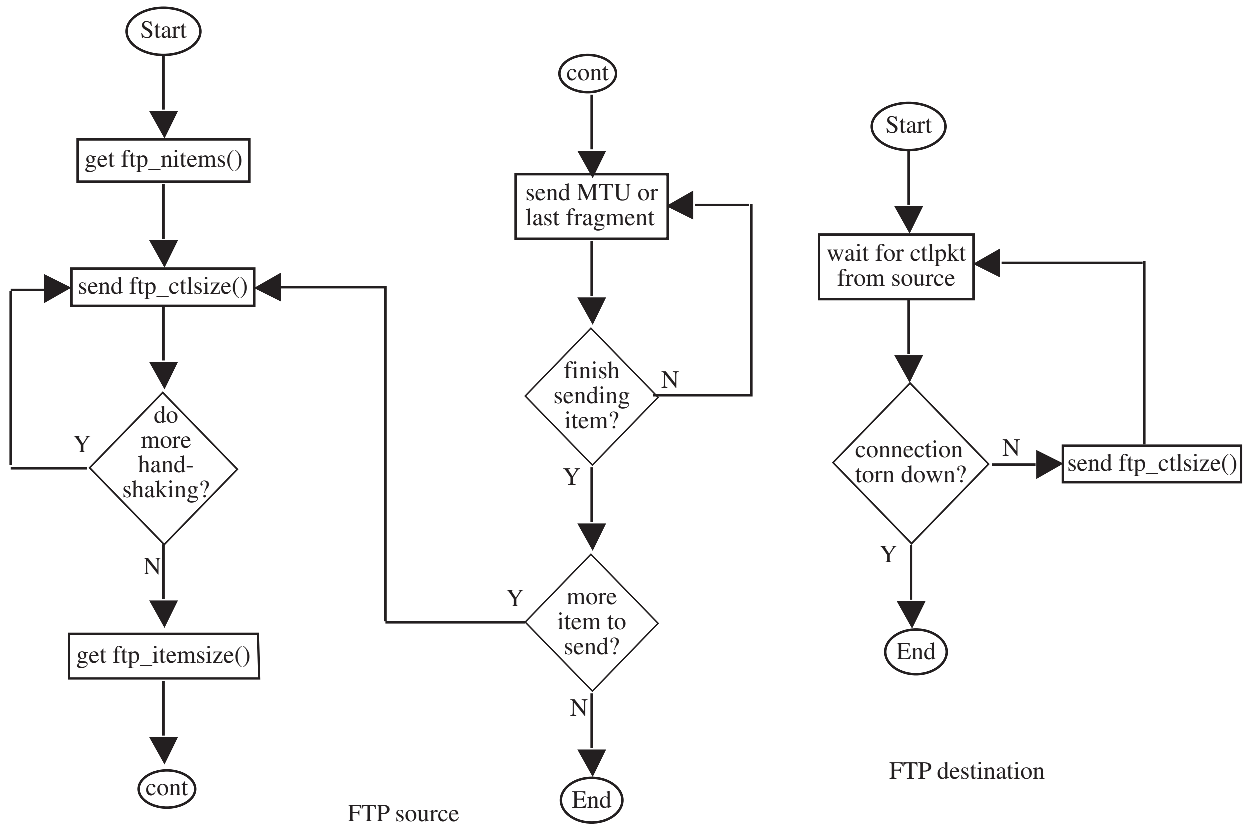

Fig.12: Flow chart for controlling FTP conversations.

The inverse transform method maps uniformly distributed 0–1 random variates through the "y-axis" of the cumulative probability distribution onto the "x-axis." With distributions fitted to analytical expressions, the inverse transform method involves inverting an equation. Consider, for example, generating an exponential random variate. If μ is a 0–1 uniform random variate and l is the parameter of the exponential, then $x=-log(1-\mu)/l$ is an exponentially distributed random variate. In our case, we built a histogram of the individual data points, and then summed the histogram bin heights to create our distribution function. Hence, our distributions are represented by arrays rather than expressions. An array index i corresponds to a particular value of the distribution. The contents of the array element at index i, $x[i]$, is the value of the cumulative distribution. Hence, to generate a random variate, we first generate a 0–1 uniform random variate μ. We then perform a binary search on the array elements until we find the element $x[k]$ into which μ falls. Finally, we linearly interpolate between $x(k)$ and $x(k+1)$ to determine our random variate x.

Rather than forming a histogram, another approach is to keep every single data point and sort the set. Then sample every 100th or 1000th element and place them in subsequent array locations that correspond to 0.01-quantile and 0.001-quantile increments. The array can then be directly indexed by μ, eliminating the search for the bin in which μ fell. We didn't adopt this approach because it takes more memory to implement than the scheme described in the previous paragraph.

5. Applying the Traffic Model

Since we are not suggesting that algorithm robustness testing should use our workload model in place of worst-case scenarios, just what good is a tool for generating realistic internetwork traffic? This section describes one problem that needs a realistic internetwork traffic model.

The problem of multiplexing application datagram traffic over wide-area virtual circuits reappears with the advent of high-speed Asynchronous Transfer Mode (ATM) networks. Assuming the existence of a reservation scheme for handling the requirements of multimedia traffic [32], we still have to accommodate the dynamics and requirements of traditional datagram traffic. When a datagram arrives at an ATM gateway, it needs to be routed onto an appropriate virtual circuit. If such a circuit doesn't exist, data transmission must wait until one is established. On the other hand, idle virtual circuits consume resources inside the ATM network. We want to find ways to multiplex TCP conversations over ATM virtual circuits that provide adequate performance while making efficient use of network resources.

We need to trade the performance costs of establishing new virtual circuits with the resource utilization advantages of closing idle circuits. Evaluating this tradeoff requires a good, average case internetwork traffic source model. With such a model we could decide how to map a set of TCP conversations onto a possibly smaller set of ATM virtual circuits, choose the queueing discipline for multiplexing datagrams onto these virtual circuits, and arrive at a timeout algorithm for reclaiming idle virtual circuits.

No previous model of wide-area traffic is appropriate for this study. To evaluate the performance of different mapping schemes, we need a realistic internetwork traffic matrix. Without accurate knowledge of application mix and behavior, we cannot predict the effect of multiplexing several different TCP conversations through a single ATM virtual circuit. To evaluate timeout schemes, we need the distribution of conversation durations and conversation interarrival times.

There are other cases where a detailed characterization of applications as presented in this paper will be required. Even for studies that aim to prove only the robustness of new designs or algorithms, using our model can show how new designs or algorithms perform on the common case.

6. Implications and Conclusions

Analysis of traffic traces collected from three different stub-networks show that the sequence of packets that application programs generate can be characterized by certain application-specific characteristics which are independent of the stub-network. We constructed an artificial workload model of a TCP/IP internetwork composed of a stub-network independent set of application source models and a stub-network dependent set of application arrival processes.

We also identified application characteristics that contradict the following commonly held beliefs regarding current wide-area traffic:

- Bulk sources transfer large amounts of data per conversation.
- Bulk sources send large packets in only one direction.
- Interactive sources send small packets in one direction, and receive echoes of comparable size in the opposite direction.
- Internetwork traffic can be modeled by either a Poisson interarrival process or a packet-train model alone.

Addressing these myths in order, we have shown that:

- Eighty percent of the time, classic bulk transfer application such as FTP transfer less than 10 kilobytes per conversation. Other applications commonly categorized as bulk traffic sources, such as SMTP, transfer even smaller amounts of data (see Figure 1a).
- Traffic generated by FTP, SMTP, NNTP, and VMNET is strongly bidirectional. Furthermore, SMTP and NNTP send as many small packets as large packets (see Figures 5b and 3b).
- Interactive applications routinely generate 10 times more data in one direction than the other, using packet sizes ranging from 1 byte to 512 bytes (see Figures 5a and 3a).
- Interactive packet interarrivals closely match a uniform plus exponential distribution (see Figure 4a).

We are continuing work on tools to create wide-area network traffic based upon our characterizations. We will also study various algorithms' responses to average case data, especially flow control and congestion control algorithms whose robustness, but not average case behavior, was evaluated in previous studies. There is more work to be done in understanding traffic reference patterns, and a better understanding of these should impact the design of future networks.

Acknowledgements

We would like to thank D. Ferrari, S. Floyd, S. Keshav, S. Morgan, J. Mogul, and M. Sullivan for their useful suggestions. M. Cooper and P. Lapsley imparted to us the inner workings of NNTP. For the data collection at Berkeley, B. Prabhakar was instrumental in collecting the traces, and C. Frost provided access to the gateway Ethernet. At USC, J. Pepin, M. Brown, and R. Kaplan provided access to the campus network, and J. Conti assisted in collecting the traces. At Bellcore, D. V. Wilson gathered the traces with his Ethernet monitor and made them available to us.

References

[1] Postel, J.B., "DoD Standard Transmission Control Protocol," RFC 761, January '80.

[2] Mark Lottor, Internet Domain System, in ACM Forum, CACM, Vol. 34, No. 11, November, 1991.

[3] Merit/NSFNET Information Services, *Link Letter*, Vol. 3, No. 6, January/February, 1991.

[4] NSFnet Backbone Statistics, Merit/NSFnet Information Services, June 1991 (obtained by anonymous FTP from nis.nsf.net).

[5] Postel, J.B., Reynolds, J.K., "File Transfer Protocol," RFC 959, October '85.

[6] Postel, J.B., Reynolds, J.K., "Telnet Protocol Specification," RFC 854, May '83.

[7] Demers, A., Keshav, S., and Shenker, S., "Analysis and Simulation of a Fair Queueing Algorithm," *ACM SIGCOMM '89*, pp. 2-12.

[8] Ramakrishnan, K.K. and Jain, R., "A Binary Feedback Scheme for Congestion Avoidance in Computer Networks," *ACM TOCS*, 8:2, May 1990, pp. 158-181.

[9] Jain, R. and Routhier, S., "Packet Trains—Measurements and a New Model for Computer Network Traffic," *IEEE JSAC*, Sep. '86.

[10] Gusella, R., "A Measurement Study of Diskless Workstation Traffic on an Ethernet," *IEEE Transactions on Communications*, Sep. 1990.

[11] Leland, W.E. and Wilson, D.V., "High Time-Resolution Measurement and Analysis of LAN Traffic: Implications for LAN Interconnection," *Proc. of INFOCOM '91*.

[12] Cáceres, R.. *Measurements of Wide-Area Internet Traffic*. UCB/CSD 89/550, University of California, Berkeley, Dec. '89.

[13] Heimlich, H., "Traffic Characterization of the NSFNET National Backbone," *USENIX Conf. Proc.*, Winter '89.

[14] Feldmeier, D., "Improving Gateway Performance with a Routing-Table Cache," *Proc. IEEE INFOCOM '88*, March '88.

[15] Jain, R., *Characteristics of Destination Address Locality in Computer Networks: A Comparison of Caching Schemes*, DEC-TR-592, Feb. '89.

[16] Crowcroft, J., *Traffic Analysis of Some UK-US Academic Network Data,* University College London TR RN/90/UK.

[17] Paxson, V., *Measurements and Models of Wide-Area TCP Conversations,* Lawrence Berkeley Lab TR LBL-30840, 1991.

[18] Danzig, P.B. and Melvin, S., "High Resolution Timing with Low Resolution Clocks and a Microsecond Timer for Sun Workstations," *ACM OS Review*, 24:1, Jan '90, pp. 23-26.

[19] Braden, B. and DeSchon, A.L., *NNStat: Internet Statistics Collection Package—Introduction and User Guide*, USC-ISI, Dec. '89.

[20] Postel, J.B., "Simple Mail Transfer Protocol," RFC 821, August '82.

[21] Kantor, B., Lapsley, P., "Network News Transfer Protocol," RFC 977, February '86.

[22] Merit/NSFNET Information Services, "New Statistics Available in Database," *Link Letter,* December 1991.

[23] Floyd, S. and Jacobson, V., "Traffic Phase Effects in Packet-Switched Gateways," *Comp. Communication Review*, April 1991, pp. 26-42.

[24] Wilder, R., Ramakrishnan, K.K., and Mankin, A., "Dynamics of Congestion Control and Avoidance of Two-Way Traffic in an OSI Testbed," *Comp. Comm. Review*, 21:2, April 1991, pp. 43-58.

[25] Williamson, C.L. and Cheriton, D.R., "Load Curves: Support for Rate-Based Congestion Control in High-Speed Datagram Networks," *Proc. ACM SIGCOMM '91*.

[26] Zhang, L., "Virtual Clock: A New Traffic Control Algorithm for Packet Switching Networks," *Proc. of SIGCOMM '90*, pp. 19-29.

[27] Zhang, L., Shenker, S., and Clark, D.D., "Observations on the Dynamics of a Congestion Control Algorithm: The Effects of Two-Way Traffic," *Proc. ACM SIGCOMM '91*.

[28] Ousterhout, J.K. et al., "A Trace-Driven Analysis of the UNIX 4.2 BSD File System," *Proc. of the 10th ACM SOSP*, Dec 1-4, 1985, pp. 15-24.

[29] Mankin, A., "Random Drop Congestion Control," *Proc. ACM SIGCOMM '90*, pp. 1-7.

[30 Mogul, J., "Network Locality at the Scale of Processes," *Proc. ACM SIGCOMM '91*.

[31] Jain, R., *The Art of Computer Systems Performance Analysis*, Wiley, 1991.

[32] Ferrari, D. and Verma, D.C., "A Scheme for Real-Time Channel Establishment in Wide-Area Networks," *IEEE JSAC*, 8:3, April '90.

Appendix 1

Comparative Data from the Three Sites

In the following figures, curves labelled *uc* represent UCB data, ones labelled *bc* represent Bellcore data, and ones labelled *sc* represent USC data.

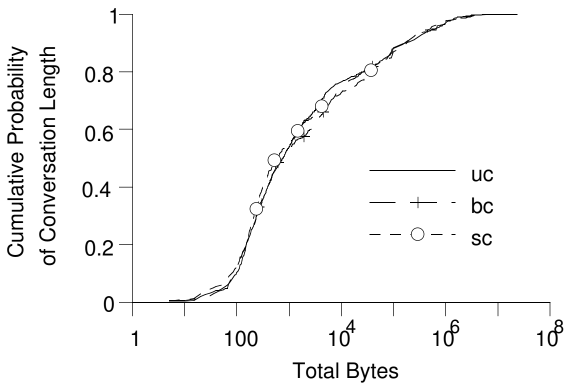

Fig. A: Total bytes transferred per unidirectional FTP conversation.

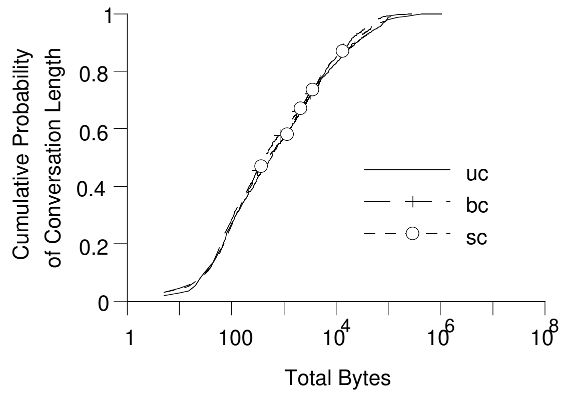

Fig. B: Total bytes transferred per unidirectional TELNET conversation.

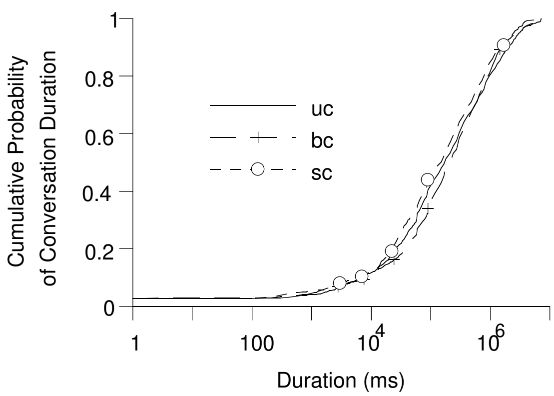

Fig. C: Duration of TELNET conversations.

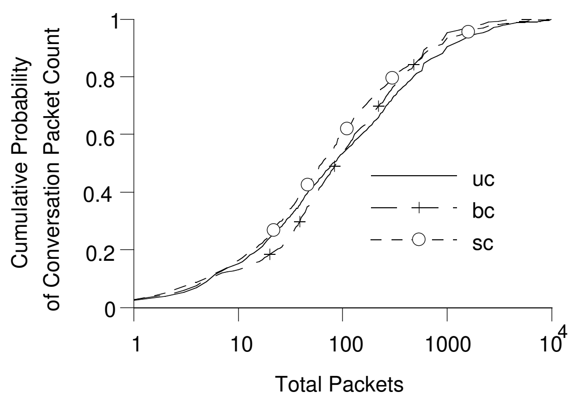

Fig. D: Packets transferred per TELNET conversation.

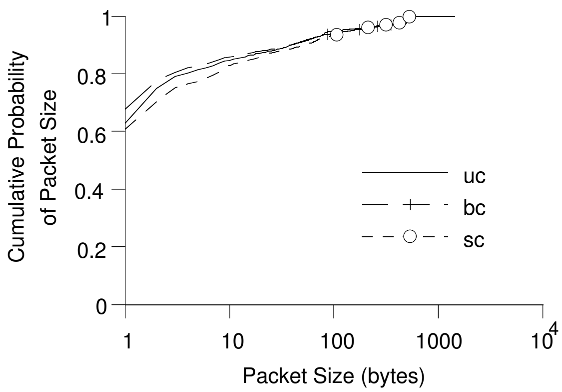

Fig. E: Distribution of TELNET packet sizes.

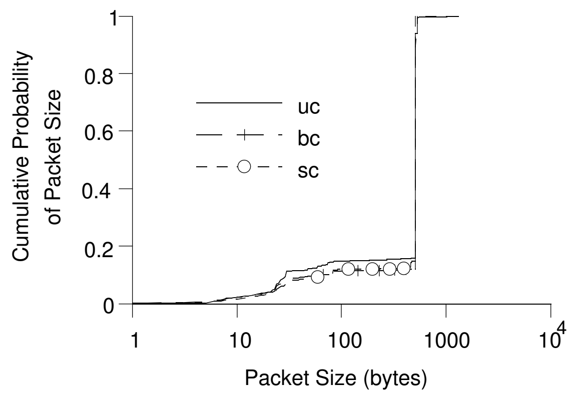

Fig. F: Distribution of FTP packet sizes.

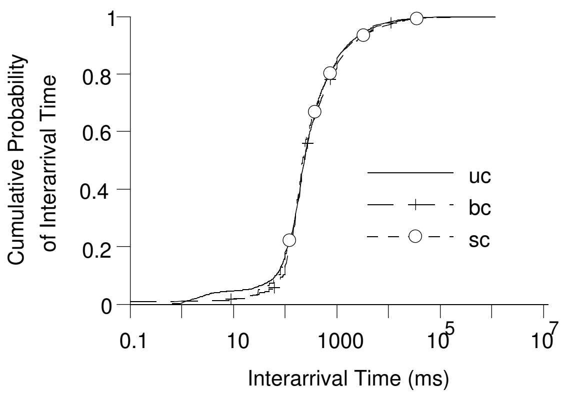

Fig. G: Distribution of TELNET packet interarrivals.

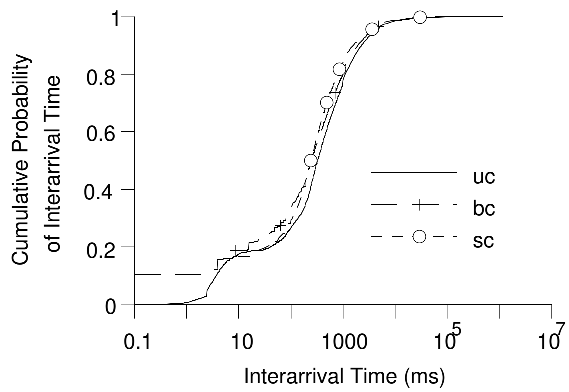

Fig. H: Distribution of FTP packet interarrivals.