Working Paper Series
ISSN 1170-487X

Current Techniques for Measuring and Modeling ATM Traffic

by: Murray Pearson, John Cleary, Brian Unger and Carey Williamson

Working Paper 96/12

June 1996

Department of Computer Science
The University of Waikato
Private Bag 3105
Hamilton, New Zealand

Table of Contents

List of Figures

Current Techniques for Measuring and Modeling ATM Traffic

Murray Pearson and John Cleary
Department of Computer Science, University of Waikato, New Zealand

Brian Unger
Department of Computer Science, University of Calgary, Canada

Carey L. Williamson
Department of Computer Science, University of Saskatchewan, Canada

1. Introduction

ATM has now been widely accepted as the leading contender for the implementation of broadband communications networks (Brinkmann, Lavrijsen, Louis, *et. al,* 1995) ATM networks are no longer restricted to research laboratories, and commercial products such as switches and interfaces manufactured by well known computer and communications companies have started to appear in the market place. The main advantage seen in ATM over other broadband networking technologies such as Synchronous Transfer Mode (STM) is its ability to transmit a wide variety of traffic types, including voice, data and video efficiently and seemlessly.

Asynchronous Transfer Mode (ATM) (Minzer, 1989) is a method by which fixed-length units called cells are individually relayed to a destination. Each ATM cell is 53 bytes in length, which includes a 5 byte header. The header includes two addressing fields: a Virtual Path Identifier (VPI) and a Virtual Channel Identifier (VCI). Other header fields include the payload type and cell priority. An error-correcting code protects the header from errors. ATM is able to support a wide assortment of services such as voice, video, and data all over the same network. The fixed size of cells and the simple header layout facilitate very fast hardware switching, making ATM ideal for high speed fiber optic networks. ATM is a connection-oriented technology. A connection (called a virtual channel) must be explicitly set up end-to-end before any data cells are transmitted. Once set up, the same path is used for all cells of the connection, with each cell carrying the same VPI/VCI. The ATM specification requires that cells be delivered in order on each VCI, as there is no sequence numbering or retransmission provided at the ATM cell level.

The wide variety of traffic types that ATM has been designed to carry can be categorised into two main groups: real time and non-real time traffic (Habib and Saadawi, 1994). Real time traffic, such as voice and video, has strict time delay and delay variability requirements that must be met, but can tolerate a small cell loss rate. Non real time traffic, on the other hand, can tolerate time delay but cannot tolerate cell loss. These differing and contradictory requirements of different traffic types combined with the estimated sizes of future ATM networks promise to make the task of network design and management an extremely difficult one. It will be important for network designers and engineers to be able to assess and predict traffic performance of current and future networks to ensure that they are used and implemented in the most economic way. Bad decisions may lead to networks that are excessively expensive and/or perform badly adversely affecting users of the network. To support the design and maintenance of these ATM networks will require the development of efficient and accurate network planning tools.

In addition to the problems associated with network design and management, a number of research issues involving ATM have yet to be resolved including scheduling, error control, switch design, multicasting, guaranteeing a quality of service level and the performance of TCP/IP applications over ATM (Unger, Gomes, Zhonge et. al., 1995) Most of these problems cannot be adequately addressed through analysis and will require the development of accurate and validated cell level simulation tools.

A vital component of these simulation tools will be the development of high fidelity traffic models as the outputs from a simulation (used for network performance estimation) are highly dependent on the inputs provided to the model. Hence without realistic traffic source models, the simulation results are of little or no value (Arlitt, Chen, Gurski and Williamson, 1995), (Frost and Melamed, 1994). Classical techniques used to model traffic in previous networks have not been found to hold for broadband networks such as ATM. The most striking reason behind this effect is that traffic sources for packetised data-flows reveal bursty and correlated statistical behavior affecting system performance in a crucial way (Kuehn, 1995).

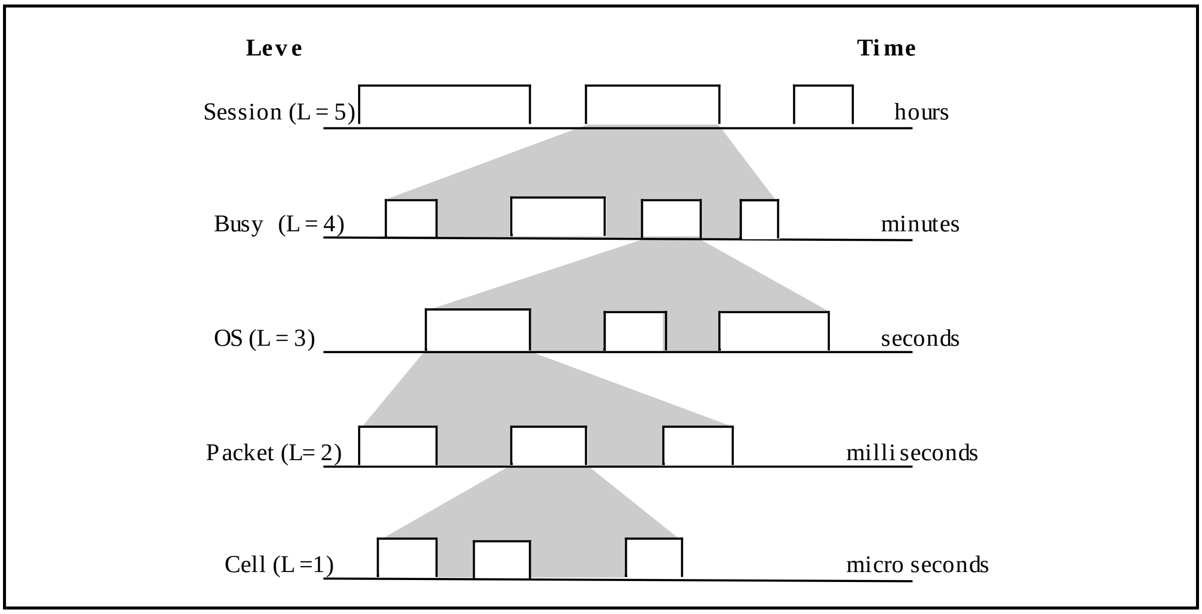

Figure 1 Multi-Layer States of Terminals

Burstiness is present in a traffic process if arrival points appear to form visual clusters on the time line. Furthermore, in many of the traffic processes where this bursty behavior is present, these bursts can be seen at multiple levels. For example, Figure 1 shows network traffic generated by a single workstation on a network at five different time scales. Over the period of a day a user may use the workstation for several sessions each of an hour or more interleaved with breaks to perform other duties (Session level, L =5, in Figure 1). During the breaks no significant network traffic is generated. Within a session there will be periods of time where a user is performing network intensive activities such as large file transfers and other periods where no network traffic is being generated such as word processing (Busy level, L = 4, in Figure 1). Each of these busy periods is made up of periods of activity followed by idle periods while the workstation is waiting for a resource (e.g., a disk drive) (OS Level, L = 3, in Figure 1). For each OS level burst, several bursts of TCP/IP packets are generated followed by breaks while packets are acknowledged (Packet Level, L = 2, in Figure 1). Lastly, in the case that the workstation is attached to an ATM network each of these IP Packets will generate a burst of ATM cells (Cell layer, L = 1, in Figure 1).

ATM exploits this bursty behavior by using a statistical multiplexing access method to increase the efficiency of links that have multiple connections switched through them as shown in Figure 2. Using statistical multiplexing, bandwidth on the shared link is allocated to a connection only when it has data to send through the shared link. To make efficient use of the shared link the aggregated peak bit rate of all the connections routed through the link may exceed the link capacity provided the average bit rate of the aggregated traffic stream does not. As the average bit rate will be less than the maximum bit rate this will potentially leave periods of time where the aggregated bit rate exceeds that of the shared link. To cope with these periods buffering techniques are used.

The size and design of these buffers is crucial to the performance of an ATM network and involves a tradeoff between cell loss rates and cell delays. When buffers are too small, excessive numbers of cells will be lost. In the case of data traffic (e.g. file transfers) the lost cells have to be retransmitted. Whereas with video traffic these cell losses result in a degradation of the quality of the image. As the size of buffers increases the cell delays through a switch also tend to increase as cells can potentially spend more time being buffered. For some types of traffic (e.g. video or voice) it is far more important to deliver cells within a specified time and failing to do so can have devastating effects on the quality of the output at the destination.

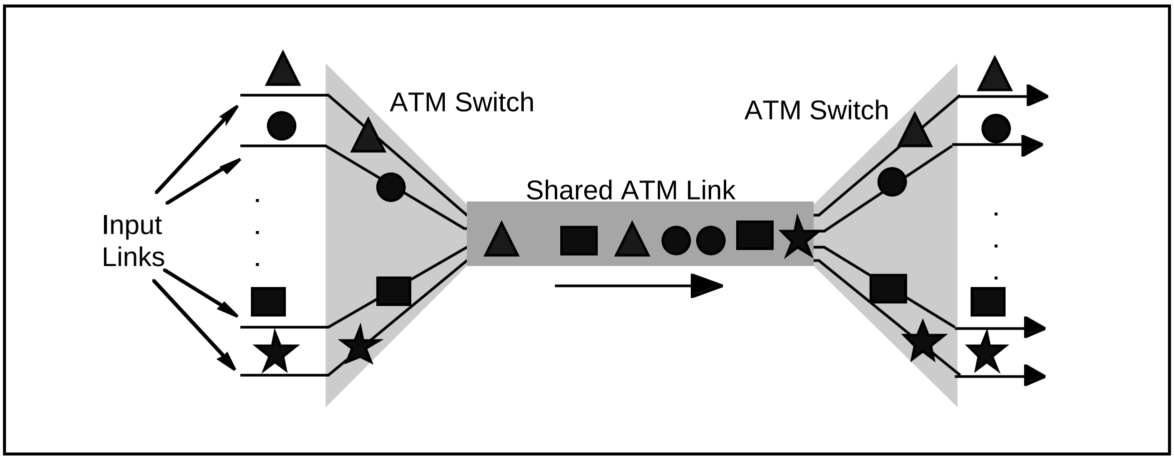

Figure 2 Statistical Multiplexing on ATM

Statistical multiplexing works well in situations where there is no statistical correlation between traffic bursts on each of the aggregated connections. In this case as the number of connections through the shared link increases the behavior of the resulting aggregated traffic tends to smooth out and become less bursty. This leads to more predictable behavior of the aggregated channel and greatly simplifies traffic management. In particular, while increasing buffer sizes does increase cell delays a much larger (exponential) decrease in cell losses is seen.

However in situations where traffic processes that show bursty and correlated statistical behavior are aggregated together, the resulting aggregated stream tends to result in even more bursty and statistical correlated behavior. This results in large variations in the aggregate bit rate making traffic management far more complex. The effect of increasing buffer sizes, and hence cell delays, produces a far smaller decrease in the cell loss rates.

As can be seen from the above discussion, different types of bursty traffic cause very different effects on traffic flows through an ATM network. If simulations of ATM networks are going to be meaningful then it is crucial that the traffic source models developed to drive these simulations are accurate. Furthermore different types of traffic exhibit very different behaviors so it is important to have a range of traffic models, with traffic mixes varying to match those of the network under study.

In this report three specific traffic types have been chosen as representative: Ethernet local area network (LAN) traffic, MPEG (Motion Pictures Experts Group) compressed video, and Internet WWW traffic. Video traffic is important because it is one of the most bandwidth-intensive applications that ATM networks are being used for. Of the different types of video, variable bit rate (i.e. compressed) video has been identified as being important because of its large and widely varying bandwidth requirements.

One of the most widely used current applications of ATM networks is backbone ethernet traffic. Important issues that need to be considered are the effects of aggregating several ethernet networks on links of an ATM network. In particular ethernet traffic has been identified as being self similar or fractal in nature, meaning that there is no natural length of traffic bursts. This self similar behavior will have major ramifications in the design of ATM. WWW traffic is considered to be important as it is currently the fastest growing and largest consumer of bandwidth on the Internet. Currently the number of WWW servers connected to the internet is doubling every six months to a year and is showing no signs of abating in the near future. Hence WWW traffic is likely to be one of the major contributors of traffic on future ATM networks.

The first stage in the development of a traffic model is to characterise the behaviour of the traffic it is to model . This usually involves collecting and analysing traffic measured on actual networks. To date most traffic measurements made to characterise different traffic sources have been obtained using existing LAN and WAN networks. While this data can provide great insight into the characterisation of a particular traffic source some care needs to be taken as results obtained by measuring traffic on real ATM networks may vary greatly. For example, differences in available bandwidths and latencies on ATM networks are different from existing technologies, which may alter the way people use them. However as test equipment capable of measuring traffic on ATM networks becomes more readily available, more experiments that involve collecting traffic off actual ATM networks are being conducted. One such set of such experiments being conducted at the University of Waikato, using modified network interface cards and PCs to provide a cheap traffic measurement system capable of collecting traffic at multiple points of an ATM network simultaneously. Further details of this work can be found in (Cleary, Pearson, Graham & Unger, 1996).

Another issue that needs to be addressed when designing traffic models is the detail that is going to be modeled for a particular traffic source. The designer of a traffic model must be aware of the trade-off between detail and simulation execution time (Frost and Melamed, 1994). For example, consider the simulation of an ATM network to estimate the cell loss ratio (CLR), which has the acceptable value of 10^{-7}. In order to have confidence in network dimensions, significant measures of CLR need to be made requiring simulation of between 10^9 and 10^{10} cells. Even using simple traffic models and small networks this implies very significant computational demands making it practical to simulate only network functions that have an appreciable impact on the desired performance metrics. For example most traffic carried on an ethernet is likely to be TCP/IP based, yet it is currently likely to be impractical to include TCP/IP detail in an ethernet model using existing network simulators.

While it may be impractical to model TCP/IP accurately in an ethernet or other traffic models, the behavior of TCP/IP on ATM networks is of great interest to network designers and researchers alike. In particular, the performance of TCP/IP traffic has been found to be very poor in some circumstances due to; a mismatch in TCP packet sizes and ATM cell sizes; and mismatch between operating system send and receive buffer scizes. For example a recent set of experiments measuring IP traffic have shown almost zero throughput on an ATM network (Manthorpe, 1995). For this reason a number of traffic models have been developed to model TCP/IP traffic on ATM networks and one is described later in this report.

As part of the Telesim project (Unger, et al, 1995) a state-of-the-art high performance simulator called ATM-TN has been developed. The simulator incorporates a number of ATM traffic source models including an aggregated ethernet model, an MPEG model, a Mosaic client/server model and a TCP/IP model. These models will be used as examples to illustrate current modeling techniques for each of the traffic types that are discussed in this report.

The rest of this report is orgainised as follows. Sections 2 - 4 discuss issues of modeling the three traffic sources, ethernet, video and WWW that were identified as being important. Section 5 gives details of the issues associated with modeling TCP/IP traffic. Finally, section 6 gives a set of conclusions from this review along with suggestions of future work that needs to be carried out in developing and validating models for ATM traffic sources.

2. Video Traffic

Video applications will be one of the largest consumers of bandwidth (e.g., 1-10 Mbps for a single compressed video stream) on high speed networks, and thus are an important class of applications to model. Furthermore, video applications are both delay sensitive and loss sensitive; if the cells are lost, the quality degrades, and if cells arrive too late for their playback point, the effect is the same as if the cells were lost.

Video streams have traditionally been transmitted using a fixed bandwidth in synchronous transfer mode (STM) circuit switched networks (Habib et. al., 1994). To improve the efficiency of transmitting constant bit rate (CBR) streams compression technique are used. When large portions of an image change from one frame to the next (e.g. scene changes) the bandwidth required to transmit these changes can exceed the fixed bandwidth of the channel. To cope with these situations, techniques that reduce the quality of the image and/or employ buffering are used.

The introduction of technologies such as ATM make it possible to use variable bit rate (VBR) coding techniques which offer the following advantages:

1. *improved transmission efficiency* - Bandwidth allocated at a particular time depends on the information content to be transmitted at that time. For example around scene changes, where most of the image is changing, large bandwidths are allocated. In the middle of a scene where little is changing much lower bandwidths are allocated
2. *constant video quality* - Bandwidths are varied to cope with differing amounts of information to be transmitted rather than altering quality.
3. *reduction in delay* - Again bandwidths are varied rather than buffering at the source.

VBR sources generate very bursty traffic with high bit rates for high motion. Early work on modeling VBR was based on ether very short sequences of empirical records consisting of a small number of scenes (often one) or conference. However these modeling attempts have not been found to capture the bursty and correlated statistical behaviour of full motion video. Two main factors contributing to traffic burstiness are (Frost and Melamed, 1994)

1. *marginal distribution of traffic*
2. *auto correlation function of traffic.*

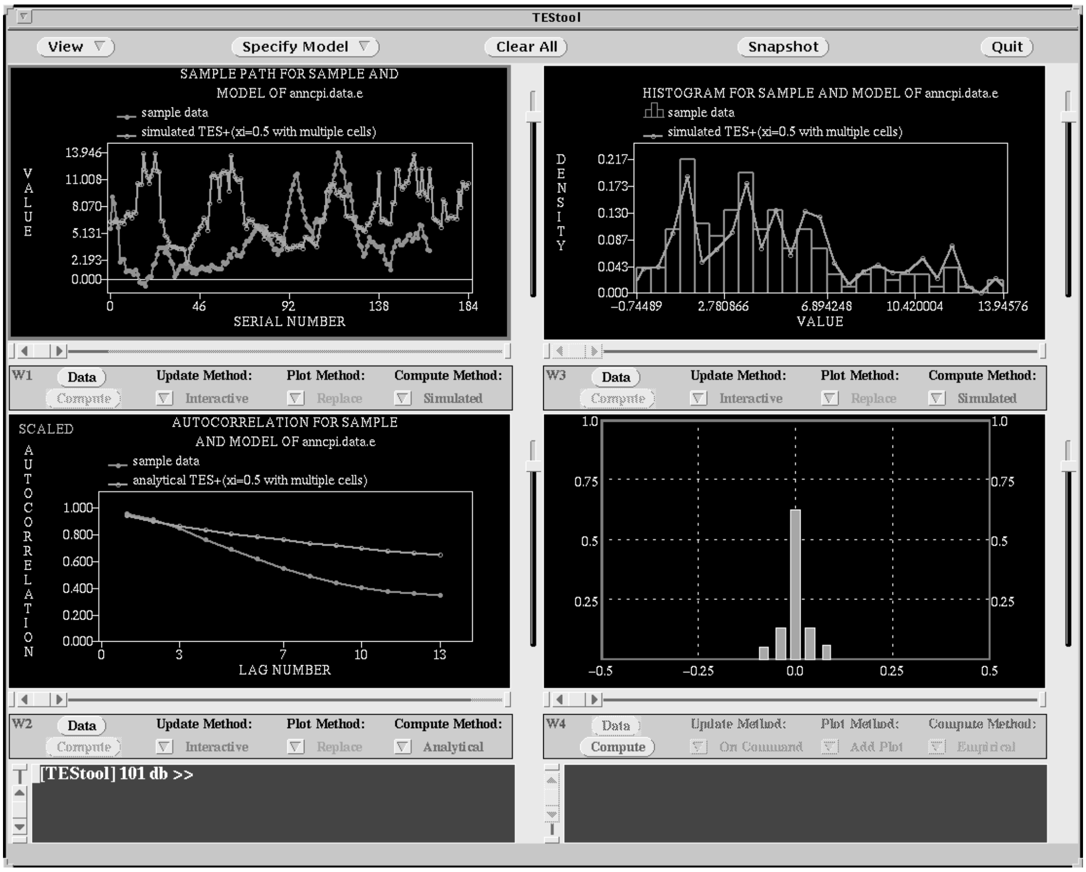

Figure 3 Snapshot of a typical TEStool screen

One modeling methodology that has been successfully applied to modeling bursty and correlated data streams such as video is the TES (Transform, Expand and Sample) methodology (Melamed and Hill, 1993). TES provides a very general approach to modeling first-order (i.e., frequency histogram) and second-order (i.e., autocorrelation function) statistics of time series. Using TES the modeller aims to fit a prescribed marginal distribution and autocorrelation function via a stationary process. This involves the modeller in a heuristic search attempting to fit the statistics of TES models to their respective empirical counterparts.

The construction of TES models is computationally intensive and requires software support to be carried out effectively. A computer based environment called TEStool has been developed to support the TES modeling methodology (Melamed and Hill, 1993). It uses state of the art visualisation techniques and allows the modeller to construct and modify TES models then superimpose the corresponding TES statistics back on their empirical counterparts.

Figure 3 contains a screen snapshot of a typical TEStool screen composed of four canvases. On the top left canvas is displayed a sample path generated for the TES model which has been superimposed on a sample path of the empirical data. The top right and bottom left canvases show histograms and autocorrelation functions respectively for the TES model superimposed on the counterparts for the empirical data. Finally the bottom right canvas allows the visual specification of a TES model. Once a change has been made to the specification a new TES model is created and the resulting sample path, histogram and autocorrelation function are updated to the appropriate canvases. This provides an interactive modeling environment that allows the user to iteratively attempt to find a TES model that best matches the empirical data. A major advantage of the TES methodology is that it is possible to calculate the statistics directly without having to rely on lengthy simulations. This considerably shortens the loop of specifying and evaluating TES models against an empirical counterpart.

The use of TEStool has been likened to “playing an arcade game” where the player is able to concentrate on the animated display without having to worry about the underlying software details. Users of TEStool have reported that it speeds up the modeling process and alleviates much of the tedium of repetitive search.

2.1 MPEG-Compressed VBR Video

MPEG (LeGall 1991) defines a standardised method for compressing full-motion video for storage as digital data. The MPEG standard is intended primarily for use with stored video applications (e.g., entertainment video), but it can also support some limited real-time (i.e., “live") video applications, such as desktop video-conferencing, depending on the compression hardware available. Because the encoding of an MPEG video sequence usually takes much longer than decoding, a video sequence is usually encoded only once and stored in compressed form. Playback can then be done as many times as desired. The decoding of MPEG video sequences can be done in real time.
The MPEG video stream consists of a sequence of images called frames that are displayed one after the other, at short periodic intervals (e.g., 30 frames per second). The MPEG standard defines three types of compressed frames:

- **I:** (Intraframe) frames are encoded using only the image data available in the current frame. Since I frames represent a complete image, they provide an absolute reference point for the other two image types in the MPEG sequence, which are encoded using interframe coding (i.e., they express the relative differences from one frame to the next).

I frames are compressed using Discrete Cosine Transform (DCT), similar to the JPEG (Joint Photographic Experts Group) standard for single image compression. I frames take the least time to compress, but have the least compression.

P: (Predictive) frames In addition to DCT compressed data, contain motion-compensated data predicted from the preceding (I or P) frame. P frames take longer to encode than I frames, are faster to decode than I frames, and achieve higher compression than I frames.

B: (Bidirectional interpolative) frames are similar to P frames, but they contain motion-compensated data from both the previous and the next frame (I or P). B frames take the longest time to encode, but offer the highest levels of compression.

An MPEG sequence then consists of a pattern of I, P, and B frames, called a Group Of Pictures (GOP). The GOP pattern must be specified at the time of encoding, and must start with an I frame. The same encoding pattern is then used repeatedly for the duration of the video sequence, with each new I frame starting a new GOP. An example of a typical MPEG sequence is: `IBBPBBPBBIBBPBBPBBI`: where the GOP pattern is `IBBPBBPBB`.

Significant correlations (i.e., autocorrelation and cross-correlation) have been observed in the sizes of the MPEG frames in a video sequence (Pancha and El Zarki 1994).

2.1.1 The ATM-TN traffic model

The MPEG video traffic model (Arlitt, Chen, Gurski and Williamson, 1995) in ATM-TN simulates the cell-level ATM network traffic generated by an MPEG video stream when transmitted to a viewer. The model generates a given combination of I, P, and B frames at a set frame rate. The MPEG video sources are modeled using the TES modeling methodology.

Three separate TES processes are used - one for each type of MPEG frame. The marginal distribution (i.e., frequency histogram) and autocorrelation function of frame sizes (done separately for I, P, and B frames) are the key inputs to the TES modeling process. These inputs have been used to derive a separate TES model for each frame type. Each of these models is saved in a TES model specification file that is used by the MPEG source model. The three frame types are then interleaved as determined by the GOP pattern. Dependence between frame types (as has been empirically observed in MPEG sequences) is created by duplicating the TES background variate of the I frame at the beginning of each GOP. The resulting cross-correlation between frame sizes increases the accuracy of the (Melamed, 1992; Garret and Willinger, 1994).

Thus the final MPEG model is a composite TES model. The model has been parameterised based on empirical studies of MPEG sequences (i.e., using MPEG data provided by Melamed). The MPEG model uses four parameters to characterise simulated MPEG traffic:

GOP: The structure of the GOP needs to be specified. This specification indicates the number of frames in a GOP, the number of P frames in a GOP, the number of B frames in a GOP, and the interleaving of the I, P, and B frames in the GOP.

The GOP structure is specified using two integer parameters, N and M. N specifies the number of frames in the GOP. M specifies the length of the substructure in the GOP (i.e., after each I frame or P frame, there are M - 1 consecutive B frames before the next I or P frame). N must be an integer multiple of M.

Changing the GOP changes the average bit rate and burstiness of the MPEG sequence. For example, a GOP with mostly B frames will have a lower average bit rate than a GOP with just I frames and P frames. Note that simple desktop video conferencing applications (which often use just JPEG compression for each image) can be modeled with a GOP pattern of I (i.e., N =1, M = 1), producing only I frames in the MPEG sequence. The default GOP pattern is IBBPBBPBB (i.e., N =9, M = 3).

Frame Rate: The frame rate specifies the number of frames per second to be transmitted and displayed by the MPEG sequence. This is usually 30 frames per second (the North American NTSC standard) or 25 frames per second (the European PAL standard).

Smoothing: The smoothing parameter controls the actual transmission of the ATM cells for each image in the MPEG sequence. Since the exact size of an image is known prior to its transmission, and the deadline for delivering the entire image to the viewer is known (e.g., 33 msec for NTSC), two transmission schemes are possible for the traffic source. In the first approach, the source might transmit all the cells of the current image at the peak rate of the network , to get the cells to the destination as fast as possible. In the second approach, the source might spread out the transmission over the interval, reducing the peak rate of transmission, but still getting all cells to the destination on time. The choice of strategy has a significant effect on the workload characteristics presented to the network.

The smoothing parameter is an integer in the range [0..N] where N is the length of the GOP in frames, that specifies the desired transmission behavior, A value of 0 means no smoothing. That is, cells are transmitted at the maximum link rate immediately after each frame is generated (i.e., the worst-case stress test of bursty traffic for the network). A value of 1 causes cells of each

	individual frame to be transmitted evenly over the entire frame time. A value of *N* causes cells of all frames in the GOP (which are generated and known in advance of transmission) to be transmitted evenly over the entire GOP time. Values between 1 and N result in the cell transmissions of groups of frames being spaced evenly over the corresponding frame times.
Scaling:	The scaling parameter allows the simulation user to adjust the network load by linearly scaling the frame sizes of the MPEG sequence, and thus the mean bit rate of the MPEG source. The scaling parameter is a real number in the range [0..1]. The default value of the scaling parameter is 0.8, corresponding to a mean bit rate of approximately 1.4 Mbits/sec.

2.1.2 Validation and Testing

The validation experiments for the MPEG model involved testing each of the parameters in turn, using a one-factor-at-a-time experimental design.

2.1.3 GOP structure

Figure 4 shows a plot of the output of the MPEG model using the default GOP pattern (IBBPBBPBBI). As can be seen from the plot this pattern clearly manifests itself in the frame sizes of the video sequence as a repeating pattern of large, small and medium sized peaks corresponding to I frames, B frames and P frames respectively. The outputs of simulations using different GOP structures revealed the corresponding changes in the jaggedness of the output.

Reducing the GOP structure to only contain I frames produced the output shown in Figure 5 where the jaggedness has almost been completely removed as expected. However a cyclic pattern is evident in this output (detected by viewing the sequence over a longer time period) which is unlikely to be present in real MPEG sources, revealing a weakness of the TES model used. Further work is require to determine the implications of this behavior on the accuracy of future ATM-TN simulation results.

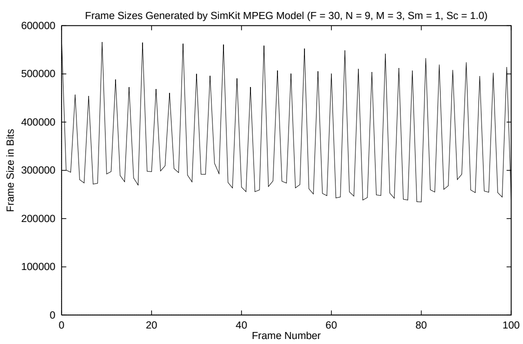

Figure 4 Frame sizes for MPEG model with default GOP (N=9, M= 3)

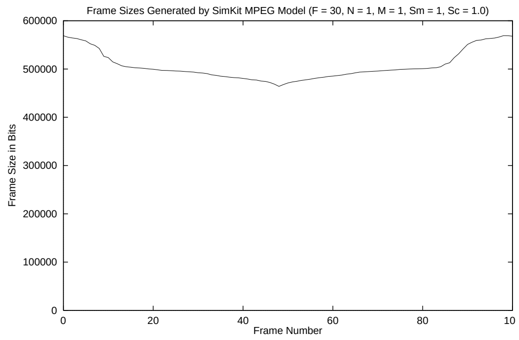

Figure 5 Frame sizes for MPEG model with only I frames (N= 1, M = 1)

2.1.4 Scaling and Frame Rate

Scaling and frame rate are two parameters that can be used to adjust the average bit rate of the MPEG video source. Simulations that used different values for these parameters produced the expected changes in the output in terms of the average bit rates.

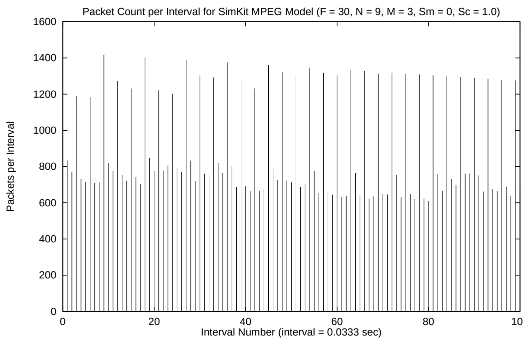

Figure 6 Packets per time interval for default MPEG GOP at 30 fps, no smoothing

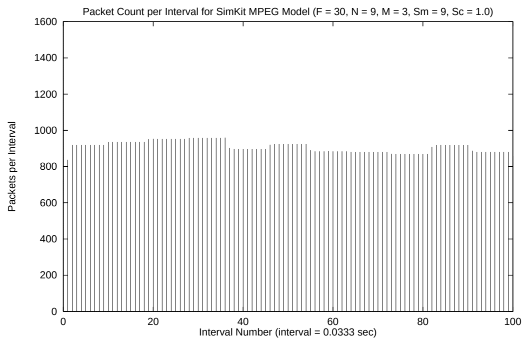

Figure 7 Packets per time interval for default MPEG GOP at 30 fps, smoothing = 9

2.1.5 Smoothing

The smoothing parameter can be used to alter the actual transmission of the cells in the frame. Figure 6 shows when the parameter is zero and Figure 7 shows the result when the smoothing parameter is 9. The peaks are significantly truncated, as they are transmitted over several frame times. These results were found to be consistent with the MPEG smoothing results discussed in (Pancha and El Zarki, 1994)

3. Ethernet Traffic

Currently one of the most widely used applications of ATM is to carry backbone traffic between existing ethernet LANs. Early modeling studies of Ethernet performance made many simplifying assumptions, such as assuming that the packet arrival process is Poisson. The results of some recent measurement studies, however, have shown that traffic is very much not Poisson (Leland et al. 1994; Paxson and Floyd 1994). In particular, Ethernet LAN traffic has been found to have a “fractal" self-similar behavior in the sense that there is no natural length of a burst: that is there are significant traffic bursts evident at time scales ranging from milliseconds to seconds to minutes to hours.

To demonstrate this self similarity pictorially Figure 8 depicts a sequence of simple plots of packet counts (i.e. number of packets per time unit) verses time from a set of traffic measurements that monitored a segment of the Bellcore ethernet network for 27 consecutive hours in August 1989 (Leland et al. 1994). These plots show the same data plotted using 5 different choices of time units. Starting with a time unit of 100 seconds (Figure 8 (a)), each subsequent plot is obtained from the previous by increasing the time resolution by a factor of 10 and concentrating on a randomly chosen sub interval (as indicated by the darker shade). The time unit corresponding to the finest time scale is 10 milliseconds (Figure 8 (e)). Observe that all plots look intuitively very "similar" to one another (in a distributional sense) and are distinctively different from white noise (i.e., an independent and identically distributed sequence of random variables). Notice also the scaling property (y-axis) and the absence of a natural length of a "burst": at every time scale ranging from milliseconds to minutes and hours, bursts consist of bursty sub-periods separated by less bursty sub-periods. This scale invariant or “self similar” behavior is drastically different from conventional telephone traffic and stochastic models for packet traffic used traditionally. The latter typically produce plots of packet counts that are indistinguishable from white noise after aggregating over a few hundred milliseconds (Leland et al. 1994).

Mathematically speaking, a stochastic time series process $X = (X(t) : t = 0;1;2; : ::)$ is said to be self-similar with self-similar Hurst parameter H if the process is covariance stationary (that is, the process has constant mean and finite constant variance) and the corresponding aggregated process has the same correlation structure as the original process (exactly self-similar) or agrees asymptotically with the correlation structure of the original process (asymptotically self-similar) over large intervals. The Hurst parameter expresses the behavior of the rescaled adjusted range statistic (R/S statistic) for large sample sizes n (Leland et al. 1994). For many naturally occurring time series, including Ethernet traffic, $E[R(n)=S(n)]$ is proportional to n^H , with H “typically" about 0.7. For models with only short-range dependence, H is 0.5. This discrepancy is called the Hurst effect (Leland et al. 1994).

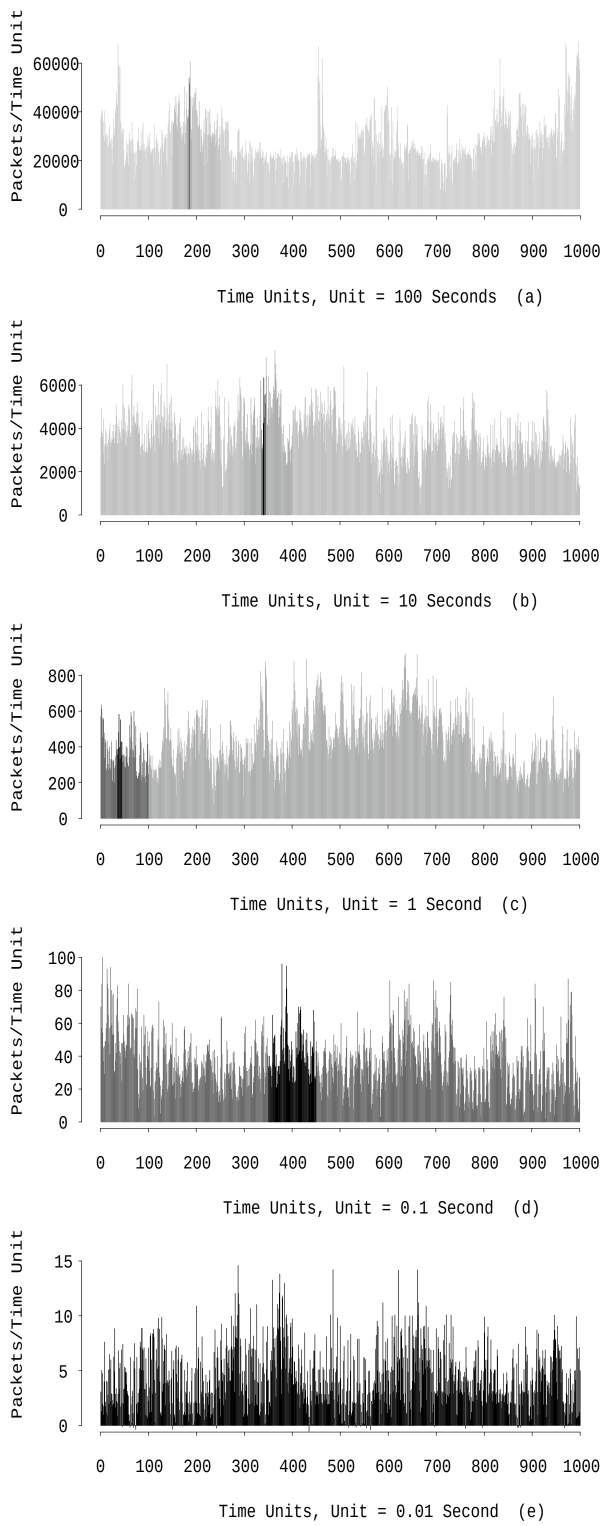

Figure 8 Pictorial "proof" of self-similarity: Ethernet traffic on 5 different time scales. (Different gray levels used to identify the same segments of traffic on the different time scales (Leland et. al., 1994)

Self-similarity manifests itself in a number of ways. One manifestation is a slowly decaying variance. That is, if you take the original time series X and compute a new smoothed time series by averaging every non-overlapping group of m observations to get a new time series $X^{(m)}$, then the variance of the sample mean for the new time series $X^{(m)}$ decreases more slowly than the sample size m (i.e., $\mathrm{Var}(X^{(m)}) \approx am^{-\beta}$ as $m \to \infty$, where a is a constant, and $0 < \beta < 1$). On a log-log plot of the variance of $X^{(m)}$ versus m, the result is a straight line with (negative) slope β. The Hurst parameter H can be estimated using $H = 1 - \frac{\beta}{2}$. A second manifestation is a hyperbolically decaying autocorrelation function of the form $AF(k) \approx ck^{-\beta}$, for some constant c. If X is self-similar, then the autocorrelation function for $X^{(m)}$ is indistinguishable from the autocorrelation function for X, for any value of m. A third manifestation is a straight line of slope β on a log-log plot of the R/S statistic.

3.1 Effects of self-similar traffic on ATM networks

Ethernet traffic has been found to have this self similarity property, and the degree of self-similarity tends to increase with the load level on the ethernet (Leland et al. 1994). Furthermore, the self-similarity property has been found to hold both in the traffic within a LAN, as well as in the traffic leaving a LAN destined for the Internet.

One of the important effects observed with aggregating multiple streams of correlated traffic together is that the resulting aggregated stream does not smooth as predicted by traditional traffic models. Instead the bursty behavior (i.e. the level of self-similarity) is found to increase. Previous studies that have compared the results of simulations based on actual traffic measurements with those based on traditional traffic models have produced very different results. In particular, overall packet losses decrease very slowly with increasing buffer capacity, in sharp contrast to Poisson driven models where losses decrease exponentially fast with increasing buffer size (Leland et al. 1994). In addition they also found that packet delay always increases with buffer capacity, again in contrast to the traditional models where the delay does not exceed a fixed limit regardless of buffer size. These properties of self-similarity also explain results of a measurement experiment performed by (Fowler and Leland, 1991) who observed the ineffectiveness of buffering to manage congestion and went on to observe that when congestion occurs, losses are severely concentrated and are far greater than the background loss rate.

In short, the self-similarity property is pervasive enough to merit its inclusion in a traffic model for aggregate LAN packet traffic. The next section describes the ATM-TN traffic model (Chen, Deng and Williamson, 1995) for self-similar ethernet traffic.

3.2 The ATM-TN Ethernet Traffic Model

The Ethernet LAN traffic model (Chen, Deng and Williamson, 1995) in the ATM-TN TeleSim project is designed to represent the workload characteristics of aggregate data packet traffic on existing local area networks, such as campus LANs. Since one of the anticipated uses of ATM is as a backbone interconnect between existing LANs, LAN-LAN traffic may form a significant fraction of the background load on an ATM network. The purpose of this traffic model is to model the aggregate LAN data traffic workload as accurately as possible. The model produces a unidirectional flow of ATM cells, with statistical characteristics matching those of aggregated traffic.

The first step of the modeling process used an S (Venables 1990) program to generate a pseudo-self-similar time series based on Fast Fourier Transform inversion. The S program produces as output a zero-mean time series with self-similarity controlled by an input parameter H (the Hurst parameter). Time series were generated for five different values of H (namely, H = 0.5, 0.6, 0.7, 0.8, and 0.9). The output time series from the S program were then translated and scaled (within S) to produce non-negative Ethernet packet count time series with a specified mean network utilization U. These synthetically generated time series were then used as input to the second step of the modeling process.

The second step of the modeling process used the TES (Melamed 1992) modeling methodology. In this case, the time series of interest is the “packet count per interval" time series from the first step above. The TEStool (Geist and Melamed 1992) program was used to construct an abstract TES model that had the same frequency histogram and autocorrelation function as the (synthetic) Ethernet time series. One TES model was constructed for each value of H considered. The resulting TES models were converted (automatically) to C ++, and incorporated into the ATM-TN model.

The Ethernet traffic model in the ATM-TN TeleSim project has two parameters:

H: The Hurst parameter 0:5 < H <1 expresses the degree of self-similarity in the traffic. Data sets with higher self-similarity parameter H have higher “burstiness", in the intuitive sense.

Empirical measurements of Ethernet traffic (both in (Leland et al. 1994) and at the University of Saskatchewan) suggest that the value of H for aggregated LAN traffic is $0.7 < H < 0.9$.

U: The network utilization $0 < U < 1$ determines the average load level offered by the Ethernet LAN, as a fraction of its total 10 Mbps bandwidth. Most ethernets operate at low to moderate levels of load (e.g., 1% to 30% utilization).

The utilization U has been empirically observed to have an effect on the degree of self-similarity observed in the traffic (Leland et al. 1994). The higher the network load, the higher the degree of self-similarity usually observed. Despite this observation in the literature, the Ethernet model in the TeleSim project treats H and U as two independent and orthogonal parameters.

3.2.1 Model Validation

To validation the ethernet model a set of simulation runs were carried out to generate a set of synthetic ethernet traces, which were then analysed. Each of the simulations used different values for the network load and Hurst parameters. Using H = 0.5 the behavior observed was consistent with the results using traditional modeling methods which do not take the self similar behavior into account. With $H = 0.9$ the traffic generated by the model remains bursty across three or four orders of magnitude in the time domain, similar to Leland, et. al. (1994).

3.2.2 Performance implications

Using the validated ethernet model a simple experiment was conducted to assess the performance implications of self similar traffic in terms of cell delay, cell loss and buffer requirements for a simulated ATM network. The network consisted of a single simulated (shared memory) switch with two input ports and two output ports. A single Ethernet traffic source was used to feed cells into the ATM switch using a Permanent Virtual Channel (PVC). All Ethernet cells were destined for a low bandwidth output link of the switch (1.5Mbps) The switch had B buffers at the output port for this link where cell queuing and cell loss may occur. Experiments were repeated for five different settings of the Hurst parameter for the traffic source, and for six different settings of the output port buffer size B ranging from 10 to 1000.

Figure 9 shows the cell delay results for this simulation model. For this experiment the number of buffers at the output port was fixed at B = 100. Figure 9 shows that the average queue size (in cells) at the switch output port increases with link utilization as expected. Queue size results are also shown for five different values of H. The larger the value of H the greater the degree of self similarity in the traffic, and the larger the average queue size at the switch. Thus self-similar traffic implies higher average queuing delays within the network.

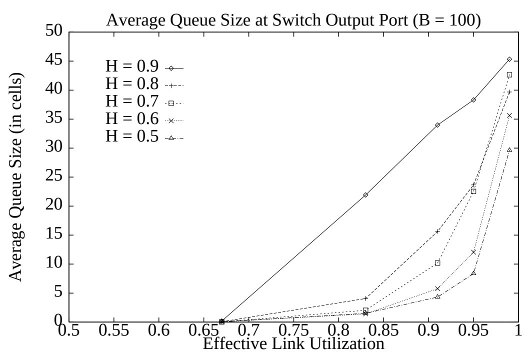

Figure 9 Average Queue Size at Switch Output Port for Ethernet Traffic (Chen, et. al., 1995)

The results of these experiments also show how self-similar traffic affects cell delay variation (i.e., jitter). Figure 10 shows the standard deviation of queue size at the switch output port for the same experiment as in Figure 9. The variation in the queue size increases with load and increases with *H*. Thus the presence of self-similar traffic on a network can potentially affect the cell delay variation (jitter) of other traffic streams using the network.

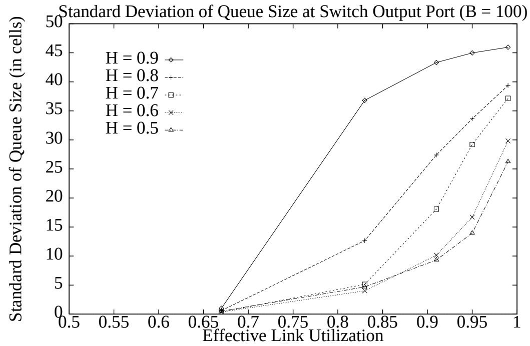

Figure 10 Standard deviation of queue size at switch output port for ethernet traffic (Chen, et. al., 1995)

Self-similar traffic has serious implications on cell performance in the network as well. Figure 11 depicts the average cell loss ratio experienced by the Ethernet traffic source, as a function of the number of buffers at the switch output port. These experiments are all conducted at a high link utilization (90%). Figure 11 shows that adding buffers at the output port always reduces the cell loss ratio, as expected. However, additional buffers are much less effective when the traffic source has a high degree of self-similarity. For example, for $H = 0.5$, increasing the number of buffers from $B = 10$ to $B = 100$ reduces the cell loss ratio from about 10^{-2} to below 10^{-4}. For $H = 0.9$, on the other hand approximately 1000 buffers are needed to bring the cell loss ratio below 10^{-2}.

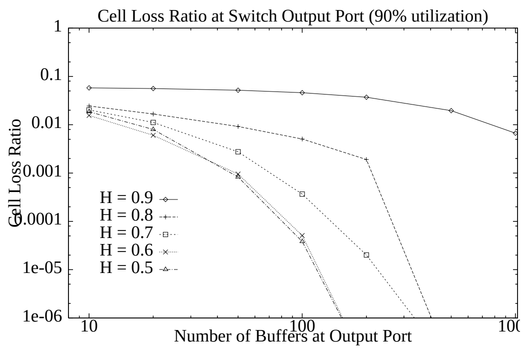

Figure 11 Cell loss ratio as a function of buffer size Traffic (Chen, et. al., 1995)

As can be seen from the results from these experiments (which agree with observations made in earlier studies), the effects of self-similar traffic have potentially serious implications on ATM networks. Cell delay, cell jitter, and cell loss may be significantly higher than predicted by simple traffic models. For example, simple Poisson traffic models, as opposed to self-similar traffic models, may underestimate ATM cell loss by two or three orders of magnitude depending on the network utilisation. Self similar traffic models are thus a valuable tool in the arsenal of network simulation providing for “worst” case stress testing of ATM networks.

4. WWW Traffic

One of the most popular resource discovery tools for navigating the Internet are World Wide Web browsers applications (e.g. Netscape and Mosaic). These browsers provide an easy-to-use graphical user interface allowing users to browse the information resources of the Internet (e.g., technical reports, software, news groups, phone lists, images, and weather maps), using a wide array of document types (e.g., text, image, graphics, audio, video, hypermedia).

In particular, browsers allow users to navigate sites that are part of the World Wide Web The World Wide Web (WWW) defines a global naming convention for all documents that are part of the Web, a standard format for defining documents in the Web, and a standard protocol for exchanging documents between Web sites.

Navigation in WWW browsers is done by going from one WWW “page" to another. Each of these pages (also called “home pages" or “Web pages") is a hypermedia document that can contain a combination of text, graphics, images, audio, and video, as well as links to other pages. Users move from one page to the next using simple “point and click" on the links (which are usually highlighted blue text in mosaic). Each page has a globally unique name in the WWW name space, called a Uniform Resource Locator (URL). Navigation can be done one step at a time (hierarchically) through a series of pages using links, or can be done directly using a URL. An example of a URL is

http://alf.usask.ca/personal/Arlitt/home.html.

WWW pages are constructed using a language called the HyperText Markup Language (HTML). Since each page consists of one or more files, HTML defines which files should appear on the page, what format they are in, where they should appear, and which items should be highlighted as links to other pages. Links may lead to another file on the same site, or to another file at a different site. When specific WWW pages are requested by users, the pages are transmitted to other sites using a standard protocol called the HyperText Transfer Protocol.

The growth in popularity of the World Wide Web has been phenomenal. WWW traffic is now the largest and by far the fastest growing component of the aggregate packet and byte traffic on the current NSFNET backbone (NSFNET 1995) as shown in Figure 12. Another indicator of the rate of growth of WWW traffic is the increase in the number of WWW servers connected to the internet, which is currently doubling every six months to a year. as shown in Figure 13 (Gray, 1996). Understanding the workload characteristics of these WWW browser sessions is therefore deemed important in any simulation study of the Internet or future high speed networks such as ATM.

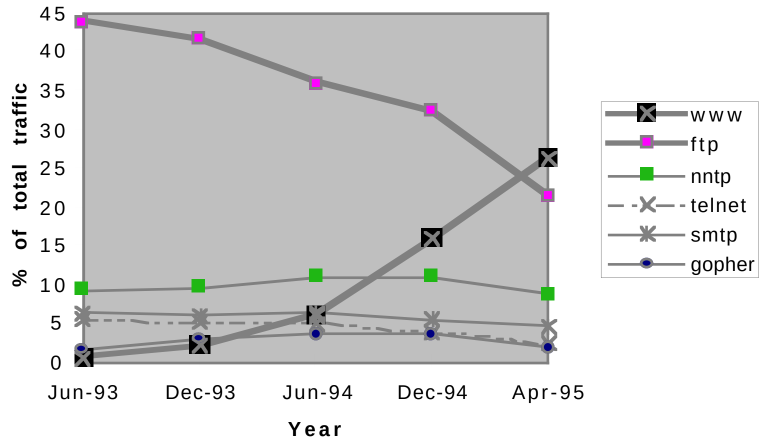

Figure 12 Percentage of total traffic on the Internet for the 6 largest contributors over time

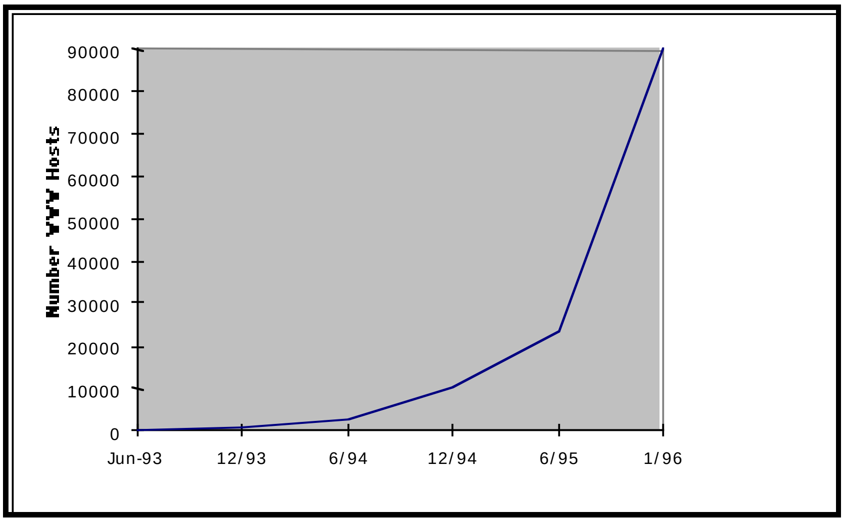

Figure 13 Estimate of the Number of WWW servers connected to the Internet between June 1993 and January 1996

The following section describes the workload characterisation, and modeling of the Internet WWW browser model (Arlitt and Williamson, 1995) that has been done as part of the ATM-TN Telesim project.

4.1 Design, implementation and validation of the ATM-TN model

In order to characterize the network packet traffic workload generated by Internet mosaic traffic, empirical network traffic measurements were collected at the University of Saskatchewan. Measurements were collected using techniques similar to those reported in the literature (Claffy 1994; Paxson 1994; Paxson and Floyd 1994). For all network traffic measurements, the trace collection facility was tcpdump (Jacobson et al. 1989). Tcpdump is a software network monitor that provides user- level control of trace collection, including filtering on a host, protocol, or port basis. The traces were set up to record only TCP packets with the SYN (start) or FIN (finish) bits set, and only the headers of these packets were recorded. These traces provide enough information to deduce the start time, end time, duration, and bytes transferred in each direction by each TCP connection, as well as the source host, destination host, and port number used. Two different sets of traces were used for workload characterization. The first set of traces recorded all TCP traffic destined from the campus Ethernet to the Internet, or from the Internet to the campus Ethernet. The traces were collected for a four day period in April 1994.

A total of 7070 WWW-related TCP connections were recorded in the first set of traces. These connections accounted for 97 Megabytes of data transferred on the network, for a mean of 14,380 bytes per TCP connection. While the total volume of mosaic-generated traffic is fairly low in this trace, the measurement results are quite consistent with those reported for the NSFNET at that time (NSFNET 1995). The second set of traces recorded only mosaic- generated activity for a single workstation on the campus Ethernet. Several traces were collected of mosaic sessions initiated from this workstation, ranging in du- ration from several minutes to several hours. A total of 1985 TCP connections were recorded in these traces. These connections accounted for 30 Megabytes of data transferred on the network, for a mean of 15,646 bytes per TCP connection. The combined traces have 9055 TCP connections with a mean size of 14,659 bytes. The largest number of data bytes transferred by a TCP connection was 2.2 Megabytes. Over 95% of the TCP connections transferred fewer than 42,000 bytes.

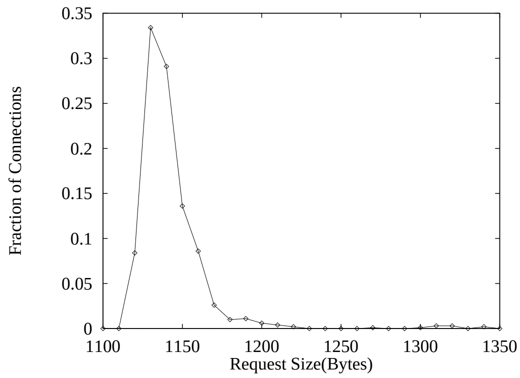

Figure 14 Distribution of bytes per TCP connection for WWW requests

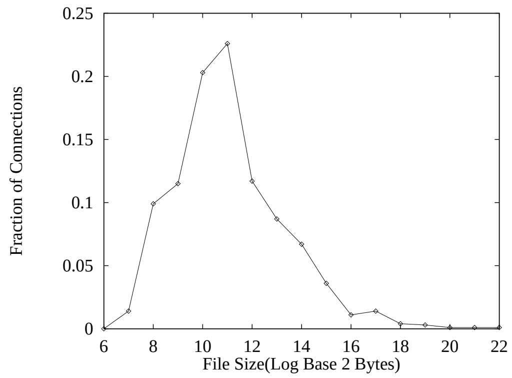

Figure 15 Distribution of bytes per TCP connection for WWW responses

Figure 14 shows the distribution of the number of bytes per TCP connection for requests for WWW pages and has a peak about 1130 bytes. Figure 15 shows the distribution of the number of bytes per TCP connection for the responses to requests for WWW pages. Note that the horizontal axis is logarithmic, showing the log (base 2) of the number of bytes. As can be seen in Figure 15 most of the files that appear on WWW pages are small. Approximately 50% of the files are in the 1-4 Kilobyte range, and over 90% of the files are less than 16 Kilobytes in size. Note, however, that there is a significant tail to the distribution, representing larger file sizes.

Figure 16 shows the distribution of the time duration of each TCP connection. The duration of a TCP connection is clearly correlated with the number of bytes sent by the connection. However, several other factors affect the connection duration as well, such as the geographical location of the destination host to which the connection is made, and the traffic load on the portion of the Internet traversed by the connection.

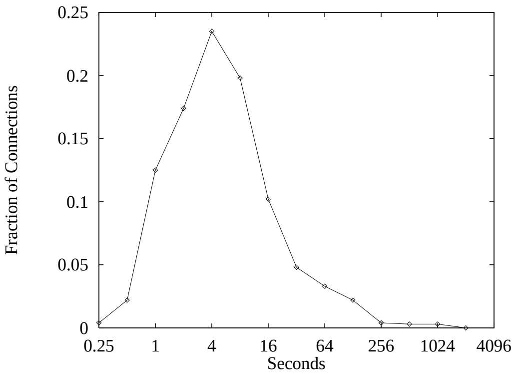

Figure 16 Distribution of TCP connection duration for WWW traffic

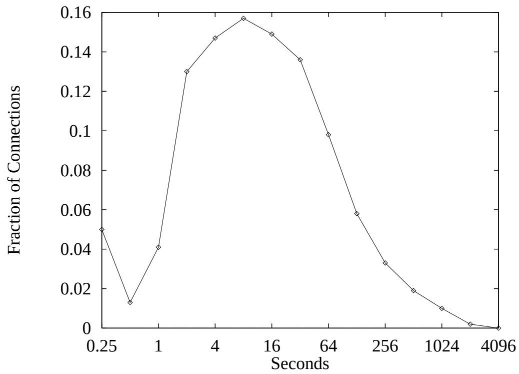

Figure 17 Distribution of TCP connection interarrival time for WWW traffic

Figure 17 shows the interarrival times between connections (i.e., the time from the start of one connection to the start of the next connection). These are obviously related to the connection duration’s shown in Figure 15. However, the connection interarrival time also depends on the “think time" between connections. This think time could be machine generated (e.g., to access the next file that is part of the current Web page, as soon as the previous file is completely received) or human generated (e.g., the user reads the current Web page and then decides which link to traverse next).

Figure 18 shows more clearly the connection gap times (i.e., the time between the end of one connection and the start of the next one). There are many short (machine- generated) connection gap times when there are multiple files for one Web page, and there are many longer (human-generated) connection gap times when new Web pages are selected. The dividing point between the two regions appears to be around 4 seconds. Further analysis of the latter (human-generated) region has found these connection gap times to be exponentially distributed and independent.

The above data was found to be extremely useful in constructing the synthetic workload model for internet WWW traffic.

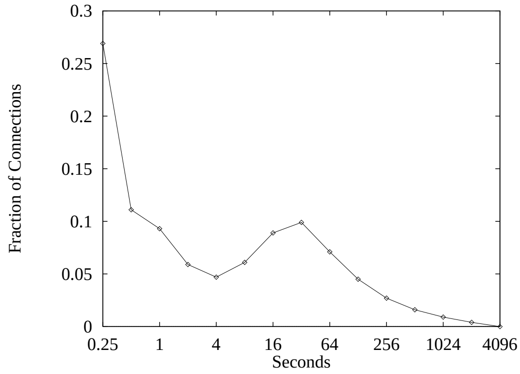

Figure 18 Distribution of TCP connection gap time for WWW traffic

4.1.1 Model Overview

The Internet mosaic traffic model is designed to represent the wide area network traffic workload characteristics generated by Internet mosaic users.

The key part of the mosaic traffic model is modeling a mosaic session (i.e., a single human user seated at a single workstation, using the mosaic graphical user interface to explore and search for information all over the Internet). A single mosaic session can generate one or more conversations with different hosts on the Internet, where a conversation consists of a bidirectional exchange of packets with a single destination host. A conversation, in turn, may consist of one or more TCP connections, where each connection involves a bidirectional exchange of packets with a specific host. Each connection transfers the data bytes of an information unit, such as a text file, an image, graphics, etc. There may be several successive connections to the same destination host as part of a conversation. The next conversation may use the same destination host again, or a different one.

A typical mosaic session may last minutes or hours. Within the mosaic session, the model generates the number of conversations, the arrival time for each conversation, the number of connections for each conversation, the arrival time for each connection, the destination for the connection, and the number of bytes to be transferred in each direction for the connection.

The mosaic session model has six parameters:

1. **Number of conversations per session :** This parameter specifies the average number of conversations initiated during one mosaic session. Because of the wide variability in the mosaic sessions of individual users, it is extremely difficult to determine a proper setting for this parameter. For simplicity, our model assumes that the number of conversations is geometrically distributed, with a mean of 50 conversations per session.

2. **Conversation arrivals:** Each conversation corresponds to a complete Web page. Once a conversation is complete, the arrival time of the next conversation must be determined. The next conversation may follow immediately after the previous conversation if the user is selecting successive

links in a list of items, or is traversing links that have been previously explored. The next conversation may follow after a pause if the user takes some time to read the current document before selecting the next link, or if the user leaves the session idle, to return to it later.

Empirical measurements have found both the conversation interarrival times (i.e., the time between the start of one conversation and the start of the next conversation) and the conversation gap times (i.e., the time between the end of one conversation and the start of the next conversation) to be exponentially distributed and independent. As a result, a simple Poisson arrival model is used for conversations. The mean conversation gap time in our model is 33 seconds.

3. **Conversation destination:** When a conversation is to be started, the destination for the conversation must be determined. Several factors must be considered when making the choice of the next destination.

   First, when a conversation is established with a particular destination, there is a strong possibility that there will be more conversations to the same destination within the near future. This “locality of reference" phenomenon, called temporal locality or persistence in the literature, is common in many areas of computer systems, and can be modeled quite easily with Least-Recently-Used (LRU) stack methods. Analysis of the empirical trace data shows strong persistence in conversations, but with only the topmost stack element seeing significant reuse. Thus persistence is modeled with a single real valued parameter DEST SAME PROB that specifies the probability of a conversation going to the same destination site as the last conversation. This parameter is set to 0.36.

   A second factor in determining the choice of the next destination is the distribution of network traffic workload across the Internet. Since mosaic makes it easy to connect to sites across the Internet, and individual users have their own preferences in what sites they visit, it seems reasonable to expect that mosaic will generate traffic widespread around the world. However, many studies of network traffic have identified the non-uniform distribution (i.e., concentration) of traffic on most networks. For example, since many users begin their mosaic sessions with the default NCSA home page, there will be many references to common hyperlinks. As another example, certain ftp sites on the Internet tend to be more popular than others. Standard techniques for modeling spatial and temporal locality have been used to incorporate these characteristics into the mosaic model.

4. **Number of connections per conversation:** This parameter specifies the average number of TCP connections per conversation. In our model, the number of connections per conversation is geometrically distributed with a mean of 2.5.

5. **Connection arrivals:** Each connection represents the transfer of one file, which may be part of a Web page (conversation). Within each conversation, the arrival time of each connection must be determined. The next connection may follow immediately after the previous connection (e.g., the graphics to appear on a Web page are transferred immediately after the transfer of the text of the page), or may follow after a pause (e.g., when the user selects a link to another page on the same server). Only the former are modeled with connection arrivals. The latter are modeled as conversation arrivals.

   As with the conversation arrivals, empirical measurements have found that connection gap times (i.e., the time between the end of one connection and the start of the next connection) are exponentially distributed and independent. Thus a simple Poisson arrival model is used for connection arrivals. The mean connection gap time in our model is 0.5 seconds.

6. **Bytes Exchanged per connection:** Finally the mosaic traffic model needs to select the amount of information that needs to be exchanged in each direction. Empirical measurements show that traffic is bidirectional. The source sends relatively few bytes (typically 1136 plus or minus a few) and the destination sends relatively more bytes (anywhere between a few hundred bytes and a few megabytes). Source bytes are modeled using a Normal distribution. Destination bytes are modeled using an Erlang distribution.

4.1.2 Model Validation

The model has been parameterized to generate synthetic workloads that closely match the empirical data collected at the University of Saskatchewan. Close agreement (often within 2%, and nearly always within 10%) has been obtained on all key characteristics of the model. Comparisons with mosaic workloads generated at other sites (e.g., Calgary, Waterloo) remains to be done. Further testing and validation of the model continues.

This model describes a single mosaic session and is likely to have little effect on the performance of an ATM network. Currently it is not clear how this model will scale to the number of concurrent WWW browser sessions on a typical future ATM network. To be useful it will need to scale to 100,000 plus users

5. TCP/IP Traffic

The Transmission Control Protocol (TCP) (Postel, 1981) is often described as the "4-wheel drive" of transport-layer protocols, because of its ability to "go anywhere that you need to go". Because TCP is a general purpose protocol, it can operate successfully in any network environment: local area networks (LAN's), metropolitan area networks (MAN's), and wide area networks (WAN's). Furthermore, TCP typically offers reasonable end-to-end performance to users in these environments regardless of the bandwidth and error characteristics of the particular network technology used at the physical layer (e.g., Ethernet, FDDI, satellite, packet radio). The robustness of TCP has contributed to its success in the Internet environment, where it provides the reliable byte stream delivery required by many higher level network applications, such as file transfer (ftp), remote login (telnet, rlogin), electronic mail (SMTP), network news (NNTP), and the World Wide Web (HTTP).

There is one network environment, however, where TCP has failed to “go" successfully: high speed ATM (Asynchronous Transfer Mode) networks (Minzer, 1981). Several recent studies (Moldeklev and Gunningberg, 1994; Romanow and Floyd, 1994) have identified the failings of TCP in ATM network environments. In fact, the study reported in (Romanov and Floyd, 1994) shows that the effective throughput achieved by TCP over a 140 Mbps ATM link can approach zero (0.16 Mbps, to be exact) under certain conditions in the configuration and operation of TCP.

5.1 Introduction to TCP/IP

An example of a protocol stack is shown in Figure 19. This protocol stack shows the application interfacing to TCP, which is the end-to-end protocol. After processing by several other levels, data is eventually transmitted over an ATM network. This process is referred to as TCP over ATM, or TCP/ATM. In Figure 19, the lowest layers, Asynchronous Transfer Mode (ATM) and ATM Adaptation Layer (AAL), are used together to prepare data for transmission over an ATM network. The Transmission Control Protocol (TCP) and the Internet Protocol (IP) comprise the popular TCP/IP protocol suite. TCP/IP provides reliable transmission and addressing for many applications including World Wide Web, telnet, and ftp. These applications interface with TCP, which transfers data to IP, which in turn uses the underlying network driver (such as AAL/ATM) to transmit and receive data.

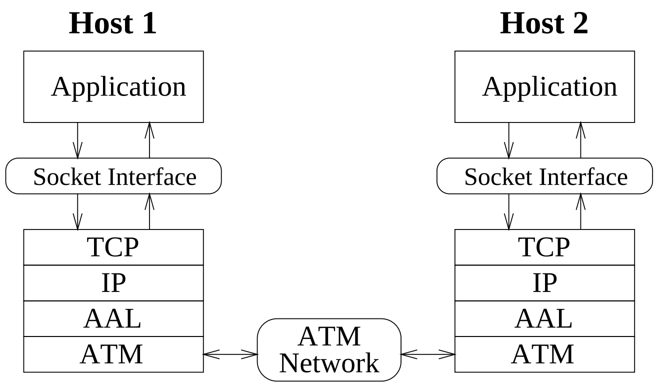

Figure 19 TCP over ATM protocol stack (Gurski and Williamson, 1995)

The basic unit of data exchange in TCP/IP is the packet, where a packet consists of zero or more bytes of user data prefixed by TCP and IP headers. The total length of these combined headers is 40 bytes, though each may include additional protocol options. Hosts along the network path use these headers for routing (IP) and end-to-end control (TCP). ATM uses a basic unit of a cell, which is of fixed length (53 bytes) and includes routing (switching) information. Conversion between packets and cells is performed by the AAL.

The Internet Protocol (IP) is a connectionless network layer protocol (Tanenbaum, 1988) and is responsible for addressing individual datagram packets so they can be routed to their destination. IP makes no attempt to detect errors or ensure that packets are delivered to the destination in the correct order, both of which are left to a higher level protocol such as TCP.

TCP is a transport-layer protocol designed to provide reliable, error-free transmission of data between two hosts. TCP uses IP as a means of transporting an application's data across a potentially unreliable network, and adds facilities to ensure that all transmitted data arrives error-free at the receiving host and is reassembled in order.

When a packet arrives at its destination, the TCP receiver sends an *acknowledgment* (ACK) back to the sender, which in turn allows another packet to be sent. If a packet arrives out of order (i.e. an earlier packet has been lost) or contains an error the sender sends out an acknowledgment that requests the retransmission of the missing or corrupted packets. Receivers are permitted to delay the sending of acknowledgments until additional segments arrive so that multiple TCP segments can be acknowledged at the same time.

In addition to error control, TCP provides flow control. To accomplish this, a sliding window method is used. The sender maintains a send window corresponding to the number of segments that it can transmit and have outstanding (i.e., unacknowledged) at one time. As segments are acknowledged, the window "slides" and new segments can be transmitted. In this manner, the source keeps the network full of packets without waiting for immediate acknowledgments, which may be subject to long round trip time delays.

Congestion control is also present in TCP. TCP attempts to estimate how busy the network is and adjust accordingly by maintaining a congestion window. The maximum number of packets a sender can have outstanding is the lesser of the send and congestion windows. A standard algorithm that makes use of the congestion window is called slow-start (Jacobson, Braden and Borman, 1992). When a packet is lost, the send window is set to one segment and increased by one segment every time an acknowledgment is received. When the congestion window approaches its previous size (i.e., half the size at which the packet loss occurred), TCP opens the congestion window more slowly (i.e., one segment per round trip time). This is known as the congestion avoidance phase.

5.2 Problems with TCP/IP over ATM

Traditionally, TCP/IP traffic is carried over packet-switched networks, where a packet is the basic unit. Routing and buffering are performed on the packet level. If the network experiences congestion and switch buffers become full, an entire packet is dropped. In contrast, ATM networks use a cell as the basic unit, and switching and buffering occur on the cell level. When switch buffers full, single cells are dropped. In ATM networks, then, it is possible for the network to drop only some cells from a packet and deliver the rest of the cells. The remaining cells in such a packet are essentially “dead", since the receiver cannot reconstruct the packet because of the lost cells. These cells only waste bandwidth on the already congested network. The TCP source will then retransmit the lost packet, resulting in even more wasted bandwidth. The end result is a low throughput for TCP over ATM. Some strategies for dealing with “dead" cells include Partial Packet Discard, where all remaining cells in a packet are dropped following a lost cell, and Early Packet Discard, where switches attempt to drop all cells from a packet when experiencing congestion. These result in greatly improved performance in comparison to TCP over plain ATM (Romanow and Floyd, 1994). Another problem is a result of the TCP slow-start algorithm. When several sources share a common link and experience congestion, cells from all sources may be dropped. All sources shrink their congestion windows to one segment, and the link is mostly idle while all sources perform slow-start. This idle time results in low throughput for all sources. Other problems with TCP over ATM are a result of the particular implementation. In 4.3 BSD, for example, the socket buffer copy rules and delayed acknowledgments can cause very low throughput over ATM networks (Moldeklev and Gunningberg, 1994). The coarse granularity of round trip and retransmission timers can also decrease performance, particularly in high speed networks.

5.3 The ATM-TN Simulation Model

The TCP model (Gurski and Williamson, 1995) in ATM-TN is designed to simulate bulk data transfer between two hosts over an ATM network, similar to the FTP protocol. Data transfer can occur in one or both directions. The primary sender is referred to as the source and the other host is the sink. These terms are somewhat misleading, as both hosts can create and consume traffic. TCP connection handshaking is not supported, as the model assumes a single connection lasts for the full length of the simulation. A number of options and parameters allow full configuration of TCP and ATM. The model is closely based on the TCP/IP networking code from the 4.4 BSD-Lite release, developed by the University of California, Berkeley and its contributors (April 1994).

This implementation is essentially the popular 4.3 BSD Reno release with support for high- bandwidth, long-delay paths (Jacobson, 1988). The networking implementation in 4.4 BSD-Lite has also been called Net/3 (Wright and Stevens, 1995). The simulation model of TCP is written in SimKit, a C ++ based framework with support for sequential and parallel simulation. A full protocol stack is modeled, from the application level down to the ATM level (see Figure 19). The model is divided into four sections: the application level, socket interface, TCP/IP protocols, and AAL/ATM layers.

5.3.1 Application Layer

The application level performs simulated data writes through the socket interface. Each host is assigned an amount of data to write during a given simulation (or unlimited data). If either host is configured to send zero bytes, it will only send acknowledgment packets and the model functions as a one-way traffic source.

5.3.2 Socket Layer

The socket layer performs buffering of simulated data and communicates with the TCP/IP layer. The socket model includes facilities for simulating the kernel mbuf data structure (Leffler, Mc Kusick, Karels and Quarterman, 1989). Socket copy rules and mbufs have been shown to cause low throughput in TCP over ATM in some situations (Comer and Lin, 1994; Crowcroft, Wakeman, Wang and Sirovica, 1992; Moldeklev and Gunningberg, 1994). The model includes parameters for specifying the send and receive buffer sizes and other socket options.

5.3.3 TCP/IP Layer

TCP is modeled in great detail. The model implementation of TCP is virtually identical to Net/3 and includes the same features such as slow-start, fast retransmit, fast recovery, and high performance extensions (Jacobson, Braden and Borman, 1992). It also supports full-duplex data communication to investigate the effects of piggybacked acknowledgments on data packets [23]. The IP layer is not explicitly modeled; instead, all routing takes place at the ATM level, and the TCP model adjusts packet sizes to account for IP headers and fragmentation.

5.3.4 AAL/ATM Layer

The AAL and ATM layers are combined. The model accepts TCP/IP packets, fragments them into ATM cells including AAL5 overhead, and queues them for transmission over the simulated network. Received cells are reassembled into packets, checked for completeness, and passed up to the TCP/IP level if they are valid. ATM level cell pacing is supported and is specified as the maximum number of cells to transmit per second.

5.3.5 Miscellaneous

Almost all socket, TCP/IP, and AAL/ATM features can be controlled through model parameters or compile options. These include timer granularity, maximum segment size (MSS), maximum transmission unit (MTU), and transmission start time. Features can also be disabled, such as the TCP high performance extensions (Jacobson, Braden and Borman, 1992) and Nagle's algorithm (Nagle, 1984). A detailed report for each source/sink is generated after each simulation, including bytes sent and received, retranslated packets, duplicate packets, and dropped cells. Detailed tracing at the TCP or cell level is available for post-simulation processing.

5.4 Model Validation

A number of experiments with the TCP model under various configurations and network loads has been conducted to

1. To validate the TCP model and ensure that it was accurately modeling the behavior of a TCP source over a real ATM network.

2. To investigate the interactions of multiple TCP sources when competing for bandwidth, by comparing the individual and aggregate throughput with the network capacity.

Figure 20 shows the network topology used for all the simulation experiments. This topology is intended to represent a typical ATM wide area network. One or more TCP sources connect to their sinks through a switch and a shared link. All sources are configured for unlimited unidirectional data transfer. That is, each source sends packets and each sink sends only acknowledgments. Network congestion occurs at the output port of the switch.

All links have a bandwidth of 1 Mbps and a propagation delay of 3.7 μs per km, resulting in a round trip time of approximately 30 ms for each source. The simulated network does not qualify as a high-bandwidth, long-delay link (Jacobson, Braden and Borman, 1992), so the TCP high performance extensions were disabled for all experiments. The traffic sources are configured for 64 kilobyte send windows in all experiments.

The switch is a simple single-stage output-buffered model (Gburzynski, Ono-Tesfaye and Ramaswamy, 1995). Outgoing cells are buffered if the output port is busy; if the buffer is full, any new cells for the port are discarded. The output buffer size is set as a simulation parameter. Cells passing through the switch are subject to a simulated lookup delay (1μs) and internal delay (2μs), as well as (variable) queuing delays at the output port.

To minimize conflicts between the sources when starting the simulation, the transmission start times of all sources were staggered 100 ms apart. All simulations were run for a 10 second warmup period, during which no statistics were gathered, and then for 300 seconds of simulated time. This allowed for a total data transfer of more than 700,000 cells.

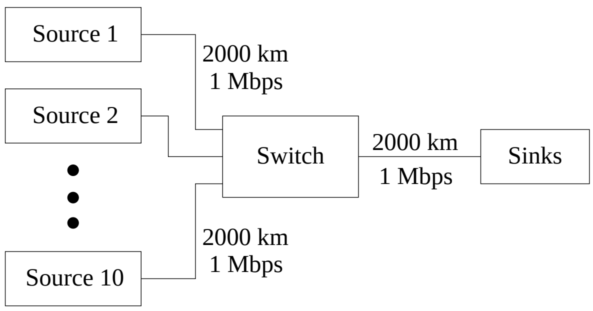

Figure 20 Simulation network topology

5.4.1 Experiment 1

The first experiment was conducted to examine the behavior of the TCP model for a single source in a congested network. A single TCP source was connected to the switch by a 1 Mbps link in the network shown in Figure 20. For this experiment only, the bandwidth of the switch output link was lowered to 500 kbps. The switch buffer was set to 1000 cells and the MSS (the maxium size of a TCP packet) was 9140 bytes. No simulation warmup period was used in order to view the source start-up dynamics.

In this situation, the TCP source will quickly fill the switch buffer due to the mismatch between the input and output link speeds (1 Mbps vs. 500 Kbps). The buffer can hold at most 5 complete TCP packets. Beyond this backlog, the switch will begin to drop cells as necessary, and the source will have to detect and retransmit lost packets.

Figure 21 shows the first ten seconds of the simulation where the upper plot shows the transmitted and dropped cells, the middle plot shows the sender's congestion window and the bottom plot shows the switch buffer occupancy. The results of this simulation show a number of interesting features of the TCP Model. First, it is possible to see the delayed acknowledgment option in action with an acknowledgment being sent every 200 ms (indicated by 'X' characters on the top plot).

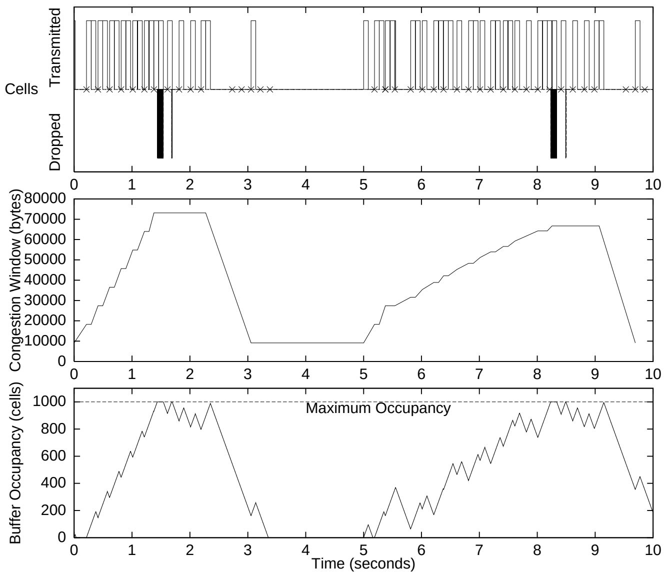

Figure 21 Dynamics of a Single TCP Source

Second, looking at the top and bottom plots, it is possible to see that packets are dropped as the switch buffer becomes full. Also when the sender has detected a lost packet the congestion window (shown in the middle plot) has been set to one and then opened at slower rate than it did previously. When another segment is dropped the process repeats. In this experiment an average throughput of 61% is achieved (meaning that 39% of the time the link is idle or transmitting cells that have to be retransmitted).

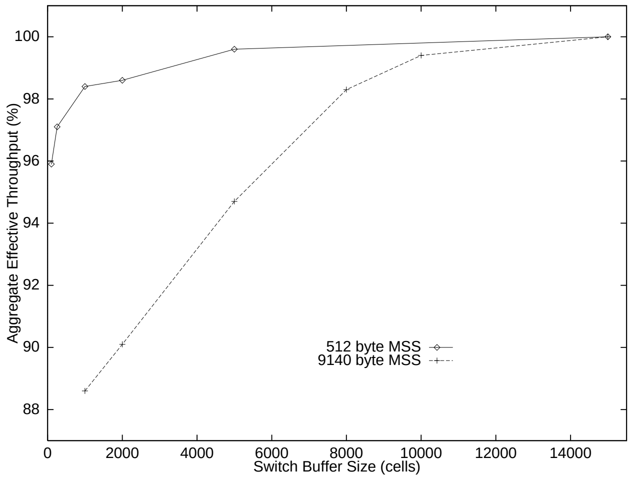

Figure 22 Effective Throughput vs. Switch Buffer Size

5.4.2 Experiment 2

A final experiment was conducted to measure the effect of switch buffer size on the aggregate throughput of the network. Ten sources were connected to the switch and the switch buffer size was varied from 100 to 15000 cells. Simulations were performed for MSS (the maximum size of IP packets) sizes of 512 and 9140 bytes. The results of these simulation runs is shown in Figure 22. As expected, throughput improves with improving buffer size are more traffic bursts can be accommodated by larger buffers and fewer cells are dropped (at the expense of longer cell delays).

At all buffer sizes, effective throughput is higher with a smaller MSS. This was an expected result as the consequence of dropping a single cell (meaning the whole packet has to be retransmitted) is less. These results are similar to those in Romanow and Floyd (1995).

5.5 Conclusions

While several solutions to TCP/ATM performance problems are suggested in the literature, clearly more work is required in order to understand and improve the performance of TCP over ATM. Solutions to the performance problems may require changes to ATM switches within the network, revised implementations of TCP, or specialized TCP protocol support at the edges of an ATM network. Because the ATM-TN TCP model is detailed enough to simulate many of the performance problems identified by other researchers, it can also be used to evaluate potential solutions to TCP/ATM performance problems. Thus, the TCP model provides a valuable tool for networking researchers involved in the design and analysis of high speed ATM networks.

6. Conclusions and Future Work

ATM is a leading contender for the implementation of future broadband communications networks because of its touted ability to transmit a variety of different traffic types efficiently and seamlessly. However, a number of recent studies have shown that most of the major traffic types reveal bursty and correlated statistical behavior. This behavior coupled with the anticipated size of future ATM networks is going make the task of network design and management an extremely difficult one.

Currently the only practical method to support the design and management of ATM networks and resolving outstanding research issues is the use of simulation. A vital component of these simulation tools are the traffic source models. The key steps in developing accurate models are: empirical network measurement, traffic characterisation and model creation.

To date very little literature has been located on empirical network measurement on ATM networks. It seems that this reflects the lack of research in this area which in turn reflects the paucity of installed ATM networks available for measurement. Most of the traffic measurements conducted to develop ATM traffic models are based on measurement of traffic on other types of networks such as LANs. While this forms a good basis for developing traffic models, measurements based on actual ATM networks are required so that these models can be charcterised and validated more accurately.

Of the literature located on modeling of ATM networks, LAN, video and WWW traffic have been identified as being important and complex traffic sources. There are a number of studies that show that LAN and video sources have self similar properties that cannot be modeled accurately using traditional Poisson or Markov based models. These self-similar properties have strong implications for traffic management as failing to take them into account can lead to predictions of network performance that are off by several orders of magnitude. This will require the development of traffic models that are characterised using empirical traffic measurements that span multiple days or months. In particular a lot of effort is still required to accurately characterise VBR video traffic and to validate the resulting traffic models.

The third traffic type identified as being important is WWW traffic because of the phenomenal growth of this type of traffic in recent years. It is currently the largest component of any traffic type on the internet and continues to be the fastest growing. Understanding the workload characteristics of these WWW sessions is therefore deemed very import in any study of the internet or high speed networks. From the literature it would appear that a great deal of research has yet to be done, particularly on the effect of aggregating multiple WWW sessions.

Because of the need to simulate large networks and the desire to minimise simulation times it is generally impractical for simulation models to model every aspect of a particular traffic type. For example, while all of the traffic types discussed above use the TCP/IP protocol for end to end communication on most existing networks, none of the models reviewed for each of them includes a TCP/IP component. However there has been quite a lot of discussion about the mismatch between TCP/IP packet sizes and ATM cell sizes. For this reason a number of traffic models have been developed to model TCP/IP traffic on ATM networks. The model reviewed have been able to reproduce and explain the very low throughputs experienced in real networks. As TCP/IP is likely to be a very widely used protocol over ATM a lot of research has yet to be done to improve its efficiency over ATM. Traffic models such as the ATM-TN traffic model will provide n very good vehicle for this research as it is sufficiently detailed to capture most of the finer details of TCP/IP and different scenarios can be tested relatively quickly.

As can be seen from the above discussion a large amount of research has yet to be performed to produce the high fidelity traffic models necessary to accurately predict actual network performance. The current research has however shown the importance of simulation and how it can used by traffic engineers and researchers to achieve their goals.

7. Bibliography

Arlitt, M. and Williamson, C.L., "A Synthetic Workload Model for Internet Mosaic Traffic", *Proceedings of the 1995 Summer Computer Simulation Conference (SCSC'95)*, Ottawa, Ontario, pp. 852-857, July 1995.

Arlitt, M., Chen, Y. , Gurski, R. and Williamson, C.L., "Traffic Modeling in the ATM-TN Telesim Project: Design, Implementation, and Performance Evaluation", *Proceedings of the 1995 Summer Computer Simulation Conference (SCSC'95)*, Ottawa, Ontario, pp. 847-851, July 1995.

Brinkmann, A., Lavrijsen, C., Louis, S., Macfadyen, R., De Vleeschauwer, D., de Vries, R. and Winstanley, S., "Performance Evaluation - Test Configurations and Results", *First International "ATM Traffic Expert" Symposium,* Basel/Switzerland, April, 1995.

Cáceres, R., Danzig, P.B., Jamin, S. and Mitzel, D.J. "Characteristics of Wide-Area TCP/IP Conversations, *ACM SIGCOMM ,* vol. 21, no. 4, 1991

Chen, Y., Deng, Z. and Williamson, C.L., "A Model for Self-Similar Ethernet LAN Traffic: Design, Implementation, and Performance Implications", *Proceedings of the 1995 Summer Computer Simulation Conference (SCSC'95),* Ottawa, Ontario, pp. 831-837, July 1995.

Claffy, K., "Internet Traffic Characterization", Ph.D. Thesis, Department of Computer Science, University of California, San Diego, 1994.

Cleary, J., Pearson, M., Graham, I. and Unger, B., "High Performance Simulation for ATM Network Development", Report to New Zealand Telecom, 1996

Comer, D. and Lin, J., “TCP Buffering and Performance over an ATM Network", *Technical Report CSD-TR 94-026*, Purdue University, West Lafayette, Indiana, 15 pages, March 1994.

Crowcroft, J., Wakeman, I., Wang, Z. and Sirovica, D., “Is Layering Harmful?", *IEEE Network Magazine,* Vol. 6, No. 1, pp. 20-24, January 1992.

Fowler, H.J., and Leland, W.E. “Local Area Network Traffic Characteristics, with implications for Broadband Network Management”, *IEEE Journal on Selected areas in Communications 9,* pp 1139 - 1149, 1991.

Frost, V. and Melamed, B. "Traffic Modeling for Telecommunications Networks", *IEEE Communications*, Vol. 32, No. 3, pp. 70-81, March 1994.

Gburzynski, P., Ono-Tesfaye, T. and Ramaswamy, S., “Modeling ATM Networks in a Parallel Simulation Environment: A Case Study", *Proceedings of the 1995 Summer Computer Simulation Conference (SCSC'95)*, Ottawa/Ontario, pp. 869-874, July 1995.

Geist, D. and Melamed, B., “TEStool: An Environment for Visual Interactive Modeling of Autocorrelated Traffic”. *Proceedings of the 1992 International Conference on Communications,* Chicago/Illinois, June, 1992

Gray, 1996. http://www.netgen.com/info/growth.html

Gurski, R. J. and Williamson, C. L., “TCP over ATM: Simulation Model and Performance Results”, *Working Paper Series in Distributed Systems Performance*, No. 95-11, Department of Computer Science, University of Saskatchewan, Canada, 1995

Habib, I. B. and Saadawi, T.N. “Multimedia Traffic Characteristics in Broadband Networks” *IEEE Communications Magazine,* pp 4-9, 1994

Jacobson, V., “Congestion Avoidance and Control", *Proceedings of the 1988 ACM SIG-COMM Conference on Communications Architectures and Protocols*, Stanford, CA, pp. 314-329, August 1988.

Jacobson, V., Braden, R. and Borman, D., “TCP Extensions for High Performance", *RFC 1323*, 37 pages, May 1992.

Kuehn, P.J., " Reminder on Queuing Theory for ATM Networks", *First International "ATM Traffic Expert" Symposium,* Basel/Switzerland, April, 1995.

LeGall, D., “MPEG: A Video Compression Standard for Multimedia Applications”, *Communications of the ACM*, Vol. 34, No. 4, pp 45-58, April, 1991

Leland, W. , Taqqu, M., Willinger, W. and Wilson, D., "On the Self-Similar Nature of Ethernet Traffic (Extended Version)", *IEEE/ACM Transactions on Networking,* Vol. 2, No. 1, pp. 1-15, February 1994.

Leffler, S., McKusick, M., Karels, M. and Quarterman, J., The Design and Implementation of the 4.3BSD UNIX Operating System, Addison-Wesley, Reading, Massachusetts, 1989.

Manthorpe, S., "Buffer and Bandwaidth Allocation for TCP Data Traffic: Experimental Results", *First International "ATM Traffic Expert" Symposium,* Basel/Switzerland, April, 1995.

Melamed, B. "TES Modelling of Video Traffic", *IEICE Trannsactions on Communications,* Vol. E-75-B (June), pp 1292 - 1300, 1992

Melamed, B. and Hill, J.R., "Applications of the TES Modeling Methodology" *Proceedings of the 1993 Winter Simulation Conference.* pp 1330 - 1338, 1993.

Minzer, S. "Broadband ISDN and Asynchronous Transfer Mode (ATM)", *IEEE Communications Magazine,* Vol. 27, No. 29, pp 17 - 24, September, 1989.

Moldeklev, K. and Gunningberg, P., "Deadlock Situations in TCP over ATM", *Proceedings of the 4th International IFIP Workshop on Protocols for High Speed Networks (PfHSN'94)*, Vancouver, B.C., pp. 219-235, August 1994.

Nagle, J., "Congestion Control in IP/TCP Internetworks", *ACM Computer Communication Review*, Vol. 14, No. 4, pp. 11-17, October 1984. (Nagle's algorithm is also described in RFC 896, January 1984).

NFSNET, 1995. Data available by anonymous ftp from nis.nfs.net:/nfsnet/statistics

Pancha, P. and El Zarki, M., "MPEG Coding for Variable Bit Rate Video Transmission", *IEEE Communications Magazine,* Vol. 32, No. 5, pp 56-64, May, 1994

Paxson, V. and Floyd, S. "Wide-Area Traffic: The Failure of Poisson Modeling", *Proceedings of ACM SIGCOMM '94* , London, UK, pp. 257-268, August 1994.

Paxson, V., "Empirically Derived Analytic Models of Wide-Area TCP Connections", *IEEE/ACM Transactions on Networking*, Vol. 2, No. 4, pp. 316-336, August 1994.

Paxson,V., "Growth Trends in Wide-Area TCP Connections", *IEEE Network*, Vol. 8, No. 4, pp. 8-17, August 1994.

Postel, J., editor, "Transmission Control Protocol", *RFC 793*, 85 pages, September 1981.

Romanow, A. and Floyd, S., "Dynamics of TCP Traffic over ATM Networks", *IEEE Journal on Selected Areas in Communications*, Vol. 13, No. 4, pp. 633-641, May 1995. (An earlier version appeared in the Proceedings of the 1994 ACM SIGCOMM Conference on Communications Architectures, Protocols, and Applications, London, UK, pp. 79-88, August 1994.

Stevens, W., TCP/IP Illustrated, Volume 1, Addison-Wesley, Reading, Massachusetts, 1993

Tanenbaum, A., Computer Networks, Second Edition, Prentice-Hall, Englewood Cliffs, NJ, 1988.

Unger, B., Gomes, F., Xiao, Z., Gburzynski, P., Ono-Tesfaye, T., Ramaswamy, S., Williamson, C.L. and Covington, A., "A High Fidelity ATM Traffic and Network Simulator". *1995 Winter Simulation Conference*, December 1995.

Technical Report, Department of Statistics, University of Adelaide, Dec., 1990

Wright, G. and Stevens, W., TCP/IP Illustrated, Volume 2: The Implementation. Addison-Wesley, Reading, Massachusetts, 1995.

Not Cited

Aalto, S., van den Berg, J.L., Norros, I., Smeitink, E. and Virtamo, J.T., "Architectural issues of connectionless overlay network", *First International "ATM Traffic Expert" Symposium,* Basel/Switzerland, April, 1995.

Adas, A. and Mukherjee, A. "On Resource Management and QoS Guarantees For Long Range Dependent Traffic", *Technical Report GIT-CC-94/60,* Geogia Institute of Technology, Altlanta, GA 30332, USA, December 1994.

ANSI, "AAL5 - A New High Speed Data Transfer AAL", ANSI T1S1.5 91-449, November 1991.

Atkinson, R., "Default IP MTU for use over ATM AAL5", *RFC 1626*, 5 pages, May 1994.

Blondia, C. and Casals, O., "Traffic Profiles in ATM Networks", *First International "ATM Traffic Expert" Symposium,* Basel/Switzerland, April, 1995.

Cáceres, R. "Efficiency of Asynchronous Transfer Mode Networks in Transporting Wide-Area Data Traffic", *Proceedings IEEE INFOCOM '92*, 1992.

Cidon, I., Guérin, R., Khamisy, A. and Sivarajan, K.N., "Cell verses Message Level Performances in ATM Networks", *First International "ATM Traffic Expert" Symposium,* Basel/Switzerland, April, 1995.

Comer, D., "Internetworking with TCP/IP, Volume 1: Principles, Protocols, and Architecture", Second Edition, Prentice-Hall, Englewood Cliffs, NJ, 1991.

Floyd, S. and Jacobson, V., "On Traffic Phase Effects in Packet-Switched Gateways", *Internetworking: Research and Experience,* Vol. 3, No. 3, pp. 115-156, September 1992.

Gallassi, G., "Traffic Issues for the Delivery of Video Services", *First International "ATM Traffic Expert" Symposium,* Basel/Switzerland, April, 1995.

Garrett, M. and Willinger, W., "Analysis, Modeling and Generation of Self-Similar VBR Video Traffic", *Proceedings of ACM SIGCOMM '94,* London, UK, pp. 269-280, August 1994.

Georgiadis, L., Guérin, R., Peris, V. and Sivarajan, K.N., "The Effect of Traffic Shaping in Efficiently Providing End-to-End Performance Guarantees", *First International "ATM Traffic Expert" Symposium,* Basel/Switzerland, April, 1995.

Gomes, F., Franks, S., Unger, B., Xiao, Z., Cleary, J. and Covington, A., "SimKit: A High Performance Logical Process Simulation Class Library in C ++ ", *1995 Winter Simulation Conference*, Washington, D.C., December 1995.

Gravey, A., Romoeuf, L., Sevilla, K. and Blaabjerg, S., "Models for Setting ATM parameter Values", *First International "ATM Traffic Expert" Symposium,* Basel/Switzerland, April, 1995.

Klivansky, S. K., Mukherjee,A. and Song, C. "On Long Range Dependence in NSFNET Traffic" *Technical Report GIT-CC-94/61,* Geogia Institute of Technology, Altlanta, GA 30332, USA, December 1994.

Kvols, K., "Combine - Interworking in Broadband Networks", *First International "ATM Traffic Expert" Symposium,* Basel/Switzerland, April, 1995.

Mark, B.L. and Ramamurthy, G., "UPC Based Traffic Descriptions for ATM: How to Determine, Interpret and Use Them.", *First International "ATM Traffic Expert" Symposium,* Basel/Switzerland, April, 1995.

Mignault, J., Gravey, A. and Rosenberg, C., "A Survey of Straightforward Statistical Multiplexing Models for ATM Networks", *First International "ATM Traffic Expert" Symposium,* Basel/Switzerland, April, 1995.

Mitrou, N.M., Kontovasilis, K.P. and Protonotarios, E.N., "ATM Traffic Engineering for ABR Service Provisioning", *First International "ATM Traffic Expert" Symposium,* Basel/Switzerland, April, 1995.

Ramaswami, V. and Wang, L., "A Tractable Queuing Model for ATM with Application to CBR Traffic Performance", *First International "ATM Traffic Expert" Symposium,* Basel/Switzerland, April, 1995.

Rao, S., Volery, F., Potts, M. and Riese, M., "Traffic Considerations for the Design of ATM Switch Architectures", *First International "ATM Traffic Expert" Symposium,* Basel/Switzerland, April, 1995.

Zhang, L., Shenker, S. and Clark, D., "Observations on the Dynamics of a Congestion Control Algorithm: The Effects of Two-Way Traffic", Proceedings of the 1991 ACM

SIGCOMM Conference on Communications Architectures and Protocols, Zurich, Switzerland, pp. 133-147, September 1991.

Zhu, W., "Characterizing Wide Area Conversations on the Internet", M.Sc. Thesis, Department of Computer Science, University of Saskatchewan, July 1994.