems. If a user's network connection is operating towards the limit of acceptable performance under a service level agreement, say, the presence of hot-spot octets in their IP address might be sufficient to reduce performance below this level. The non-uniformity of error hot-spotting goes against the common assumption of random errors made by higher network layers.

The occurrence of error *hot-spots* has ramifications above the MAC layer, and for systems other than Gigabit Ethernet (indeed, any system where an optical physical layer where errors may occur is combined with a block code). Zorzi&Rao [165] discusses the performance of TCP over wireless channels, and observes that TCP was designed for an environment where packet loss is primarily due to congestion, rather than frame corruption. [165] also notes that

"the sensitivity of the performance at the upper layers of a packet switched protocol stack to higher order channel error characteristics and physical layer design is not fully understood." Stone *et al.* [149] discusses the impact non-uniform errors have on the checksum of TCP. These results indicate that not only increased packet loss may result; Stone and Partridge note certain "unlucky" data would rarely have errors detected [139]. A further example of related work is Jain [131], which illustrates how errors impacted the coding layer of FDDI and had required specific error detection/recovery in the data link layer.

6.2 Transmission Experiments With Real Network Traffic

The potential effects of error hot-spotting on Internet traffic have been examined. Now, how the non-uniform errors observed in Chapter 5 affect the packet loss for real traffic is investigated. This section determines what proportion of frames is received with the payload octet damage examined previously, as opposed to other types of frame error.

A set of experiments, measuring packet loss as well as examining octet damage for payload error cases, is undertaken using Gigabit Ethernet on fibre in a state of reduced power margin to induce errors.

It has been shown that data damage may occur on attenuated links, leading in almost all cases to a link layer CRC failure. However, so far this work has only considered cases where the data payload was damaged in such a way that the frame was still received (using the modified equipment to view frames which would fail the CRC). It would be interesting to know the ratio between transmitted packets, those received with a damaged payload and those "lost" entirely. These dropped packets could be due to damage to the framing code-groups, or invalid code-groups being received (as observed in Chapter 4). In a switched system there is also the risk of packet loss due to incorrect routing, but is outside the scope of this study.

6.2.1 Experimental Method

A set of tests were performed using the existing Gigabit Ethernet test platform, Figure 5.1, together with the suite of software tools for analysis of errors in received frames (Section 5.1.1). This time, however, the *day-trace* was used as the test traffic; by virtue of this data being taken from real networks, this offers both a frame size profile and payload profile which is typical of Internet traffic. A range of measurement points was used in the system, as shown in Figure 6.8.

Traffic is sent from the *tcpfirediff* software over a 10/100BASE-T Ethernet link (`eth0`) to a Cisco switch (Switch 1). This forwards all test traffic from its Ethernet interface to the appropriate Gigabit Ethernet port, where a 1000BASE-ZX transceiver is in place. The

optical link from this back to the host computer contains a variable optical attenuator; the host PC contains a SysKonnect 1000BASE-LX Gigabit Ethernet NIC (`eth2`). Frames where the Ethernet FCS does not match the data, which would normally cause a frame drop, are preserved by the modified NIC driver. All other configuration is unaltered for the NIC and switch. Data is recorded for a range of receiver power levels, each being sampled for at least 40 loops through the complete *day-trace*, for a minimum data sample of 65 million packets. For many tests, including those at low error rates, the system was run for much longer than this.

Measurements can be taken at a sufficient number of points in the system to determine the quantities of switch traffic (frames sent periodically by the switch, including ARP requests) and other application traffic at various points in the system. This additional traffic can therefore be eliminated from the analysis. The redundancy of measurement implied by the multiple test points validates these results. The number of test traffic frames which are lost at the various stages can be deduced, and the received frames can be compared to their transmitted counterparts to check for errors. The software tools also allow analysis of any errors in the Ethernet payloads.

6.2.2 Types of Frame Damage

To fully understand the measurements, all the possible outcomes when a packet has been transmitted must be considered.

It is assumed that all packets successfully leave the network layer at the transmitter in the correct format, and that errors only affect the frames after this point, i.e., in software or hardware at the data link layer or below.

1. **Lost Frame** The frame never reaches the receiver. It may fail to leave the transmitter on the network medium, or be be discarded or misrouted at a switching stage or other intermediate system. Alternatively it may suffer so much degradation due to noise or interference that when it does reach the receiver, the receiver fails to detect that any packet has arrived.

2. **Malformed Frame at the Coding Layer** A frame without the correct starting and ending delimiters will be discarded at the coding layer. (If the starting delimiter /S/ is not received, the rest of the frame will also not be received and the effect is as if the frame was lost *en route*.) If the ending delimiters do not appear in the correct sequence, and then IDLE symbols are observed, the receiver should indicate an error and the frame will not be received.

3. **Incorrectly Encoded Frame** This type of frame contains one or more of the following:

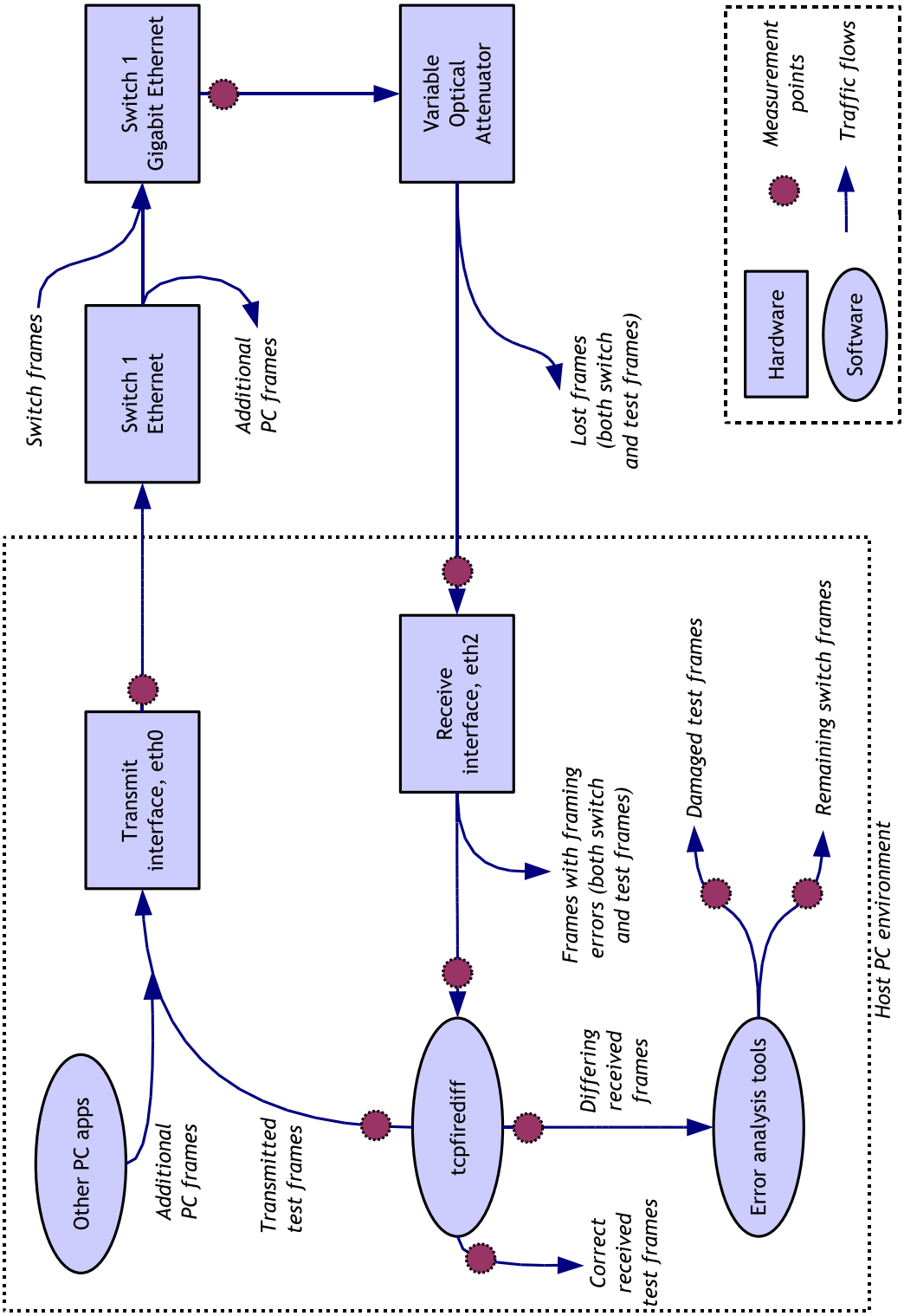


Figure 6.8: Diagram of hardware and software used for experiments with real network traffic

control code-groups where data is expected, invalid code-groups anywhere, incorrect disparity sequence code-groups anywhere. (The latter error form does not occur if *relaxed* decoding, Section 4.1.3, is used.) These frames are discarded by the coding layer.

4. **Malformed Frame at the MAC Layer** If the decoded octets do not form a correct MAC Frame (as per the criteria in Section 4.2.3) the frame will be discarded.

5. **Frame Fails FCS check at the MAC Layer** If the FCS for the frame is not correct, the frame will be discarded. This may be caused by errors in the data block, errors in the FCS octets themselves, or both.

6. **Frame Contains Entirely Undetected Errors** If the error pattern in the data block and FCS is such that the CRC cannot detect it, the frame will be passed up the network stack despite containing errors.

7. **Frame Received Correctly** In which case the frame, identical to that transmitted, is passed up the network stack, ultimately to the relevant application.

This work distinguishes between outcomes 4 and 5, as the test equipment allows the observation of received frames which fail the FCS check. Outcome 6, an undetected error, is extremely unlikely in most cases (except those where long frames such as jumbo frames are transmitted through an error-prone channel). As the test setup compares the entire payload of each received frame with the transmitted version it is possible to detect this type of frame, but none are observed in these experiments. It is included in this list for completeness.

6.2.3 Link Measurements

The primary test software, *tcpfirediff*, permits the specification of how many frames of the traffic trace to transmit, and reports back how many frames were received. A *diff* is recorded whenever a frame is received which does not match the most recently transmitted one. A *timeout* is counted when no frame matching the transmitted one was received during a given time after transmission. This means that either the frame was lost along the way to this point, or that it was damaged. (In the test system some switch frames had not been filtered out by this point, which added to the received, diff and timeout totals; these were removed by the error analysis tools.)

At this point it is interesting to reflect on the link performance measures used in practice when specific network measurements are not being taken. For the "average" user working on a Windows PC, say, the network interface status available to the user is a simple count of packets transmitted and received. In this case, the above outcomes collapse down to two: either a frame is lost (outcomes 1 to 5 inclusive) or it arrives successfully (6 and 7) for use by the application in

question. The user does not care what actually happened to his missing frames, as if they make up a small proportion of his traffic the data they carried will in most cases be retransmitted automatically by an intermediate network layer (TCP). If many frames are lost this user has only limited alternatives to attempt to fix his connection, such as looking for a physical failure (an unplugged cable). A more sophisticated user may use a tool such as *ifconfig* to assess the performance of a network interface at the data link layer. As well as counts of transmitted and received frames/bytes, for an Ethernet interface this tool also presents counts of `errors, dropped, overruns` and `frame` for the receiver and `errors, dropped, overruns, carrier` and `collisions` for the transmitter. Whether or not these values are useful (beyond any non-zero value being indicative of a problem) depends on the implementation of the network interface software and hardware. Some of these values are determined by the kernel, driver or hardware and may be inconsistent between implementations (see Section 7.2).

At the transmit interface of the host PC, `eth0`, *ifconfig* can be used to count outgoing frames; *ifconfig* is also used at the receive end, but only in terms of the count of total frames received. Although *ifconfig* provides a range of other counters, these are of limited use. None of these tests revealed any `dropped` or `overrun` frames. The former would be the result of a routing decision at the current host, which is not applicable to this configuration, so this is as expected. The latter count, of `overrun` frames, implies frames longer than expected; this could be the result of an incorrect frame length field, a buffer overflow or a loss of physical layer carrier. In the test link, a frame length error would require the code-groups representing the MAC frame length to be damaged in such a way that the received code-groups were data code-groups and which represented a valid frame length value; this is unlikely. The number of `frame` errors is similar to the separate measure of payload damaged test frames, being on average 2% greater than this. This value probably includes frames where the MAC framing and/or FCS are incorrect; however it is of limited use as some switch frames will be included by it as well as the test traffic frames. The same applies to the count of `errors`. This analysis does not include these values. As noted in Section 7.2, all these measures may not be reliable in any case due to hardware and software implementation dependencies.

From all the information available, the total number of *lost* packets (outcomes 1 to 5 inclusive) can be determined. The total number of test frames transmitted (encompassing all possible outcomes) is also known. It is possible to identify frames which would fail the FCS (outcome 5), and these *damaged* frames are distinguished from the other types of dropped frame (which cannot be distinguished between) which are collectively referred to as *missing* frames. Using the software which analyses the octets within a frame, the actual octet errors for frames with damaged Ethernet payloads can be examined. It is also possible to identify cases where the data payload itself is undamaged, but where one or more of the FCS octets have been received in error.

Another performance metric pertinent to this work is the bit error rate at the physical

layer. Those testing physical transmissions usually assess this by experimentally transmitting a pseudo-random bit sequence, counting the number of bits in error, and thus obtaining the probability of a bit error. Alternative measurement techniques include estimation from eye diagrams (see Section 3.1.2). BER is not discussed until Section 6.3.

6.2.4 Results

The results of the tests described in Section 6.2.1 are shown in Table 6.1, to two significant figures. *Missing* frames include all frames which are lost on the line, or rejected at the receiver due to coding errors or MAC framing errors (except for FCS failures). *Damaged* frames are those which would be detected by an FCS failure in an unmodified system; they include one or more errors in the data payload and/or FCS fields which have produced other valid data code-groups. The distinction between these two types is somewhat artificial, as in practice both lead to a lost frame; however it will help with the development of the packet loss / BER mapping in the next section. In addition, the separation is perhaps interesting in light of the analysis of Chapter 4, where the theoretical probabilities of different types of frame error were calculated for the case where a frame suffered a single bit channel error. The theoretical and experimental ratios between frames with payload damage and frames with other forms of error can be compared. In the theoretical case, the outcome of a *missing* frame encompasses the probabilities P_f, P_u and P_m; for an MTU-sized frame, using the relaxed decoding scheme implemented in the test system, the overall probability of a lost frame is the sum of these, 5.5×10^{-9}. The probability of a *damaged* frame, P_d, is 9.8×10^{-9} — approximately twice as likely as the other error types. In contrast, the experimental results (admittedly for a range of frame sizes) show that payload errors are orders of magnitude more likely than other errors, regardless of attenuation.

Receiver Power/dBm	Missing frames/%	Damaged frames/%	Correct frames/%
-23.3	0.00013000	0.43000000	99.57000000
-23.2	0.00004100	0.20000000	99.80000000
-23.0	0.00002800	0.16000000	99.84000000
-22.8	0.00000500	0.04000000	99.96000000
-22.6	0.00000230	0.01400000	99.98600000
-22.4	0.00000067	0.00370000	99.99630000
-22.2	0.00000000	0.00028000	99.99972000
-22.0	0.00000000	0.00000680	99.99999320
-21.8	0.00000000	0.00000130	99.99999870
-21.6	0.00000000	0.00000030	99.99999970

Table 6.1: Results for *day-trace* test traffic at a range of attenuations, as percentages of the total number of test frames transmitted

It is worth noting that although Table 6.1 gives actual receiver power levels, they are not in themselves particularly relevant. The specific values relate only to the sensitivity of the specific receiver used in these tests. An optical attenuator was used to limit the receiver power, not because of an interest in attenuation, but in order to simulate the performance of a link which is operating with a low power budget margin.

6.2.5 Discussion

When the octet-level payload damage was examined, it was found to be similar to that for pseudo-random traffic (Chapter 5). Allowing for the non-uniform octet distribution of the *day-trace*, the error probability distribution per-octet was approximately the same as for the random data. Similar numbers of octets in error per frame were also found, and this distribution does not vary much with receiver power. (There was a smaller proportion of frames with more than 6 octet errors in the network traffic case than the random one, as the experimental setup had been improved to filter out the remaining switch frames. In Chapter 5, frames with more than 6 errors were not included in the analysis as an inspection of these cases showed that the overwhelming majority of these were not the result of line errors.)

The proportion of damaged frames where the FCS must be relied upon to detect the error is higher than might be expected, regardless of overall packet error rate. One would hope that a link subject to a low power budget margin would suffer from enough lost packets that a user (or automatic checking system) might notice the problem. In most cases this will be the case, but in systems operating with very small power budget margins and carrying long frames, there is a small but finite risk that the link layer CRC32 may not detect some errors.

6.3 Connecting Packet Loss Rate to Bit Error Rate

In this Section the relationship between packet loss and channel bit error rate is investigated, using the experimental results for packet loss from the test using Internet traffic over an attenuated Gigabit Ethernet link and an understanding of error behaviour in uniform data payloads. Firstly, this section describes the motivations behind this work, and some of the decisions made in the development of the mapping. Some assumptions and limitations of the work are then noted in Section 6.3.2. The path of an error from the code-group at the physical layer to octet at the Ethernet MAC layer is traced out, then broadened to consider entire frames and both uniform and Internet traffic in Section 6.3.2. Finally, a complete mapping for average performance cases between channel BER, packet loss, and data link layer bit error rate is presented in Section 6.3.3.

6.3.1 Motivations

One of the original objectives of this work was to detail the relationship between channel bit error rate and packet loss. While the results of Section 5.1.2 show that these two metrics are not directly related, owing to significant data-dependencies, it is possible to provide a relationship mapping between the two for two particular data cases which are of interest. The first is uniform, pseudo-random data, which is similar in character to the pseudo-random bit sequences used by the photonics community to test optical systems. The second traffic type is real Internet data, represented by the *day-trace* data.

In the previous section, packet loss rates were measured for a range of receiver power levels for real Internet data, and it is reasonable to assume that random data with the same frame size profile will suffer similar levels of loss. (The *day-trace* does not contain a large quantity, overall, of any of the most error-prone hot-spot octet values.)

There are various ways in which the mapping could be developed. One would be to perform a worst-case analysis. This would require knowledge of what, in this context, "worse" meant: a poor packet loss for a given BER, a great many data bit errors for few channel errors, channel errors which might cause errors which are undetectable by the CRC, or an illustration of some aspect of non-uniformity. The latter could involve considering the relative probabilities of error for two IP frames containing the same payloads, one with addresses made up of high error probability octets, and the other of low error risk ones. The most obvious, and simplest, "worst case" is that where each single channel bit error causes a frame to be dropped. This gives a 1:1 ratio of BER to packet loss, regardless of receiver power levels and coding scheme, but is not true in all cases as shown previously: often frames suffer from multiple errors. This could mean that fewer frames are lost than the BER would suggest. It would be interesting to know what BER must be specified for the channel if a network is to offer a guaranteed level of service in terms of packet loss. This is addressed here, for the case of Gigabit Ethernet or any similar system using 8B/10B line coding and similar data link framing.

So, this section will derive a metric which will apply well to the "average" situation, not an extreme case. (These will exist, but are not as applicable as a more general example. However, it is hoped that this work will have highlighted potential problems and raised awareness of these more extreme non-uniformities.) Wherever a range of possible values occurs, the expected value is therefore selected.

6.3.2 Developing the Mapping

The objective of this work is to connect three things together. Firstly, the channel BER and the packet loss rate (covering all lost frames, including ones with a data payload error which fail

the FCS). Since cases where the CRC might not detect the error are of interest, and because the number of data link payload bits in error is not necessarily what would be expected, the rate of error of the data link layer bits is also considered.

Mappings are produced for the various attenuation values for which data about packet loss is available, and for both the *day-trace* and uniform data samples.

Assumptions

Firstly, the types of error which are not going to directly contribute to the mapping should be noted. Any errors in the *Idle* sequences between frames are entirely ignored here (unless they lead to a damaged frame, which is possible if synchronisation is lost; see Section 5.1.8). Errors within any part of a frame, including framing code-groups, which give rise to invalid or misplaced control code-groups are not overtly examined. However, experimental measurements are available of the number of *missing* frames, which are damaged in some way other than payload errors, and this will include all frames with code-group errors; these measurements will be taken into account in the mapping. Any errors in framing code-groups will apply with equal probability to both uniform and Internet data. There may be some difference in the probabilities of data code-groups being received as invalid ones for the two traffic types, but as these lost frames make up a small proportion of the overall loss rate this differential is not considered.

The errored frames are divided into two types, as in the experimental work in the previous Section. One is missing frames, where the exact error which occurred is unknown. The other, damaged frames, only contain payload error(s) which have produced valid data code-group(s). A great deal is known about the latter type of error, but rather less about the former (which conveniently has been shown to occur less often). It would be interesting to know how many channel bits were received in error to cause these frames to be lost. Several approaches could be taken here. One would be to assume, as framing bits make up such a small proportion of the overall frame, that errors within them will occur relatively infrequently. Therefore the only errors worth considering here would be data code-groups which were received as invalid or control groups. However, this probability is only fully understood from the theoretical model which assumed a single bit line error. Given the number of multiple bit channel errors within payloads (observed in Section 5.1.6) this is possibly not a good assumption. There are two alternative methods. One would be to assume that a single bit channel error causes each of these dropped frames. This has the advantage of simplicity and also being a reasonable "worst case" value. The alternative is to use the mean number of channel bit errors observed in the data payload errors; this is 1.29 bit errors per code-group, and allows for a number of cases where multiple bit errors might occur. The mapping uses this value (the overall results are not significantly changed if 1 is substituted).

As the mapping is partially derived from experimental data, it is assumed that the *relaxed* decoding scheme is used when considering the transmitted and received code-groups, as this decoding method is the one implemented in the test equipment used to obtain these results (Section 4.1.3). It should be noted that the number given in the previous paragraph for mean channel bit errors was derived using a *minimum* channel bit error assumption. This was that, of the up to 4 possible channel error patterns which could have occurred (between two possible transmitted code-groups and two possible received ones), the one with the fewest errors actually happened. This assumption is used again here, in the derivation of the mapping at the physical layer.

Frames of differing lengths are not distinguished in this analysis. Both the uniform and Internet traces have the same frame size profiles; it is known that error probability varies with frame length, but the focus here is on the average error rates, so this variation is not considered here.

In the previous Chapter, it was shown that both the previous octet X_{i-1} and the value of the current octet X_i affect the error probability of X_i. In the development of the mapping described here, this additional correlation is ignored for clarity. The effect of the octet sequencing is extremely small (except in cases of severe *hot-spotting*) and becomes averaged out during the derivation of the mapping. The results with X_{i-1} included are indistinguishable from those given here.

Because experimental measurements of packet loss rate and payload damage rate are to be used, there is no need to consider variables such as the number of channel bits per frame. The *rate* part of the packet loss rate, bit error rate and so forth is already integrated into the data; bit error rate is the rate over the number of transmitted bits, and packet error rate over the total number of transmitted packets in an interval.

Mapping Errors from Channel to Ethernet MAC Layer

First, the metrics required are expressed in terms of the information available. It is possible to see how the two types of error, causing missing and damaged frames, can be linked.

The overall packet loss rate for each receiver power value is given by:

$$\text{Packet Loss Rate} = P(\text{Damaged frame}) + P(\text{Missing frame}) \tag{6.1}$$

where P(Damaged farme) and P(Missing frame) are the respective probabilities from the experiment with Internet traffic (Section 6.2).

The channel bit error rate is:

$$\begin{aligned}\text{Channel BER} = &\{P(\text{Damaged frame}) \times E[\text{Channel payload bits in error}]\} \\ &+ \{P(\text{Missing frame}) \times E[\text{Other damage bits in error}]\}\end{aligned} \tag{6.2}$$

for each receiver power. The notation uses $E[X]$ for the expected value of X, and $P(X)$ for the probability of X. The expected values are per-frame; this corresponds to the per-frame loss rates measured experimentally. It was previously decided to use 1.18 for the expected number of channel bits in error per dropped frame (E[Other damage bits in error]).

The data link layer error rate, which measures the rate at which data link layer bits which are received in error, does not need to consider the missing frame case. Again, the expected value below is per-frame:

$$\text{Data Link Error Rate} = \{P(\text{Damaged frame}) \times E[\text{Data link layer payload bits in error}]\} \tag{6.3}$$

Expressions will now be derived for the expected values in these equations, by following the error path up from the physical layer, considering the errors per-code-group/per-octet to begin with.

At the Physical Layer It is known that channel errors are data-dependent and the probability of error depends on the code-group used to represent the octet; code-groups containing strong high frequency components are more susceptible to error. A two-dimensional, 256×256 matrix P is defined which represents the error probabilities in terms of each possible transmitted and received octet pair. P is derived from the complete set of octet probability information for uniform data accumulated in previous experiments (Chapter 5). This matrix is scaled such that

$$\sum_{TX} \sum_{RX} P = 1 \tag{6.4}$$

where TX is the set of 256 transmitted octet values and RX is the set of 256 received octet values.

It was shown in the previous Chapter that multiple bit errors can occur in a single damaged code-group at the physical layer (Section 5.1.6). As before, for each non-matching transmitted and received octet pair the number of channel bit errors which occurred in the 10 bit code-group can be deduced. Once more, the pair of code-groups which give the minimum number of channel bits in error is selected. This count of errored channel bits for each possible transmitted and received octet is stored in a 256×256 matrix C.

At the Data Link Layer By the data link layer, the data has been decoded from its 10 bit format on the line back to octets. Where errors occurred, between 1 and 8 bits per octet will be in error. The number of data link bit errors for each transmitted and received octet pair is recorded in a 256×256 matrix B.

At the octet level, it should be noted that the Internet traffic consists of some octets more than others. The results for Internet data are therefore weighted using this octet distribution, and the frequencies of each octet in the *day-trace* are recorded in a 256 element vector D. This vector is scaled such that the sum of its elements is 256.

All this information is now put together to derive the intermediate mapping terms.

Mapping for Uniform Data The expected number of channel bits in error per damaged octet can be expressed as:

$$E[\text{Channel payload bits in error}] = E[C \cdot P] = \sum_{TX} \sum_{RX} (C \cdot P) = 1.2877 \tag{6.5}$$

where $C \cdot P$ is the Hadamard, or *entrywise*, matrix product.

To make this into a per-frame term, one simply multiplies by the expected number of damaged octets per frame, which is 1.18 (it was observed in Section 5.1.7 that 88% of damaged frames have one octet in error, nearly 10% have two, and 2% had more than that).

The expected number of data link layer bits in error per octet received in error is:

$$E[\text{Data link layer payload bits in error}] = E[B \cdot P] = \sum_{TX} \sum_{RX} (B \cdot P) = 2.4390 \tag{6.6}$$

and can similarly be scaled up to a per-frame value.

Mapping for Internet Data To convert the above expressions for Internet data requires an additional step, to weight the matrix products $C \cdot P$ and $B \cdot P$ by the octet distribution vector D.

$$E[\text{Channel payload bits in error}] = E[C \cdot P \cdot D] = \sum_{TX} (C \cdot P \cdot D) = 1.2792 \tag{6.7}$$

$$E[\text{Data link layer payload bits in error}] = E[B \cdot P \cdot D] = \sum_{TX} (B \cdot P \cdot D) = 2.4433 \tag{6.8}$$

These values can be scaled to apply per-frame using the 1.18 factor in the same way as for the uniform data.

All the information required to determine the full mapping is now in place.

6.3.3 Results

The mapping between channel BER, packet loss rate and data link layer BER for uniform data is given in Table 6.2; the equivalent for Internet data is shown in Table 6.3. The receiver powers at which the link measurements were made are not repeated here, as the absolute values themselves are not relevant to this work which focuses on the BER and packet loss relationships. The power levels are merely illustrative of the effects of various reduced power budget margins, which was simulated using an optical attenuator.

Channel BER	Total Packet Loss Rate	Data link BER
6.60×10^{-03}	4.35×10^{-03}	1.25×10^{-02}
3.10×10^{-03}	2.04×10^{-03}	5.87×10^{-03}
2.45×10^{-03}	1.61×10^{-03}	4.64×10^{-03}
6.00×10^{-04}	3.95×10^{-04}	1.14×10^{-03}
2.09×10^{-04}	1.38×10^{-04}	3.96×10^{-04}
5.58×10^{-05}	3.67×10^{-05}	1.06×10^{-04}
4.28×10^{-06}	2.82×10^{-06}	8.11×10^{-06}
1.03×10^{-07}	6.80×10^{-08}	1.96×10^{-07}
2.05×10^{-08}	1.35×10^{-08}	3.89×10^{-08}
4.56×10^{-09}	3.00×10^{-09}	8.63×10^{-09}

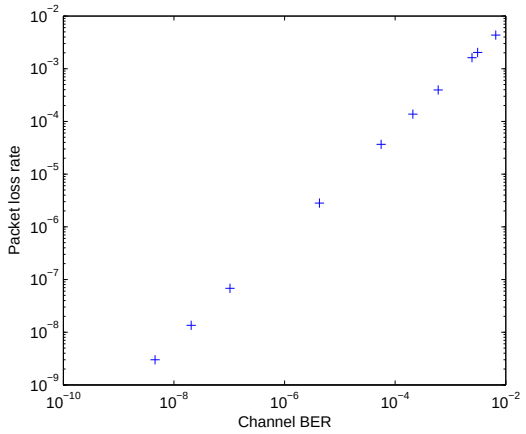
Table 6.2: Mapping between channel BER, packet loss rate and data link error rate, for uniform data

Channel BER	Total Packet Loss Rate	Data link BER
6.56×10^{-03}	4.35×10^{-03}	1.25×10^{-02}
3.08×10^{-03}	2.04×10^{-03}	5.88×10^{-03}
2.43×10^{-03}	1.61×10^{-03}	4.65×10^{-03}
5.96×10^{-04}	3.95×10^{-04}	1.14×10^{-03}
2.08×10^{-04}	1.38×10^{-04}	3.97×10^{-04}
5.54×10^{-05}	3.67×10^{-05}	1.06×10^{-04}
4.25×10^{-06}	2.82×10^{-06}	8.12×10^{-06}
1.03×10^{-07}	6.80×10^{-08}	1.96×10^{-07}
2.04×10^{-08}	1.35×10^{-08}	3.89×10^{-08}
4.53×10^{-09}	3.00×10^{-09}	8.65×10^{-09}

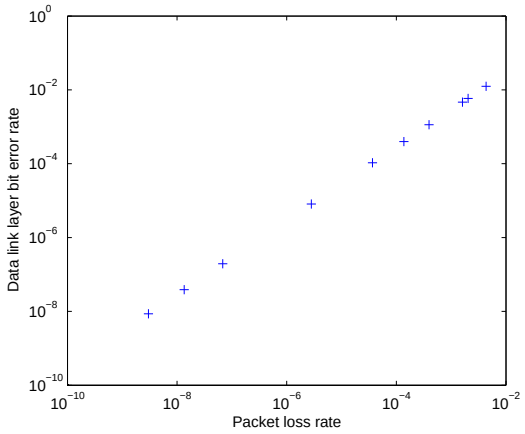
Table 6.3: Mapping between channel BER, packet loss rate and data link error rate, for the Internet data sample

These results are also displayed in Figure 6.9, for the uniform data case, and Figure 6.10 for the Internet data sample. These Figures relate each pair of metrics in turn.

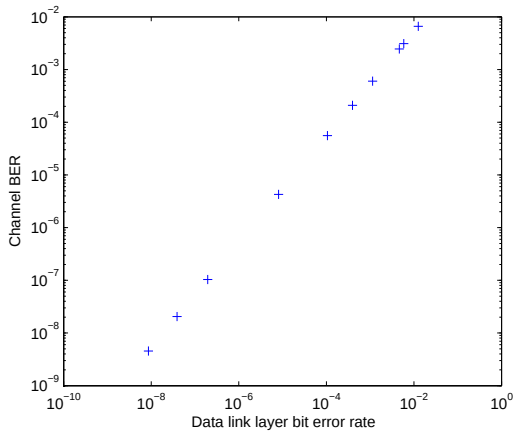
The three causes of error non-uniformity are disguised by these overall results. The data-dependent channel error probability distribution, the number of channel errors actually occurring, and the resulting data layer error rate after "amplification" by the decoding scheme each bring octet-dependent distortions to the error rate, but in this mapping the distortions to some extent cancel each other out. The same applies to the overall octet distribution of the Internet data, so that it gives similar results to the pseudo-random data.



(a) Packet loss rate versus channel BER



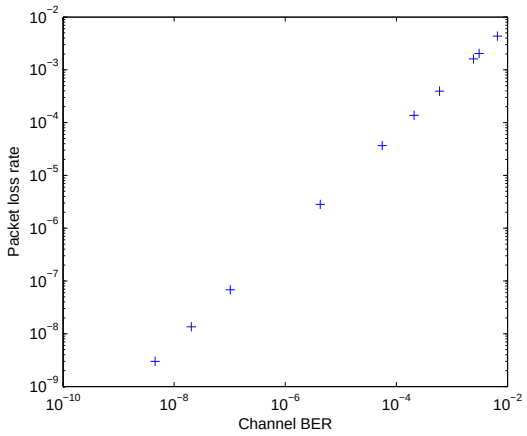
(b) Data link layer BER versus packet loss rate



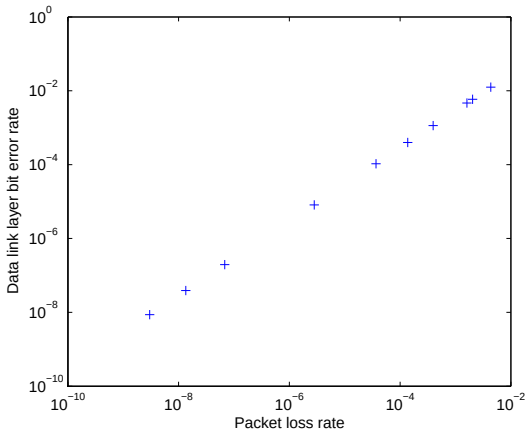
(c) Channel BER versus data link layer BER

Figure 6.9: Comparative mappings between channel BER, packet loss rate and data link error rate, for uniform data

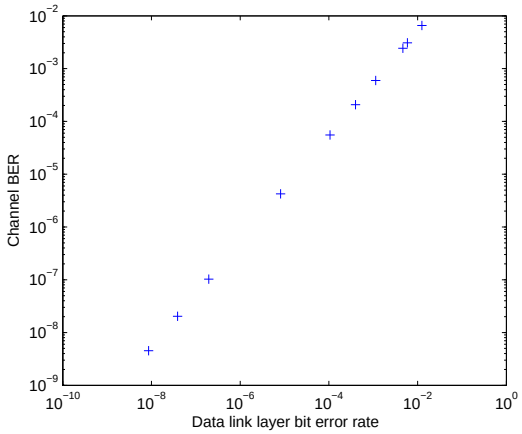
The averaging process removes some information, particularly the extreme cases where individual octets or octet sequences distort the error rate. A more extreme example of this is given in Section 6.4, which investigates the cause of the the data-dependent relationship between BER and packet loss rate which was found experimentally in Section 5.1.2.



(a) Packet loss rate versus channel BER



(b) Data link layer BER versus packet loss rate



(c) Channel BER versus data link layer BER

Figure 6.10: Comparative mappings between channel BER, packet loss rate and data link error rate, for the Internet data sample

6.3.4 Conclusion

A mapping has been derived relating channel BER, packet loss rate and data layer payload bit error rate for the two cases of uniform data and Internet data. This mapping illustrates the average characteristics of an 8B/10B block coded system carrying Ethernet-style frames.

6.4 Explaining Relative Bit Error Rates and Packet Loss Rates for Different Payloads

In the previous section it was shown that although various non-uniformities affect the error path from the physical layer channel to the Ethernet MAC layer, when considered across bulk data most of these are averaged out. Here, a more extreme specific case is examined where the relationship between packet loss and channel BER is both data-dependent and not what would be expected. The case of frames containing repeated instances of single octet values is considered, BER and packet loss are compared for different octets, with the aim of establishing their relative sizes, not their absolute values. Expressions are dervied for the relative value of the expected number of channel bit errors for this type of frame and the relative packet loss rate. Experimental data about octet errors in uniform traffic is then examined to identify an example case of two octet values, one giving a higher BER but a lower packet loss rate than the other. Finally, the implications of this result are discussed.

A previously described experiment (Section 5.1.2) attempted to relate bit error rate to packet loss. In an ideal case, one would do this using the same end-to-end setup. However, this is extremely difficult to do experimentally, unless one can obtain specialised and expensive test equipment which allows both metrics to be assessed for real network data. Instead two different setups were used to measure BER and packet loss. The main result was that, depending on data payload, some frames would suffer from a higher BER than others but a simultaneously lower packet loss, and vice versa. With insight into the systems involved, this section explains this result.

A great deal of data about octet errors in attenuated Gigabit Ethernet links carrying uniform data has been amassed (Chapter 5), so this can now be used in the explanation. Firstly, the channel bit error rate is estimated.

6.4.1 An Expression for Relative Channel Bit Error Rates

Examining the large sample of errored octets in uniform data, it is possible to determine the expected number of channel errors for each transmitted octet value, whenever it is received

in error. This is done by deducing the pair of transmitted and received code-groups for that octet (considering all possible incorrect received octets which may be the result of decoding a damaged code-group) and selecting the pair which offers the fewest channel bit errors. A range of expected values is found, from the obvious minimum, of a single bit channel error, for the octet 0x23, to 3.4 channel bits in error (the octet 0xDF). These are illustrated in Figure 6.11.

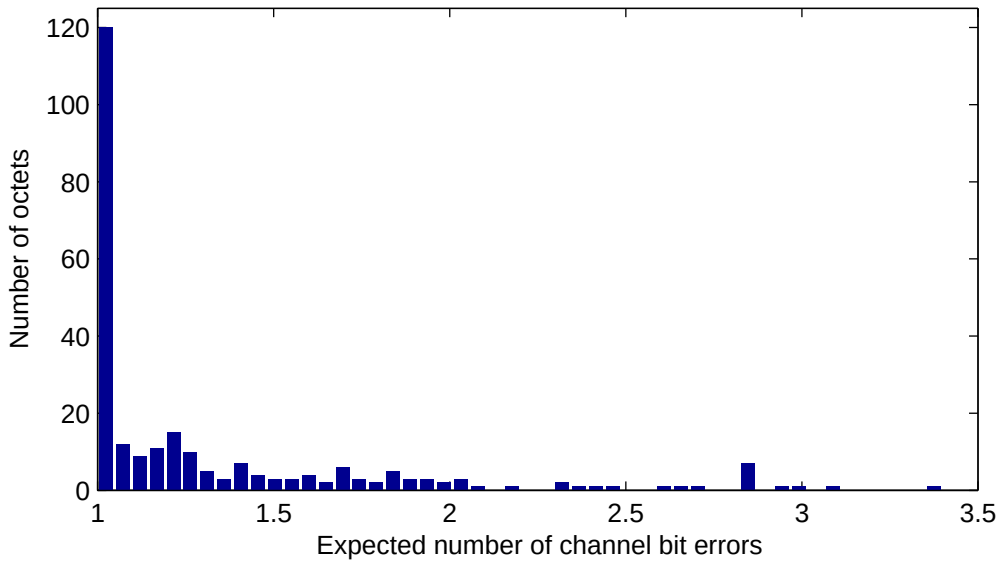


Figure 6.11: Histogram of expected channel bit errors for each possible transmitted octet value

The bit error rate is an average, or expected value, itself. For a frame containing 1500 octets all of the same value, the BER is given by:

$$\text{BER} = E[\text{channel bit errors}] \times 1500 \times P(\text{error in that octet}) \tag{6.9}$$

Here, cases of missing frames with other error types are ignored; it is assumed that since the framing will be identical for any octet payload that the probability of this can be neglected in the relative BER expression.

We do not have an absolute value for the probability of any given octet being received in error. However, this work is interested in comparing the BERs for two octets, so a relative measure should be adequate. The proportion of errored octets in the large sample which correspond to a transmitted octet of the value we are interested in should suffice. The relative BER can therefore be estimated:

$$\text{Relative BER} = E[\text{channel bit errors}] \times 1500 \times \left(\frac{\text{times the octet is received in error}}{\text{total number of errored octets}} \right) \tag{6.10}$$

6.4.2 An Expression for Relative Packet Loss Rates

An expression is also required for the packet loss rate; this can also be relative, as this work aims to compare values for frames containing different octet values. Now, for any 1500 octet frame there is some proportion of all damaged frames which will be lost due to damage in the framing sequences at MAC or coding level; this proportion will be the same independent of data payload. There is also a loss rate due to data code-groups being received as invalid or control code-groups, and this proportion is quite likely to vary depending on the payload (some octets can be damaged by a single bit line error into an invalid code, whereas others will instead be received as another data code, for example). However, for the moment it is assumed that both of these loss rates are fixed proportions and the same regardless of payload. They can therefore be ignored in the analysis, as only a relative measure of packet loss is required. Now, a packet will be lost if the FCS detects an error; this will happen in almost all cases (the small probability of the CRC failing to detect an error pattern is neglected here, as this requires multiple octets to be received in error). Any number of octet errors in the frame will then be identified by the FCS. It is known that single octet errors in a frame are the most common, but that in nearly 10% of damaged frames two octets will be damaged, and 2% will have even more errors (Section 5.1.7). A factor could be determined to work out how many errors would occur within a single frame (this of course affects the bit error rate) but again since only an approximate relative measure is required, these multiple-error cases can be neglected and it is assumed that a single octet is damaged in each errored frame.

So, if again a 1500 octet frame filled with identical octets is assumed, the relative packet loss rate is (ignoring packets dropped due to reasons other than FCS error):

$$\text{Relative packet loss rate} = 1500 \times \left(\frac{\text{number of times the octet is received in error}}{\text{total number of errored octets}} \right) \quad (6.11)$$

Simplifying further, and removing terms common to all octet cases, two expressions are obtained:

$$\text{Relative BER} = E[\text{channel bit errors}] \times \text{number of times the octet is received in error} \quad (6.12)$$

$$\text{Relative packet loss rate} = \text{number of times the octet is received in error} \quad (6.13)$$

6.4.3 Identifying an Exemplar Case

The sample of pseudo-random data frames with errors is searched for pairs of octets where these two values are sharply contrasting. To verify the original result, a case should be identified where:

octet A is very likely to be received in error (causing a high packet loss rate) but only suffers one bit of line error each time this happens (making for a relatively low BER), and

octet B is not so likely to be received in error (a low packet loss rate) but which suffers multiple channel error bits frequently, generating a high BER.

As expected, cases are found where this is so. Take for example the octets 0x43 (relatively high packet loss rate, low relative BER) and 0xDF (comparatively low packet loss rate, high relative BER). These are shown in Table 6.4; remember that the terms for packet loss and bit error rate are relative and that the absolute values are not meaningful.

Octet	E[channel bit errors]	Relative packet loss rate	Relative BER
0x43	1.0004	65566	65592
0xDF	3.3998	30179	101603

Table 6.4: Relative bit error rate and packet loss rate for frames with single-octet payloads

A simple interpretation of this result is that the channel error rate variation is due to different error types observed for different data. If a data type is prone to suffering a synchronisation type of error (Section 5.1.8), rather than "random" bit damage, this might cause a large number of bits to be damaged but still only one frame to be affected. This would give a relatively high number of expected channel bits in error, but a lower packet loss. Other data where commonly only one channel bit is affected by error, will give a dropped packet for each bit error, and thus a relatively high packet loss rate.

6.4.4 Discussion

Some variables have of course been ignored in this analysis. No consideration was made of whether one octet might be more likely to produce a control code-group when damaged on the line rather than a data one (the number of channel bits damaged is unknown unless a data code-group is generated). The CRC32 values for the two frames which would be appended to the data have also been ignored (it is reasonable to assume identical octets representing address and other header fields, but the CRCs would be different). If one frame generated a CRC octet which was very prone to error, this could skew the packet loss rate and alter the observed bit error rate.

Transceiver properties are also a potential cause of the data-dependent discrepancies between bit error rate and packet error rate. Different equipment was used for BER and packet loss assessments in Section 5.1.2, and these variations could lead to slightly differing actual results

for BER and packet loss relationships. However, this section has shown that even for a single link setup, diverging values for BER and packet loss are possible.

However, although some factors have not been considered, it has been shown that bit error rate and packet error rate are data dependent, and that a relatively high value for one does not necessarily indicate a high value for the other.

Chapter 7

Error Non-uniformities and Layer Abstraction

This chapter summarises the work described in this dissertation, and discusses its implications in terms of the use of layer abstraction and the design of new optical networks. The main results about error behaviour in Gigabit Ethernet on fibre, the relationship between packet loss and bit error rate, and the non-uniform nature of the errors (which may not agree with the assumptions of higher network layers) are detailed. The error channel through the network stack, and the ways in which errors are measured and handled by different network layers, are then outlined. Finally the implications of this work for the designs of the next generation of optical networks are considered.

Section 7.1 reviews the work covered in this dissertation. Further reflections on the implications this work has for layer abstraction in networks are documented in Section 7.2. Section 7.3 details the potential error behaviour due to complex optical systems which will be used in future optical packet switched networks, and considers the issues highlighted by this work, which should be taken into account by network designers. Finally, Section 7.4 concludes this work.

7.1 Summary of Work

In Chapter 1, the field of optical networking for computing applications was introduced, and the motivations for this work in terms of the increased potential for errors in future optical systems were described. Chapter 2 reviewed some technologies which are likely to be present in the next generation of optical networks and interconnects, and described a prototype packet switched optical local area network as an example. Some relevant theory on line coding, error detection

and issues arising from the use of layer abstraction was discussed in Chapter 3. Chapter 4 was a theoretical analysis of the effects of a single bit channel error in Gigabit Ethernet on fibre (1000BASE-X), a system using 8B/10B line coding and frame format similar to the SWIFT network of Chapter 2. In Chapter 5 experimental work was used to investigate the behaviour of a real Gigabit Ethernet system in a state of reduced link power margin, paying particular attention to errors in the octets of uniform data payloads. Chapter 6 extended this work to consider packet loss in real network traffic, and developed a novel mapping connecting channel bit error rate and packet loss.

This Section reviews the theory, results and conclusions of this dissertation.

7.1.1 Context

It is postulated that future optical systems where the power budget margin is reduced are likely to suffer errors more frequently than has been the case to date.

One reason for this is the use of complex optical devices and systems; an example of this is the prototype optically switched packet network, "SWIFT", of Chapter 2, or links containing long runs of fibre or many splitters. Other reasons include the desire for cheap networking hardware, which will often be operated at or near the specification limits, and the difficulty of building reliable equipment for serial transmission speeds beyond 10Gbps. In addition, the increase in speed alone forces greater transmission powers to be used if the existing error rate is to be maintained — this is not always possible if optical components limit the maximum power. (The SWIFT prototype system uses multiple parallel wavelengths at lower bit-rates, which reduces this latter problem.)

7.1.2 Principal Results and Implications

A number of interesting error characteristics were revealed by the theoretical and experimental investigations of 8B/10B encoded optical packet data.

- It was identified that channel bit error rate and packet loss were both data-dependent, and only weakly deterministically related. Some data payloads suffer from high BERs and low packet loss rates, compared to others with lower BERs and yet higher packet losses.
- It was determined that implementations of Gigabit Ethernet do not always behave exactly as specified in the standard. In particular, this was seen in terms of the method used for 8B/10B decoding, and when the link was maintained despite the receiver optical power level being low enough to induce frequent errors.

- The use of layering abstraction, whilst bringing advantages to network design, may allow the error channel to be obscured such that system behaviour in the presence of channel errors is not necessarily what might be assumed at higher levels of the network stack.

- A variety of non-uniform error characteristics at the physical and data link layers of Gigabit Ethernet and similar systems were observed, which in a network operating towards the limit of the optical power budget margin could cause notable performance differences targeting specific users, applications and networks.

- A mapping between channel BER and Ethernet layer packet loss, representing average system behaviour, was developed. This provides information for both uniform and Internet traffic over a range of loss levels.

Error Characteristics in Gigabit Ethernet

Here an outline of the individual error characteristics, from the physical layer channel up to the top of the data link layer, which contribute to these overall effects, is presented.

Physical Channel Long packets, such as the 1500 byte Ethernet MTU and jumbo frames, are more likely to suffer errors, due partly to their length and partly to an increasing probability of errors throughout the frame transmission time. This also increases the risk of multiple errors per frame; more cases of two or more errors per frame were observed than would be expected from an assumption of independent error probabilities. Channel errors damaging a single code-group consisted of 2 or more bits 13% of the time. This type of error may be due to partial synchronisation loss and not merely bit damage caused by noise. The errors themselves tend to focus on code-groups with a strong high frequency component (alternating bits, `0101`), leading to error *hot-spotting*: some octets having unusually high error probabilities. The application of a scrambler as a form of data-whitener, while not improving the underlying loss-rate, eliminated the *hot-spotting* effect (restoring the uniformity of error which may be assumed by higher network layers). It was found that Gigabit Ethernet on copper UTP, 1000BASE-T, exhibited a very different failure mode, with all octets equally susceptible to damage, and frame loss occurring due to loss of data through much of a frame, rather than individual errored octets. This was due to the different line coding scheme used.

Physical Coding Sublayer While the 8B/10B block coding scheme may offer some data protection, it can also cause error *amplification*, where a single bit error on the line is multiplied to between 1 and 4 data link layer bit errors. The number of data link layer error bits arising from a given channel error pattern is thus data-dependent. The *relaxed* 8B/10B decoder observed in real implementations of Gigabit Ethernet is more likely to

cause an errored payload octet to be decoded as a valid data octet than the specified decoding scheme in the standard.

Data Link Layer Investigations into packet loss for real network traffic showed that the majority of frames are dropped due to errors in the data payload which are detected by the Ethernet MAC layer frame check sequence (a 32 bit CRC). The error amplification worsens the risk of a combination of errors occurring which the CRC32 cannot detect (at least 4 data layer bit errors, spread out by more than 32 bits, are required for this). This, together with the risk of multiple errors per frame, particularly for longer frames (which are both more prone to error and offered reduced protection by the CRC), causes the probability of the CRC not detecting an error pattern to be higher than might be expected. As few as two damaged code-groups within a jumbo Ethernet frame could cause the CRC32 to fail to detect the errors.

7.1.3 Conclusions

Examining the 8B/10B line code, used in Gigabit Ethernet and elsewhere, this work has documented the form and cause of failures that occur in a regime of limited optical power margin. These induce at best poor performance, and at worst, undetected errors that may focus upon specific networks, applications and users, causing their frames to suffer greater loss rates than the norm. This content specific effect will be difficult to diagnose because it occurs without a total failure of the network. The pattern-related failure is made more serious by the non-uniform nature of application data. Although this study concentrates upon the specific case of 8B/10B systems, the error characteristics described will be similar for any system using a block code. The overall non-uniformity is caused by the interaction of various effects occurring at different points within the physical and data link layers. The combination of these distorts the frame loss rate relative to both the frame content and the channel error rate. Section 6.1.2 and the references therein indicate that error uniformity has been assumed by higher network layers in the past.

The IEEE 802.3z specification defines a robust network; and in installations obeying the specification, engineers will not see the issues documented here. It is considered that the future of optical networks will implicitly alter the environment as systems may become more susceptible to errors, as they are operated with low optical power budget margins. The non-uniformities reported here will affect the behaviour of future physical and data link layers, which may not be the same as those now deployed. These errors may not manifest themselves in the ways which are assumed by those working at the packet layer through to the application layer.

This work therefore points to a need for network designers to understand the flow of errors

through the network stack. In the following Section, issues relating to layer abstraction and error handling are reviewed.

7.2 Layer Abstraction and Error Behaviour

In Section 3.3, a number of cases where problems arose when layers interacted in unexpected ways were noted. Those working at the upper layers of the stack can make poor assumptions (such as expecting uniform, random error behaviour) which ignore the nature of the lower layers. The design and implementation of those lower layers is adequate — the functional requirements of IEEE 802, for example, do not require these layers to offer uniform error behaviour — but in combination may give rise to performance which the upper layers simply may not anticipate. Error models do form a part of each layer's defined interfaces, but this does not necessarily mean that error behaviour through the network stack is integrated and clear.

Different communities working within networking each have their own methods of assessing errors and network performance. This Section reflects upon these issues as they apply to various sections of the stack.

7.2.1 At the Physical Layer Channel

At the optical level, channel bit error rate is measured for transmitted pseudo-random bit sequences (PRBSs), using bit error rate test equipment (BERTs). Some potential problems with BER measurements were noted in Section 3.1.2. In general, though, the BERT has a number of advantages: it is synchronised end to end, sends a custom data sequence repeatedly and automatically assesses the BER. It is possible to replicate some features of a given line coding scheme in the test data; commonly, the maximum runlength of the PRBS is selected to match that of the coding scheme. This allows the characteristics of a coding scheme on a given physical channel to be assessed, but does not necessarily incorporate other implementation details which could affect performance. (For example, Woodward *et al.* [166] observes the differences between the spectra of a multiplexed 8B/10B data stream and a PRBS signal, and notes that the use of repeated Idle codes in Gigabit Ethernet, for example, would distort this further.)

However, at best the BER gives only an average performance measurement. For some systems this is undoubtedly adequate, but for others pattern-dependent errors may severely affect performance at higher network layers. One example of unusual pattern dependent failure was given in Section 5.1.5, where a certain data sequence could not be transmitted over a specific section of an FDDI ring, as well as the error *hot-spotting* described here.

Although errors can be introduced into a system at other stages, this dissertation is concerned with the errors which occur at the physical layer. In the case of optical links, it is possible that some installations susceptible to error could be identified by a simple measurement of receiver optical power level when the link is set up. However, the usefulness of this depends on system variability over time, whether a representative end to end link can be measured, and the skill of the operator. Some pattern-dependent error causes, such as the FDDI case mentioned above, would not be detected by this method.

7.2.2 In the Line Coding Scheme

When selecting a line coding scheme, consideration is sometimes made of the input BER / output BER ratio, or fault tolerance, of potential encodings. However, this is often a minor concern in comparison to issues of channel compatibility (e.g., ease of clock recovery) and implementation complexity. Some considerations which should be made in light of this work when choosing a coding scheme are discussed in Section 7.3.2.

When analysing the performance of such coding schemes, understandable simplifications, such as the assumption of a single bit line error, are made. This was the case for the theoretical analysis of 8B/10B coding in Chapter 4; however, experimentally it was found that channel errors were more complex, and requires more sophisticated error models. The form of channel errors in a real system and the performance of the line coding scheme are invariably closely connected, and so any thorough systems analysis must consider them both in conjunction.

7.2.3 Implementations of Physical and Data Link Layers

In the implementation of a network interface card, where the physical and data link layers are the pertinent ones, various error indications will be given in hardware and firmware. In the specific example of Gigabit Ethernet on fibre, error methods implemented at one layer are not seen at another. At the physical layer, the standard specifies a `signal_detect` flag; if this is set low, then code-groups should not be read in. In the experiments of Chapters 5 and 6 it was found that the optical signal could be attenuated in order to induce errors and that frames were still received. This is just one case which illustrates that information about errors is easily lost within the network stack, either inadvertently or through design choice. In addition, actual implementations of systems may not fully adhere to the specification, leading to unanticipated behaviour (an example being the *relaxed* 8B/10B decoding observed in Section 4.1.3). If this type of variation from the standard occurs in a common device or chipset, the variation may become widespread without the NIC developers being aware of it.

Reasonably detailed error information may be available to the software driver for a NIC.

The driver author must then decide what to do with this; the level and usefulness of the messages in logs and/or signalled may vary widely across hardware and platforms. Clearly it is to be hoped that most error indications which are available to the developer will be used appropriately; however, the nature of the error channel through the network stack does not always permit all the information about errors to be passed up to the next layer. A colleague of the author was surprised to find different information displayed by the `ethereal` tool on PC and Mac platforms when testing a poorly performing network link. On the Mac, ethereal reported a number of errored frames which were tagged to say they had failed the Ethernet CRC; in contrast, on the PC, far fewer damaged frames were displayed and no mention was made of the CRC. The varied error handling of the Ethernet interfaces, which superficially appeared similar, was unexpected. Another data link layer tool is `ifconfig`, and although this offers a range of error counters, their value (beyond providing a simple indication of error/no error) is entirely implementation-dependent (Section 6.2.3).

Although functions are carefully specified in the reference model or standard, actual implementations may not have the clearly distinguishable sublayers detailed in these documents. Many layers of software, hardware and firmware will interact to create a working system, each dealing as best it can with the information provided at its interfaces. Since an overall system is likely to be made up of components from different sources, working in ignorance of the decisions made elsewhere, it is possible for a full system to function adequately without necessarily meeting all the detailed requirements of the model. Even if implementations entirely adhere to the specifications provided by a standard, and are correctly installed and operated as designed, the uniformity of error anticipated by higher levels may not be present.

7.2.4 At Higher Network Layers

The type and quality of error information passed between layers, when those layers have fundamentally different viewpoints of the network, is also questionable. For example, at the IP layer, the network is viewed as a hop-by-hop, unreliable "best effort" datagram service. In this context, performance is essentially binary: either a good frame arrives, or it doesn't. In contrast, at the TCP layer a reliable, end-to-end service is offered. A TCP connection is designed to be robust in the face of failures (particularly those caused by network congestion); the connection is full duplex, allowing acknowledgements to be returned and retransmissions of data to be used if necessary. The performance of TCP in response to errors, such as those described in previous chapters, does not form part of this work. The state of a TCP connection can be hard to assess; *heartbeats* may be used to measure the quality of the connection, but at the risk of detecting faults which would not in most circumstances directly affect the connection in question. TCP and IP are bound together in today's network architectures, but the dichotomy between their two paradigms can present problems for performance assessment.

For engineers working at higher levels in the system, it is perfectly understandable (and indeed predicated by the layered design principle) that assumptions are made about lower layers. However, it is desirable that these are somewhat informed by an understanding of the realities of performance at these layers. One simple illustration is the error hot-spotting in Gigabit Ethernet. The non-uniformity of this may surprise those working at higher network layers; they would undoubtedly also be surprised to discover that Gigabit Ethernet on copper UTP (1000BASE-T) does not have this particular characteristic (Section 5.4). The physical media choice also determines the coding scheme.

Altering the application does not necessarily provide a solution to the problems of lower layer non-uniformity. In some circumstances, performance will be poor regardless of higher layer attempts to correct this. For example, adding additional checksums or CRCs will not only add implementation costs, but will not fix all performance problems. As well as the desire for uncorrupted data, applications hope to receive good throughput and latency performance; detecting an error at the transport layer will require a retransmission if TCP is used, which takes up valuable time and bandwidth. Whether or not the addition of CRCs is an acceptable solution is therefore highly application dependent.

The higher layers of the network stack implicity assume, and depend upon, lower layer reliability. As data rates increase, though, the lower layer features added in previous designs are sometimes optimised out so that faster performance can be obtained at acceptable cost. On a wider scale, Internet systems run over a huge range of physical channels, from conventional wireline systems to wide area mobile networks where errors are more common, and whose link layers use different methods to handle them (such as strong forward error correction, and link layer retransmission schemes [167]). Applications may therefore be subject to varying error performance, and assumptions about this should be made with care.

7.2.5 Summary

This work has illustrated how there is scope for problems to develop in the design, development and evolution of the physical components, algorithms, protocols and applications that constitute our networks. These problems are caused by undesirable interactions between network modules. From the physical layer up to the data link layer of 8B/10B encoded optical links, a variety of non-uniform error behaviours have been described, from pattern dependent channel errors to error amplification. Potentially exacerbated by the characteristics of real network traffic, actual protocol implementations and the observed higher than expected probability of multiple line errors, these effects may interact to produce unanticipated performance problems. Issues with layer interactions have been noted in other contexts before (see Section 3.3).

We define *naïve* layering: the use of protocol layers in situations and combinations beyond

the scope of their original specification. Often this layer evolution is combined with assumptions by developers about the behaviour of the layers below and above. Best described by example, naïve layering can result in the construction or specification of a network module without sufficient understanding of the layers with which it must inter-operate. This, together with the inadvertent loss of information between layers, can lead to unexpected errors; particularly in the case of optical networks operating at higher data-rates with increasing complexity. Even at the specified bit error rate of 10^{-12}, a single Gigabit Ethernet link will on average experience an error every 800s.

A conjecture is made that there is a need to build converged systems, with the combinations of physical, data link, and network layers optimised to interact correctly; thorough cross-layer analyses will enable enhanced system reliability to be achieved. In the mean time, what will become increasingly necessary is both an identification of the potential for failure and the need to plan around it. The non-uniformities observed in this work have a subtle effect on the assumptions about error behaviour made by higher layers, leading to unexpected failure modes. These systematic errors are challenging to detect as they are caused by the interference between functions at different layers. Since networks must often continue to work with legacy layers which cannot be changed or redesigned, the ability to work around their characteristics through the use of *shim* layers, such as the scrambler of Section 5.3, becomes increasingly necessary. This work does not propose alterations to the network stack, for example an explicit error communication path from the link layer to the transport layer. The complications of violating the layered architecture in theory and implementation would have to be traded off against the small potential performance gain, which is limited by the ability of any higher layer to use enhanced error information[168].

Those who are designing systems susceptible to errors must select components, sub-systems and protocols to optimise the performance of the relevant application. For high performance networking systems this should include consideration of how the application will respond to errors, as well as overall throughput, latency and so forth. This work has noted that errors are measured in different ways and have different characteristics at each network layer or sub-layer. For a given network, the propagation of errors (whether they take the form of corrupted payloads or dropped frames) through the network stack must be considered in light of the tolerance of the top level application. An awareness of these issues is particularly critical for those developing line coding, error detection and correction and test systems, regardless of what layer of the network stack the individual system is to operate at.

The next section reflects on how these considerations can be applied to the next generation of optical communications systems under development.

7.3 Designing Future Optical Networks

Chapters 1 and 2 illustrated that optical networks are seeing increasing application in computing networks such as LANs and interconnects. The next generation of higher speed systems (some of which are optically switched) is under development, making this a good opportunity to reflect on the implications of this work for these networks.

As well as the effects of reduced optical power margins, errors may be introduced by the use of optical devices for switching; some such devices are reviewed in Section 7.3.1, along with the impact of multi-wavelength systems. Some of the factors which must be considered when selecting an appropriate line coding scheme are then discussed in Section 7.3.2. Section 7.3.3 considers the advisability of protecting against the effect of error *hot-spotting* previously described. Finally, Section 7.3.4 discusses the implications of this work on error detection effectiveness, and whether additional detection/correction may be required.

7.3.1 Errors in Complex Optical Devices and Systems

As illustrated earlier (Chapter 2), future optical networks are likely to contain a range of optical devices to perform switching and amplification. These may contribute different error effects to those observed for a simple link in a regime of low receiver power. A brief outline of anticipated error types and causes in some of these additional systems is given here. In particular, semiconductor optical amplifiers are discussed, as they are used in a range of optical packet switching testbeds (e.g., the Data Vortex [54]) as well as the SWIFT network prototype (Section 2.2). Multiple wavelength systems are also mentioned, as they provide a way of utilising more of the bandwidth offered by fibre, as well as permitting multiple, lower line-rate, channels to be used. Many other optical switching technologies exist,

Mach-Zehnder Devices

The Mach-Zehnder interferometer is popularly used for modulation and other network functions, including switching. The input signal is split between two arms; an applied voltage induces a phase difference between the signals. The two signals are then recombined leading to constructive or destructive interference, and a new output signal. Alternative implementations, such as passive Mach-Zehnder interferometers, use length differences between the arms instead of external modulation, and can be used to convert frequency-modulated signals to intensity modulated ones [169, §24.2]. This technique is largely linear, and it is anticipated that a Mach-Zehnder device will not noticeably affect errors in a network, beyond the reduced power due to insertion loss.

Semiconductor Optical Amplifiers

One very simplified view of semiconductor optical amplifiers (SOAs) could be that they are laser diode structures without mirroring at the ends but with fibre connections instead, so that light is amplified as it travels through if a bias current is applied. As well as the obvious application of straightforward amplification, SOAs may also be used for switching purposes (Chapter 2). SOAs are broadband amplifiers, and a single SOA can amplify light with a typical spectral width of 30–50nm, supporting multi-wavelength operation [107]. These devices can introduce additional error behaviour from that observed for the regime of limited receiver power. One common problem with SOAs is patterning due to carrier depletion. This is a particular problem for high speed networks, with line-rates of 10Gbps; the bit period here is of the same order of magnitude as the recovery time of the SOA. These gain dynamics, coupled with the saturation behaviour of SOAs, can induce pattern-sensitive distortion, and consequent errors. Saturation occurs at high gain levels or high input power levels, and can lead to crosstalk between wavelengths and subsequent errors. These additional error characteristics may contribute to the pattern dependency of channel errors in a system. Also, when multiple wavelengths are used, care should be taken that SOA saturation does not cause errors (see Section 7.3.2).

As noted in Chapter 3, serial transmission beyond 10Gbps is challenging in practice. To attain higher overall link speeds, using a number of parallel transmissions may prove beneficial. This can be done with individual fibres (in the form of ribbon fibre, perhaps) or with multiple wavelengths on a single fibre, depending on application-specific criteria for the implementation, such as switching requirements (Sections 2.1.2 and 2.2.1).

Multiple Wavelength Systems

Multiple wavelength systems will suffer similar effects (depending on the devices and conditions present in the network) per-wavelength, as their single wavelength counterparts. Crosstalk between wavelengths is a possible cause of errors if the system is not well designed. In multichannel systems with extremely high spectral efficiencies, burst errors are more likely, as beating may occur between signals of differing phases [170].

Further effects may arise in parallel wavelength systems, and are likely to depend on the method in which data is distributed across the available wavelengths (see Section 7.3.2).

7.3.2 The Line Coding Scheme

In any communications system, some bits on the line will be more prone to error than others. It may be that a certain pattern is affected by the frequency response of the link, or causes

saturation in a device such as an amplifier, or that bits at the start of frames are susceptible to being received in error if the receiver clock is not well synchronised with the incoming bitstream.

By selecting an appropriate line coding scheme, it may be possible to even out or protect against these errors; this may be take the form of optimising for the best ratio of input BER to output BER, or a more complex analysis in terms of network layer interactions. In addition, a coding scheme should allow as much data as is required to be transmitted across the channel in a suitably timely manner (in terms of both absolute speed and latency). Other factors which may influence the choice of coding scheme include the complexity of implementation (perhaps in terms of the type, availability and cost of electronics hardware), and the range of transmit (and hence receiver) powers which can be used. Some low level characteristics which may be desirable for a coding scheme were reviewed in Section 3.1.

Issues for High Speed Networks

Section 4.1 showed that the 8B/10B block code has a range of features which make it a good choice for use in fast optical links. However, it was not selected for use in 10 Gigabit Ethernet and the reason for this is, perhaps, surprisingly prosaic. One form of 10 Gigabit Ethernet, 10GBASE-LX4, does use 8B/10B on each of four wavelengths. However, the other two forms, 10GBASE-W (for wide area networking) and 10GBASE-R (three optical variants) use 64B/66B coding [36]. This is a scrambler system, not a block code as might be expected from the name; it offers poorer running disparity and transition density properties than 8B/10B, permits a runlength of up to 66 bits, making synchronisation more challenging, and was not as well understood at the time of adoption [171]. However, it has one major advantage over 8B/10B: a much lower overhead. The number of redundant bits was not so much the concern as the line bit-rate required to support 10Gbps at the data link layer. With 8B/10B coding, a line-rate of 12.5Gbps would be needed; at the time when the standard was being discussed, this was pushing at the limits of suitable laser technology (particularly when component cost was also considered). Using 64B/66B, the laser need only operate at 10.3Gbps which was felt to be more manageable. Section 5.2.2 noted the difficulties of operating at such high serial transmission speeds electronically; Jajszczyk [28] points out that currently manufacturable systems are limited to 40Gbps per wavelength by the electronic interfaces, not the optics. This supports the belief that future, faster systems will use multiple parallel channels, each operating at an achievable bit-rate, perhaps in a way similar to the SWIFT network (Chapter 2). In packet switched systems, the need for burst mode clock recovery requires the coding scheme to have strong run length and transition density properties (Section 2.2.5).

Potential Modifications to the 8B/10B Scheme

Other than the overhead, then, 8B/10B is generally a good line coding scheme in many ways. One potential disadvantage of it might be the range of code-group characteristics which give rise to error *hot-spotting*. If the set of data code-groups contained fewer codes with strong high frequency components, which have been observed as more susceptible to error, for example, the overall level of hot-spotting could perhaps be reduced. The code book only uses a small proportion of the potential 1024 10 bit code-groups; clearly this selection is partly to maintain the properties of runlength and so on, but there are many codes currently not in use which could be brought in to a modified scheme. However it should be noted that the problematic (in terms of hot-spotting) code-groups containing alternating bits also have high transition densities which assist with clock recovery. For some applications, perhaps in point-to-point links where clock recovery is not a major problem, the code book could be adjusted to reduce the effect of error hot-spotting. It is even possible that a coding scheme could be designed which measured (either at power-on, or on an on-going basis) channel characteristics and adapted to them, where the desired compromise point between hot-spotting and other coding scheme features is known. Given the availability of comparatively low cost memory parts, and the number of memory stages on a standard network interface card, additional processing like this might be acceptable economically. In these future networks, potentially limited power levels at the receiver might not be well suited to the bit-by-bit detection and decoding used by standard 8B/10B systems. A purpose-optimised decoding scheme, using minimum-distance decoding perhaps, might allow the correction of some errors which would, in the normal case, be received as an invalid code-group causing a whole frame to be dropped. The investigation of these possibilities is left for future work.

Scramblers

This work has not considered the implications for error multiplication of scrambler-based coding schemes. These would not suffer from error hot-spotting, but may have other problems associated with their use. One example is that a malicious user may be able to transmit sequences which break normal operation of the scrambler in some circumstances [146]. Scramblers also add considerable implementation complexity; this increases cost and perhaps power, as additional logic processing may be required. If a scrambling code is to work together with a cyclic redundancy check, it should be selected to ensure that the polynomials associated with both codes do not interact in a problematic way [172]. More bit errors may appear at the output of a scrambling decoder than occurred on the line; this is similar to the effect of error *amplification* described in Chapter 4, although Boudreau *et al.* [172] calls the newly generated errors *false errors*. Errors which occur within a frame may also "spill over" and continue after the end of that frame; for this reason it may be advisable to restart the scrambler for each new

packet.

Multiple Wavelength Coding Schemes

This work has noted that the use of parallel connections for high speed systems may be beneficial, as the bit-rate on each individual serial link will be reduced to a more manageable level. Parallelism in optical links may take the form of wavelength striping (Section 2.1.2), or the use of multiple fibres with one or more wavelengths per fibre. In the case of striped systems, novel coding schemes may be used which take advantage of the additional dimension of wavelength in addition to time.

Asynchronous coding is a way of utilising the parallel wavelengths to improve clock and data recovery performance. Using techniques derived from asynchronous circuit design, a coding scheme can be implemented where a number of wavelengths will undergo a level transition at each clock period. Providing that the individual wavelengths are sufficiently synchronised, a single clock recovery unit can recover a bit clock which will apply to all of the wavelengths, with a reduced preamble compared to a standard PLL for a serial link. Wavelength synchronisation at the receiver can be achieved using a delay line to de-skew the wavelengths; the delay configuration is established during initial transmission of a training pattern [74, 173]. This scheme could potentially also be used in a multiple-fibre system.

Special coding methods may also be used to handle other systematic effects. One example would be compensation for the gain change which occurs in optical amplifiers in response to a change in the number of wavelengths passing through them [67]. The coding scheme in Roberts *et al.* [67] limits the number of wavelength channels which are active (transmitting a `1`) and inactive (transmitting a `0`) at any given time, to maintain a narrow range of levels of total optical power at the amplifier. In these systems, the normal per-wavelength requirements for clock recovery must also be considered in the code design.

Summary: Impact of Line Coding Scheme

This dissertation has shown that, for links which may be subject to errors due to limited receiver power margins or other causes, the selection of a line coding scheme should take into account its failure modes and susceptibility to certain error types. An important part of this is to consider the application performance in the presence of whatever residual errors and/or dropped frames will occur after channel errors have been processed through the network stack. Of course, the normal characteristics of runlength, transition density and so on, which determine the performance of a coding scheme over a given channel, must of course be taken into account as well. The right compromise between all these properties must be sought for each network example.

7.3.3 Eliminating Hot-Spotting

As well as non-uniformities caused by the coding scheme, pattern-dependent line errors were observed.

If the line coding scheme does not include a scrambling element, then the data represented by code-groups which interact with the channel characteristics to create a higher than usual error probability will invariably suffer from proportionately greater probability of error. This work has shown how this could cause certain frames to suffer damage more than others, and hence difficult to detect network performance problems.

If the differential probability of error between these worst case code-groups and others is greater than can be tolerated for a given application, the use of a data-whitener may be worthwhile. It was shown in Chapter 5 that a scrambler used in this way eliminated the *hot-spotting*. Some applications may already have whitening processes somewhere in the network stack; encrypted packets, for instance. However, higher level functions may not protect the MAC address fields which would then still be left vulnerable to hot-spotting.

A low level data whitener could be implemented for applications or links where hot-spotting could cause problems. A scrambler used for this function will of course also bear the disadvantages of scramblers which were observed in Section 7.3.2 (implementation complexity, cost, etc.). As with any encryption system, both communicating hosts must be aware of and able to perform the whitening/dewhitening.

7.3.4 More Powerful Error Detection/Correction

This dissertation has shown that some frames may be at risk of error patterns going undetected by the Ethernet frame check sequence (CRC32). Indeed, no CRC can provide protection against all errors; there is always an error pattern sufficiently complex which could defeat the CRC for a given frame size. However, the error detection mechanism is selected to provide acceptable protection for a specific class of frame in a network, when weighed against the implementation cost and overhead of the mechanism.

In Chapter 5, it was found that frames were more likely to be subject to multiple independent error incidents (more non-consecutive damaged octets) than a simple understanding of the error probability would suggest. Nearly 10% of the damaged frames observed contained 2 octets in error, and 2% had between 3 and 5 errored octets. All these frames were of the Ethernet MTU length or less. Longer frames were observed to be more subject to error, with more errors occurring progressively throughout the frame transmission. It is believed that this is due to a gradual loss of symbol synchronisation, as the length of time since an indication of symbol clock was transmitted grows (Section 5.1.3). It is conjectured that jumbo frames, with

9000 octet payloads (six times as long as the normal MTU), will suffer from comparatively even more errors in a link with a similar line error rate. In Section 5.1.7, the ability of the CRC32 to detect multiple error events in jumbo frames was tested, and it was found that as few as 2 damaged octets within the payload could be sufficient to defeat the CRC.

The data transmitted in a real network may be more prone to some specific errors than others, which may also increase the risk of errors being undetected. However, even in error-prone links, the probability of two code-groups (in one of the the relevant pairs of positions) being errored in such a way as to generate the relevant data link layer error patterns will be extremely low. (Similar error probabilities were calculated in Sections 4.2.6 and 5.1.7.) Nonetheless, it has been shown that various factors contribute to an overall error probability higher than might be expected. The use of longer frames provides improved performance due to reduced overheads. For systems carrying such frames of important data over links susceptible to error, additional error protection may be desirable, although it will clearly add both implementation complexity and cost.

Koopman [138] suggests a stronger CRC polynomial for use in Gigabit Ethernet and faster systems, particularly for jumbo frames; this polynomial provides a Hamming distance of 6 for Ethernet MTU frames, and of 4 for longer lengths up to at least 9000 octet payload frames. This new polynomial could be used for high-speed messages whilst retaining the current, legacy CRC for backwards compatibility with slower messages [138].

New optical systems have been proposed, where, for links where a higher channel error rate is anticipated, forward error correction schemes are used to bring this down to an acceptable data bit error rate [29]. The work of Chapter 5 showing the data-dependent and weakly deterministic relationship between channel bit error rate and packet loss may suggest that an optimisation for acceptable packet loss might be more appropriate. Certainly the error channel in such a system must be well understood, and the choice between low overall packet loss, or "even" packet loss regardless of data, must be carefully made for the given application. If error correcting codes are to be used in conjunction with line codes, it is vital that they are selected to interoperate correctly; Immink [174] discusses the order in which channel and error-correcting coding schemes are applied, in terms of error propagation reduction.

7.4 Overall Conclusions

This dissertation described investigations into the physical and data link layer performance of optical links carrying 8B/10B encoded data in the presence of errors. Both experiments and theoretical analysis were used, and a range of non-uniform error characteristics in real systems were observed. Their potential effects on network traffic and implications for optical network design were discussed.

I assert that future optical networks will be more susceptible to errors in received data, and the response of their sub-systems to these errors should be considered using a cross-layer analysis to ensure that otherwise unanticipated interactions do not adversely affect application performance.

Bibliography

[1] P Molinero-Fernandez, N McKeown, and H Zhang. Is IP going to take over the world (of communications)? In *ACM HotNets-I*, Princeton, NJ, Oct 2002.

[2] Y Xin, G N Rouskas, and H G Perros. On the Physical and Logical Topology Design of Large-scale Optical Networks. *IEEE Journal of Lightwave Technology*, 21(4):904–915, Apr 2003.

[3] M J O'Mahony, D Simeonidou, D K Hunter, and A Tzanakaki. The Application of Optical Packet Switching in Future Communication Networks. *IEEE Communications Magazine*, pages 128–135, Mar 2001.

[4] S Yao, S J Ben Yoo, B Mukherjee, and S Dixit. All-Optical Packet Switching for Metropolitan Area Networks: Opportunities and Challenges. *IEEE Communications Magazine*, pages 142–148, Mar 2001.

[5] N Frigo, K Reichmann, and P Iannone. Whatever Happened to Fibre-to-the-Home? *Optical Fiber Communication Conference and Exhibit (OFC)*, Mar 2003. Paper TuR1.

[6] D M Chiarulli, S P Levitan, R G Melhem, et al. Optoelectronic buses for High Performance Computing. *Proceedings of the IEEE*, 82(11):1701–1710, Nov 1994.

[7] F Jia and B Mukherjee. A High-Capacity, Packet-Switched, Single-Hop Local Lightwave Network. *IEEE GLOBECOM*, pages 1110–1114, Nov 1993.

[8] M Listanti, V Eramo, and R Sabella. Architecural and Technological Issues for Future Optical Internet Networks. *IEEE Communications Magazine*, pages 82–92, Sep 2000.

[9] J Comellas, R Martinez, J Prat, V Sales, et al. Integrated IP/WDM Routing in GMPLS-Based Optical Networks. *IEEE Network*, 17(2):22–27, Mar/Apr 2003.

[10] S Yoo. Optical-Packet Switching and Optical-Label Switching Technologies for the Next Generation Optical Internet. *Optical Fiber Communication Conference and Exhibit (OFC)*, 2:797–798, Mar 2003. Paper FS5.

[11] Y Chen, C Qiao, and X Yu. Optical Burst Switching: a new area in optical networking research. *IEEE Network*, 18(3):16–23, 2004.

[12] L Xu, H G Perros, and G Rouskas. Techniques for Optical Packet Switching and Optical Burst Switching. *IEEE Communications Magazine*, pages 136–142, Jan 2001.

[13] K V Shrikhande, I M White, M S Rogge, et al. Performance Demonstration of a Fast-tunable Transmitter and Burst Mode Packet Receiver for HORNET. *Optical Fiber Conference*, Mar 2001.

[14] J S Turner. Terabit Burst Switching. *Journal of High Speed Networks*, 8(1), 1999.

[15] A Ackaert, D Colle, P Demeester, P Lagasse, and M O'Mahony. IST-OPTIMIST view on Technology Trends in Optical Networking. *European Conference on Optical Communications (ECOC)*, pages 150–153, Sep 2001. Paper Tu F 2.3.

[16] A Jourdan, D Chiaroni, E Dotaro, et al. The Perspective of Optical Packet Switching in IP-Dominant Backbone and Metropolitan Networks. *IEEE Communications Magazine*, pages 136–141, Mar 2001.

[17] D Wonglumsom, I M White, S M Gemelos, et al. HORNET – A Packet-Switched WDM Network: Optical Packet Transmission and Recovery. *IEEE Photonics Technology Letters*, 11(12):1692–1694, Dec 1999.

[18] P Gambini, M Renaud, C Guillemot, F Callegati, I Andonovic, et al. Transparent Optical Packet Switching: Network Architecture and Demonstrators in the KEOPS project. *IEEE Journal on Selected Areas in Communications*, 16(7):1245–1259, Sep 1998.

[19] M Duelk, J Gripp, et al. Fast Packet Routing in 2.5Tbps optical switch fabric with 40Gbps duobinary signals at 0.8b/s/Hz spectral efficiency. *Optical Fibre Communication Conference*, 2003. Post-deadline paper.

[20] A S Tanenbaum. *Computer Networks, Fourth Edition.* Pearon Education International, 2003.

[21] J D Day and H Zimmermann. The OSI Reference Model. *Proceedings of the IEEE*, 71:1334–1340, Dec 1983.

[22] J D Day. The (Un)Revised OSI Reference Model. *Computer Communication Review*, 25:39–55, Oct 1995.

[23] V Cerf and R Kahn. A Protocol for Packet Network Interconnection. *IEEE Transactions on Communications*, COM-22:637–648, May 1974.

[24] R M Metcalfe and D R Boggs. Ethernet: distributed packet switching for local computer networks. *Communications of the ACM*, 19(7):395–404, 1976.

[25] D Cheriton. The Internet Architecture: Its Future and Why It Matters. *SIGCOMM keynote*, 2003.

[26] G Chiruvolu, A Ge, D Elie-Dit-Cosaque, M Ali, and J Rouyer. Issues and Approaches on Extending Ethernet Beyond LANs. *IEEE Communications Magazine*, 80-86 2004.

[27] IEEE. IEEE 802.3z — Gigabit Ethernet, 1998. Standard.

[28] A Jajszczyk. Optical Networks - the Electro-optical Reality. *Optical Switching and Networking*, 1:3–17, Jan 2005.

[29] IEEE. IEEE 802.3ah — Ethernet in the First Mile, 2004. Standard.

[30] P Dyke and J Heath. Gigabit Ethernet Passive Optical Networks. *Networks and Optical Communications*, Long-Haul and Access Networks, Optical Metro and WDM:337–344, Jun 2001.

[31] B Atterbury and M Kelsey. Gigabit Ethernet Passive Optical Networks. *Networks and Optical Communications*, Long-Haul and Access Networks, Optical Metro and WDM:345–354, Jun 2001.

[32] J Wolde, U Bigalk, D Zriny, A Dupas, et al. Optical Ethernet metro access network prototype : implementation and results. *European Conference on Optical Communications (ECOC)*, Sep 2003. Postdeadline paper.

[33] IEEE. IEEE P802.3ap — Backplane Ethernet Task Force, 2005. Pre-Standard.

[34] A Myers, E Ng, and H Zhang. Rethinking the Service Model: Scaling Ethernet to a Million Nodes. *Proceedings of HotNets III*, Nov 2004.

[35] John Leyden (The Register). Ethernet forum plots death of SONET. URL="http://www.theregister.co.uk/2005/04/12/carrier_ethernet".

[36] IEEE. IEEE 802.3ae — 10 Gb/s Ethernet, 2002. Standard.

[37] S J Vaughan-Nichols. Will 10-Gigabit Ethernet have a Bright Future? *IEEE Computer*, pages 22–24, Jun 2002.

[38] R I Killey, P M Watts, V Mickhailov, M Glick, and P Bayvel. Electronic Dispersion Compensation by Signal Predistortion Using Digital Processing and a Dual-Drive Mach-Zehnder Modulator. *IEEE Photonics Technology Letters*, 17(3):714–716, Mar 2005.

[39] P M Watts, V Mikhailov, S Savory, M Glick, P Bayvel, and R I Killey. Electronic signal processing techniques for compensation of chromatic dispersion. *Proceedings of the European Conference on Networks and Optical Communications (NOC)*, Jul 2005.

[40] P Molinero-Fernandez and N McKeown. The performance of circuit switching in the Internet. *OSA Journal of Optical Networking*, 2(4), Mar 2003.

[41] S Yao, B Mukherjee, and S Dixit. Advances in Photonic Packet Switching - An Overview. *IEEE Communications Magazine*, pages 84–94, Feb 2000.

[42] E Iannone, R Sabella, and S Binetti. Granularity in All-Optical WDM Networks. *IEEE Journal of Lightwave Technology*, 16(12):2318–2327, Dec 1998.

[43] D K Hunter and I Andonovic. Approaches to Optical Internet Packet Switching. *IEEE Communications Magazine*, 38(9):116–122, Sep 2000.

[44] T S El-Bawab and J-D Shin. Optical Packet Switching in Core Networks: Between Vision and Reality. *IEEE Communications Magazine*, pages 60–65, Sep 2002.

[45] B Lavigne, E Balmefrezol, P Brindel, B Dagens, et al. Low input power All-Optical 3R Regenerator based on SOA devices for 42.66Gbit/s ULH WDM RZ transmissions with 23dB span loss and all-EDFA amplification. *Optical Fiber Communication Conference and Exhibit (OFC)*, 2003. Postdeadline paper PD15.

[46] S Watanabe, F Futami, R Okabe, Y Takita, et al. 160 Gbit/s Optical 3R-Regenerator in a Fiber Transmission Experiment. *Optical Fiber Communication Conference and Exhibit (OFC)*, 2003. Postdeadline paper PD16.

[47] A Carena, M D Vaugn, R Gaudino, M Shell, and D J Blumenthal. OPERA: An Optical Packet Experimental Routing Architecture with label Swapping Capability. *IEEE Journal of Lightwave Technology*, 16(12):2135–2145, Dec 1998.

[48] T Dimmick, R Doshi, R Rajaduray, G Rossi, B-E Olsson, and D Blumenthal. Optically Multiplexed Transmitter for Hybrid Baseband and Subcarrier Multiplexed Signals. *European Conference on Optical Communications (ECOC)*, pages 155–156, Sep 2000. Paper 6.2.5.

[49] D J Blumenthal, J E Bowers, L Rau, S Rangarajan, W Wang, and H N Poulsen. Optical Signal Processing for Optical packet Switching Networks. *IEEE Communications Magazine*, 41(2):S23–S29, Feb 2003.

[50] T Nakahara, R Takahashi, and H Suzuki. Self-Routing of 100Gb/s Optical Packets using Self Serial-to-Parallel Conversion-Based Label Recognition. *IEEE Photonics Technology Letters*, 15(4):602–604, Apr 2003.

[51] M Zirngibl. What are the Problems we are Solving by Optical Switching? *Optical Fiber Communication Conference and Exhibit (OFC)*, pages 159–160, Mar 2002. Paper TuX3.

[52] C Guillemot, M Renard, P Gambini, et al. Transparent Optical Packet Switching: The European ACTS KEOPS Project Approach. *IEEE Journal of Lightwave Technology*, 16(23):2117–2134, Dec 1998.

[53] D K Hunter, M H M Nizam, K M Guild, J D Bainbridge, et al. WASPNET: A Wavelength Switched Packet Network. *IEEE Communications Magazine*, 37(3):120–129, Mar 1999.

[54] W Lu, B A Small, J P Mack, L Leng, and K Bergman. Optical Packet Routing and Virtual Buffering in an Eight-Node Data Vortex Switching Fabric. *IEEE Photonics Technology Letters*, 16(8):1981–1983, Aug 2004.

[55] R Srinivasan and A K Somani. A Generalised Framework for Analyszing Time-Space Switched Optical Networks. *IEEE Journal on Selected Areas in Communications*, 20(1):202–215, Jan 2002.

[56] X Cao, V Anand, and C Qiao. Waveband Switching in Optical Networks. *IEEE Communications Magazine*, pages 105–111, Apr 2003.

[57] I H White, R V Penty, J Hankey, K A Williams, G F Roberts, M Glick, and D McAuley. Optical Local Area Networking using CWDM. In *SPIE ITCom 2003*, Orlando, FL, 2003.

[58] F Callegati, G Corazza, and C Raffaeli. Exploitation of DWDM for Optical Packet Switching with Quality of Service Guarantees. *IEEE Journal on Selected Areas in Communications*, 20(1):190–201, Jan 2002.

[59] J McEntee. Solitons go the distance in ultralong-haul DWDM. Fibre Systems Europe 2003. `http://fibers.org/`.

[60] L B Aronson, B E Lemoff, L A Buckman, and D W Dolfi. Low-Cost Multimode WDM for Local Area Networks up to 10Gb/s. *IEEE Photonics Technology Letters*, 10(10):1489–1491, Oct 1998.

[61] C M C Davenport. LANL Gigabit Optical Networking Research. Technical report, LANL, Mar 1997. Rainbow II network in collaboration with IBM.

[62] C S Jelger and J M H Elmirghani. Photonic Packet WDM Ring Networks Architecture and Performance. *IEEE Communications Magazine*, 40(11):110–115, Nov 2002.

[63] A E W Phillips, R V Penty, and I H White. Integrated passive wavelength athermalisation for vertical-cavity semiconductor laser diodes. *IEE Proceedings: Optoelectronics*, 152(3):174–180, 2005.

[64] Y Liu, A R Davies, J D Ingham, R V Penty, and I H White. Nanometer Scale Thermal Drift of 4Gbit/s Uncooled Laser for Tight-Channel-Spacing Low-Cost WDM. *CLEO 2005 Conference Proceedings*, 2005.

[65] D Cotter and A D Ellis. Asynchronous Digital Optical Regeneration and Networks. *IEEE Journal of Lightwave Technology*, 16(12):2068–2080, Dec 1998.

[66] A K Srivastava, Y Sun, J L Zyskind, and J W Suloff. EDFA Transient Response to Channel Loss in WDM Transmission System. *IEEE Photonics Technology Letters*, 9(3):386–388, Mar 1997.

[67] G F Roberts, R V Penty, I H White, A West, and S Moore. Multi-Wavelength Data Encoding for Improved Input Power Dynamic Range in Semiconductor Optical Amplifier Switches. In *The 18th Annual Meeting of the IEEE Lasers and Electro-optics Society (LEOS)*, Orlando, FL, 2005.

[68] M L Loeb and G R Stilwell. High-Speed Data Transmission on an Optical Fiber Using a Byte-Wide WDM System. *IEEE Journal of Lightwave Technology*, 6(8):1306–1311, Aug 1988.

[69] B A Small, O Liboiron-Ladouceur, A Shacham, J P Mack, and K Bergman. Demonstration of a complete 12-port terabit capacity optical packet switching fabric. *Proceedings of the Optical Fiber Communications Conference (OFC)*, Mar 2005.

[70] K Vahala, R Paiella, and G Hunziker. Ultrafast WDM Logic. *IEEE Journal of Selected Topics in Quantum Electronics*, 3(2):698–701, Apr 1997.

[71] R Paiella, G Hunziker, and K J Vahala. Spectral logic gates for byte-wide WDM signal processing. *Optical Fiber Communication Conference and Exhibit (OFC)*, 1999.

[72] A Bhardwaj, P O Hedekvist, H Andersson, and K Vahala. All Optical Front End Error Correction on a Spectral Data Bus. *Conference on Lasers and Electro-optics (CLEO)*, pages 280–281, 2000.

[73] S-K Shao and M-S Kao. WDM Coding for High Capacity Lightwave Systems. *Journal of Lightwave Technology*, 12(1):137–148, Jan 1994.

[74] A West, S Moore, M Dales, and M Glick. Multi-Wavelength Coding for Packet-Switched Optical Interconnects. In *Proceedings of the London Communications Symposium*, University College London, 2005.

[75] M L Loeb and G R Stilwell. An Algorithm for Bit-Skew Correction in Byte-Wide WDM Optical Fiber Systems. *IEEE Journal of Lightwave Technology*, 8(2):239–242, Feb 1990.

[76] G Jeong and J Goodman. Long-Distance Parallel Data Link using WDM Transmission with Bit-skew Compensation. *IEEE Journal of Lightwave Technology*, 14, Jul 1996.

[77] T Sakamoto, N Tanaka, and Y Ando. Skew-Compensation Technique for Parallel Optical Interconnections. *IEICE Transactions on Communications*, E82-B(8):1162–1168, Aug 1999.

[78] D Zhou, B C Wang, R J Runser, I Glesk, and P R Prucnal. Perfectly Synchronised Bit-Parallel WDM Data Transmission over a Single Optical Fiber. *IEEE Photonics Technology Letters*, 13(4):382–384, Apr 2001.

[79] L A Bergman, C Yeh, and J Morookian. Advances in Multichannel MultiGbytes/slash s Bit-Parallel WDM Single Fiber Link. *IEEE Transactions on Advanced Packaging*, 24(4):456–462, Nov 2001.

[80] M Gupta and S Singh. Greening of the Internet. *Proceedings of SIGCOMM*, pages 19–26, Aug 2003.

[81] I Keslassy, S-T Chuang, K Yu, D Miller, M Horowitz, O Solgaard, and N McKeown. Scaling Internet Routers Using Optics. *Proceedings of SIGCOMM*, pages 189–200, Aug 2003.

[82] J Trezza, H Hamster, J Iamartino, H Bagheri, and C De Cusatis. Parallel Optical Interconnects for Enterprise Class Server Clusters: needs and technology solutions. *IEEE Communications Magazine*, 41(2):S36–S42, Feb 2003.

[83] N McKeown. Optics inside routers. *European Conference on Optical Communications (ECOC)*, Sep 2003.

[84] B N Schilit and R Want. Creating and Protecting Digital Worlds. *IEEE Computer*, pages 99–101, Feb 2005.

[85] K Oguchi, O Kamatani, and T Fujii. Large capacity contents handling over optical networks - Motion pictures on optical networks. *Optical Fiber Communication Conference and Exhibit (OFC)*, Feb 2004. Paper ThH2.

[86] Y Maeno, A Tajima, Y Suemura, and N Henmi. 8.5 Gbit/s/port Synchronous Optical Packet-switch. *Proceedings of the Fourth International Conference on Massively Parallel Processing Using Optical Interconnections*, pages 114–119, 1997.

[87] T DeFanti, M Brown, J Leigh, O Yu, E He, et al. Optical Switching Middleware for the OptIPuter. *IEICE Transactions on Communications*, E86-B(8), Aug 2003.

[88] D D Clark, T Roscoe, I Stoica, J Wroclawski, L Zhang, C Partridge, R T Braden, et al. Making the world (of communications) a different place. *ACM SIGCOMM Computer Communication Review*, 35(3), July 2005.

[89] A Liu, R Jones, L Liao, et al. A high-speed silicon optical modulator based on a metal-oxide-semiconductor capacitor. *Nature*, (427):615–618, Feb 2004.

[90] H Rong, R Jones, A Liu, O Cohen, D Hak, et al. A continuous-wave Raman silicon laser. *Nature*, (433):725–728, Feb 2005.

[91] I O'Connor. Optical Solutions for System-Level Interconnect. *Proceedings of the 2004 International Workshop on System Level Interconnect Prediction*, pages 79–88, Feb 2004.

[92] N Sugimoto, Y Suzuki, Y Sakai, et al. A Small (37cc) One-Fiber WDM Optical Ethernet PC Card for Fiber-To-The-Notebook-PC. *Optical Fiber Communication Conference and Exhibit (OFC)*, 1:143–144, Mar 2003. Paper MF114.

[93] The Fibre Channel Association. *Fibre Channel Storage Area Networks*. LLH Technology Publishing, Eagle Rock, VA, 2001.

[94] InfiniBand Trade Association. InfiniBand Technology Overview. URL="http://www.infinibandta.org/ibta/".

[95] S Pope, D Roberts, D Riddoch, K Mansley, D Clarke, T Mills, and A Hopper. CLAN Scalable High Performance User Level Networking. *IEEE Gigabit Networking Workshop*, 2001.

[96] L Dailey Paulson. The Ins and Outs of New Local I/O Trends. *IEEE Computer*, pages 13–16, Jul 2003.

[97] Edward Solari and Brad Congdon. *The Complete PCIExpress Reference*. Intel Press, Hillsboro, OR, 2003.

[98] G K Chang, G Ellinas, B Meagher, W Xin, et al. A Proof-of-Concept, Ultra-low Latency Optical Label Switching Testbed Demonstration for Next Generation Internet Networks. *Optical Fiber Communication Conference and Exhibit (OFC)*, 2000.

[99] B Meagher, G K Change, G Ellinas, Y M Lin, W Xin, et al. Design and Implementation of Ultra-low Latency Optical Label Switching for Packet-Switched Networks. *IEEE Journal of Lightwave Technology*, 18(12):1978–1987, Dec 2000.

[100] D McAuley. Optical Local Area Network. In A Herbert and K Spärck-Jones, editors, *Computer Systems: Theory, Technology and Applications*. Springer-Verlag, Feb 2003.

[101] S Iyer and N W McKeown. Analysis of the Parallel Packet Switch Architecture. *IEEE/ ACM Transactions on Networking*, 11(2):314–324, Apr 2003.

[102] P Toliver, I Glesk, R J Runser, K-L Deng, B Y Yu, and P R Prucnal. Routing of 100Gb/s Words in a Packet-Switched Optical Networking Demonstration (POND) Node. *IEEE Journal of Lightwave Technology*, 16(12):2169–2180, Dec 1998.

[103] D K Hunter, M C Chia, and I Andonovic. Buffering in Optical Packet Switches. *IEEE Journal of Lightwave Technology*, 16(12):2081–2094, Dec 1998.

[104] L B James, G F Roberts, M Glick, D McAuley, K A Williams, R V Penty, and I H White. Wavelength Striped Semi-synchronous Optical Local Area Networks. In *Proceedings of the London Communications Symposium*, Sep 2003.

[105] A Smiljanic, M Boroditsky, and N J Frigo. High-Capacity Packet-Switched Optical Ring Network. *IEEE Communications Letters*, 6(3):111–113, Mar 2002.

[106] C S Jelger. Characterisation of a Wavelength Division Multiplexing Multi-Ring Network. MPhil, University of Wales, Swansea, UK, 2000.

[107] K A Williams, G F Roberts, T Lin, R V Penty, I H White, M Glick, and D McAuley. Integrated Optical 2x2 Switch for Wavelength Multiplexed Interconnects. *IEEE Journal Of Selected Topics In Quantum Electronics*, 11(1):78–84, Jan/Feb 2005.

[108] X Ma and G-S Kuo. Optical Switching Technology Comparison: Optical MEMS vs Other Technologies. *IEEE Optical Communications*, pages 16–23, Nov 2003.

[109] Y Shibata, Y Yamada, K Habara, and N Yoshimoto. Semiconductor Laser Diode Optical Amplifiers/Gates in Photonic Packet Switching. *IEEE Journal of Lightwave Technology*, 16(12):2228–2235, Dec 1998.

[110] Y Maeno, Y Suemura, and N Henmi. A Novel Interconnection Network using Semiconductor Optical Amplifier Gate Switches for Shared Memory Multiprocessors. *Proceedings of the International Conference on Massively Parallel Processing Using Optical Interconnections*, pages 239–246, 1996.

[111] M Glick, M Dales, D McAuley, T Lin, K A Williams, R V Penty, and I H White. Swift: A testbed with optically switched data paths for computing applications. *Proceedings of the 7th International Conference on Transparent Optical Networks (ITCON)*, Jul 2005.

[112] J Y Wei. The Role of DCN in Optical WDM Networks. *Optical Fiber Communication Conference and Exhibit (OFC)*, 2000. Paper FI1.

[113] C-C Chen, L A Wang, and S-Y Kuo. A Wavelength Encoded Multichannel Optical Bus for Local Area Networks. *IEEE Journal of Lightwave Technology*, 14(3):315–323, Mar 1996.

[114] M Glick, L B James, D McAuley, G F Roberts, and K A Williams. SOAPS Network Interface Card Specification v1.0. Technical report, University of Cambridge and Intel Research, 2003.

[115] J Scourias. Overview of GSM: The Global System for Mobile Communications, 1996. url = "http://citeseer.nj.nec.com/scourias96overview.html".

[116] M Glick, D McAuley, K A Williams, et al. Wavelength Striped Semi-synchronous Optical Local Area Networks. *Proceedings of the European Conference on Optical Communications (ECOC)*, 2003.

[117] S Han and M-S Lee. Burst-Mode Penalty of AC-Coupled Optical Receivers Optimized for 8B/10B Line Code. *IEEE Photonics Technology Letters*, 16(7):1724–1726, Jul 2004.

[118] R T Hofmeister, C-L Lu, M-C Ho, P Poggiolini, and L G Kazovsky. Distributed Slot Synchronisation (DSS): A network-wide slot synchronisation Technique for Packet-switched Optical Networks. *Journal of Lightwave Technology*, 16(12):2109–2115, Dec 1998.

[119] G I Papadimitriou, P A Tsimoulas, M S Obaidat, and A S Pomportsis. *Multiwavelength Optical LANs*. Wiley, 2003.

[120] E Modiano. WDM-Based Packet Networks. *IEEE Communications Magazine*, pages 130–135, Mar 1999.

[121] D Patterson. Latency Lags Bandwidth. *Communications of the ACM*, 47(10), Oct 2004.

[122] S Finkler and D Sidhu. Performance Analysis of IEEE 802.3z Gigabit Ethernet Standard. *Global Telecommunications Conference (GLOBECOM)*, pages 1302–1306, 1999.

[123] C E Shannon. A Mathematical Theory of Communication. *Bell Systems Technical Journal*, 27:379–423, 623–656, 1948.

[124] R W Hamming. Error Detecting and Correcting Codes. *Bell Systems Technical Journal*, 29:147–160, 1950.

[125] S B Wicker. *Error Control Systems for Digital Communication and Storage*. Prentice Hall, 1994. ISBN 0-132-00809-2.

[126] W W Peterson and E J Weldon. *Error-Correcting Codes*. The MIT Press, 1972. ISBN 0-262-16039-0.

[127] D J C MacKay. *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, 2003.

[128] D J Torrieri. The Information-Bit Error Rate for Block Codes. *IEEE Transactions on Communications*, COM-32(4):474–476, Apr 1984.

[129] D J Torrieri. Information-Bit, Information-Symbol and Decoded-Symbol Error Rates for Lienar Block Codes. *IEEE Transactions on Communications*, 36(5):613–617, May 1988.

[130] C Desset, B Macq, and L Vandendorpe. Computing the Word-, Symbol- and Bit-Error Rates for Block Error-Correcting Codes. *IEEE Transactions on Communications*, 52(6):910–921, Jun 2004.

[131] R Jain. Error Characteristics of Fiber Distributed Data Interface (FDDI). *IEEE Transactions on Communications*, 38(8):1244–1252, 1990.

[132] D Fiorini, M Chiani, V Tralli, and C Salati. Can We Trust in HDLC? *ACM Computer Communication Review*, pages 61–80, 1994.

[133] A X Widmer and P A Franaszek. A DC-Balanced, Partitioned-Block, 8B/10B Transmission Code. *IBM Journal of Research and Development*, 27(5), 440–451 1983.

[134] T Fujiwara, T Kasami, and S Lin. Error Detecting Capabilities of the Shortened Hamming Codes Adopted for Error Detection in IEEE Standard 802.3. *IEEE Transactions on Communications*, 37(9), Sep 1989.

[135] R Braden, D Borman, and C Partridge. Computing the Internet Checksum. *ACM SIGCOMM Computer Communication Review*, 19(2), Apr 1989.

[136] R Black. Fast CRC32 in Software. *University of Cambridge Computer Lab "Blue Book"*, Feb 1994. URL="http://www.cl.cam.ac.uk/Research/SRG/bluebook/21/crc/crc.html".

[137] J L Hammond, J E Brown, and S S Liu. Development of a transmission error model and error control model. Technical Report RADC-TR-75-138, Rome Air Development Center, 1975.

[138] P Koopman. 32-Bit Cyclic Redundancy Codes for Internet Applications. *Proceedings of Dependable Systems and Networks (DSN)*, 2002.

[139] J Stone and C Partridge. When the CRC and TCP Checksum Disagree. In *Proceedings of ACM SIGCOMM 2000*, pages 309–319, 2000.

[140] B Manning and P Vixie. Operational Criteria for Root Name Servers. RFC 2010, 1996, (Format: TXT=14870 bytes) (Obsoleted by RFC2870) (Status: INFORMATIONAL).

[141] L Jolitz and W Jolitz. *Inside the Internet Data Center*. Wiley, 2002.

[142] D L Tennenhouse. Layered multiplexing considered harmful. In *Protocols for High-Speed Networks*. North Holland, Amsterdam, Based on a presentation at IFIP WG 6.1/WG6.4 International Workshop on Protocols for High-Speed Networks, Zurich, May 1989 1989.

[143] R Chakravorty et al. Performance Optimizations for Wireless Wide-Area Networks: Comparative Study and Experimental Evaluation. *Proceedings of ACM Mobicom*, Sep 2004.

[144] J Crowcroft, I Wakeman, Z Wang, and D Sirovica. Is Layering Harmful? *IEEE Network Magazine*, 6(1):20–24, 1992.

[145] D D Clark and D L Tennenhouse. Architectural considerations for a new generation of protocols. *Proceedings of the ACM symposium on Communications architectures & protocols*, pages 200–208, 1990.

[146] J Manchester, J Anderson, B Doshi, and S Dravida. Ip Over SONET. *IEEE Communications Magazine*, 36(5), May 1998.

[147] W Simpson. PPP over SONET/SDH. RFC 1619, 1994, (Format: TXT=8893 bytes) (Obsoleted by RFC2615) (Status: PROPOSED STANDARD).

[148] A Malis and W Simpson. PPP over SONET/SDH. RFC 2615, 1999, (Format: TXT=18708 bytes) (Obsoletes RFC1619) (Status: PROPOSED STANDARD).

[149] J Stone, M Greenwald, C Partridge, and J Hughes. Performance of Checksums and CRC's over Real Data. volume 6, IEEE/ACM Transactions on Networking (TON) 1998.

[150] IEEE. IEEE 802.3ab — Physical Layer Parameters and Specifications for 1000 Mb/s Operation over 4 pair of Category 5 Balanced Copper Cabling, Type 1000BASE-T, 1999. Standard.

[151] IEEE. IEEE 1394b — High-Performance Serial Bus, 2002. Standard.

[152] D G Cunningham and W G Lane. *Gigabit Ethernet Networking*. New Riders, 1999.

[153] A Lauck. An Analysis of CSMA/CD Undetected Error Rates. Technical report, Presented to IEEE Project 802 Committee, Aug 1982. URL="http://www.madriver.com/users/tlauck/sfd.html".

[154] tcpfire, 2003. http://www.nprobe.org/tools/.

[155] Cisco. 1000BASEZX GBIC Specification.

[156] L B James, A W Moore, and M Glick. Structured Errors in Optical Gigabit Ethernet. In *Passive and Active Measurement Workshop (PAM 2004)*, Sep 2004.

[157] R Ramaswami and K N Sivarajan. *Optical Networks*, pages 258–263. Morgan Kaufmann, 2002.

[158] A Lesea. Bit Error Rate: What is it? What Does it Mean? *Xilinx TechXclusives*, 2004. URL="http://www.xilinx.com".

[159] Various. The End-To-End Interest Mailing List Archive, Jun 2005. URL="http://www.postel.org/pipermail/end2end-interest/".

[160] ANSI. T1.105-1988, Synchronous Optical Network (SONET) — Digital Hierarchy: Optical Interface Rates and Formats Specification, 1988.

[161] IEEE. IEEE 802.15.4 — Wireless Personal Area Network, 2003. Standard.

[162] A W Moore, J Hall, E Harris, C Kreibech, and I Pratt. Architecture of a Network Monitor. In *Proceedings of the Fourth Passive and Active Measurement Workshop (PAM 2003)*, pages 309–319, 2003.

[163] D C Feldmeier. Fast Software Implementation of Error Detection Codes. *IEEE/ACM Transactions on Networking*, 3(6):640–651, Dec 1995.

[164] W W Plummer. TCP Checksum Function Design. *Internet Experiment Note 45*, Jun 1978.

[165] M Zorzi and R M Rao. Effect of Correlated Errors on TCP. *Conference on Information Sciences and Systems*, Mar 1997.

[166] T K Woodward, A L Lentine, J D Fields, G Giaretta, and R Limacher. First Demonstration of Native Ethernet Optical Transport System Prototype at 10Gb/s Based on Multiplexing of Gigabit Ethernet Signals. *IEEE Photonic Technology Letters*, 12(8):1100–1102, Aug 2000.

[167] J Chesterfield et al. Exploiting Diversity to Enhance Multimedia Streaming Over Cellular Links. *Proceedings of INFOCOMM 2005*, Mar 2005.

[168] Architectural Implications of Link Indications, Aug 2005. Internet Draft, URL="http://www.iab.org/documents/drafts/draft-iab-link-indications-03.txt".

[169] J Gowar. *Optical Communications Systems*. Pearson, Second Edition, 1993.

[170] T Healy, F C Garcia Gunning, and A D Ellis. Performance Evaluation of FEC Codes in Highly Spectrally Efficient 42.6 Gbit/s Coherent WDM Optical Transmission System. *Proceedings of the European Conference on Optical Communications (ECOC)*, Sep 2005.

[171] Various. IEEE P802.3ae 10Gb/s Ethernet Task Force and Higher Speed Study Group notes, 1999-2002. URL = "http://grouper.ieee.org/groups/802/3/ae/public/index.html".

[172] P E Boudreau, W C Bergman, and D R Irvin. Performance of a cyclic redundancy check and its interaction with a data scrambler. *IBM Journal of Research and Development*, 38(6):651–658, 1994.

[173] W J Bainbridge, W B Toms, D A Edwards, and S B Furber. Delay-insensitive, Point-to-Point Interconnect Using M-of-N Codes. In *Proceedings of the Ninth International Symposium on Asynchronous Circuits and Systems*, 2003.

[174] K A S Immink. Code configuration for avoiding error propagation. *IEE Electronics Letters*, 32(24), Nov 1996.