Proceedings of the 1997 Winter Simulation Conference
ed. S. Andradóttir, K. J. Healy, D. H. Withers, and B. L. Nelson

ON THE EFFECT AND CONTROL OF SELF-SIMILAR NETWORK TRAFFIC: A SIMULATION PERSPECTIVE

Kihong Park

Department of Computer Sciences
Purdue University
West Lafayette, IN 47907, U.S.A.

ABSTRACT

This paper presents a discussion of simulation-related issues arising in the study of self-similar network traffic with respect to its effect and control. Self-similar traffic has been shown to be an ubiquitous phenomenon arising in diverse networking contexts with potentially adverse effects on network performance.

In many instances, an experimental or empirical approach needs to be taken to effectively evaluate the performance impact of sophisticated control algorithms acting at various layers in the protocol stack under self-similar traffic conditions. Simulating or experimentally implementing such environments is nontrivial due to the fact that, in general, the characteristics of the observed traffic is itself influenced by the actions of the control algorithms under study. To what degree self-similarity manifests itself in network traffic may depend on the properties of the protocols employed, and trace-based simulations that rely on traffic measurements to drive simulations fail to capture this dynamic aspect.

We discuss an approach to evaluating control protocols under self-similar traffic conditions based on a simple, robust application-level causal mechanism of traffic self-similarity which is grounded in both empirical UNIX file system research and analytic traffic models involving certain renewal processes. We present a high-level discussion concentrating on simulation-related issues, with specific research results summarized or pointed to in the references.

1 INTRODUCTION

Starting with the seminal study by Leland et al. (1993), a flurry of network measurement-based work has been carried out (Crovella and Bestavros, 1996, Garret and Willinger, 1994, Huang et al., 1995, Paxson and Floyd, 1994, Willinger et al., 1995) which show that traffic self-similarity is an ubiquitous phenomenon arising in diverse networking contexts. From a queueing theory perspective, the principal distinguishing characteristic of long-range dependent traffic is that the queue length distribution decays much more slowly—i.e., polynomially—vis-à-vis short-range-dependent traffic sources such as Poisson sources which exhibit exponential decay.

A number of performance studies (Adas and Mukherjee, 1995, Addie et al., 1995, Likhanov and Tsybakov, 1995, Norros, 1994, Park et al., 1997) have shown that self-similarity has a detrimental effect on network performance leading to increased packet loss rate, delay, and a degraded delay-throughput tradeoff relation. In Grossglauser and Bolot (1996) and Ryu and Elwalid (1996), the point is advanced that for small buffer sizes or short time scales, long-range dependence has only a marginal impact. This is as expected since low frequency components—relative to buffer size and time scale—in the frequency spectrum act like "DC-components," and burstiness, which is an important factor in determining performance, is dominated by high-frequency components which collectively tend to exhibit short-range correlations.

The work by Park et al. (1996) is interesting from the perspective that it shows *why* traffic self-similarity may be such an ubiquitous phenomenon. A simple, robust application-level causal mechanisms is advanced that in a generic network environment and in the presence of a typical protocol stack invariably induces traffic self-similarity at the link level where traffic is multiplexed. In a nutshell, all that is required for self-similarity to occur is that the application objects being communicated over a network have a propensity to be extremely large with nonnegligible frequency.

More precisely, if the remote access of application objects is modeled as random sampling from a probability distribution whose random variable represents *object size*, then the *heavy-tailedness* of object size distribution in the sense of hyperbolically decaying tail is a sufficient condition for inducing traffic self-similarity. In addition to the comprehensive demonstration of the causal link via controlled simulations in Park et al. (1996), the causality relation is supported by empirical evidence from past UNIX file system research (Satyanarayanan, 1981, Smith, 1981, Ousterhout et al., 1985, Baker et al., 1991, Bodnarchuk and Bunt, 1991, Ramakrishnan et al., 1992, Peterson and Grossman, 1995, Peterson, 1996) going back to the early '80s which indicate that UNIX file systems tend to be approximately heavy-tailed. The increasing popularity of multimedia communication is expected to make the situation even worse. On the analytical side, the conclusion is supported by the ON/OFF model of Willinger et al. (1995) which has its roots in some early work by Mandelbrot (1969) which shows that aggregating a large number of 0/1 renewal reward processes with heavy-tailed interrenewal times produces self-similarity.

Given the importance of properly reflecting traffic self-similarity when performing simulation-based network performance studies, this paper discusses how this may be effectively done. This is a nontrivial problem due to the fact that, in general, the characteristics of the observed traffic is itself a function of the actions of the control protocols under study. If a network model is adopted where self-similarity is delimited to a background traffic process with a priori fixed characteristics, then it is straightforward to study the behavior of such a shared queue by generating a self-similar time series with suitable properties and feeding it to the queue independent of network state.

If, however, a system perspective is adopted whereby self-similarity is not the sole property of an externally imposed traffic source but an induced phenomenon of the totality of controlled traffic streams under study, then clearly the simplistic approach is inadequate. To what degree self-similarity manifests itself in network traffic may depend on the properties of the protocols employed, and trace-based simulations that rely on traffic measurements to drive simulations fail to capture this dynamic aspect. What we need is a way, one, to ascertain the influence of the protocol stack on the characteristics of multiplexed downstream traffic, and two, to *induce* traffic self-similarity without designating one or more traffic sources as conduits of fixed self-similar time series.

The aforementioned application-level causal mechanism satisfies both requirements and lends itself to straightforward realization in simulation- as well as implementation-based contexts. In fact, a trace-based approach can still be adopted, however, with traces collected at the *application layer* capturing access patterns to application objects. The latter can be generalized to include real-time video/audio streams as well as interactive applications.

The rest of the paper is organized as follows. In Section 2, we give a brief, self-contained overview of the main definitions needed to follow the discussion on self-similar traffic simulation. Section 3 presents the two modes of incorporating self-similarity in simulations and discusses their relative merits. Section 4 discusses protocol stack control characteristics.

2 PRELIMINARIES

Let $(X_t)_{t\in\mathbb{Z}_+}$ be a time series which, for example, represents the trace of data flow at a bottleneck link measured at some fixed time granularity. We define the aggregated series $X_i^{(m)}$ as

$$X_i^{(m)} = \frac{1}{m}(X_{im-m+1} + \cdots + X_{im}).$$

That is, X_t is partitioned into blocks of size m, their values are averaged, and i is used to index these blocks.

Let $r(k)$ and $r^{(m)}(k)$ denote the autocorrelation functions of X_t and $X_i^{(m)}$, respectively. X_t is *self-similar*—more precisely, *asymptotically second-order self-similar*—if the following conditions hold:

$$r(k) \sim \text{const} \cdot k^{-\beta}, \qquad (1)$$

$$r^{(m)}(k) \sim r(k), \qquad (2)$$

for k and m large where $0 < \beta < 1$. That is, X_t is "self-similar" in the sense that the correlation structure is preserved with respect to time aggregation—relation (2)—and $r(k)$ behaves hyperbolically with $\sum_{k=0}^{\infty} r(k) = \infty$ as implied by (1). The latter property is referred to as *long-range dependence*.

Let $H = 1 - \beta/2$. H is called the *Hurst parameter*, and by the range of β, $1/2 < H < 1$. It follows from (1) that the farther H is away from $1/2$ the more long-range dependent X_t is, and vice versa. Thus the Hurst parameter acts as an indicator of the degree of self-similarity.

A test for long-range dependence can be obtained by checking whether H significantly deviates from $1/2$ or not. We use two methods for testing this condition. The first method, the *variance-time plot,* is based on the slowly decaying variance of a self-similar time series. The second method, the R/S plot, uses the fact that for a self-similar time series, the *rescaled range* or R/S statistic grows according to a power law with exponent H as a function of the number of points included. Thus the plot of R/S against this number on a log-log scale has a slope which is an estimate of H. A comprehensive discussion of the estimation methods can be found in Beran (1994) and Taqqu et al. (1995).

A random variable X has a *heavy-tailed* distribution if

$$\Pr\{X > x\} \sim x^{-\alpha}$$

as $x \to \infty$ where $0 < \alpha < 2$. That is, the asymptotic shape of the tail of the distribution obeys a power law. The *Pareto* distribution,

$$p(x) = \alpha k^{\alpha} x^{-\alpha-1},$$

with parameters $\alpha > 0$, $k > 0$, $x \geq k$, has the distribution function

$$\Pr\{X \leq x\} = 1 - (k/x)^{\alpha},$$

and hence is clearly heavy-tailed.

It is not difficult to check that for $\alpha \leq 2$ heavy-tailed distributions have infinite variance, and for $\alpha \leq 1$, they also have infinite mean. Thus, as α decreases, a large portion of the probability mass is located in the tail of the distribution. In practical terms, a random variable that follows a heavy-tailed distribution can take on extremely large values with nonnegligible probability.

3- NETWORK MODELS AND TRAFFIC SELF-SIMILARITY

This section discusses two methods for incorporating traffic self-similarity in network performance evaluations, be they by simulation, implementation, or analytical.

3.1- Self-similar Background Traffic

The most straightforward way of evaluating a control protocol's effectiveness at managing a traffic stream is to subject the traffic stream to the interference effect of a priori fixed self-similar background traffic. This set-up is depicted in Figure 1.

The controlled traffic $\xi(t)$ is generated by source S which also modulates traffic flow as a function of observed network state via feedback. The self-similar background traffic $\zeta(t)$ is a fixed sequence, independent of network state, and it can be generated in a number of ways including traditional time series methods (Samorodnitsky and Taqqu, 1994), using the ON/OFF model (Willinger et al., 1995) to superpose a number of $0/1$ renewal reward processes with heavy-tailed interrenewal times, or the use of traces obtained from actual traffic measurements (Leland et al., 1993, Paxson and Floyd, 1994).

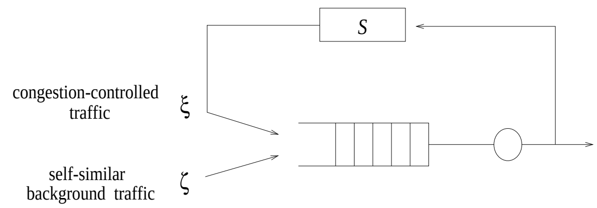

Figure 1:- Open System with Queue Shared by Congestion-Controlled Traffic ξ and Self-Similar Background Traffic ζ

The model in Figure 1 is best suited for environments where ξ has negligible influence on ζ. Such is the case, for example, when ζ represents reserved traffic and ξ is served in a best-effort manner, or ξ's traffic rate is miniscule compared to ζ and network bandwidth. An instance of the former is a study of an adaptive algorithm for packet-level forward error-correction called AFEC (Park, 1997a). The adaptive forward error-correction problem is analyzed with respect to optimality and stability, and its performance is evaluated under bursty traffic conditions when the cross traffic is treated as reserved. The effectiveness of packet-level FEC for "small" object transport vis-à-vis ARQ schemes with respect to throughput is investigated in Park (1997b). Part of the simulation set-up for incorporating self-similar traffic conditions follows the background traffic model of Figure 1.

The background traffic model is inadequate if ζ is significantly affected by ξ. From another perspective, one may wish to investigate a closed system consisting of n traffic sources $\xi_1, \ldots, \xi_n$ where each ξ_i exerts a nontrivial influence on the other traffic streams. The control protocols regulating the ξ_i's may be as complicated as TCP with its various control features (Jacobson, 1988), leading to nonlinear coupling among the n traffic streams.

3.2- Induced Traffic Self-similarity

Figure 2 shows a queueing system consisting of n traffic sources $\xi_1, \xi_2, \ldots, \xi_n$ where each ξ_i depends on network state via a feedback-loop. In addition to the requirement that each traffic stream ξ_i obey a closed-loop control encapsulated by S^i, we also require that none of the $\xi_i(t)$ be self-similar to begin with. That is, any possible self-similarity of the downstream traffic must not be attributable to one or more traffic streams $\xi_i(t)$ being outright self-similar from the outset. Compressed MPEG video, for example, has been shown to possess self-similar characteristics (Garret and Willinger, 1994, Huang et al., 1995, Heyman and Lakshman, 1996), hence, single-source generated self-similarity is clearly possible.

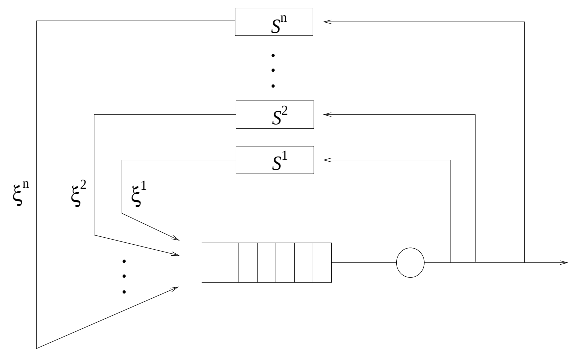

Figure 2: Closed Feedback System with Queue Shared by n Congestion-Controlled Traffic Streams $\xi^1, \xi^2, \ldots, \xi^n$

However, an interesting question arises whether self-similar traffic can arise in a network system subject to a typical protocol stack without any of the constituent traffic streams being self-similar to begin with. This has been answered in the positive by Park et al. (1996) in the form of a simple, robust application layer causal mechanism, namely, the communication of application objects with heavy-tailed object size distributions in a generic internetwork. This causal link, which was established via controlled simulations, is also supported by empirical evidence from UNIX file system research which show that UNIX file systems tend to be approximately heavy-tailed, and by the ON/OFF traffic model of Willinger et al. (1995) which shows that aggregating a large number of 0/1 renewal reward processes with heavy-tailed interrenewal times produces self-similarity.

The principal benefit of using the network model in Figure 2 is as follows. Since traffic self-similarity is a structural application layer property of generic networked distributed systems with heavy-tailed object size distributions, the control algorithms in the protocol stack can be changed and their effect evaluated without disturbing the intrinsic disposition of the system to induce self-similar traffic. Moreover, the strength of the system's disposition to induce self-similarity can be controlled by adjusting the degree of heavy-tailedness of the application layer object size distribution. In the case of the Pareto distribution, this is affected by varying the shape parameter α. In fact, in extreme situations where an overly aggressive form of transport based on UDP is employed (Park et al., 1996), self-similarity of the multiplexed downstream traffic can be destroyed leaving a traffic pattern whose Hurst parameter has significantly decreased from near 1. The separation of the propensity of a networked system to generate self-similar traffic from the control characteristics of a protocol stack and its mechanics allows protocols to be devised to combat the detrimental impact of self-similarity and evaluate their effectiveness.

Figure 3 shows throughput traces for simulations under the set-up of Figure 2 when the application layer causal mechanism was active with α being the control variable of heavy-tailedness of the Pareto distribution. The exact topology consisted of n clients making remote accesses to two servers connected at the other end of the queue or router. The top row of the figure shows a single client simulation ($n = 1$) whereas the bottom row shows traces of $n = 32$ client simulations. The first three columns depict traces as α is varied from 1.05, 1.35, to 1.95. The last column depicts throughput traces when an exponential distribution is used for the object size distribution. The means of the distributions across the different cases were held constant to achieve comparability.

The traces were recorded at 10 ms granularity, and the aggregated time series at the 100 sec level—a four-fold order of magnitude increase in aggregation—are shown. Using the notation of Section 2, if $X_i = X_i^{(1)}$ denotes the time series at 10 ms granularity, then Figure 3 shows the time series $X_i^{(10000)}$, however, not normalized by 10000, i.e., $10000 \cdot X_i^{(10000)}$. The mean object size was set at ≈ 4 KB and the service rate was set at T1 link speed. The protocol stack consisted of a transport layer emulating TCP Reno which in turn ran on top of an internetworking layer. The `ns` simulation package from Lawrence Berkeley National Laboratory, suitably modified, was used for this purpose.

The first three columns of the top row show that a single traffic stream or client produces burstiness at the 100 sec aggregation level when α is close to 1 (very heavy-tailed), however, the bursts are separated by flat plateaus which resemble the traffic characteristics of the $\alpha = 1.35, 1.95$ runs. Although not shown here for brevity, the single client run exhibits only weak self-similarity across the aggregation levels 10 ms, 100 ms, 1 sec, 10 sec, 100 sec. Not so for the 32 client runs. Aggregation over multiple clients—i.e., multiplexing—produces scale-invariant burstiness patterns (again not shown here for brevity) which are most pronounced when α is close to 1. This is can also be seen in Figure 4 which shows the H estimates for the 32 client runs based on the R/S and V-T methods described in Section 2. Clearly, we observe a linear relationship between H and the shape

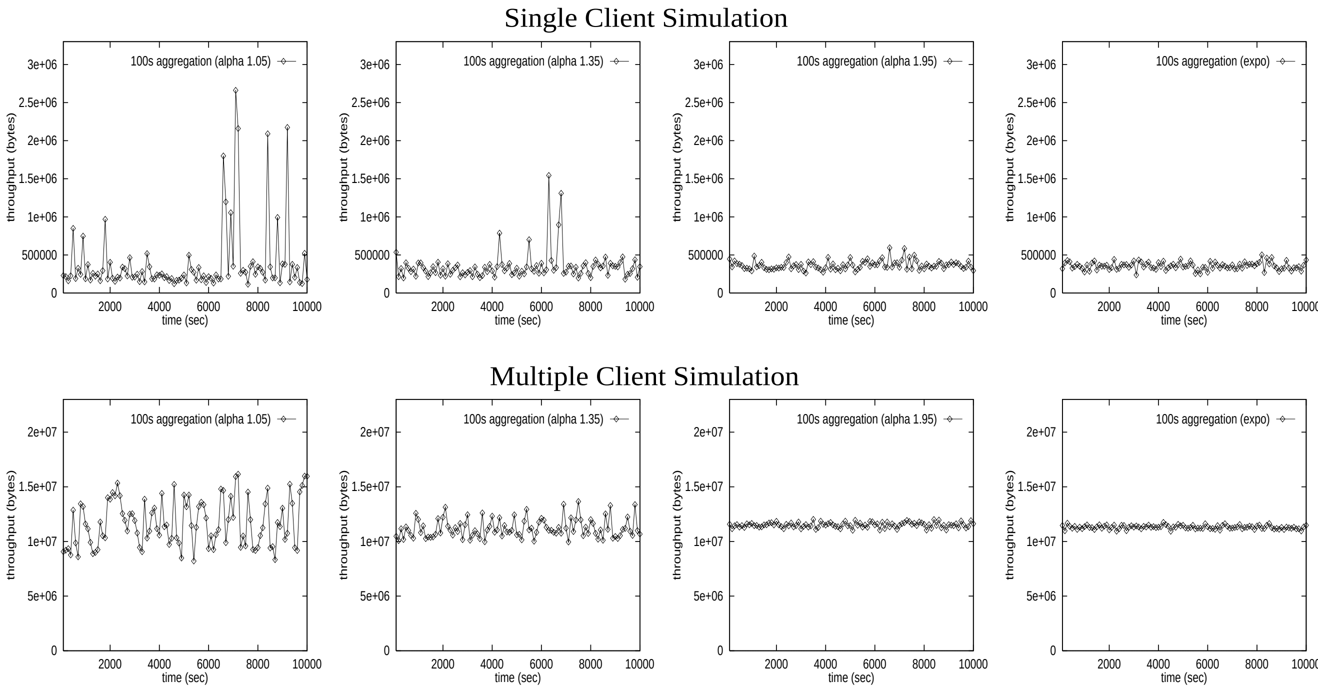

Figure 3: Throughput Trace of Simulation Under the Set-Up of Figure 2 with Application-Level Causality and TCP as Transport Protocol. Top Row: Traffic Trace of Single Client Simulation ($n = 1$). Bottom Row: Traffic Trace of $n = 32$ Client Simulation

parameter α. The more heavy-tailed the object size distribution, the more self-similar the resultant network traffic.

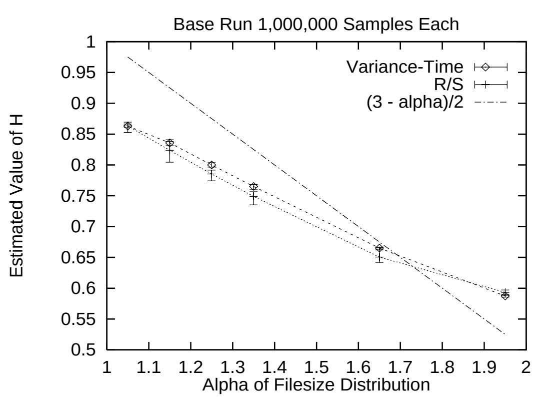

Figure 4: Hurst Parameter Estimates (R/S and V-T) for α Varying from 1.05 to 1.95

The last two columns of Figure 3 show that as far as the characteristics of aggregate traffic is concerned, there is little difference between using a "weakly" heavy-tailed distribution ($\alpha \approx 2$) and an exponential distribution of the same mean. This is also confirmed by their Hurst parameter values (Park et al., 1996). Mathematically, however, their properties are still worlds apart. This points toward the usefulness of using heavy-tailed object size distributions with α as the control parameter for determining how strongly a system is predisposed to generating self-similar traffic.

4 PROTOCOL STACK CONTROL ACTIONS AND THEIR PROPERTIES

Now that we have discussed two methods for incorporating self-similarity in network simulation models and shown the usefulness of the *Induced Traffic Self-similarity* Model at evaluating control algorithms acting in the protocol stack, we will take a closer look at issues surrounding traffic management under self-similar traffic conditions.

Figure 5 depicts the application layer-based causal mechanism of self-similarity and its mediation by the protocol stack. The predisposition of a networked distributed system to produce self-similar traffic patterns is isolated in the access characteristics to application objects in the application layer. The remote access of application objects—a high-level abstraction—is mediated by control planes in the protocol stack which can influence the induced traffic characteristics including to what degree multiplexed traffic is self-similar.

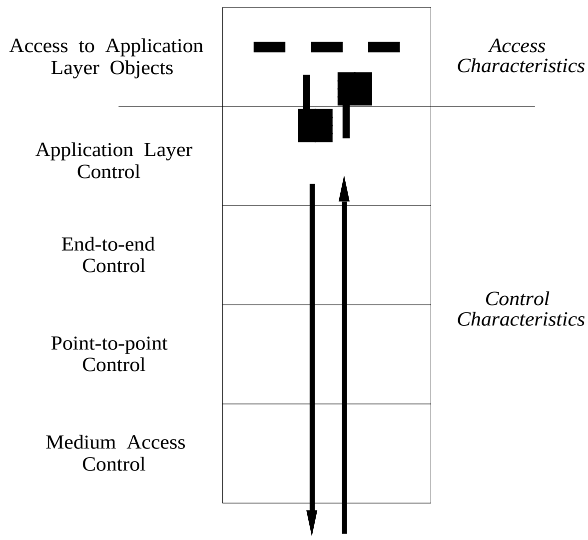

Figure 5: Application-Level Causality and Influence of Protocol Stack

4.1 Application Layer Control

The presence of scale-invariant burstiness in a network system across a wide range of time scales is determined by the degree of heavy-tailedness of application object size distributions. The closer the exponent α is to 1, the more of a factor the tail of the distribution becomes, generating larger and larger objects with nonnegligible probability.

The influence of remote access of large objects can be seen in the single client run for $\alpha = 1.05$ in Figure 3 (top row, left-most plot). The high spikes representing instances of large object transfers, when multiplexed with other traffic streams, collectively conspire to produce the bursty picture of the 32 client run in Figure 3 (bottom row, left-most plot). This, in turn, has a direct negative impact on system utilization and performance when compared to the smooth aggregation achieved for large α runs or the exponential object size distribution run (the right-most two colums of Figure 3). This leads to the heuristic

(H-1) *avoid the transfer of large objects whenever possible*

which can be implemented in the application layer using three mechanisms:

- caching,
- demand-based paging,
- information summaries.

Caching, in this context, refers to client-side caching since the objective is to minimize unnecessary object transfers. "Demand-based paging" refers to the policy of fetching a divisible object's needed segment and generalizes its usage in operating systems. This requires, in part, that network applications be able to interpret incomplete, segmented objects by padding and other techniques. Such capabilities are already present, e.g., in web browsers such as Netscape. On the surface, it may seem that prefetching is an incompatible policy with the objective (H-1) at hand. However, as long as prefetching is not excessive and only small segments of objects are transferred, they should not boost self-similarity significantly.

Information summaries are a way of creating useful information proxies of large objects such that the "information content" conveyed by a proxy object is close to that of the full-fledged object itself with respect to most application accesses. Thumbnails of images, ASCII versions of postscript files, and non-graphic versions of HTML documents are instances of this strategy.

4.2 End-to-end Control

The issue of end-to-end control from the flow control perspective has been investigated in Park et al. (1996). Transport protocols such as TCP which implement both reliability and congestion control are apt at mediating the application layer causal mechanism such that the long-range correlation structure nascent in large object transfers over resource-bounded networks is preserved. Reliability represents a form of information conservation whereas congestion control (Park, 1993) tries to achieve a form of "speediness." If the latter is too aggressive without conserving information through retransmission of dropped packets, the net effect is to erode the correlation structure of prolonged transmissions resulting in multiplexed traffic with diminished self-similarity. A related heuristic is:

(H-2) *affect speedy transfer, whenever possible, to reduce long-range dependence.*

Conversely, slowing down the rate of transmission of the bits of an object does not reduce traffic self-similarity.

An alternative to ARQ-based transport schemes is to use packet-level forward error-correction (Biersack, 1993). The issue of adaptively applying FEC as a function of network state for the transport of QoS-sensitive multimedia data has been investigated in Park (1997a) with analysis of optimality and stability, and evaluation of performance under bursty traffic conditions. For packet loss-dominated systems, it is shown that AFEC—an adaptive forward error-correction protocol—is able to achieve near constant QoS by adaptively varying the level of redundancy. For delay-dominated systems, intrinsic limitations exist stemming from instability considerations and the level of burstiness. The effectiveness of packet-level FEC for "small" object transport vis-à-vis ARQ schemes with respect to throughput is investigated in Park (1997b). Under highly bursty traffic conditions, it is shown that employing FEC can improve both delay (i.e., response time) and throughput vis-à-vis ARQ. However, this holds under the individual traffic model shown in Figure 1. If a system perspective is taken using the model in Figure 2, then a zero-sum law applies consistent with the conclusions in Biersack (1993).

4.3 Point-to-point Control

The main feature of point-to-point control is routing (e.g., IP) and to some degree flow control which manifests itself as ARQ-based protocols in the data link layer. With respect to routing, variants of shortest-paths algorithms already seek "good" paths where bandwidth is reflected. From the self-similarity perspective, one may choose to emphasize high-bandwidth paths (even with longer propagation latencies) over low-bandwidth paths with short propagation delays assuming other things being equal. Link-based flow control issues—also recently advanced by Kung et al. (1994) for ATM congestion control—are similar in nature to congestion control issues with respect to their effect on self-similarity and are not discussed separately.

4.4 Medium Access Control

There exist a variety of medium access control (MAC) protocols from the popular CSMA/CD for Ethernet to token-based bus arbitration in FDDI to TDMA-based schemes in T1 links and wireless communication. For some contention resolution schemes such as CSMA/CD, it is theoretically possible, as with feedback congestion control protocols, to generate such high levels of contention—i.e., collision rate—so as to significantly drop the effective throughput of the LAN medium and thus its outflux of traffic to the outside world. However, such extreme scenarios are incompatible with actual operating conditions faced by most LAN systems and little additional influence on self-similarity can be attributed to medium access controls.

5 CONCLUSION

This paper has provided a high-level, overview discussion of simulation-related issues as they pertain to the evaluation of control protocols and their performance effects under self-similar traffic conditions. Many of the issues raised—e.g., application layer control and medium access control—can benefit from further simulations and experiments to provide an ever-expanding library of specific networking contexts for which the mechanics of self-similar traffic generation and its dependence on the protocol stack is verified. This paper, hopefully, serves to facilitate this process by giving some guidelines, albeit nonexhaustive, on what to watch out for when dealing with self-similar network traffic and its performance issues.

REFERENCES

Adas, A. and Mukherjee, A. 1995. On resource management and QoS guarantees for long range dependent traffic. In *Proc. IEEE INFOCOM '95*, pages 779–787.

Addie, R., Zukerman, M., and Neame, T. 1995. Fractal traffic: measurements, modelling and performance evaluation. In *Proc. IEEE INFOCOM '95*, pages 977–984.

Baker, M. G., Hartman, J. H., Kupfer, M. D., Shirriff, K. W., and Ousterhout, J. K. 1991. Measurements of a distributed file system. In *Proceedings of the Thirteenth ACM Symposium on Operating Syst em Principles*, pages 198–212.

Beran, J. 1994. *Statistics for Long-Memory Processes.* Monographs on Statistics and Applied Probability. Chapman and Hall, New York, NY.

Biersack, E. 1993. Performance evaluation of forward error correction in an ATM environment. *IEEE J. Select. Areas Commun.*, 11(4):631–640.

Bodnarchuk, R. R. and Bunt, R. B. 1991. A synthetic workload model for a distributed system file server. In *Proceedings of the 1991 SIGMETRICS Conference on Measurement and Modeling of Computer Systems*, pages 50–59.

Crovella, M. and Bestavros, A. 1996. Self-similarity in world wide web traffic: Evidence and possible causes. In *Proceedings of the 1996 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems.*

Garret, M. and Willinger, W. 1994. Analysis, modeling and generation of self-similar VBR video traffic. In *Proc. ACM SIGCOMM '94*, pages 269–280.

Grossglauser, M. and Bolot, J.-C. 1996. On the relevance of long-range dependence in network traffic. In *Proc. ACM SIGCOMM '96*, pages 15–24.

Heyman, D. and Lakshman, T. 1996. What are the implications of long-range dependence for VBR-video traffic engineering? *IEEE/ACM Transactions on Networking*, 4(3):301–317.

Huang, C., Devetsikiotis, M., Lambadaris, I., and Kaye, A. 1995. Modeling and simulation of self-similar variable bit rate compressed video: a unified approach. In *Proc. ACM SIGCOMM '95*, pages 114–125.

Jacobson, V. 1988. Congestion avoidance and control. In *Proc. ACM SIGCOMM '88*, pages 314–329.

Kung, H. T., Blackwell, T., and Chapman, A. 1994. Credit-based flow control for ATM networks: credit update protocol, adaptive credit allocation, and statistical multiplexing. In *Proc. SIGCOMM '94*, pages 101–114.

Leland, W., Taqqu, M., Willinger, W., and Wilson, D. 1993. On the self-similar nature of ethernet traffic. In *Proc. ACM SIGCOMM '93*, pages 183–193.

Likhanov, N. and Tsybakov, B. 1995. Analysis of an ATM buffer with self-similar ("fractal") input traffic. In *Proc. IEEE INFOCOM '95*, pages 985–992.

Mandelbrot, B. B. 1969. Long-run linearity, locally gaussian processes, h-spectra and infinite variances. *Intern. Econom. Rev.*, 10:82–113.

Norros, I. 1994. A storage model with self-similar input. *Queueing Systems*, 16:387–396.

Ousterhout, J. K., Costa, H. D., Harrison, D., a Kunze, J., Kupfer, M., and Thompson, J. G. 1985. A trace-driven analysis of the UNIX 4.2 BSD file system. In *Proceedings of the Tenth ACM Symposium on Operating System Pr inciples*, pages 15–24.

Park, K. 1993. Warp control: a dynamically stable congestion protocol and its analysis. *Journal of High Speed Networks*, 2(4):373–404.

Park, K. 1997a. AFEC: an adaptive forward error-correction protocol and its analysis. Technical Report CSD-TR-97-038, Department of Computer Sciences, Purdue University.

Park, K. 1997b. Self-similar network traffic and its control. To appear in *Proc. 25th Allerton Conference on Communication, Control and Computing*.

Park, K., Kim, G., and Crovella, M. 1996. On the relationship between file sizes, transport protocols, and self-similar network traffic. In *Proc. IEEE International Conference on Network Protocols*, pages 171–180.

Park, K., Kim, G., and Crovella, M. 1997. On the effect of traffic self-similarity on network performance. To appear in *Proc. SPIE International Conference on Performance and Control of Network Systems*.

Paxson, V. and Floyd, S. 1994. Wide-area traffic: the failure of Poisson modeling. In *Proc. ACM SIGCOMM '94*, pages 257–268.

Peterson, D. and Grossman, R. 1995. Power laws in large shop DASD I/O activity. In *CMG Proceedings*, pages 822–833.

Peterson, D. L. 1996. Data center I/O patterns and power laws. In *CMG Proceedings*.

Ramakrishnan, K. K., Biswas, P., and Karedla, R. 1992. Analysis of file I/O traces in commercial computing environments. In *Proceedings of the 1992 SIGMETRICS Conference on Measurement and Modeling of Computer Systems*, pages 78–90.

Ryu, B. and Elwalid, A. 1996. The importance of long-range dependence of VBR video traffic in ATM traffic engineering: myths and realities. In *Proc. ACM SIGCOMM '96*, pages 3–14.

Samorodnitsky, G. and Taqqu, M. 1994. *Stable Non-Gaussian Random Processes: Stochastic Models with Infinite Variance*. Chapman and Hall, New York, London.

Satyanarayanan, M. 1981. A study of file sizes and functional lifetimes. In *Proceedings of the Eighth ACM Symposium on Operating System P rinciples*.

Smith, A. J. 1981. Analysis of long term file reference patterns for application to file migration algorithms. *IEEE Transactions on Software Engineering*, 7(4):403–410.

Taqqu, M. S., Teverovsky, V., and Willinger, W. 1995. Estimators for long-range dependence: an empirical study. Preprint.

Willinger, W., Taqqu, M., Sherman, R., and Wilson, D. 1995. Self-similarity through high-variability: statistical analysis of Ethernet LAN traffic at the source level. In *Proc. ACM SIGCOMM '95*, pages 100–113.

AUTHOR BIOGRAPHY

KIHONG PARK is an Assistant Professor in the Department of Computer Sciences at Purdue University. He received his B.A. from Seoul National University in 1988, and a Ph.D. degree in computer science from Boston University in 1996. His research interests center around the design, analysis, and implementation of algorithms for controlling network resources in large-scale, high-speed network environments. Of special interest are optimality, stability, and efficiency issues arising in congestion control, routing, and quality of service provision.