Performance Evaluation of the ServerNet® SAN under Self-Similar Traffic

D. R. Avresky and V. Shurbanov
ECE Dept., Boston University
8 St.Mary's St.
Boston, MA 02215
{avresky, vash}@bu.edu

R. Horst and P. Mehra
Compaq Tandem Labs
19333 Vallco Parkway
Cupertino, CA 95014
{bob.horst, pankaj.mehra}@compaq.com

Abstract

Self-similar traffic distributions have been observed in a wide range of networking applications and models such as LANs, WANs, telnet, FTP, WWW, ISDN, SS7 and VBR traffic over ATM. Therefore, it has been suggested that many other theoretical protocols and systems need to be reevaluated under this different type of traffic before practical implementations potentially show their faults.

The ServerNet SAN is a new core technology for server architectures that focuses on moving data. It is a wormhole-routed, packet-switched, point-to-point network with special attention paid to reducing latency and assuring reliability. In this paper we investigate the implications of self-similar traffic distributions in the ServerNet SAN, and compare the results with those obtained on the basis of the Poisson assumption.

1. Introduction

The innovative work of Leland et al. [7], which was subsequently repeated by many other researchers around the world, used long, high-resolution traces of Ethernet packets to show that their arrival rates exhibit self-similar behavior, i.e., traffic is bursty over a wide range of aggregation scales, and does not degrade into uniform noise when averaged over a long period. This disparity with the traditional buffering and queuing models that rely on the Markov assumption may have a significant impact on them. The literature mentions high cell loss problems in first ATM switches due to the assumption of Poisson arrivals, which suggested buffer sizes too small for the actual traffic.

Self-similar traffic has been observed in Ethernet, ISDN, and ATM LANs and WANs for a range of traffic patterns generated by networking applications such as telnet, FTP, and World-Wide-Web [4], as well as for signaling (SS7) information traffic and variable bit rate (typically video) traffic over ATM. Therefore, it has been suggested that many other protocols and systems need to be reevaluated under this different type of traffic distribution before practical implementations potentially show their faults.

Previously we have performed many studies, based on the Poisson assumptions, of ServerNet SAN's performance characteristics and have proposed a method for improving them by optimizing the arbitration policy of routers. It is now necessary to reevaluate the results of these studies with self-similar traffic distributions. The purpose of this is to provide a basis for fine-tuning the ServerNet routers and end devices, and to modify the optimization method accordingly. Such a study also provides insight into the performance discrepancies that may be generally observed between self-similar and Poisson traffic distributions.

1.1. Modeling Self-Similar Traffic

It has been shown through trace data [7, 10] and also proven [9] that modeling of self-similar traffic can be achieved by the aggregation of a large number of ON-OFF packet train sources. The ON-OFF states are strictly alternating where ON represents a state where packets are generated according to some regular rate, and OFF represents a state where no packets are generated. The length of time in which each train spends in either the ON or OFF state should be selected according to a distribution which has long-range dependence, i.e, the time spent in a state can be very large with a non-negligible probability. The Pareto distribution ($F(x) = 1 - x^{-\alpha}$, with $1 < \alpha < 2$) [6] has been found to fit well to the empirically observed packet distributions. The long-term correlations that result from using this distribution are the main difference from traditional traffic models, such as Poisson distributions.

1.2. Testing for Self-Similarity

There are three methods typically used for verifying self-similarity in a time series: the “visual” test, aggregated variance plots, and rescaled R/S statistic plots. The first of these

is merely a set of plots of the original time series aggregated over a variety of scales, which enables simple visual inspection to determine that the process is bursty on many time scales. Fig. 1 shows the difference between a Poisson arrival process and a self-similar arrival process on a scale of 100 time units. It can be seen that the variance of the self-similar is much higher than that of the Poisson process. The data presented in this section was produced to verify the code and parameters, which were used to generate packets for the simulations.

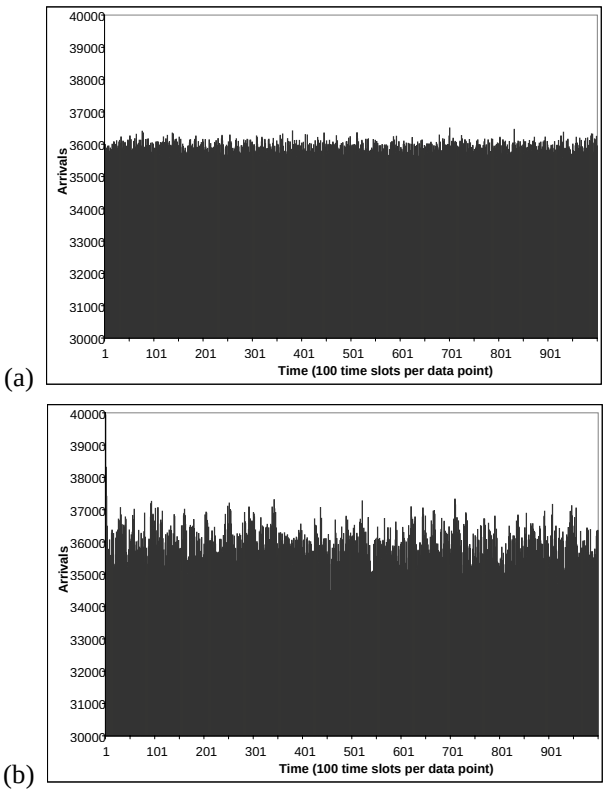


Figure 1. Arrival Processes: (a) Poisson, (b) Self-Similar

The method of aggregated variances visualizes the variance of the discrete self-similar process. The logarithm of the variance of an aggregated self-similar process, X_t, decreases linearly with the logarithm of the aggregation size [8], m:

$$log[Var(X_t^{(m)})] \equiv log[Var(X_t)] - \beta log[m]. \qquad (1)$$

As shown in Fig. 2, a log-log plot which fits a solid line to the aggregated variance points allows us to verify the self-similar nature of the process, as well as estimate the degree of self-similarity $H = 1 - \beta/2$, where $-\beta$ is the slope of the line. The degree of self-similarity of a process is typically described by the *Hurst* parameter (H) [8]. H is between 0.5 and 1, where 0.5 represents non self-similar behavior and the closer to 1, the more long-range dependent the process is. From the figure it is clear that the slope of the variance plot line formed by the generated data is much less than 1, therefore $H > 0.5$ for this data, i.e., the process is self-similar. The figure also displays a line with a slope of -1, which corresponds to the aggregated variances which would be seen from a process with short-term dependence (e.g, Poisson).

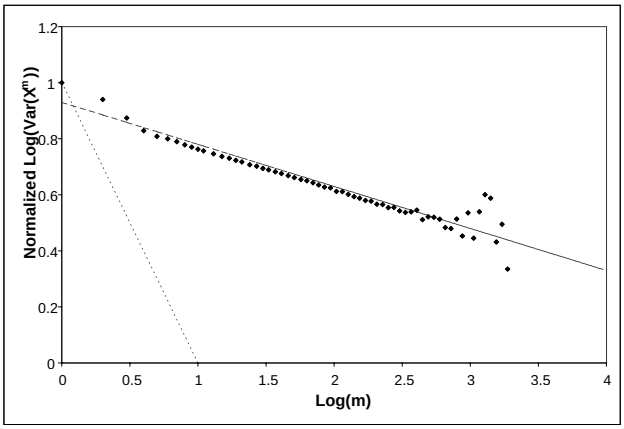


Figure 2. Normalized Variances of Aggregated Versions of a Self-Similar Time Series

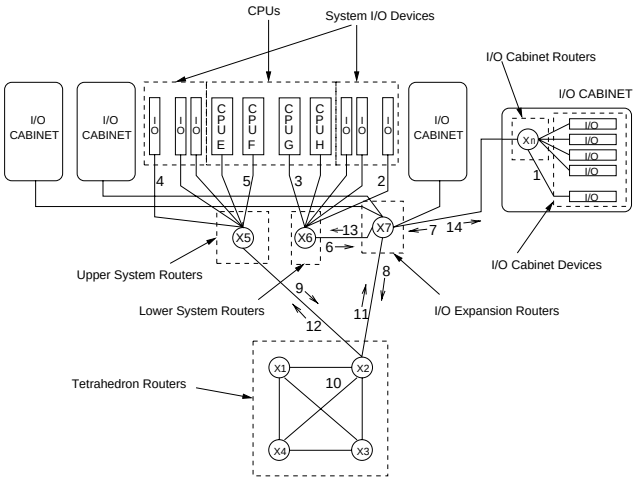
2. Modeling ServerNet

The ServerNet SAN is a wormhole-routed, packet-switched, point-to-point network with special attention paid to reducing latency and assuring reliability [5]. It uses multiple high-speed, low-cost routers to rapidly switch data directly between multiple data sources and destinations. Because it provides the intelligent switching that could previously be supplied only by a processor, ServerNet eliminates the need for a processor in every data path.

There are four types of packets defined in the ServerNet SAN: Read Request, Read Response, Write Request, and Write Response [5, 3]. They consist of header, address, checksum information, and a data payload of 64 bytes. The generation of requests is limited by the maximum number of outstanding requests allowed (8 was used in this study).

2.1. Simulation model

ServerNet is simulated by a discrete-event unit-time model [2, 6] where each device is allowed to enter a particular state during each time step. The state that each device enters is dependent on the condition of its state variables.



Note: This device grouping is mirrored at each of the four tetrahedron routers.

Figure 3. 16-CPU ServerNet SAN Topology (16 CPUs, 104 I/O devices) and Link Categories

The order in which each device is given a chance to enter a state is randomly generated during each clock step.

The simulation tool was written in C++ and compiled for Solaris 2.5. It has been validated by comparing its results with those collected in an experimental ServerNet testbed implemented at Compaq Tandem Labs. The testbed consisted of 8 routers and 24 end devices performing bulk data transfers which result in bursty traffic distributions. Less than 10% discrepancies were observed between the simulation and the empirical results.

2.2. Traffic model

The modeled topology is shown in Fig. 3. It was chosen because: it is one of the large topologies supported in the family of ServerNet-based NonStop® Himalaya® S-series servers; we have extensively studied it previously using the Poisson model; and it has the valuable properties of being deadlock-free and easily scalable.

The traffic load is determined by the Request Generation Rate (λ), defined in units of packets·device/ms. In the self-similar traffic simulations (λ) determines the number of packet-train sources that are aggregated to generate requests. While in the *ON* state each source generates a request every 10 clock ticks (200 ns). A Pareto distribution was used to control the time intervals for the packet trains to remain in each of two states (ON or OFF). Thus a packet train remains in the *ON* state for $\triangle t_{ON} = (1-R)^{\frac{-1}{\alpha_{ON}}}$, and in the *OFF* state for $\triangle t_{OFF} = (1-R)^{\frac{-1}{\alpha_{OFF}}}$, where R is uniformly distributed random number between 0 and 1, $\alpha_{ON} = 1.9$, and $\alpha_{OFF} = 1.25$ [10]. For the Poisson simulations the request generation is controlled by a Poisson process, the generation interval is determined using $t_{next} = \frac{-1}{\lambda} ln(1-R)$, where R is a uniformly distributed random number between 0 and 1. The type of request (Read of Write) is also determined by a uniformly distributed random variable and the probabilities of generating either type of request are equivalent.

Three types of traffic patterns (i.e., destination distributions) were employed in our studies [2]: Common Subtree (CS) approximates applications where the interprocessor traffic is primarily contained within a 4-CPU subtree; Uniform (U) approximates applications where traffic is evenly distributed among all devices in the network; and Transpose (T) approximates the worst case traffic in which traffic is almost exclusively remote.

2.3. Performance characteristics

The network performance characteristics used in this paper are as follows. **Actual Generation Rate**, in flits/clock tick, is the average number of flits generated during a single simulation clock step over the life of the simulation. **Throughput**, in Flits/clock tick is the average number of flits consumed at a destination during each simulation clock tick [6]. **2-Way Delivery Time**, in ms, is the difference between the time a request is created and the corresponding response is consumed by the original sender; it is a measure of the instantaneous round-trip time. Each of these parameters is averaged over a large number of packets during the course of the simulation until the required accuracy and confidence level are achieved.

3. Results and Analysis

In previous studies of ServerNet network characteristics were obtained based on a Poisson traffic-generation model. This data was used to estimate the maximum two-way delivery time of the network [1], to determine the occurrence of hot spots [2] and to eliminate them by improving the router arbitration policy. Here the Poisson data is compared with the new results obtained with the Self-Similar model.

Fig. 4a shows that self-similar traffic saturates the network at lower average data rates, compared to the Poisson traffic. It is noticeable that before saturation the network exhibits higher average throughput under self-similar traffic. The higher throughput is accompanied by higher average delivery times. Graph (b) indicates that the two-way delivery time is approximately four times higher for self-similar traffic, compared to Poisson. The link utilization, illustrated in Fig. 5, shows the percentage of time links in each given category (see Fig. 3) were in one of three states: *transmitting, stalled (blocked) or idle*. All link categories

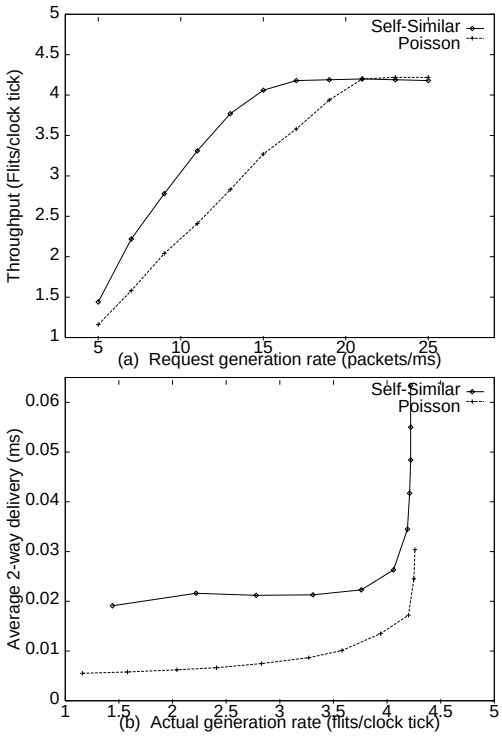


Figure 4. Network Characteristics: Poisson and Self-Similar; Uniform Destination Pattern

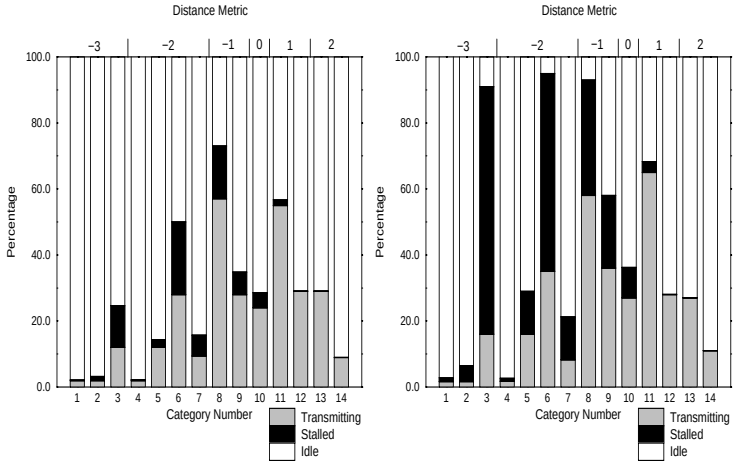


Figure 5. Utilization of Link Categories (Fig.3): Poisson (left) and Self-Similar (right), Uniform Destination Pattern

display an increased stalled time. This is particularly evident for categories 3, 6, and 8. The stalling of these links is caused by a "domino" effect. It begins at link 8, which is the most loaded one in the topology. The stalling of link 8 propagates through link 6 to link 3. Link 3 is highly utilized because it carries traffic to and from the CPUs. CPUs generate more traffic than IOs for two reasons: (1) CPUs are allowed more outstanding requests; (2) CPUs have to generate large amounts of responses, because they process all requests from the numerous IOs. Thus, the combination of the highly active CPUs and the stalling of the most loaded link 8, causes link 3 to stall more than 70% of the time. Notice that this is not the case for link 5, which is also connected to CPUs and hence has the same load as link 3, but this load does not need to pass through link 8. This significant increase in the fraction of time that links are blocked is related to the bursty nature of self-similar traffic. Bursts of data result in blocking conditions at the routers. In turn, these blocking situations result in higher average delivery time, as it was shown in Fig. 4. The increase in average delivery times is of the same degree as the increase in blocking time.

Similar results were obtained under CS and T traffic patters. Link stalling occurs more often under self-similar traffic – two to three times the values obtained under Poisson traffic, which shows the direct connection between the blocking time of links and the latency. Indeed, the average two-way delivery time for CS traffic are again two to three times higher under self-similar data streams. CS traffic is not as demanding on network bandwidth as U traffic is, hence, a lower degree of blocking was observed. The increase of latencies for the T pattern was of higher magnitude than for the other two. The reason for this is that under the T pattern packets make a more hops on average and, thus, the link stalling propagates to produce higher latencies.

It is also important to estimate the maximum two-way delivery time, i.e. the worst-case round-trip time. This is performed under the worst traffic conditions [1]: T pattern and maximum generation rate. In this estimation the number of samples necessary to achieve a reasonable percentile and confidence level were determined by the formula [1]:

$$n = z_{1-\frac{\alpha}{2}}{}^2 \frac{p}{1-p}. \qquad (2)$$

where p is the sought percentile, and $z_{1-\frac{\alpha}{2}}$ is the normal quantile for a confidence interval of $100(1-\alpha)\%$. Thus 660,484 samples were needed to estimate the 99.999 percentile of the maximum two-way delivery time with a confidence interval of 99%.

It is reasonable to expect that the maximum two-way delivery time should not increase under self-similar traffic generation. The premises for such expectation is that to estimate the maximum latency it was necessary to use the worst possible traffic conditions, which cannot be deteriorated by any factr, including bursty generation. Fig. 6 shows that in fact self-similar traffic does not produce a higher value for the maximum two-way delivery time. However, the self-similar traffic leads to high delivery times occurring more

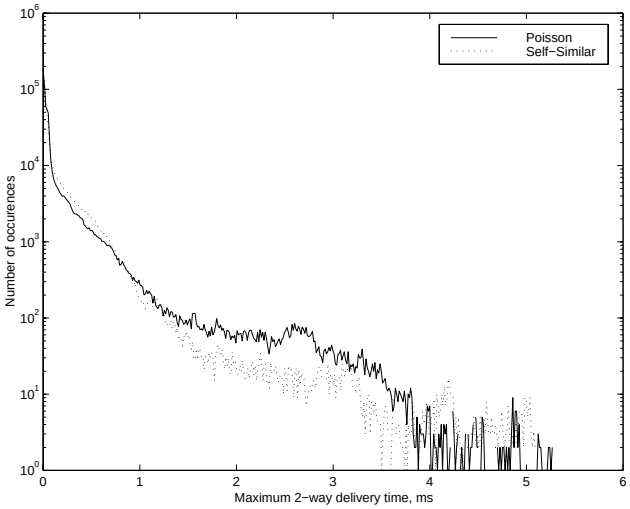


Figure 6. Maximun two-way delivery time histogram, 660,484 samples

often. This also explains the significant increase of the average delivery times.

4. Conclusions

This study has demonstrated that making assumptions about the nature of data traffic, in particular whether the packets are generated according to a Poisson or a self-similar process, can strongly influence performance evaluation results. Simulation data showed that the occurrence of blocking situations within the network is up to four times higher when packets are injected into the network by a self-similar process compared to when packets are generated by a Poisson process. The higher degree of blocking, results in higher latencies, which also increase by up to four times. This significant increase of blocking situations is attributed to the bursty nature of self-similar traffic. The bursts of data cause significant congestion in the network at much lower average traffic loads than for Poisson traffic.

Thus, it is reasonable to expect that in general buffer overflows will occur at lower average generation rates than anticipated based on the Poisson assumptions. However, such a phenomenon cannot be observed in ServerNet SAN because it is prevented by the flow control mechanism which stalls transmission when buffers approach their limits. On the other hand, when average data generation rates are very high and the network is saturated, the type of traffic generation does not significantly influence the performance characteristics, since enough data is generated (whether bursty or not) to maintain constant congestion. Therefore, the maximum two-way delivery times estimated using the Poisson assumption retain their validity.

It is foreseeable that some modifications of network components will be required to better accommodate self-similar traffic distributions. One such modification would be to increase buffer/queues sizes, so that bursts of data could be stored effectively on a single device. This would reduce the congestion caused by burst of data which fill the buffers of several consecutive devices and block entire portions of the network. Another means of controlling the network performance is to reduce the maximum number of outstanding requests and thus to limit the size of the data bursts and reduce congestion. Finally, it will be necessary to devise suitable router arbitration policies which will maximize the performance of the network.

Acknowledgments The authors are thankful to the graduate students J. Redi and I. Bulyko for their research of self-similar traffic generation and for conducting some simulations.

References

[1] D. Avresky, V. Shurbanov, R. Wilkinson, R. Horst, W. Watson, and L. Young. Maximum delivery time and hot spots in ServerNet™ topologies. *J. of Computer Networks and ISDN Systems*, 1999. Elsevier Science, The Netherlands.

[2] D. Avresky, V. Shurbanov, R. Horst, W. Watson, and L. Young, and D. Jewett. Performance modeling of ServerNet™ topologies. *The J. of Supercomputing*, 1999. Kluwer Acad. Pub.

[3] W. Baker, R. Horst, D. Sonnier, and W. Watson. A flexible ServerNet-based fault-tolerant architecture. In *Proc. of the 25th Int. Symp. Fault-Tolerant Computing*, pages 2–11, Pasadena, CA, U.S.A., June 1995.

[4] M. Crovella and A. Bestavros. Self-similarity in world wide web traffic: evidence and possible causes. *IEEE/ACM Transactions on Networking*, 5(6):835–46, Dec. 1997.

[5] R. Horst. TNet: A reliable system area network. *IEEE Micro*, pages 37–45, Feb. 1995.

[6] R. Jain. *The Art of Computer Systems Performance Analysis*. John Wiley & Sons, Inc., 1991.

[7] W. Leland, M. Taqqu, W. Willinger, and D. Wilson. On the self-similar nature of Ethernet traffic. *ACM/SIGCOMM Comp. Comm. Review*, 23:183–193, 1993.

[8] M. Taqqu. A bibliographical guide to self-similar processes and long-range dependence. In E. Eberlein and M. S. Taqqu, editors, *Dependence in Probability and Statistics*, pages 137–165. Birkhauser, 1985.

[9] M. Taqqu, W. Willinger, and R. Sherman. Proof of a fundamental result in self-similar traffic modeling. *ACM/SIGCOMM Comp. Comm. Review*, pages 5–23, 1997.

[10] W. Willinger, M. Taqqu, R. Sherman, and D. Wilson. Self-similarity through high-variability: statistical analysis of Ethernet LAN traffic at the source level. *IEEE/ACM Transactions on Networking*, 5(1):71–86, Feb. 1997.