1-1997

Protocols Can Make Traffic Appear Self-Similar

Jon M. Peha
Carnegie Mellon University, peha@andrew.cmu.edu

Published In
Proceedings of IEEE/ACM/SCS Communication Networks and Distributed Systems Modeling and Simulation Conf.,, 47-52.

Expansion of research published as
"Retransmission Mechanisms and Self-Similar Traffic Models," *Proceedings of IEEE/ACM/SCS Communication Networks and Distributed Systems Modeling and Simulation Conf.,* Jan. 1997, pp. 47-52.

Protocols Can Make Traffic Appear Self-Similar[1]

Jon M. Peha[2]
Carnegie Mellon University

Abstract

Empirical studies have shown that self-similar traffic models may better describe traffic in many of today's computer networks than traditional Markovian models. The causes of this apparent self-similar behavior must be identified to determine how widely applicable these models are, and how network designers should respond. While some researchers have argued self similarity is an inherent property of traffic as generated by the typical applications, it is also possible that the network's own protocols may cause or at least contribute to this phenomenon. In this paper, it is shown that even if packets were to arrive according to the well-behaved Poisson process, simple retransmission mechanisms can make traffic appear self similar over time scales of engineering interest. Moreover, some techniques intended to decrease the likelihood of congestion also have the effect of prolonging congestion when it does occur. This increases burstiness over large time-scales, reinforcing the appearance of self-similarity.

Key words:
ARQ, chaos, fractal, long-range dependence, self similar, retransmission, traffic model

Section 1: Introduction

Empirical evidence has been discovered [LEL94,PAX95,ERR94,CRO97,LUC97,JER97] that the traffic typically carried on local-area and wide-area packet-switched networks could be better represented with *self-similar* models which incorporate *long-range dependence*, rather than the traditional Markovian models. Given the significance of this phenomenon, it is important that we understand its causes, and determine the circumstances under which it can occur. This paper investigates the possibility that network protocols could be a cause.

It is likely that there will be multiple factors contributing to the self-similar behavior observed in network traffic. There are two general categories of causes. First, self similarity may be an inherent property of traffic streams that enter a network. Indeed, this is the most common explanation, and it has led many researchers to seek methods of simulating self-similar input traffic. Some important supporting evidence has been uncovered [PAX95,CRO97,MEI91,WIL97,PAR96]. For example, the actual time between user-initiated calls might be heavy-tailed, which would imply infinite variance and possibly infinite mean in the user's "think time." Or there could be infinite variance in the duration

[1] This research was supported under Grant NCR-9706491 from the National Science Foundation.

[2] Jon M. Peha, Professor, Carnegie Mellon University
Carnegie Mellon University, Dept. of Electrical & Computer Engineering, Pittsburgh, PA 15213-3890
(412) 268-7126, peha@stanfordalumni.org, www.ece.cmu.edu/~peha

of a call before the user decides to terminate. Or the messages transmitted could be heavy-tailed in length; this has been observed in web pages [CRO97], which constitute a large portion of today's Internet traffic (although the appearance of self-similarity was observed before the web became popular [LEL94]). In all of these cases, the applicability of self-similar models entirely depends on the applications, and possibly the specific user population. If application-level traffic is the primary cause of the observed self-similarity phenomenon, then application-level traffic shaping may be the most promising response for protocol developers [CHR97].

Alternatively, or additionally, self-similar behavior could be a side effect of the network's own protocols, a possibility that has not received as much attention. If some protocols prolong traffic burstiness to create or strengthen self similarity, then application-layer traffic shaping may not be the most effective method to improve throughput. This paper extends the work of [PEH97], which argued that basic retransmission mechanisms may create long-range dependence and self similarity over timescales of engineering interest, *even when the input traffic exhibits no long-range dependence or self similarity.*

The next section presents the intuition behind focusing on retransmission mechanisms as a possible source of long-range dependence. The specific model employed throughout this paper is presented in Section 3. Section 4 shows how this model can lead to the appearance of self similarity, and Section 5 shows the impact of various design parameters on the phenomenon. The paper is concluded in Section 6.

Section 2: Retransmission Algorithms in Congestible Networks

Intuitively, a system is prone to chaotic behavior (e.g. long range dependence) if a slight change in initial conditions can impact performance into the distant future. *Congestible* systems are likely candidates for this phenomenon. In a congestible system, as input load increases from 0, the useful output first increases, then reaches a maximum, and finally falls. (Figure 1 is an example.) When useful output (e.g. throughput) is decreasing with respect to input load, the system is considered to be congested. When output is at its maximum, a slight perturbation can cause the system to operate in either the congested or uncongested region for extended periods.

Retransmission mechanisms can make a network congestible, because these mechanisms often cause network inefficiencies which degrade throughput to occur specifically in periods when load is already high. For example, each packet lost to buffer overflow often causes multiple packets to be retransmitted, thereby wasting more link capacity with redundant retransmissions. This can occur because of a go-back-N retransmission mechanism. It can also occur when one packet is made up of multiple ATM cells [FLO94], IP fragments, or link-layer frames, and the entire packet must be retransmitted when one piece is not received. Further inefficiencies are possible if queueing delay can exceed the time-out, so unnecessary retransmissions add to the load when delays are already large. Lost acknowledgments due to large queueing delays or buffer overflow also lead to unnecessary retransmissions. With an end-to-end retransmission mechanism, when a packet is lost at the last queue in a multi-hop source-destination path, that packet must be retransmitted through the first queues in the path as well. Finally, if the queue is really a shared ethernet with contention rather than a dedicated contention-free link, inefficiency due to collisions increases with load. Whenever a spike in load causes collisions or buffer overflow, the retransmission mechanism can create another spike one time-out later, possibly repeating the problem. One dropped packet can therefore trigger prolonged congestion, so it is reasonable to investigate the possibility of long-range dependence.

It has been shown [PAR97] that when input traffic exhibits long-range dependence and self similarity, TCP sustains these properties. In contrast, UDP does not sustain long-range dependence. The basic retransmission mechanism could be the reason for this difference (or it could conceivably be slow start, or other TCP features.) But what if the input traffic were not self similar? Can these protocols cause rather than just sustain the appearance of chaotic behavior?

The impact of TCP has also been examined [VER00] when the input load is infinite, and throughput is limited only by TCP's flow control mechanisms in a single bottleneck. It was shown that aggregate traffic through a bottleneck has *no long-range dependence,* presumably because congestion control mechanisms tend to hold traffic levels relatively steady under infinite loads. But would this result hold if loads were not infinite? It was also shown that, in some instances, individual streams could appear to have long-range dependence. This burstiness could spread to non-bottleneck links - a possible phenomenon that warrants further investigation.

This paper, in contrast, assumes a finite and fixed input load. We employ no dynamic window mechanism, or equivalently, a window size of 1. This is appropriate at the TCP layer when traffic consists of many low-data-rate streams, which are content with small window sizes. It is also appropriate for some non-TCP retransmission mechanisms, such as those at the link or application layer. Finally, even when only considering TCP traffic, the only way to separate the influence of the retransmission mechanism from the effect of other aspects of the protocol is to implement the former without the latter.

Section 3: The Model

To observe how retransmissions can introduce long-range dependence (over time scales of interest) in any data stream, we will use an extremely simple model. In particular, we will assume that the *new packets* (i.e. excluding retransmissions) arrive according to the simple Poisson process. We must also make the system congestible by choosing one of the causes of congestion described in the previous section.

Focusing on a simple scenario allows us to isolate the effect of retransmissions. (Some simplifying assumptions are being relaxed in subsequent work, which includes congestion control across multiple links.) We consider a single finite-buffer first-in-first-out (FIFO) queue, and constant length packets. Each packets consists of 10 cells. Without loss of generality, the transmission time of one packet is defined to be 1. We will assume that all packets that are transmitted are also successfully acknowledged before the time-out. In our simple model, packets each consist of 10 cells which arrive together. Regardless of whether 1 or all 10 cells are lost in a buffer overflow, the entire packet must be retransmitted a fixed time-out period T_{to} later. (This makes the system congestible.) After N failed attempts at transmission, a packet is considered lost. (Thus, the duration of an "on period" for any source is always less than N T_{to}.)

When load becomes too high, much of the capacity is spent on retransmissions, and many packets are simply lost. Figure 1 shows average throughput as a function of total load, i.e. including new arrivals and retransmissions. These performance results were determined via discrete event simulation, and the 95% confidence interval is within 0.2% of the values shown. The maximum number of attempts N=5, the time-out T_{to} =25 packet transmission times, and the buffer can hold 10 full packets (100 cells). Throughput is maximized when the load from new arrivals is .77, and the total load is .848. Assuming throughput is the most important measure of performance (which is not always the case), this is the point at which the system should operate when possible.

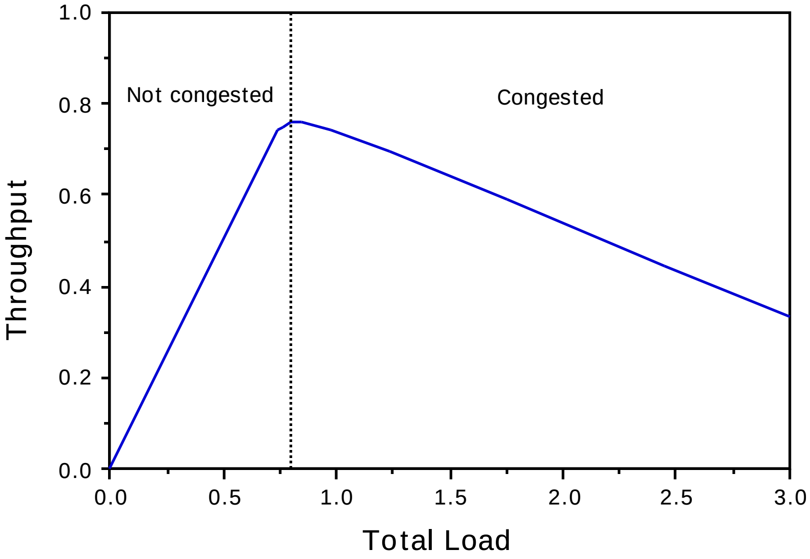

Figure 1: Throughput versus total load, including retransmissions. N=5. T_{to} =25. Buffer size = 10.

Section 4: Self Similarity Over Broad Timescales

Figure 2 shows the total load including retransmissions as a function of time, where each point describes load over a different averaging duration D. The load from new arrivals is .77, so throughput is maximized. As D increases, the variance of load from new arrivals grows small, but the burstiness from retransmissions remains. These load fluctuations from retransmissions look somewhat similar until D grows extremely large. When D gets as high as 78,125, there is a noticeable reduction in the variance, although load still frequently surges over 1.

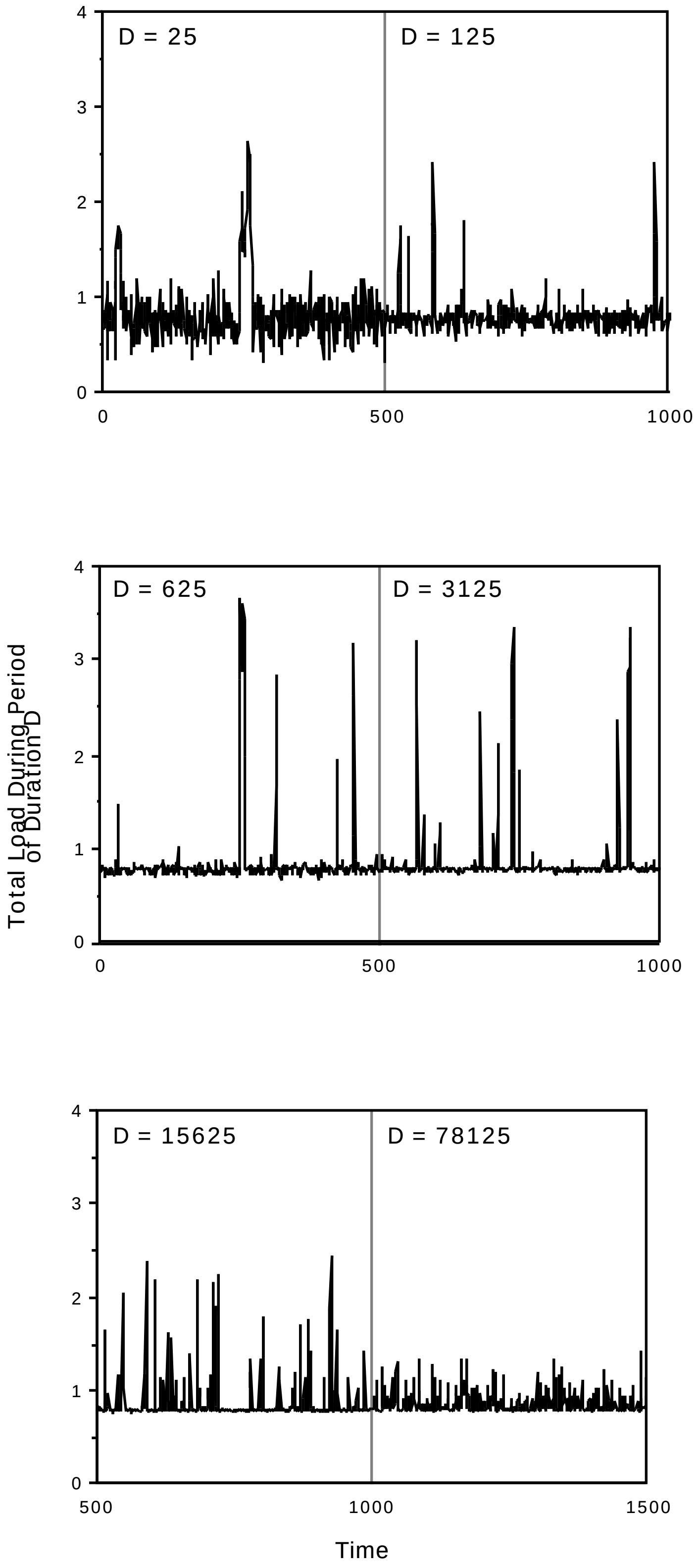

Figure 2: Total load as a function of time. Load from new arrivals = .77. N=5. T_{to} =25. D=25, 125, 625, 3125, 15625, and 78125. Buffer size = 10.

This observation can be more formally demonstrated through other means. A useful tool is a log-log graph of sample variance of load during an averaging period of duration D, as a function of D.[3] (Throughout this paper, sample variance is determined via simulation over a period of 390,625,000, not including the initial start-up phase.) In the absence of any autocorrelation, or with an exponential decay of autocorrelation, variance would be inversely proportional to D; this yields a slope of -1 on this log-log scale. A slope $-\beta$: $0 \leq \beta < 1$ for large D indicates long-range dependence, and if the slope β happens to be constant for all D, then the traffic is second-order self similar with Hurst parameter $H=1-\beta/2$

Figure 3 shows the variance of load from new arrivals, load from retransmissions, total load, and packets lost, under the same network conditions as in Figure 2. Thus, throughput is maximized. Figure 3 shows that when D exceeds around 10 packet transmission times, retransmissions account for almost all of the variance observed, so we can focus our attention on retransmissions when considering self similarity. (The Bellcore data [LEL94] includes time scales of 10 ms and greater on a 10 Mb/s ethernet, so even with maximum-length packets, their time scales all exceed 10 packet transmission times.) For a time scale D under 10^4, variance of load from these retransmissions is largely unaffected by D, indicating that traffic would appear to be highly self-similar even at fairly large time scales. (The slope from D=1 to D=15625 averages just -.18.) As D grows even larger, variance finally begins to fall at a slope that later approaches -1. It is also interesting to note that the loss rate closely mirrors the retransmission rate on the log-log scale, i.e. the ratio of loss rate and retransmission rate remains relatively constant.

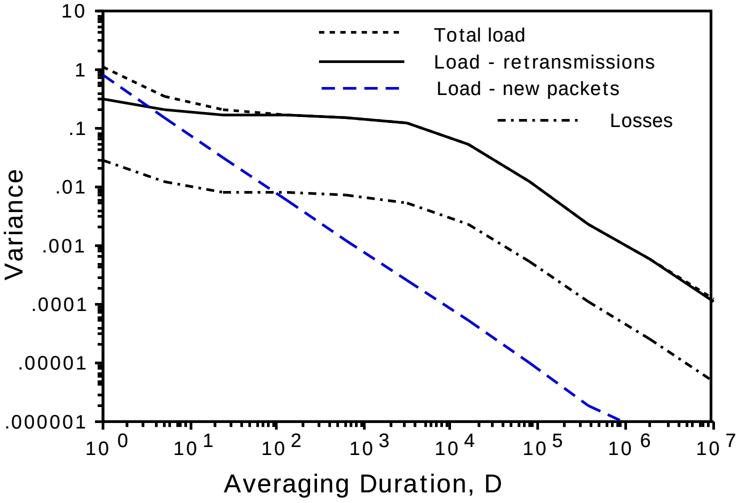

Figure 3: Variance of load from new arrivals, load from retransmissions, total load, and packet losses, versus averaging duration D. Load from new arrivals = .77. N=5. T_{to} =25. Buffer size = 10.

[3] This method (and our POX graphs) can overestimate values of H when H is high. However, our goal is not to derive an exact value for H. Variance-time graphs are valuable because they produce estimates of H that are accurate enough to determine whether a self-similar model would be appropriate (H>0.5), and more importantly, to view this effect at multiple time-scales. This approach also does not require assumptions regarding the underlying stochastic process (e.g. whether it is fractional Gaussian noise.)

Figure 4 shows the variance of load from retransmissions as a function of the averaging duration D for loads from new arrivals that range from .74 to .81. The pattern is essentially the same at all of these loads, although the appearance of self-similar behavior persists at slightly greater time scales when load (and therefore congestion) is greater.[4] Thus, there is no significant traffic smoothing when the number of independent traffic sources is increased. Each curve can be approximated as follows. For D ranging from 1 to around 10^4, there is a line with small slope, roughly -.15 to -.2. This would correspond to a Hurst value of approximately .9 if it persisted. For greater values of D, the slope becomes -1.

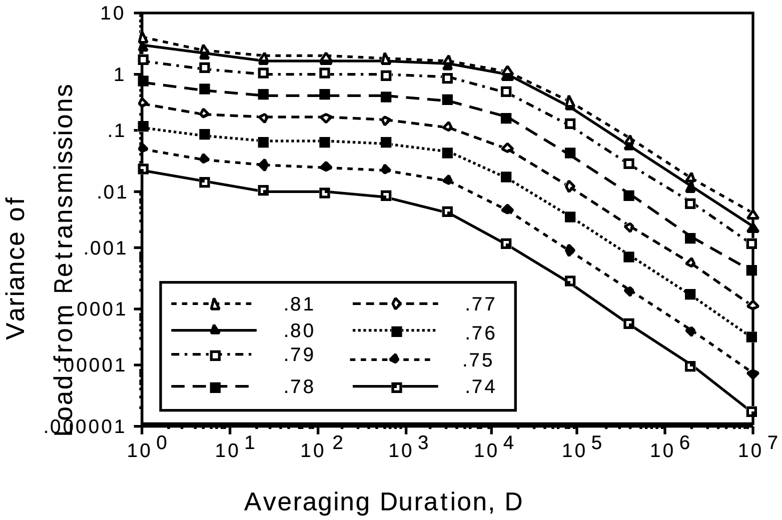

Figure 4: Variance of load from retransmissions versus averaging duration D. Load from new arrivals ranges from .74 to .81. N=5. T_{to} =25. Buffer size = 10.

Because slope appears to go to -1 as duration D goes to infinity, there is no indication that this meets the true definition of long-range dependence. Indeed, it is impossible to show empirically whether traffic on any of the networks observed is truly self similar overall time scales, because that would require examining variance over an infinitely long period. However, for time scales less than some large threshold, retransmission mechanisms create the appearance of self-similar behavior. If that threshold is sufficiently large, network performance would be roughly the same as if traffic were self similar [GRO99,NEI98]. If this threshold is at least two orders of magnitude greater than the time required to transmit a buffer's worth of packets, then based on [NEI98], this appears to be large enough.

Section 5: Impact of Key Parameters

Section 5.1: Buffer Size

One approach to reducing loss rate is to increase buffer size. Figure 5 shows the variance of load from retransmissions with buffer sizes of 5, 10, and 20. Loads are scaled

[4] This pattern obviously would not persist when congestion became so great that most packets are retransmitted the maximum number of times, but this change is not apparent until load is so high that the majority of packets are ultimately lost. Admission control and congestion control mechanisms try to prevent such conditions from occurring.

to maximize throughput, keeping retransmission and loss rates roughly constant for all three buffer sizes. While it is true that increasing buffer size from 10 to 20 allows the network to support 6.5% more traffic before the onset of congestion, the duration of a congested period increases dramatically with buffer size. This increases the time scales over which traffic appears to be self similar by a full order of magnitude. This is another reason why increasing buffer size is a relatively ineffective method of preventing congestion when traffic is self similar. Obviously, real networks have buffers much larger than 20, even if it is only to accommodate simple Markovian (e.g. Markov-Modulated Poisson Process - MMPP) traffic.

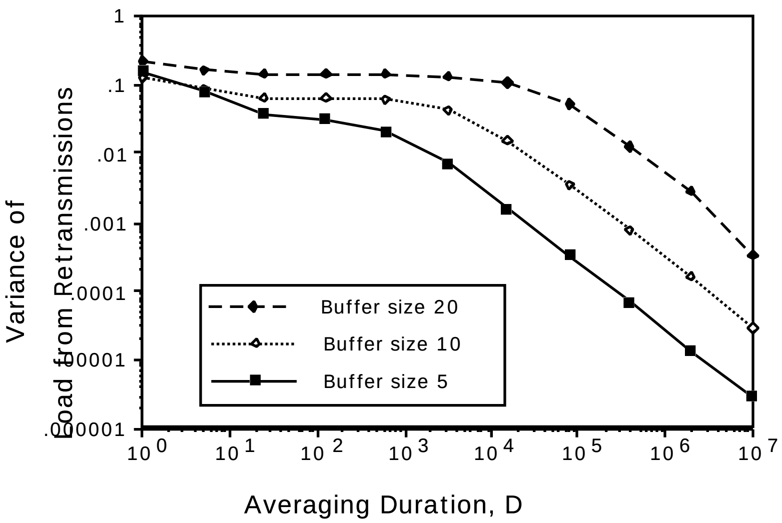

Figure 5: Variance of load from retransmissions versus averaging duration D. N=5. T_{to} =25. Buffer size = 5, 10, and 20, with loads of .69, .76, and .81.

Section 5.2: Number of Retransmission Attempts

The impact of the number of attempts N on self-similar behavior is also great. Increasing N while holding input load constant will increase the total number of retransmissions, thereby consuming more network capacity. Thus, with larger N, throughput is maximized at lower load. Figure 6 shows the variance of load from retransmissions as a function of D with N=5 and load from new traffic = .76, and with N=10 and load from new traffic = .66, both of which lead to comparable retransmission and loss rates. With N=10, traffic behavior appears self-similar for time scales up to roughly 10^5 or 10^6 - much greater than with N=5. Thus, merely doubling the maximum number of retransmission attempts prolongs the range where traffic appears to be self-similar by roughly two order of magnitude.

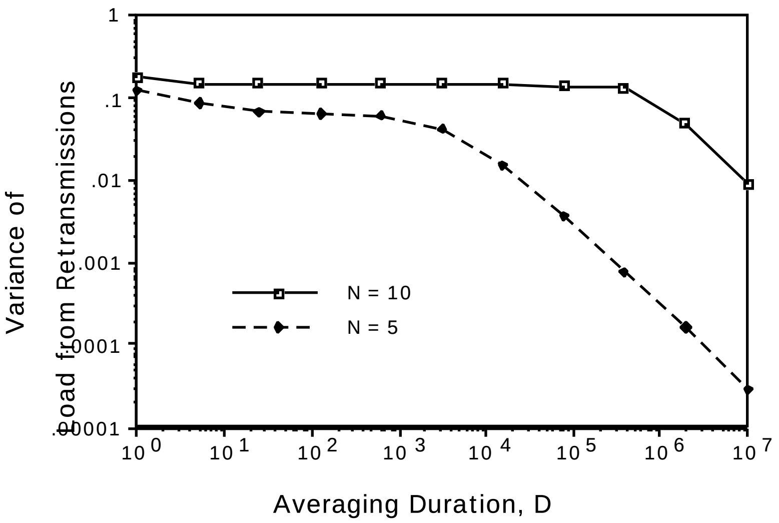

Figure 6: Variance of load from retransmissions versus averaging duration D. T_{to}=25. N=5 and N=10, with loads of .76 and .66. Buffer size = 10.

Section 5.3: Time-out Duration

Protocol designers have greater flexibility in setting time-out duration, and T_{to} =25 would be considered low in many networks. Time-out must exceed one round-trip time. For a wide-area network with propagation delay across the diameter of 30 ms, 3000-bit multi-cell packets, and 1.5 Mb/s links, T_{to} > 30. With 150 Mb/s links, T_{to} > 3000.

Figure 7 shows the variance of load from retransmissions as a function of the averaging duration D for time-out durations T_{to} of 25, 125, 625, and 3125, and loads from new arrivals of .78, .815, .82, and .82, respectively. These loads were chosen to keep throughput high, and to keep the mean load from retransmissions and loss rate roughly fixed across all time-out values. Consequently, the variance for very small values of D are also roughly the same for each curve, but they differ when D grows larger. For larger time-outs, self-similar behavior is observed at much greater time scales. Indeed, with T_{to}=3125, the slope still has not stabilized at -1 by D=10^7. For sufficiently large values of the time-out T_{to}, the slope of variance forms something closer to a step function; slope is in the typical -.2 neighborhood when D is small, until it briefly goes to -1, and then the curve flattens out again. One possible explanation is as follows: there is temporal autocorrelation of load in the short term because once the buffer is full, it tends to stay close to full for a while. Thus, if a packet arriving at time t must be retransmitted, chances are good that a packet arriving at time close to t will also be retransmitted. There is longer-term autocorrelation because if packet arriving at time t must be retransmitted, then its retransmission causes load to be higher at time t+T_{to}, so chances are greater that a packet arriving at time t+T_{to} must be retransmitted. When T_{to} is much larger than the duration of a short-term spike in load, these two effects are distinguishable as two separate plateaus in a variance-time plot. That is probably why when T_{to} = 3125, the second plateau in Figure 7 begins at D=3125. Of course, this second plateau occurs at a timescale that probably has much less impact on throughput for a system with a buffer this size. Thus, burstiness is less problematic with T_{to}=3125 than with T_{to}=25.

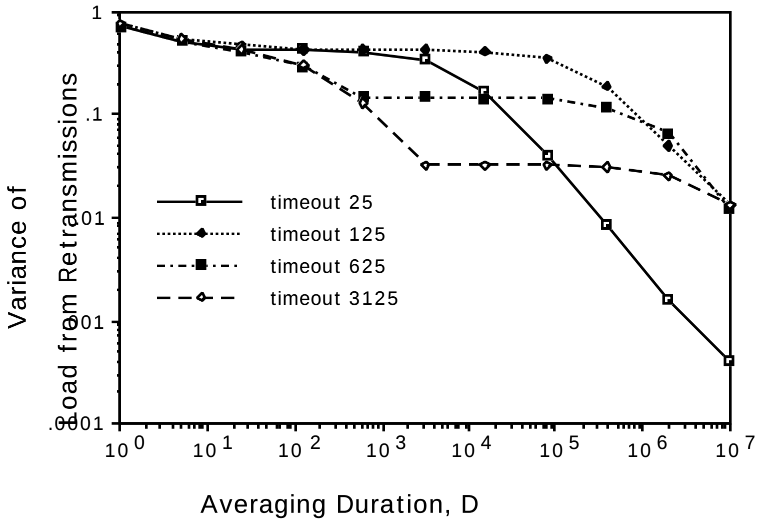

Figure 7: Variance of load from retransmissions versus averaging duration D. N=5. T_{to} =25, 125, 625, and 3125, with loads of .78, .815, .82, and .82. Buffer size = 10.

Time-out duration need not be fixed. One common approach is to use a binary exponential back-off, where the i'th consecutive time-out lasts T_{min} 2^{i-1} for some constant T_{min}. Indeed, when traffic comes from a small number of sources, this has the useful effect of cutting the traffic arrival rate until congestion subsides. However, we have addressed a scenario where traffic comes from a large number of low-data rate sources. Since sources that have not yet experienced loss do not slow down, this traffic reduction effect has little impact.

The other result of binary exponential back-off is quite apparent: to spread the traffic out over time when congestion occurs. This has the unfortunate effect of prolonging congestion and increasing the timescales over which traffic appears self similar. Figure 8 shows the variance of load as a function of duration D in such a system, as compared to deterministic time-outs of 25 and 125. Loads are again scaled to increase throughput and make retransmission and loss rates comparable. This binary exponential back-off scheme has a minimum timeout of 25 and a mean timeout of 88 in this scenario. Even though mean timeout is between the deterministic values of 25 and 125, the system with the binary exponential back-off cannot carry as heavy a load as the other two.[5] (In the case of a random access system with collisions such as ethernet or Aloha, it is possible that reinforcing the appearance of self similarity is an asset rather than a liability [ARA97], so this could sometimes be an advantage of binary exponential back-off.)

[5] Similarly, we found that a deterministic back-off of 125 allowed the system to support a greater load (81%) than a binary exponential back-off with T_{min} = 125 (78%).

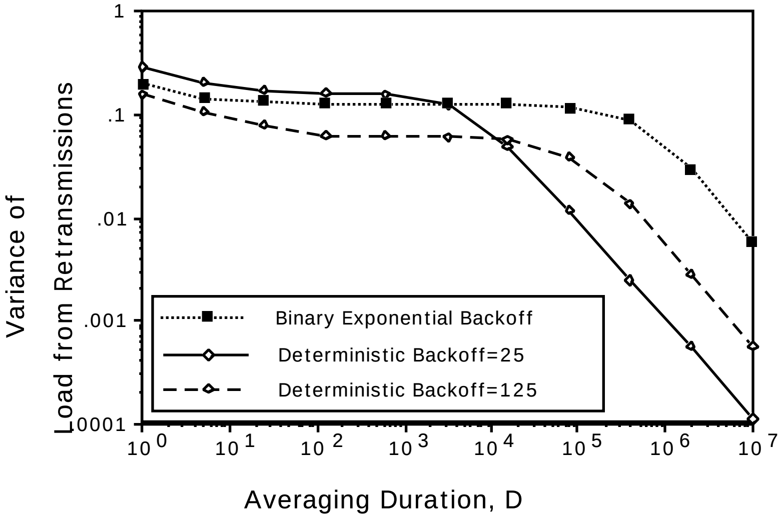

Figure 8: Variance of load from retransmissions versus averaging duration D. N=5. Deterministic backoff with T_{to} =25 and 125 and loads of .77 and .81. Binary exponential backoff with T_{min}=25 and load of .72. Buffer size = 10.

Another variation is to make timeout stochastic. Random time-outs can also have the effect of spreading retransmissions out over time. This has advantages and disadvantages. It reduces the likelihood that congestion will occur. However, when congestion does occur, it is steady, rather than in periodic bursts, so it lasts much longer. Figure 9 shows the variance of load in four systems where the mean timeout duration is 25: deterministic, exponentially distributed, uniformly distributed between 0 and 50, and uniformly distributed between 20 and 30. Because random time-outs prolong congestion, all three random time-outs appear self-similar over larger time scales than the deterministic time-out. This is especially true for the uniformly distributed timeout from 0 to 50; its curve is still flat at time scales of 5 10^7. In this scenario, the disadvantages of spreading retransmissions over time outweigh the advantages, so a heavier load can be supported with a deterministic time-out. In contrast, with the higher minimum timeout of 625, timeout variance helps: a deterministic timeout of 625 supports a load of about 82%, a timeout uniformly distributed between 500 and 750 supports a load of about 84.25%, and a timeout uniformly distributed between 0 and 1250 supported a load of about 85.75%.

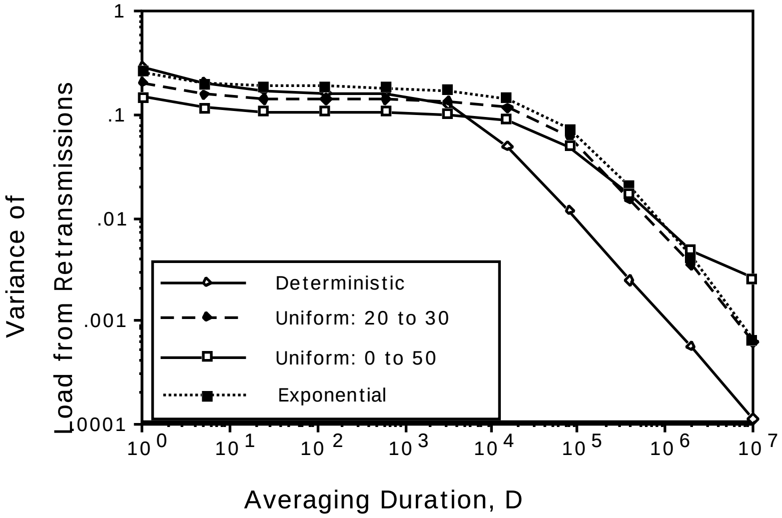

Figure 9: Variance of load from retransmissions versus averaging duration D. N=5. Buffer size = 10. Mean T_{to} =25. Deterministic timeout, exponentially distributed timeout, uniformly distributed timeout from 20 to 30, uniformly distributed timeout from 0 to 50. Loads of .77, .7, .71, and .6975.

Section 6: Conclusion

Empirical results have shown that network designers should consider the stochastic nature of the traffic carried by networks. This simulation study has shown that designers should also consider how the network's own protocols influence the nature of that traffic.

Even if input traffic is simple and Markovian, retransmissions are not. Retransmissions may be a small fraction of total traffic, but when load is set to maximize throughput, retransmissions are the primary source of burstiness. Retransmission mechanisms have the ability to make traffic appear self-similar over extended periods. We have seen those periods exceed the timeout period by more than six orders of magnitude. This phenomenon could have contributed to the apparent self-similar behavior that has been observed empirically in network traffic.

We deliberately assumed a simple and well-behaved system to see if self similarity or long-range dependence could become apparent even with Markovian inputs and mechanisms. Simplifying assumptions included Poisson arrivals, constant-length packets, a single queue rather than a network of queues, first-come-first-served scheduling which is less effective but reduces queueing delay variance, a simple retransmission mechanism without congestion control mechanisms, and no contention or collisions while waiting in the queue. We know that real input traffic is not Poisson, and that violation of this assumption and many of the other assumptions above have the potential to increase load variance, and/or prolong the chaotic behavior. Thus, the actual range of time scales at which self-similar behavior may be even worse than observed in this paper.

The possibility that protocols may contribute to the appearance of self similarity could force protocol designers to make difficult tradeoffs, particularly when some of the tools designers can use to make congestion less frequent or severe also increase the duration when congestion does occur. This is more important in networks carrying traffic for which average performance is a poor measure. For example, it is better to lose every tenth voice packet than to lose ten consecutive voice packets out of every hundred.

References

[ARA97] J. Aracil and L. Munoz, "Performance of Aloha Channels Under Self-Similar Input," *IEE Electronic Letters*, vol. 33, no. 8, Apr. 10, 1997.

[CHR97] K. J. Christensen and V. Ballingam, "Reduction of Self-Similarity by Application-Level Traffic Shaping," *Proc. 22nd IEEE Local Computer Networks Conf.*, Nov. 1997, pp. 511-8.

[CRO97] M. E. Crovella and A. Bestavros, "Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes," *IEEE/ACM Trans. on Networking*, vol. 5, no. 6, pp. 835-46, Dec. 1997.

[ERR94] A. Erramilli, W. Willinger, and P. Pruthi, "Fractal Traffic Flows in High-speed Communications Networks," *Fractals*, vol. 2, no. 3, pp. 409-12, 1994.

[FLO94] S. Floyd and A. Romanow, "Dynamics of TCP Traffic Over ATM Networks," *IEEE J. Selected Areas in Communications*, vol. 13, no. 4, pp. 633-41, May 1995.

[GRO99] M. Grossglauser and J. Bolot, "On the Relevance of Long-Range Dependence in Network Traffic," *IEEE Trans. Networking*, vol. 7, no. 5, pp. 629-40, Oct. 1999.

[JER97] J. L. Jerkins and J. L. Wang, "A Measurement of ATM Cell-Level Aggregate Traffic," *Proc. IEEE Globecom*, Nov. 1997, pp. 1589-95.

[LEL94] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, "On the Self-Similar Nature of Ethernet Traffic (Extended Version)," *IEEE/ACM Trans. on Networking*, vol. 2, no. 1, pp. 1-14, Feb. 1994.

[LUC97] M. T. Lucas, D. E. Wrege, B. J. Dempsey, and A. C. Weaver, "Statistical Characterization of Wide-Area IP Traffic," *Proc. 6th IEEE Intl. Computer Communications and Networks Conf.*, Sept. 1997, pp. 442-7.

[MEI91] K. S. Meier-Hellstern, P. E. Wirth, Yi-Ling Yan, and D. A. Hoeflin, "Traffic Models for ISDN Data Users: Office Automation Application," *Proc. 13th Intl. Teletraffic Congress*, June 1991, pp. 167-72.

[NEI98] A. Neidhardt and J. L. Wang, "The Concept of Relevant Time Scales and Its Application to Queuing Analysis of Self-Similar Traffic," *Proc. ACM Sigmetrics*, 1998, pp. 222-32.

[PAR96] K. Park, G. Kim, and M. Crovella, "On the Relationship Between File Sizes, Transport Protocols, and Self-Similar Network Traffic," *Proc. IEEE Intl. Conf. on Network Protocols*, Oct. 1996, pp.171-80.

[PAR97] K. Park, "On the Effect and Control of Self-Similar Network Traffic: A Simulation Perspective," *Proc. Winter Simulation Conference,* 1997, pp. 989-96.

[PAX95] V. Paxson and S. Floyd, "Wide Area Traffic: The Failure of Poisson Modeling," *IEEE/ACM Trans. on Networking*, vol. 3, no. 3, pp. 226-44, June 1995.

[PEH97] J. M. Peha, "Retransmission Mechanisms and Self-Similar Traffic Models," *Proc. IEEE/ACM/SCS Communication Networks and Distributed Systems Modeling and Simulation Conf.*, Jan. 1997, pp. 47-52.

[VER00] A. Veres and M. Boda, "The Chaotic Nature of TCP Congestion Control," *Proc. IEEE Infocom,* March 2000, pp. 1715-23.

[WIL97] W. Willinger, M. S. Taqqu, R. Sherman, and D. V. Wilson, "Self-Similarity Through High-Variability: Statistical Analysis of Ethernet LAN Traffic at the Source Level," *IEEE/ACM Trans. on Networking*, vol. 5, no. 1, pp. 71-86, Feb. 1997.