Chasing Errors through the Network Stack – A Testbed for Investigating Errors in Real Traffic on Optical Networks

Andrew W. Moore*, Laura B. James†, Madeleine Glick‡, Adrian Wonfor†
Ian H. White†, Derek McAuley‡ and Richard V. Penty†
*University of Cambridge, Computer Laboratory, andrew.moore@cl.cam.ac.uk
†University of Cambridge, Department of Engineering, Centre for Photonic Systems
{lbj20, aw300, ihw3, rvp11}@eng.cam.ac.uk
‡Intel Research, Cambridge, {madeleine.glick, derek.mcauley}@intel.com

***Abstract*— A testbed is described which allows both physical layer errors to be observed and analysed, as well as monitoring network performance via frame loss. Real network traffic loads can be used for testing, so that all measurements taken are representative of what would be seen in a deployed system.**

We illustrate our testbed with an examination of the behaviour of a well-known networking standard, Gigabit Ethernet, in conditions of reduced receiver power on optical fibre. Our testbed results show that the line codes used to represent the data in the network affect the bit error rate for that data. Along with the previously reported result that bit error rate and packet error rate have only a weakly deterministic relationship, this highlights the need for testing of all network layers within a complete system carrying real world traffic.

I. Introduction

Many modern networks are constructed as a series of layers. The use of layered design allows for the modular construction of protocols, each providing a different service, with all the inherent advantages of a module-based design. Network design decisions are often based on assumptions about the nature of the underlying layers. For example, the design of an error-detecting algorithm, such as a packet checksum, will be based upon a number of premises about the nature of the data over which it is to work and assumptions about the fundamental properties of the underlying communications channel over which it is to provide protection. We believe that it is important for network designers at all levels to appreciate the layers above and below the ones they are concerned with, and that this will become especially important in the design of optical networks in the future.

Researchers and engineers often test optical components and sub-systems using pseudo-random bit sequences, and base their performance evaluations of these overall systems on the bit error rate (BER) thus obtained. However, most networks will be carrying real traffic very different in character to these short, artificial sequences, thus making the frame loss, data integrity, and other high level network metrics more important to the network's operators and users than BER.

Our work considers the effects of errors on data at various levels in the network stack. The OSI reference model [1] is illustrated in Figure I, using the example of a Gigabit Ethernet system being used for web access.

A. Optical Networking

Current work in all areas of networking has led to increasingly complex architectures: we have focused upon the field of optical networking. To take advantage of capacity developments offered by optical systems at the short timescales relevant to local area networks, packet switching and burst switching techniques have seen significant investigation. An example is the project to investigate Optical Packet Switching, involving the construction of a switched optical data path based upon semiconductor optical amplifiers [2]. This work attempts to minimise latency, and avoids both optical buffering and all-optical signal processing. This architecture uses high speed optical switch fabrics for routing, and combines this with wavelength striping and a separate control channel. The data path between the sending and receiving end-systems has a significant number of devices such as amplifiers and wavelength multiplex units in the path.

Deployments with longer runs of fibre may use large numbers of splitters for measurement and monitoring, as well as active optical devices; the overall system loss in these new, complex systems is therefore greater than in today's point-to-point links. Other examples include Ethernet in the last mile, and passive optical networks [3].

The Power vs. Speed Problem: In competition with the reduced available operating power is the desire to increase network 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. Doubling the bit rate means that the number of photons sent per unit time must thus be doubled (doubling the transmission power) to maintain the BER.

The ramifications of this are that a future system operating at twice the current bit rate 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. However, fibre nonlinearities impose limits on the maximum optical power that can be used in an optical network. Subsequently,

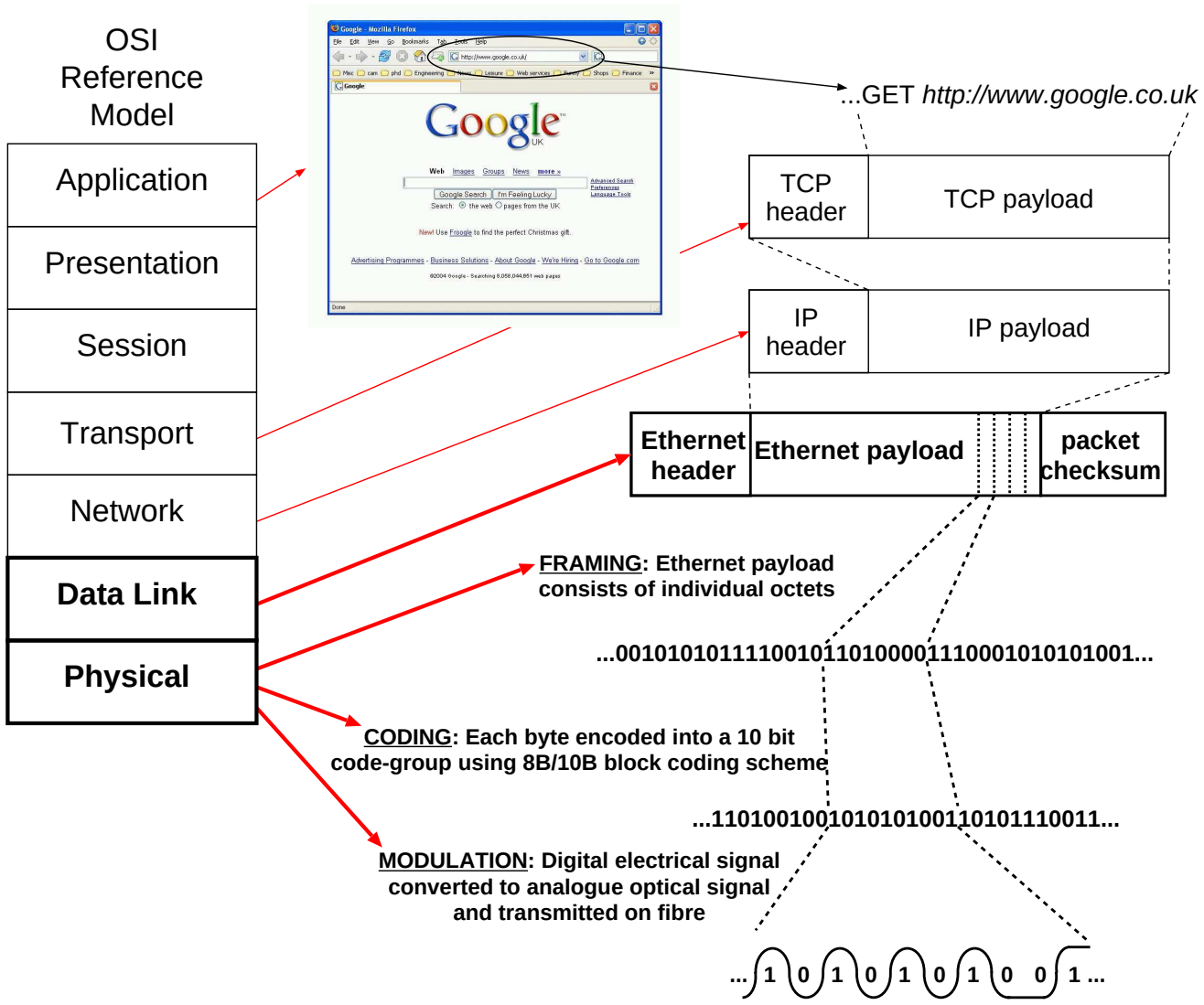


Fig. 1. The framing, coding and modulation of data, Gigabit Ethernet in this example, in the context of the OSI Reference Model

we maintain that a greater understanding of the low-power behaviour of network systems, where errors are more likely to occur, will provide invaluable insight for future designs.

With both increased system complexity and higher speeds in future optical networks reducing the available power, the receivers will have to operate at lower optical powers, and be tolerant of errors. To design for this situation, a thorough understanding of the nature and effects of these errors is crucial.

B. Bit Error Rate and Packet Error Rate

Previous work showed that the relationship between bit error rate and power at the receiver could not offer a prediction of the outcome for packet error rate versus receiver power [4]. This is due to the processing of the data through the network stack, and in particular the effects of line coding at the physical and data-link layers.

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 recovery of transmitted bits; often this requires a minimum number of bit transitions to occur for successful clock and data recovery. In most systems, transceivers are AC coupled, which can lead to distorted pulses and baseline wander (as the DC component of the signal builds up), so an equal balance of `0`s and `1`s is important to counteract this. Numerous mechanisms exist to convert data to be communicated into a form suitable for transmission; two of the more common types are scramblers and block codes.

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 `0`s and `1`s along with sufficient transitions to maintain clock synchronization, a scrambler need only *whiten* the input data, ensuring that there are suitable numbers of `0`s and `1`s for transmission. The operation of a scrambler may be considered as the multiplication of the input data with a random number; the receiver divides 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 `0`s or `1`s, 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: are another popular choice; they translate s bits of data into x bits for transmission, where $s \leq x$. 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 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.

The coding scheme used will affect the way in which bit errors on the line will propagate up the network stack.

C. Higher Layer Implications

Our previous work showed that the physical conditions, line coding and particular data for transmission through a network interact so as to cause non-uniform distribution of packet errors. We also found that certain data values had a substantially higher probability of being received in error than others: error *hot-spotting* [4]. Further sets of wide-ranging experiments allowed us to conclude that Ethernet frames containing a given octet of certain value were up to 100 times more likely to be received in error (and thus dropped), when compared with a similarly sized packet that did not contain such octets [5]. This type of behaviour can lead to failures inducing, at best, poor performance and, at worst, undetected errors that may focus upon specific networks, applications and users. This content-specific effect is particularly insidious because it occurs without a total failure of the network. These effects highlight the need for thorough testing of the whole network system in realistic and representative conditions.

In addition to increasing the chances of frame-discard due to data-contents, error *hot-spotting* also has implications for higher level network protocols. Packet checksums assume that there will be a uniformity of errors within the frame, justifying detection of single-bit errors with a given precision. While Jain [6] demonstrates that the checksum as used in Ethernet is sufficiently strong as to detect all 1, 2 and 3 bit errors for frames up to 8 KBytes in length, problems may be encountered for certain combinations of errors above this. We note that in some coding schemes, single-bit errors on the physical layer will translate into multi-bit errors following decoding (see Section III). Also, Stone *et al.* [7] discuss the impact this non-uniformity of error has for the checksum of TCP, further up the network stack. This may call into question our assumption that only increased packet-loss will be the result of the error *hot-spots*. Instead of just lost packets, Stone *et al.* noted certain "unlucky" data would rarely have errors detected. However, the probability of 4 or more bits of data error occurring in a pattern which would defeat the CRC's detection ability is low [6].

In this paper, having outlined the motivations for this work, we describe the construction of our testbed. This allows network performance to be measured at various levels in the network stack, enabling us to fully understand the behaviour of a system under real world load. We can monitor bit errors on the line due to physical conditions at the same time as packet loss, which affects network application performance. We then present some example results obtained using the testbed, and note how these illustrate the importance of understanding the impact of errors on several layers of the networking stack. These effects would be virtually impossible to investigate using traditional Bit Error Rate techniques, and this demonstrates the true flexibility of our testbed.

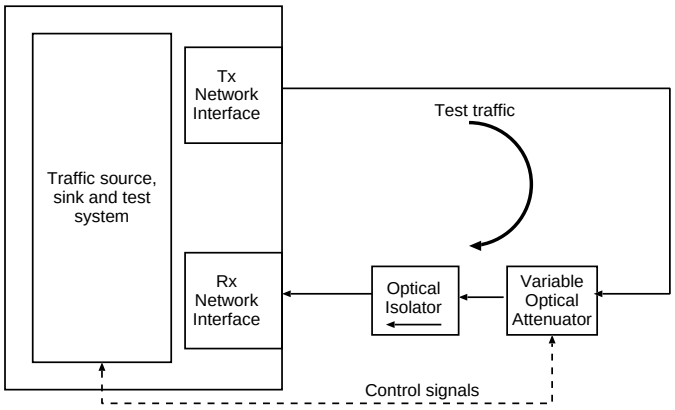


Fig. 2. Main test environment, set up for low receiver power testing

II. Experimental Method

Our testbed allows us to compare two commonly used metrics: bit error rate, as used to describe the physical layer performance, and packet error rate: a measurement of network, transport and application layer performance.

A. Testbed Environment

Figure 2 illustrates our testbed. A variable optical attenuator and an optical isolator are placed in one direction of the test link. A traffic generator feeds this link, and the link terminates in a traffic sink and tester. We use software to control the test bed and to generate and evaluate test data.

A packet capture and measurement system is implemented within the traffic sink using an enhanced driver for the network interface card. The modified driver allows application processes to receive error-containing frames that would normally be discarded.

We use a special-purpose traffic generator and comparator, which is combined into one real-time software module. This system transmits pre-constructed test data frames in a standard format. Transmitted frames are compared to their received versions and if they differ, both original and error frames are stored for later analysis. This testbed software, *tcpfirediff*, is based upon *tcpfire*[1]. We also use purpose-built code for error analysis in the receiving system.

While a variable optical attenuator is shown in this figure, we are not limited to testing reduced power cases, and the

[1]http://www.cl.cam.ac.uk/Research/SRG/netos/nprobe/downloads/

modular nature of the test environment allows us to substitute other devices as needed. The attenuation in the link can be varied automatically for new measurements, based on previous results. We had previously noted interference due to reflection and the isolator allows us to remove this aspect from the results.

Clearly this testbed is a simplified version of a true network with multiple nodes, each with multiple input sources. However, the effects observed in such a network will be comparable to those seen here, with more complexity added by, for example, power and clock synchronisation variation between packets.

B. Lower OSI Layer Testing

As well as measuring packet loss, our testbed hardware and software allows us to examine errors at lower layers in the network stack. The received data just above the physical line coding layer is easily observed and recorded. Depending on the system under test, actual line errors are either observed or can be deduced from the decoded data.

The ability to monitor network application performance at the same time as observing lower level errors and how they propagate through the network layers allows us to fully test and understand the interactions between network layers. The nature of the modular, layered design of network stacks has sometimes worked against the architects, implementers and users. There exists a tension between the desire to place functionality in the most appropriate sub-system, ideally optimised for each incarnation of the system, and the practicalities of modular design intended to allow independent developers to construct components that will inter-operate with each other through well-defined interfaces. This can lead to problems when layers do not behave as the designers of other system components expect. The ability to test a network at multiple points in the stack is the contribution of our testbed.

C. Real Traffic

Our testbed can use any available traffic, including traces from genuine networks. Tests are conducted either with manually constructed test-frames or with real network traffic, such as the *day-trace* referred to in our example. This network traffic was captured from the interconnect between a large research institution and the Internet [8]. We consider it a representative sample of network traffic for an academic/research organisation of approximately 150 users.

Other traffic tested included *pseudo-random data*, consisting of a sequence of frames of the same number and size as the *day-trace* data — preserving packet-size characteristics — although each is filled with a stream of octets whose values were drawn from a pseudo-random number generator.

III. Example Results obtained using the Testbed

Our previous work examined Gigabit Ethernet on fibre (1000BASE-X), and identified that a uniformly-distributed set of random data, after encoding according to the specification [9] using 8B/10B block code, will not suffer code-errors with the same uniformity.

This led to the investigation of the impact of the line coding scheme upon physical layer errors, when those errors would be observed in the data-link layer. Here we investigate Gigabit Ethernet on optical fibre, under conditions where the received power is sufficiently low as to induce errors in the Ethernet frames. The results described here are drawn from several sets of data taken at a range of attenuation values.

We assume that while the packet checksum within Ethernet is sufficiently strong to catch the errors, the dropped frames and resulting packet loss will result in a significantly higher probability of packet errors than the norm for certain hosts, applications and perhaps users.

8B/10B Block Coding: Gigabit Ethernet uses 8B/10B block coding, which converts 8 bits of data for transmission (ideal for any octet-orientated system) into a 10 bit line code. Although this adds a 25% overhead, 8B/10B has many valuable properties: at least 3 transitions in each 10 bit code group and a maximum run length of 5 bits (both important for clock recovery), and virtually no DC spectral component. In addition to being the standard Physical Coding Sublayer for Gigabit Ethernet [9], it is used in Fibre Channel, the 800Mbps extensions to the IEEE 1394 / Firewire standard, and is the basis of coding for the electrical signals of PCI Express.

The 8B/10B codec defines encodings for data octets and control codes which are used to delimit the data sections and maintain the link. Individual codes or combinations of codes are defined for Start of Packet, End of Packet, line Configuration, and so on. Also, Idle codes are transmitted when there is no data to be sent to keep the transceiver optics, electronics, and clock-recovery active.

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, a one bit error in a code-group can be decoded to give a byte with 4 bits of difference from the original transmitted byte. In addition, the calculations for the scheme used to maintain DC balance after the code-group may be be thrown off, leading to potential errors in subsequent decoding.

A. Octet Analysis in the Testbed

For use with Gigabit Ethernet, the testbed contains a module which allows per-octet analysis of received packets, comparing to the known transmitted data octets. This allows us to observe both actual line errors in the 10 bit code-groups of Gigabit Ethernet, and also the data errors in the decoded octets as would be passed up the network stack.

Various types of code-group damage may be observed. One of these is the single-bit error caused by the low signal to noise ratio at the receiver; another results from a loss of bit clock causing a subsequent bit ro be read as having the value of the previous bit. A final example results from the loss of code-group clock synchronisation. This can lead to the code-group boundaries being misplaced, so that a sequence of several code-groups, and thus several octets, will be incorrectly recovered.

This octet-analysis software allows us to collect a great deal of information about errors in the system, and can also store

information for further study in different environments. Similar analysis is possible for other line coding schemes.

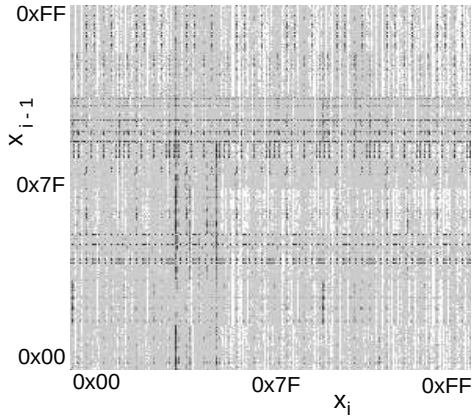
B. Effects on data sequences

We have found that individual errored octets in Gigabit Ethernet do not appear to be clustered within frames but are independent of each other. However, we are interested in whether earlier transmitted octets have an effect on the likelihood of a subsequent octet being received in error.

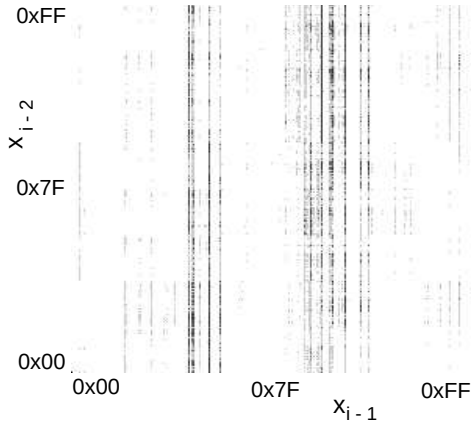
Our custom testbed software allows us to collect statistics on how many times each transmitted octet value is received in error, and also store the sequence of octets transmitted preceding this. Our software can store error counts in 2D matrices (or histograms) of size 256×256, representing each pair of octets in the sequence leading up to the errored octet. One histogram is produced for the errored octet and its immediate predecessor, one for the predecessor and the octet before that, and so on. We normalise the error counts for each of these histograms by dividing by the matrix representing the frequency of occurrence of this octet sequence in the original transmitted data. We then scale each histogram matrix so that the sum of all entries in each matrix is 1.

Figure 3 shows error frequencies with darker areas representing higher error probabilities. Figure 3(a) has the "current octet" X_i (the correct transmitted value of the octet which was then received in error) on the x-axis, versus the octet which was transmitted before each specific errored octet, X_{i-1}, on the y-axis. Figure 3(b) shows the preceding octet and the octet before that: X_{i-1} vs X_{i-2}, where X_i is the octet which was received in error. An example of this might be the transmitted sequence `1,2,3,4` which was received as `1,2,9,4`. The octet received in error, which is the one we are interested in, is X_i sent in this case as `3`. $X_{i-1} = 2, X_{i-2} = 1$ and so on. Vertical lines in Figure 3(a) are indicative of an octet that is error-prone independently of the value of the previous octet. In contrast, horizontal bands indicate a correlation of errors with the value of the previously transmitted octet; these appear as vertical lines in Figure 3(b).

It can be seen from Figure 3 that while correlation between errors and the value in error, or the immediately previous value, are significant, beyond this there is no significant correlation. The equivalent plot for X_{i-2} vs. X_{i-3} produces a featureless white square.



(a) Error counts for X_i vs. X_{i-1}



(b) Error counts for X_{i-1} vs. X_{i-2}

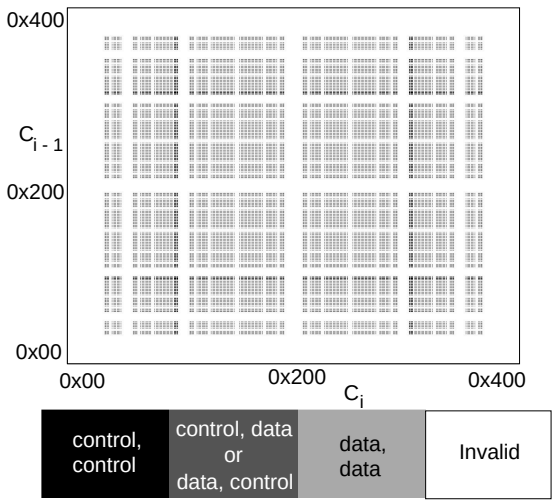
Fig. 3. Error counts for pseudo-random data octets, darker values represent higher probability of error

C. 8B/10B code-group frequency components and their effects

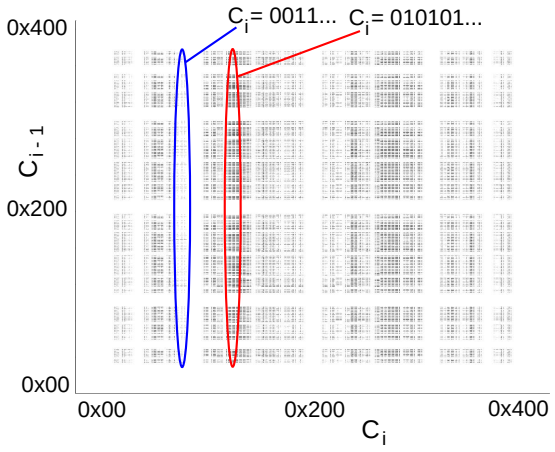
It is illustrative to consider the octets which are most subject to error, and the 8B/10B codes used to represent them. In the pseudo-random data, the following ten octets give the highest error probabilities (independent of the preceding octet value): 0x43, 0x8A, 0x4A, 0xCA, 0x6A, 0x0A, 0x6F, 0xEA, 0x59, 0x2A. It can be seen that these commonly end in A, and this causes the start of the code-group to be `01010`. The octets not beginning with this sequence in general contain at least 4 alternating bits. Of the ten octets giving the lowest error probabilities (independent of previous octet), which are 0xAD, 0xED, 0x9D, 0xDD, 0x7D, 0x6D, 0xFD, 0x2D, 0x3D and 0x8D, the concluding D causes the code-groups to start with `0011`.

Fourier Transforms (FTs) were generated for data sequences consisting of repeated instances of the code-groups of 8B/10B. Examining the FTs of the high error octet sequences shows the peak corresponding to the base frequency (625MHz, half the baud rate) is pronounced in most cases, although there is no such feature in the FTs of the low error octet sequences. The pairs of preceding and current octets leading to the greatest error (which are most easily observed in Figure 3) give much higher error probabilities than the individual octets. The noted high error octets (e.g. 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`, which serves to further emphasise that frequency component.

The 8B/10B codec defines both data and control encodings, and these are represented on a 1024x1024 space in Figure 4(a), which shows valid combinations of the current code-group (C_i) and the preceding one (C_{i-1}).



(a) Valid $C_{(i-1)}, C_i$ pairs



(b) Errors using *day-trace* as a function of code-groups

Fig. 4. The codebook for 8B/10B represented on a 1024x1024 space

In Figure 4(a) the octet errors found in the *day-trace* have been displayed on this codespace, showing the regions of high error concentration for real Internet data. It can be seen that these tend to be clustered and that the clusters correspond to certain features of the code-groups. Two clusters have been ringed, the one indicated as $C_i = 0011\ldots$ represents those codes with a low-error suffix. In contrast that indicated as $C_i = 010101\ldots$ indicates the error-prone symbols with a suffix of 0xA.

Transceiver Effects: This bit frequency-dependent error patterning is primarily due to effects in the electrical/optical interfaces. It is well known that in a directly modulated optical source it is possible that bandwidth limitations can cause *single ones* to achieve slightly less amplitude than a run of multiple ones. In normal operation, this has no effect on the error rate of the received signal and error-free operation 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`. In addition to optical issues of data-pattern, the packaging requirements (printed circuit board tracks, wires, etc.) imposed in the electrical domain can exacerbate this effect.

It is worth noting that the code-groups with multiple transitions, which we have observed as subject to increased error probability, are also beneficial to the network. Clock and data recovery systems at the receiver use these frequent transitions to recover and maintain an accurate bit clock.

These broadband limitation effects will be much more significant at the increased modulation rates required for 10 Gbps Ethernet. Our testbed system is fully modular and can use conventional network interface cards; as such it can easily be modified to test 10 Gbps Ethernet, other standard physical layers, or prototype cards.

IV. Conclusions

We have illustrated how problems may arise in the design, development and evolution of the physical components, algorithms, protocols and applications that make up our networks. These problems are caused by the undesirable interactions between network modules when carrying real traffic loads and may not be detected by the traditional tests of bit error rate in pseudo-random sequences used by the designers of optical systems.

Prototype networks, such as the optically switched system described in [2], illustrate how future optical networks will be complex, consisting of an increasingly large number of diverse elements, with greater limitations on the optical power budget. The 8B/10B codebook is widely deployed and we do not propose alterations to this (of course, specific applications may benefit from the addition of some corrective action to protect those codewords most likely to suffer from errors). However, the design of these new networks must carefully consider the physical layer and its effects on higher level network protocols to ensure that the system is adequately tolerant to errors; thorough testing and understanding of error propagation at all network layers will be vital.

Our testbed system, consisting of a special hardware setup and a suite of custom software, allows error measurements at both network and physical coding levels so that the performance of the network when subject to real traffic can be fully understood.

Acknowledgements: Andrew Moore acknowledges the Intel Corporation's generous support of his research fellowship. Laura James thanks the EPSRC & Marconi for their support of her PhD research.

References

[1] U. Black, *OSI: A Model for Computer Communication Standards.* Prentice-Hall, 1991.

[2] D. McAuley, "Optical Local Area Network," in *Computer Systems: Theory, Technology and Applications*, A. Herbert and K. Spärck-Jones, Eds. Springer-Verlag, Feb 2003.

[3] IEEE, "IEEE 802.3ah —Ethernet in the First Mile," 2004, standard.
[4] L. B. James *et al.*, "Structured Errors in Optical Gigabit Ethernet," in *Passive and Active Measurement Workshop (PAM 2004)*, Apr. 2004.
[5] L. B. James *et al.*, "Packet error rate and bit error rate non-deterministic relationship in optical network applications," in *Proceedings of OFC-2005*, Anaheim, CA, 2005.
[6] R. Jain, "Error Characteristics of Fiber Distributed Data Interface (FDDI)," *IEEE Transactions on Communications*, vol. 38, no. 8, pp. 1244–1252, 1990.
[7] J. Stone, M. Greenwald, C. Partridge, and J. Hughes, "Performance of Checksums and CRCs over Real Data," in *Proceedings of ACM SIGCOMM 2000*, Stockholm, Sweden, Aug. 2000.
[8] A. W. Moore *et al*, "Architecture of a Network Monitor," in *Passive & Active Measurement Workshop 2003 (PAM2003)*, Apr. 2003.
[9] IEEE, "IEEE 802.3z —Gigabit Ethernet," 1998, standard.