The Effect of Data Compression on Packet Sizes in Data Communication Systems

K. Pawlikowski T. Bell H. Emberson
P. Ashton
Department of Computer Science, University of Canterbury,
Christchurch, New Zealand

Abstract

Data compression is well known as a transformation of data traffic that is able to greatly increase the data rate in a network, and as the cost of processing power decreases compression is becoming more and more widely used for this purpose.

In this paper, we look at the statistical properties of this transformation, studying changes in the probabilistic characteristics of packet sizes caused by compressing packets transmitted under three different network protocols (TCP, IP, and Ethernet) and using three different compression algorithms. The analysis is performed on real data traffic recorded in a local area network environment. Our aim is to provide network designers with a model of compressed data traffic that could be useful in modelling and performance evaluation of telecommunication networks.

1 Introduction

Measurements of real data traffic have been widely recognised as an important aid in performance modelling, design, and development of telecommunication networks. Newly emerging services and new applications of data communications demand more exhaustive measurement studies to provide foundations for the development of high speed telecommunication networks of the next century. A survey of pre-1988 publications on data traffic measurements was presented by Pawlita [12]. Most of the research activity reported on is either user- or system-oriented and emphasises the dependence of data traffic on network architecture and/or communication protocols. Little attention has been given to studying the influence of various data transformations on the statistical properties of data traffic.

Data compression, used to reduce data storage requirements and to increase the efficiency of data transmission, has many applications in telecommunication. For example it is used in facsimile services [6], Usenet electronic news [7], and the PCTERM system [1], as well as with the V.42*bis* [14] and MNP protocols.

We focus our attention on data compression as a stochastic transformation of data traffic. An important feature of this transformation is the effect it has on the distribution of packet sizes. The influence of compression on characteristics of data traffic in a local interactive data communication system was reported in [10]. In this paper, we present the results of exhaustive studies of data traffic in a local area network, and its characteristics before and after data compression by three different compression algorithms (described in Section 2). The measurement experiments and the environments in which they were performed are described in Section 3. Experiments were designed to assess the effects of the compression method used and the level of the protocol stack at which compression was applied. Characteristics of uncompressed and compressed data traffic are presented in Section 4.

2 Data Compression

Applying data compression in a communications environment provides a means of trading channel bandwidth for computing power. A system using compression is less demanding on a network, but requires more processing from the devices at each end. There are various levels of tradeoff, ranging from simple compression methods that are very fast but do not reduce the size of the data by much, to very complex methods that require a lot of computing resources, but are able to reduce the amount of data by a considerable amount. Computing power is generally improving at a greater rate than communications bandwidth, so more complex compression methods are becoming viable.

At present data compression is used mainly in wide area networks. Data being transmitted between different cities and countries is often compressed because substantial cost savings can be made. At a more local level, compression is becoming common in modems (particularly where the V.42*bis* and some of the MNP protocols are supported) because it enables higher data rates to be achieved over telecommunication lines. If this trend continues, one can expect to see compression applied in local area networks.

Currently, fast compression methods are used in order to keep the cost of compressing data low. In our work, three compression methods (*Huffman*, *Compress*, and *PPMC*) have been evaluated as transformations of data traffic at the packet level in a local area network. The three methods are representative of the most popular approaches to compression. The first two are already commonly used in protocols such as V.42*bis* and MNP. The PPMC method is currently too resource-intensive to be suitable, but will become viable for many applications in the near future as processor power and memory become cheaper.

The *Huffman* method is an adaptive Huffman code that uses Gallager's method [5] for adapting the code tree. The code for each input byte is determined from the relative frequency of that byte in the prior text so that more common symbols get shorter output codes, and rare symbols get longer output codes.

The *Compress* program is a Ziv-Lempel variant, based on Welch's LZW method [15]. LZW constructs a "dictionary" based on the previously coded text. The string of text that is about to be coded is located in the dictionary and is replaced with a code to represent its location in the dictionary. This method usually gives slightly better compression than Huffman coding because it takes account of some correlations between character occurrences.

The *PPMC* method (Prediction by Partial Matching, method C) is a variation of a method proposed originally by Cleary and Witten [4] and refined by Moffat [9]. The PPMC method "predicts" a forthcoming character using a probabilistic model. The model is constructed using the prior text, so the decoder is able to construct the same model. To code a character, the compressor takes the preceding n (most often three) characters, and obtains the relative frequencies of characters that have occurred in the context of those n characters earlier in the text. This provides a probability distribution that is used to code the next character using arithmetic coding [16]. If the estimated probability of the next character is zero, then a special "escape" character is transmitted, and the compressor shifts down to a context of $n-1$ characters. This continues until the character is eventually coded, using no context at all if necessary. In practice, many characters are coded with high probabilities, and so add few bits to the output. This leads to a very economical coding, usually significantly better than that achieved by *Huffman* or *Compress*. For more details of these methods, the reader is referred to Bell, Cleary and Witten [3].

3 Description of Measurement Experiments

As already mentioned, in this paper we are interested in the changes in distributions of packet sizes caused by data compression in a local area network.

Data traffic was recorded in a LAN in the Computer Science Department of the University of Canterbury. The LAN consisted of 12 Sun workstations and about 60 X-Terminals, all connected by two Ethernet segments. One of the Suns was the departmental file-server and was the bridge between the two Ethernet segments. Some of the other Suns had local disks, but most are diskless.

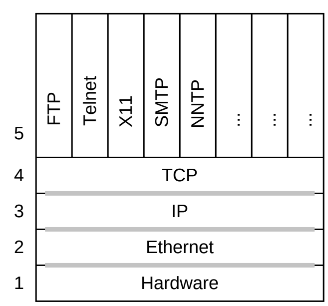

Figure 1: The TCP/IP protocol stack.

Nearly all traffic on the departmental Ethernet segments results from use of the DARPA Internet protocol suite, whose major transport protocols are TCP (a reliable byte-stream protocol) and UDP (an unreliable datagram protocol). The experiments described in this paper are based on analysis of the TCP/IP protocol stack, as shown in Figure 1. Results including an analysis of UDP packets will be published in [11].

Messages passed to TCP are split into packets of less than 64kb[1] and a TCP header of at

[1] In practice, Sun's TCP implementation splits the message into pieces that should not be fragmented when

least 20 bytes is prepended to each packet. Each TCP packet is passed to IP, which forms one or more IP packets each with an IP header of at least 20 bytes and passes them to the Ethernet layer. Finally, the Ethernet interface extends each IP packet by adding an Ethernet header and trailer before transmitting it.

To perform our experiments we collected from our Ethernet segments large numbers of packets, typically 750,000 or 1,000,000 at a time. The collection periods were spread out over a working day so that we obtained a representative sample of packets on the network. We repeated the experiments six times between April and October 1993. In the interests of privacy and data security the raw data was stored in encrypted form and discarded as soon as was practical.

Analysis was performed at the Ethernet, IP and TCP levels, requiring construction of IP and TCP packets from the Ethernet packets recorded. The IP level data was processed to remove all IP packets that contained UDP packets. This processing was not entirely accurate, but we estimate that in the worst-case less than 0.7% of the UDP packets could have escaped removal.

By compressing with three different compression methods the data portions of the Ethernet, IP and TCP packets, we simulated the effect of introducing compression at various levels of the TCP/IP protocol stack. At each level, the headers of the packets were not compressed, although the data portions contained the headers of upper layer packets. The lengths of the uncompressed and compressed data portions of the packets were stored and the packets discarded. The next section discusses our analysis of the statistical properties of these lengths.

4 Data Analysis

Typical distributions of packet sizes before and after compression by the *Huffman*, *Compress* and *PPMC* algorithms are shown in Figure 2. Arrows point to the largest packet size recorded in each case. One can see that in all cases we dealt with typical multimodal distributions observed also in previous studies of LAN teletraffic [2, 13]. Also, it is clear that for all data compression methods used the packet size distribution was flatter than that of the uncompressed packets, while the range of packet sizes after compression became wider.

Data compression is usually characterised by a compression ratio defined, in this case, as the ratio of the mean packet sizes before and after compression. To analyse the changes in statistical characteristics of data traffic caused by compression, we analysed the first three moments of the empirical distributions of packet lengths: the mean value, variance (or, equivalently, its square root, i.e. standard deviation), and third central moment. These results were used to determine the coefficient of variation (the ratio of the standard deviation and the mean) and the coefficient of skewness (the ratio of the third central moment and the cube of the standard deviation).

---

crossing the network. When transmitting over Ethernet, TCP packets are no larger than 1460 bytes, so that when a 20 byte TCP header and a 20 byte IP header are added the packet will fit exactly into a 1500 byte Ethernet packet.

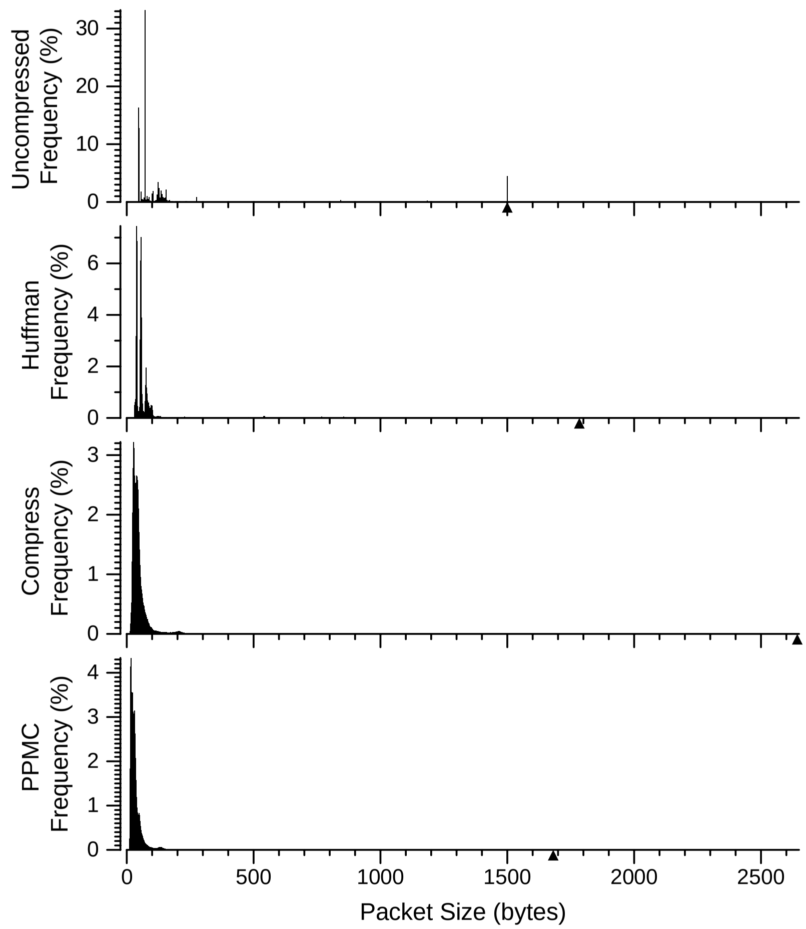

Figure 2: Distributuion of packets sizes at Ethernet level before and after compression

Estimates of the mean packet size, standard deviation, coefficient of variation and coefficient of skewness are depicted in Figures 3 and 4. Estimates in the figures are for two different measurements of real data traffic. Estimates are given for packet size distributions before compression (N) and after compression (H, C and P)

Absolute values of results are shown by numbers above bars, while the relative changes are shown inside bars (as a percentage). These results show typical changes in packet sizes caused by data compression, regardless of the compression algorithm. One can see that a decrease in the mean packet size is accompanied by a decrease of the standard deviation, but that change is not so significant as the former, and as a result the coefficient of variation and skewness of packet sizes increases. The latter change can be quite significant. For example, in Figure 4 the coefficient of skewness in packet sizes at IP level grows almost seven times if the *Compress* or *PPMC* compression method is applied. *PPMC* seems to be not only the most powerful method for reducing the mean, but also for reducing the variance of packet sizes. On the other

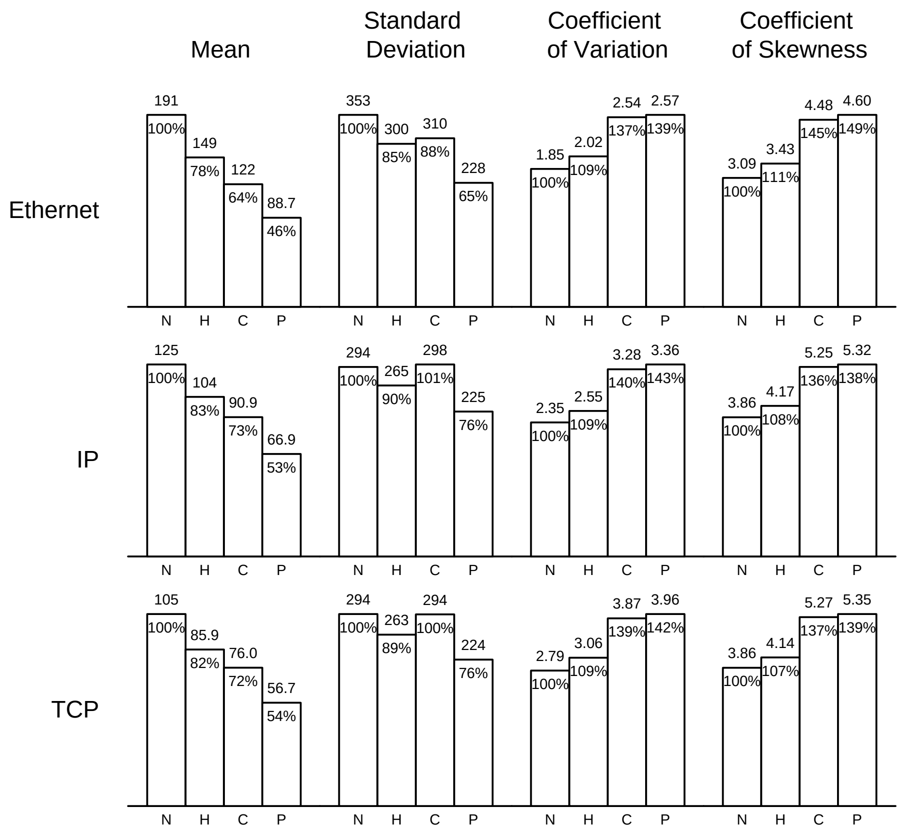

Figure 3: Real data traffic. N = *No* compression, H = *Huffman*, C = *Compress*, P = *PPMC*

hand, one can also see that the better the compression the larger the coefficients of variation and skewness of compressed packet sizes. While these effects of compression were typical for all samples of data traffic that we collected, there are instances where changes of standard deviations, or coefficients of variation, had different character (see, for example, the standard deviation at the IP and TCP levels under *Compress* in Figure 3, or the coefficient of variation at the TCP level under the same method in Figure 4). Closer investigation of these cases led us to the conclusion that the data traffic recorded in these cases was "contaminated" by data that was already compressed. To support this thesis, Figure 5 shows the effects of compression on pre-compressed data traffic. This comfirms that the anomalous behaviour could be caused by already compressed data.

Our results confirm previously reported findings that coefficients of variation greater than one are typical for data traffic in LANs [8]. Our results show too that the coefficients of skewness, both before and after compression, can assume larger positive values than the coefficients of variation.

This indicates that packet sizes both before and after compression can be modelled by the

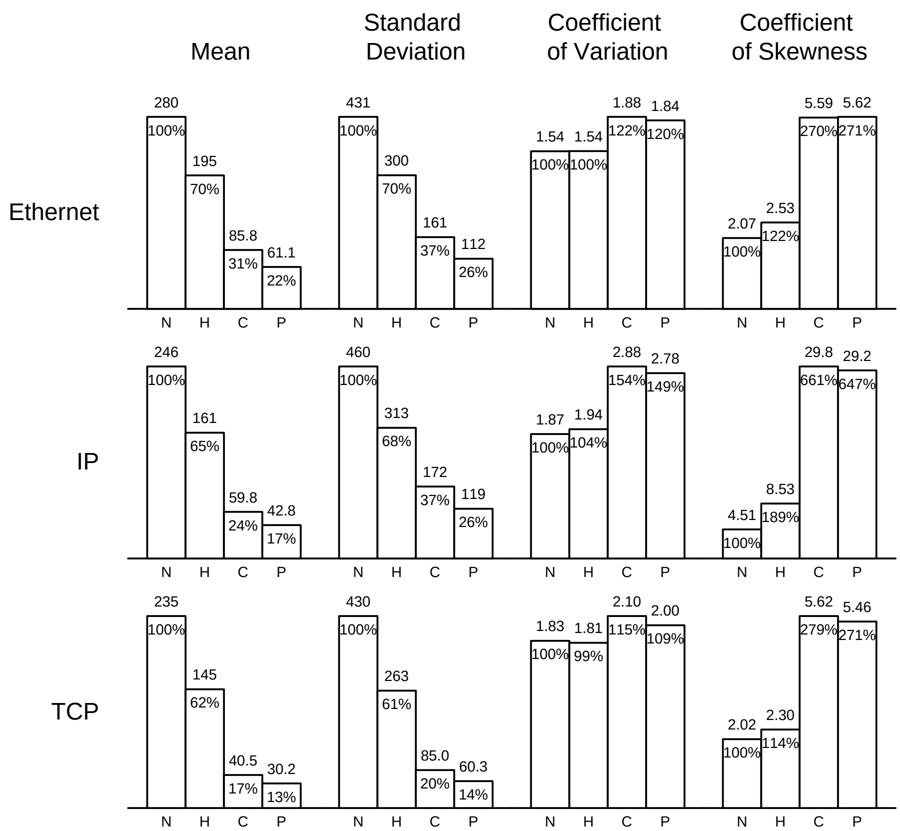

Figure 4: Real data traffic. N = *No* compression, H = *Huffman*, C = *Compress*, P = *PPMC*

same class of probability distributions with coefficients of variations greater than one, that is, a discrete homologue of the class of (continuous) hyperexponential distributions.

This was confirmed by fitting mixed Erlang distributions to our experimental distributions of packet sizes [11], leading us to the conclusion that compression of data traffic at the packet level in local area networks reduces the mean value and variance of the packet size distribution, but the resulting probability distribution remains in the same family of distributions with coefficients of variation and skewness greater than one.

5 Summary

In this paper we have considered the statistical properties of data compression as a transformation of data traffic in a local area network environment. Data traffic was measured at three different levels of the network architecture and three different compression methods were assumed. The results reported in this paper reveal new features of data compression in a network environment and should help to develop more precise models of data traffic.

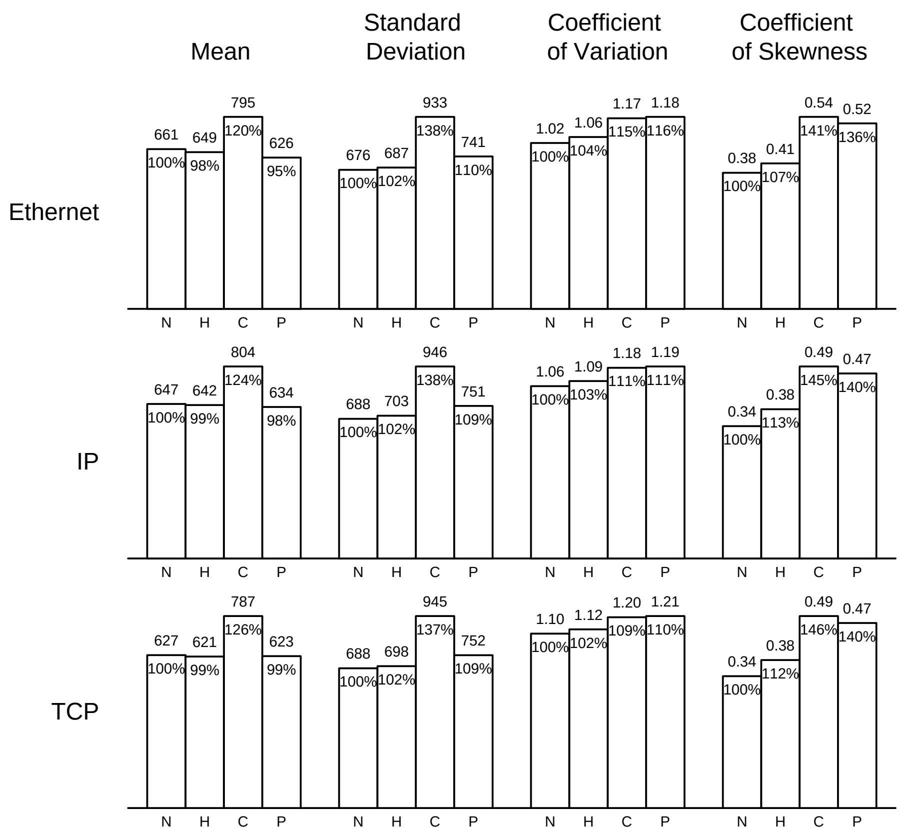

Figure 5: N = No compression, H = *Huffman*, C = *Compress*, P = *PPMC*

Acknowledgements

The authors would like to thank Tony Dale and Srividya Venkataraman for their help with measurement experiments, and the University of Canterbury for a research grant for this project.

References

[1] Auslander, M., Harrison, W., Miller, V., and Wegman, M. PCTERM: a Terminal Emulator Using Compression. In *Proc. IEEE Globcom'85* (New York, 1985), IEEE Press, pp. 860–862.

[2] Balkovich, E. E., and Morse, J. H. Performance of Distributed Software Implemented by a Contention Bus. In *Proc. Symp. on Data Comm.* (Nov. 1981), ACM SIGCOMM., pp. 39–45.

[3] Bell, T. C., Cleary, J. G., and Witten, I. H. *Text Compression.* Prentice Hall, Englewood Cliffs, 1990.

[4] Cleary, J. G., and Witten, I. H. Data Compression Using Adaptive Coding and Partial String Matching. *IEEE Trans. on Comm. 32*, 4 (1984), 396–402.

[5] Gallager, R. G. Variations on a Theme by Huffman. *IEEE Trans. on Information Theory 24*, 6 (1978), 668–674.

[6] Hunter, R., and Robinson, A. H. International Digital Facsimile Coding Standards. *Proceedings of the IEEE 68*, 7 (1990), 854–867.

[7] Krol, E. *The whole Internet user's guide & catalog.* O'Reilly & Associates, 1992.

[8] Marshall, W. T., and Morgan, S. P. Statistics of Mixed Traffic on a Local Area Network. *Computer Networks and ISDN Systems 10* (1985), 185–194.

[9] Moffat, A. Implementing the PPM data compression scheme. *IEEE Transactions on Communications 38*, 11 (Nov. 1990), 1917–1921.

[10] Pawlikowski, K., and Bell, T. C. The Effect of Data Compression on Packet Sizes in Data Communication Systems. In *Teletraffic and data traffic in a period of change*, A. Jensen and V. B. Iversen, Eds., vol. 14 of *Studies in Telecommunication.* North-Holland, Amsterdam, 1991, pp. 556–562.

[11] Pawlikowski, K., Bell, T. C., Emeberson, H., and Ashton, P. Statisical properties of data compression as a transformation of teletraffic. In preparation.

[12] Pawlita, P. Two Decades of Data Traffic Measurements: A Survey of Published Results, Experiences and Applicability. In *Proceedings of the 12th International Teletraffic Congress ITC'12 (Torino, Italy)* (1988), International Advisory Council of ITC, pp. 5.2A.5.1–5.2A.59.

[13] Shoch, J. F., and Hupp, J. A. Measured Perormance of an Ethernet Local Computer Network. *Comm. ACM 23*, 12 (1980), 711–721.

[14] Thomborson, C. The V.42*bis* standard for data compressing modems. *IEEE Micro* (Oct. 1992), 41–53.

[15] Welch, T. A Technique for High-Performance Data Compression. *IEEE Computer 17*, 6 (1984), 8–19.

[16] Witten, I. H., Neal, R., and Cleary, J. G. Arithmetic Coding for Data Compression. *Comm. ACM 30*, 6 (1987), 520–540.