The Entropy of Traffic Streams in ATM Virtual Circuits

Nina T. Plotkin and Pravin P. Varaiya
Department of Electrical Engineering and Computer Science
University of California at Berkeley
Berkeley, CA 94720

Abstract

We model an ATM virtual circuit as a tandem queueing system. At each queue, transmission of the virtual circuit traffic is interrupted by cross traffic, causing the scattering and coalescing of cells. We study this phenomenon using entropy as a traffic descriptor. We compute the entropy of traffic streams produced in simulation, via an estimation technique based on the Lempel-Ziv universal data compression method. The estimator modifies the Lempel-Ziv method to compute entropy rather than compress data. We show that the entropy of virtual circuit traffic at successive queues can either increase or decrease depending upon the types of input traffic. Even when bursty input is scattered, its entropy does not achieve maximum entropy within a reasonable number of queues. We also define a distance metric to compare the correlation structures of two output processes and observe this metric at successive queue outputs.

1 Introduction

An ATM virtual circuit includes a series of network buffering devices, including multiplexers and switches. To understand the behavior of a traffic stream over a virtual circuit, we study the behavior of traffic in a tandem queueing system. With few exceptions, notably quasi-reversible queues, the distribution of the departure processes in a tandem queueing system is not known. We study properties of these departure processes, but the main focus is on their entropy.

The system we consider, with Q queues in tandem, is depicted in Figure 1. We use G/D/1-FIFO queues to represent network devices [4, 5]. Each queue has two input classes. Input class 1 carries through traffic and input class 2 carries cross traffic. Each cross traffic stream enters a particular queue and departs the system after service by that queue. In the case of simultaneous arrivals we use a discipline which serves the cell from input class 1 first. To capture heterogeneous traffic, typical of high-speed networks, we consider traffic streams with varying amounts of correlation.

If we examine the interaction of multiple traffic streams at the micro level of individual cells, one observes both scattering and clustering of cells in the through traffic. It is this phenomenon we wish to describe using *entropy* as a measure of the "disorganization" among through traffic cells. We propose entropy as a new traffic descriptor. The entropy rate of a traffic process also provides an upper bound on the success achievable by a predictive system in predicting future traffic behavior based on past behavior. Alternative traffic descriptors, such as peak-to-mean ratio of clusters of cells, are very sensitive to the time interval over which the peak and mean sizes are calculated [6], and thus cannot serve as descriptors of "intrinsic" traffic properties. We discuss traffic descriptors further in §2.

Since we do not know the underlying joint distributions of the queue departure processes, we cannot compute the entropy analytically. Instead, we use an entropy estimation technique that is motivated by the Lempel-Ziv (LZ) universal data compression algorithm. This algorithm can compress a data sequence derived from any finite alphabet. The *compression ratio* is given by the ratio of the length of the compressed string over the length of the original string. It is well known that the LZ algorithm asymptotically achieves a compression ratio proportional to the entropy of the dataset [3]. We are thus motivated to use LZ to determine the entropy of a dataset. The basic LZ method, and the variations on LZ compression, when used to compute entropy rather than compress data suffer from large error terms that are a function of the size of the dataset. Based on a recent result in [9], we have implemented an entropy estimator which has a small constant error term. For a detailed comparison of these methods see [8].

Other studies that investigate properties of the departure process of through traffic from a simple queue with interspersing cross traffic are reported in [2] and

Appeared in: IEEE Infocom Proceedings, June 1994

[7]. In [2] they study the amount of distortion in the queue output process relative to the input process for individual departure classes. The distortion is measured in terms of percentage change in relative burst sizes and inter-burst periods. They find conditions under which the distortion is negligible and the output is practically the same as the input process.

Ohba et al. [7] find the stationary distribution for the interdeparture intervals of through traffic streams. Their description assumes that successive departure instants are independent. We are interested in the correlation structure of departure processes and define a distance metric which allows us to compare the difference between the correlation structures of two traffic processes. We observe how the distance metric changes at successive queue outputs. They also consider a system with four tandem queues and study the behavior of the coefficient of variation of departure processes and average end-to-end delay, as a function of load and cross traffic type. In our results, we find cases of input traffic in which the average queue delay decreases in successive tandem queues (which concurs with their findings), but in addition we also find cases of input traffic in which the average queue delay increases in successive queues.

In this paper we examine the entropy of successive departure processes. We have developed an implementation of the entropy estimator and written the simulator code all in C. Our simulator is fast and can be used to examine the behavior of thousands of tandem queues. In §2 we discuss ATM traffic models, traffic descriptors and the scattering and clustering phenomenon. In §3 we present the entropy estimator. The accuracy of the estimator is discussed in §3.1. The simulation results on the entropy, the queue behavior and the correlation structure at tandem queue outputs are presented in §4, §5, §6, respectively. We conclude with suggestions for future research in this area in §7.

2 Traffic

We consider both an "on-off" bursty traffic generator and and a Bernoulli traffic generator. The former is a two-state discrete time Markov Chain, as depicted in Figure 2. In state 1 we generate a cell and in state 0 we generate an empty slot. A cluster of back-to-back cells, with no spacing between them is called a *burst*. The burst length B and idle period length I are geometric random variables with parameters p and q respectively; $E(B) = \frac{1}{p}$ and $E(I) = \frac{1}{q}$. Bursty traffic streams generated this way exhibit correlations over successive slots. We generate traffic

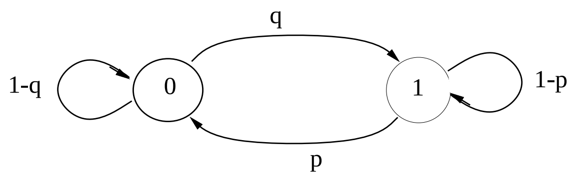

Figure 2: On-Off Traffic Model

streams with varying amounts of correlation by adjusting p and q. Note that the correlations here are only among the cells of a given burst; there is no correlation between different bursts. In a Bernoulli traffic stream, the probability that a slot contains a cell is p and the probability that a slot is empty is $1 - p$. The Bernoulli traffic is a special case of the on-off model when $q = 1 - p$.

Because we consider a system with fixed sized time slots which may or may not contain a cell, a traffic stream can be represented by a binary sequence of 0's and 1's, where a '1' indicates the presence of a cell and a '0' indicates an empty slot. This observation provides us with a representation of traffic streams for which it is then possible to apply entropy estimation techniques.

The cells in the through traffic can either be scattered or clustered as they traverse each queue. We say that a group of cells become more *scattered* (*clustered*) when the distances, measured in time slots, between individual cells increase (decrease), respectively. Examples of cell scattering and clustering are given in Figures 3 and 4. The figures depict snapshots in time. The input streams and queue depict the state of the system at some time t. The output streams depict the through traffic at a later time, after all the cells in the example have been served. The cells in the through traffic are numbered, at both the input and the output, to show the change in their relative positions. In the cell scattering example, the queue is empty immediately before the arrival of cell #1. Since the cross traffic cells leave the system after service, they effectively create *holes* in the original input stream, scattering the initial group of back-to-back cells. In the cell clustering example, the queue has 7 cells in it immediately before the arrival of cell #1. During the next 6 time slots, cells numbered #1-#4 will arrive and be positioned back to back in the queue. After they have all been served they will be clustered together, with no holes between them.

In the initial input to a tandem queue system, the beginning and ending boundaries of a burst are well

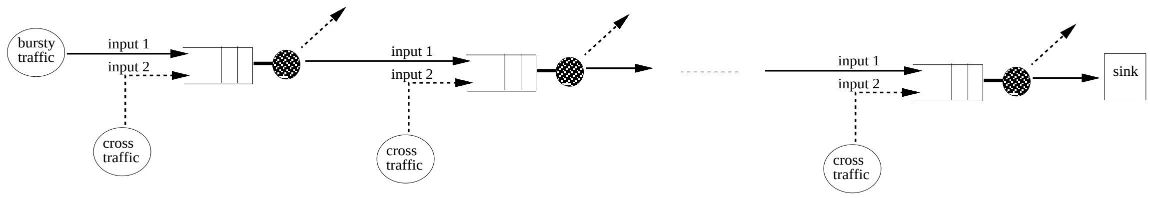

Figure 1: Tandem Queueing System

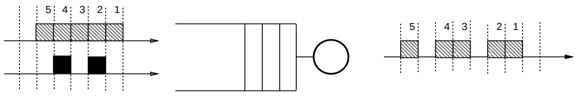

Figure 3: Cell Scattering

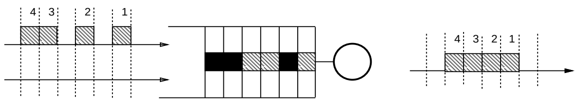

Figure 4: Cell Clustering

defined. Due to the scattering and clustering of cells in such a system, these boundaries can be lost entirely in output processes *inside* the network. Due to this blurring of burst boundaries and the creation of small holes within bursts, the notion of burst is ill-defined inside the network. The entropy descriptor can capture this notion of cell scattering and clustering without being tied to any notion of burst. The entropy descriptor is one dimensional and also captures, although indirectly, frequently occurring patterns in traffic streams. However the entropy descriptor is limited in that different underlying traffic distributions can generate the same entropy.

3 Entropy

We briefly review some basic definitions. (For a full coverage see [3].) The entropy is a measure of the uncertainty of a random variable. The entropy of a random variable X with a probability mass function $p(x)$ is defined by

$$H(X) = -\sum p(x) \log_2 p(x) \tag{1}$$

We use logarithms to base 2, and thus measure the entropy in bits. The *entropy rate* (in bits per unit time) of a stochastic process $\{X_i\}$, $i = 1, 2, \ldots$ is defined by $H = \lim \frac{1}{n} H(X_1, X_2, \ldots, X_n)$ when the limit exists where $H(X_1, X_2, \ldots, X_n)$ is defined as in (1) with $p(x)$ replaced by the joint probability distribution $p(x_1, x_2, \ldots, x_n)$. For a stationary Markov chain, the entropy rate is given by $H(X_i/X_{i-1})$, or $H(X_2/X_1)$. Theorem 4.2.4 in [3] states that for a stationary Markov chain, $\{X_i\}$, with stationary distribution μ and transition matrix P, the entropy rate is

$$H = H(X_2/X_1) = -\sum_{ij} \mu_i P_{ij} \log P_{ij} \tag{2}$$

Using equation (1) the entropy of a Bernoulli traffic source with load p is given by

$$H(p) = -p \log p - (1-p) \log(1-p) \tag{3}$$

Using equation (2) the entropy of our bursty on-off traffic source is

$$\frac{p}{p+q} H(q) + \frac{q}{p+q} H(p) \tag{4}$$

Although we can find, analytically, the entropy of the traffic processes input to our tandem queueing system, we do not know the distribution of the underlying stochastic processes $\{X_i\}$ at the queue outputs. Therefore we estimate the entropy of these processes from empirical data produced in simulations, using the following entropy estimation method. Let m denote the length in bits of a of a data sequence. (In our application the data sequence comes from our 0-1 traffic streams.) We call the first n bits, where $(n < m)$, the *database*. Starting with the $n+1$st bit, we look along the data sequence until we find the longest string which also exists in the database. This is called the *longest match*. The longest match, L, is a random variable with $1 \leq L \leq n$. For example, consider the database 01000111011 and the input sequence 00101011011. The sequence of matches found is 001,010,11011. More formally, consider the random process $X_{\{-n+1\}}^{\infty}$, where $X_i \in \{0, 1\}$. For any fixed positive n, let $D_n = X_{\{-n+1\}}^{0}$ be the database. We look along the input sequence, $X_1, X_2, \ldots$ sequentially, looking for the longest match in the database. Let L_i denote the ith longest match found. The result in [9] states

$$\left| E(L_i) - \frac{log n}{H} \right| \leq O(1) \quad (5)$$

This implies that we can calculate the entropy of a dataset by computing the longest matches. It is required that the database be formed from the first n elements of the random process, because the theorem only holds if the distribution of the database is the same as the distribution of the input sequence. In addition, the result requires that the $\{X_i\}$ process be stationary, ergodic and Markov. Although the error term, $\frac{O(1)}{\log n}$, can be reduced by increasing the size of the database, we cannot increase n arbitrarily. We are constrained in our choice of n by the memory required and the running time of the estimator algorithm. The implementation of our algorithm is described in [8] and has a running time of $O(n+m)$. At the time of this writing, little is knows about the $O(1)$ error constant; however based on experience as described in the following section, it is believed to be very small.

3.1 Accuracy of the entropy estimator

To examine the accuracy of this entropy estimator, we observe its performance on two processes, namely the Bernoulli and the on-off traffic processes, whose entropies are known analytically (equations (3) and (4), respectively). First we empirically generated a Bernoulli traffic stream with parameter p. In Figure 5, we plot both the estimated entropy ($\hat{H}$) and the true entropy (H) versus p. We plotted $\hat{p}$ on the abscissa, which is the actual coin bias produced by the random number generator. This figure shows that the error is largest for the maximum entropy case, $p = 0.5$. For $0.2 < p < 0.8$, the entropy is biased on the low side, and for $p < 0.2$ and $p > 0.8$, the estimator is slightly biased on the high side.

For the datum in Figure 5 a database of 20,000 bits and an input of 30,000 bits was used. For a given process, the error is a function of the database size n, the input size m and the process parameter(s). On one hand, it turns out we want to choose m and n large enough to reduce errors, but on the other hand we want m and n small enough to minimize the running time and memory space used by the algorithm. We need m large enough to generate a sufficient amount of data to achieve a 98% confidence interval in our estimate. We also need n large because, as we will see, our estimator has an inherent bias, which can be partially reduced by increasing n. Intuitively, this occurs because the database must contain sufficiently many bit patterns to represent the true process entropy and generate a meaningful $E(L_i)$.

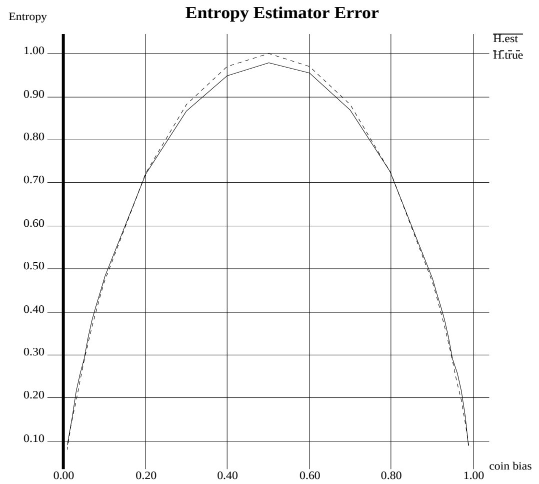

Figure 5: Entropy Estimator Error for Bernoulli Traffic

We choose the smallest n and m which are large enough to minimize the average RMSE for the Bernoulli process which generated the worst case error, namely $p = 0.5$. With $n = 20,000$ and $m = 30,000$ the $RMSE = 0.001$. Increasing either n or m did not reduce the RMSE any further. For this n and m we obtained $H - \hat{H} = 0.021$ (where H denotes the true process entropy and $\hat{H}$ denotes our estimate) which indicates that the estimator itself has an inherent bias.

We performed the same test for on-off traffic streams. The error is now a function of m, n, p and q. In order to bring the RMSE down to 0.001 we needed to increase m to $200,000$. In this case the inherent bias (for worst case p and q) was 0.027. Hence we chose $n = 20,000$ and $m = 200,000$ in all our simulation runs. For a $RMSE = 0.001$ we can be 98% confident that the true mean of our estimate lies within ± 0.0022 of the sample mean. Even though the traffic inside the virtual circuit is neither Bernoulli nor an on-off stream, these tests give us some confidence that the error in our entropy estimator will be small.

4 Entropy properties

Since we are only interested in the properties of the through traffic we use the term *input* to denote input stream 1 and the term *output* to denote departure class 1. Four combination of input types were tested and are indicated in Figures 6 - 9. The notation $EB/EI = x/y$ specifies the average burst size EB, and the average idle period size EI. The load for the on-off model is thus given by $\frac{EB}{EB+EI}$. The notation x_load indicates that the cross traffic is a Bernoulli process with parameter equal to x_load.

In Figures 6 and 7 both the through traffic load and the cross traffic load are fixed at 0.4 each. The cross traffic streams represent streams of increasing correlation. In Figure 6 the entropy grows at successive queue outputs, whereas in Figure 7 the entropy decreases at successive queue outputs. Furthermore, the entropy is sensitive to the amount of correlation in the cross traffic. For Bernoulli cross traffic we see in Figure 8 (9) that the entropy increases (decreases) at successive queue outputs, respectively. These graphs demonstrate that the entropy of a single departure class can either increase or decrease depending upon the input types. In Figures 8 and 9, the cross traffic load was varied. From all these figures we see that opposite effects for uncorrelated inputs and highly correlated inputs occur, for a given cross traffic stream.

For a through traffic load of 0.4, the maximum entropy possible occurs when the process is Bernoulli, $H(0.4) = 0.971$ (see equation (3)). Clearly from these figures the through traffic doesn't achieve "maximal randomness" within the first thirty queues. Thirty queues is certainly an upper limit on a typical number of queues in a virtual circuit.

5 Queue behavior

Here we first examine the average queue behavior of successive queues and then examine the relation between entropy and average queue size. For each of the two cases considered in Figures 10 and 11, we tested 6 cases of cross traffic starting from cross traffic with no correlation up to correlations lasting over 100 cells. In both cases, this data indicates that the average queue process is sensitive to the amount of correlation in the cross traffic (the load is the same in all tests). One of the results in [7] shows that average queue sizes decrease in successive queues. Our results demonstrate that actually average queue sizes can increase or decrease with successive queues depending upon the types of input traffic.

To observe the average queue process in relation to the entropy process we consider a single case of correlated cross traffic, namely $EB/EI = 40/60$. We consider two through traffic types, one whose entropy increases in tandem queues with correlated cross traffic (i.e., bursty), and one whose entropy decreases in tandem queues with correlated cross traffic (i.e., Bernoulli). The entropy for this test is plotted in Figure 12 and the related average queue size is plotted in Figure 13. For bursty through traffic an increase in entropy occurs in conjunction with a decrease in average queue size. The opposite effect (decreasing entropy with increasing average queue size) can be seen in the case of Bernoulli through traffic. The same behavior occurs for the other specific cases of cross traffic. If one could express the rate of increase (decrease) of H as a simple function of the rate of decrease (increase) of average queue size for case of bursty (Bernoulli, respectively) through traffic, then then entropy could be said to "track" the average queue process. These results are encouraging as they indicate that the entropy measure may be useful in the design of high-speed telecommunications networks (i.e., resource allocation, traffic shaping).

6 Correlation structures

Another common type of traffic descriptor is one in which the correlations among successive cells is captured. The correlation coefficient $\rho(k)$ is given by

$$\rho(k) = \frac{E(X_i X_{i+k}) - E(X_i)E(X_{i+k})}{\sqrt{Var(X_i)Var(X_{i+k})}}$$

where X_i is defined as in §3. The notation "Qn" in Figures 14 and 15 denotes the output process of the nth queue. For the case studied in Figure 14, in the initial input $Q0$ all the correlations were zero, in the output processes we see that some correlations have been introduced. The correlations are small, and only exist over 4 contiguous slots. The correlation between any 2 contiguous slots is negative indicating that a cell is more likely to be followed by an empty slot than another cell.

For the case studied in Figure 15 we see that there is more correlation over a length of 80 or 90 in later output processes than in the initial bursty input. This occurs because the holes introduced by the cross traffic effectively spread the burst out over a greater number of slots. Once there are holes in the burst, new cross traffic cells can essentially go through the system unfelt, if they temporarily occupy those holes. This explains why the bulk of the change in the traffic stream occurs during the first few queues, and subsequent changes are smaller.

In comparing Figures 14 and 15, we observe that the correlation structure of bursty through traffic is approaching the correlation structure of Bernoulli through traffic as the number of queues grows. In order to measure the "distance" between these corre-

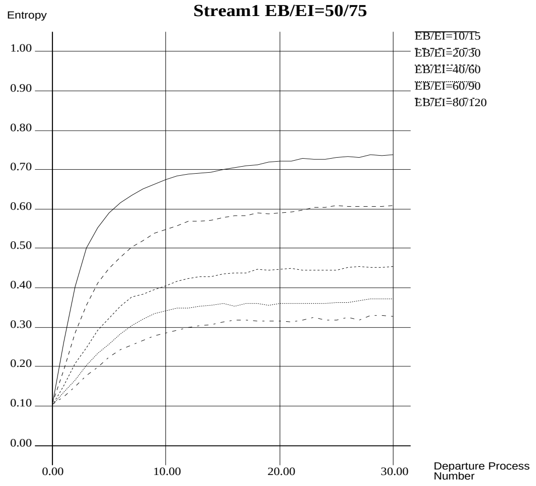

Figure 6: Bursty through traffic, Bursty cross traffic.

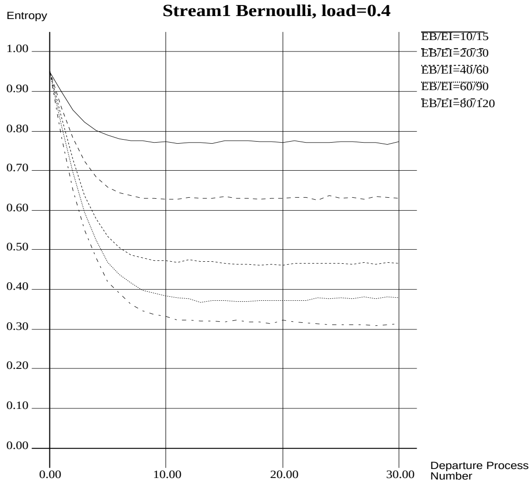

Figure 7: Bernoulli through traffic, Bursty cross traffic.

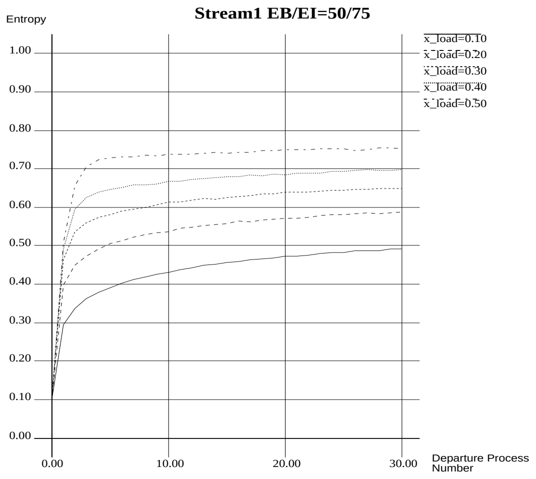

Figure 8: Bursty through traffic, Bernoulli cross traffic.

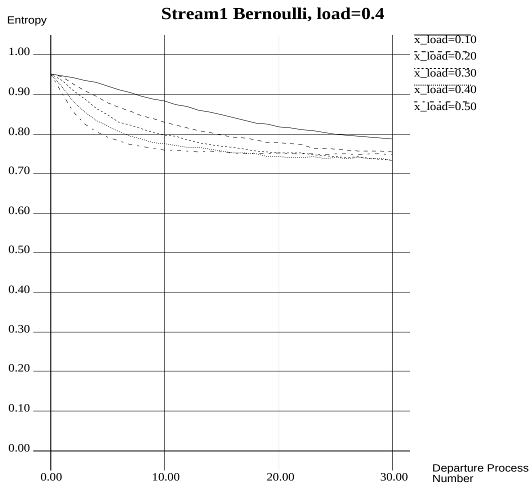

Figure 9: Bernoulli through traffic, Bernoulli cross traffic.

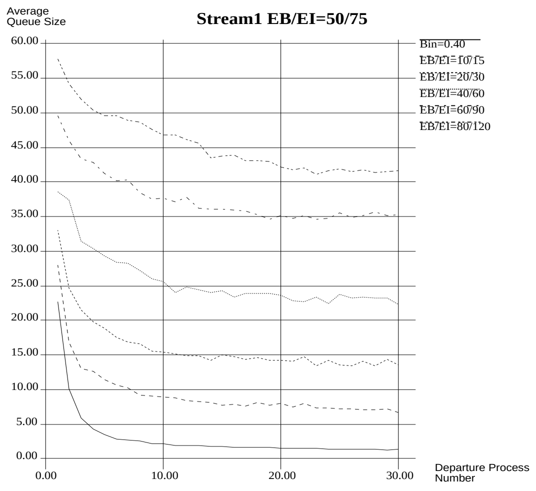

Figure 10: Bursty Through Traffic

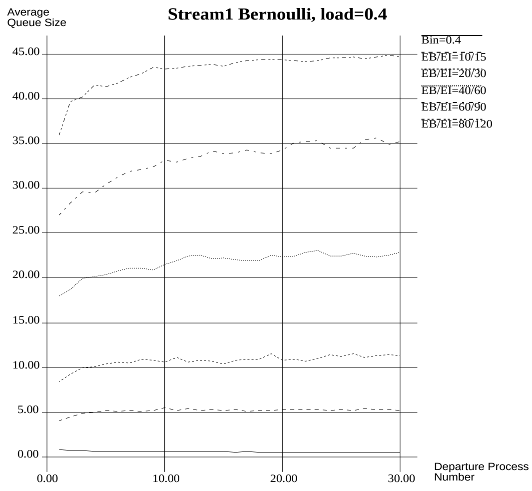

Figure 11: Bernoulli Through Traffic

Appeared in: IEEE Infocom Proceedings, June 1994

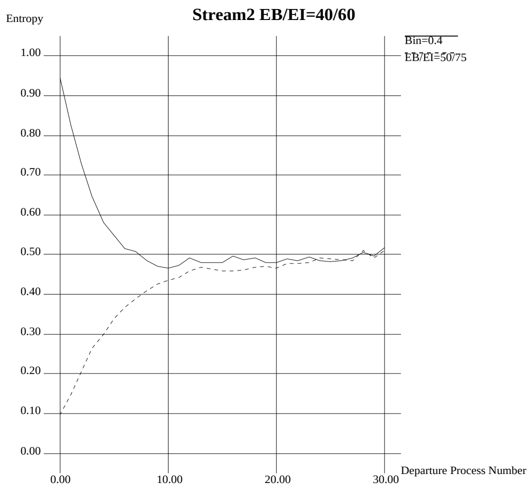

Figure 12: Entropy. For Figure 13.

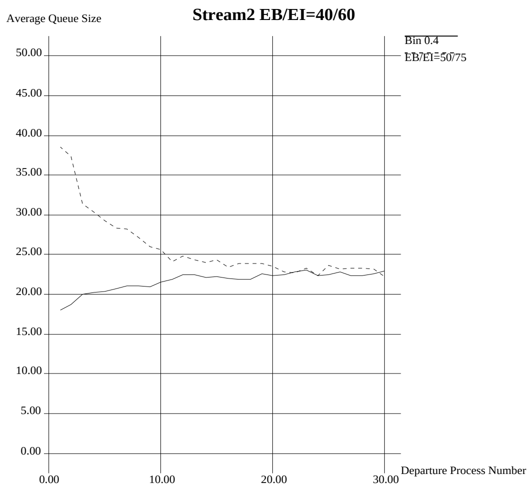

Figure 13: Average Queue Size. For Figure 12.

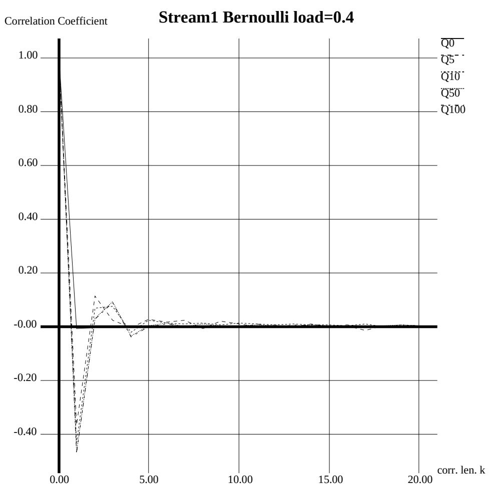

Figure 14: Bernoulli through traffic, Bernoulli cross traffic.

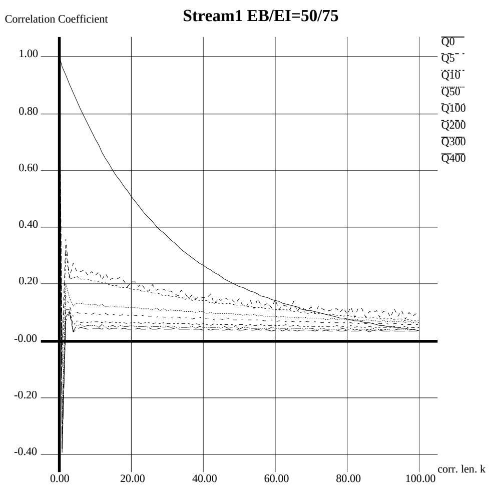

Figure 15: Bursty through traffic, Bernoulli cross traffic.

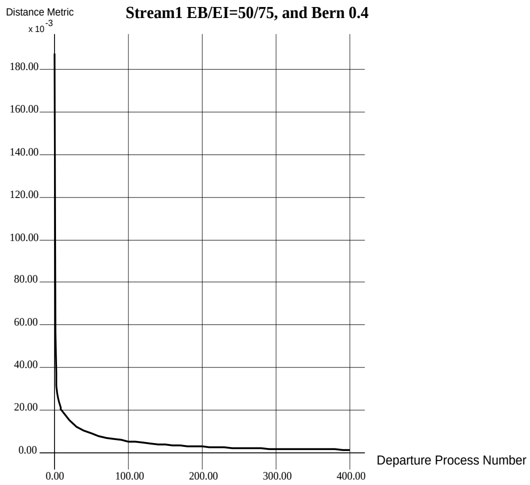

Figure 16: Distance Metric for Correlations

lation structures, we use the following metric.

$$D = \frac{\sum_{k=0}^{l} [\rho_1(k) - \rho_2(k)]^2}{\sqrt{\sum_{k=0}^{l} \rho_1(k)^2 \sum_{k=0}^{l} \rho_2(k)^2}}$$

where l is the length of a traffic stream, $\rho_1(k)$ denotes correlations of length k in the Bernoulli through stream, $\rho_2(k)$ denotes correlations of length k in the bursty through stream, and the denominator is a normalizing constant such that $0 \leq D \leq 1$. In Figure 16 we plotted this distance metric for the first 400 queues. At the input the two streams are 18% different with respect to this measure. They are 2% different at the 10th queue, and 1% different by the 40th queue.

7 Summary

The focus of this study has been twofold. First, we have implemented an entropy estimation algorithm to compute the entropy of a binary dataset. We used this algorithm to study the entropy of queue departure processes for various combinations of correlated and uncorrelated inputs. Second, we studied the behavior of traffic entropy, average queue length and traffic correlation structures at successive queue outputs.

We have presented an initial step in the use of information theoretic ideas in characterizing network traffic and queueing behavior. More experimentation on the linkage between entropy and queue length properties is of interest. Characterizing aggregate streams in which each individual stream is characterized by its entropy and the resulting queueing behavior is a direction of future research. In order to better understand the usefulness of the entropy measure for network design (e.g., resource allocation or traffic shaping), a study of the relation between entropy and other quality of service parameters such as delay jitter, sustainable rate and burst tolerance is needed.

Acknowledgements

The authors wish to thank Abraham J. Wyner for many helpful discussions on entropy.

References

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

[2] Wing cheong Lau and San qi Li. Traffic Analysis in Large-Scale High-Speed Integrated Networks: Validation of Nodal Decomposition Approach. *IEEE Infocom Proceedings*, March 1993.

[3] T. Cover and J. Thomas. *Elements of Information Theory.* John Wiley & Sons, Inc., 1991.

[4] M. Hluchyj and M. Karol. Queueing in High-Performance Packet Switching. *IEEE JSAC*, December 1988.

[5] Joseph Hui. *Switching and Traffic Theory for Integrated Broadband Networks.* Kluwer Academic Publishers, 1990.

[6] W. Leland, M. Taqqu, W. Willinger, and D. Wilson. On the Self-Similar Nature of Ethernet Traffic. *ACM-SigComm Proceedings*, September 1993.

[7] Y. Ohba, M. Murata, and H. Miyahara. Analysis of Interdeparture Processes for Bursty Traffic in ATM Networks. *IEEE Journal on Selected Areas in Communications*, April 1991.

[8] Nina T. Plotkin. *High-Speed Network Traffic: Charaterization and Control.* PhD thesis, University of California at Berkeley, 1994.

[9] Abraham J. Wyner. *String Matching Theorems and Applications to Data Compression and Statistics.* PhD thesis, Stanford University, 1993.