Self-Similar Network Traffic: Implications to Real-Time Applications and QoS

Kihong Park
Department of Computer Sciences
Purdue University
West Lafayette, IN 47907
park@cs.purdue.edu

Abstract

Recent measurements of network traffic [3, 8] have shown that scale-invariant burstiness—self-similarity—is a ubiquitous phenomenon found in both LANs and WANs as well as many types of traffic including compressed video. In [5], a generic high-level (application layer) causal mechanism was advanced for inducing self-similar link traffic when incorporating the effects of the transport layer. The performance impact was investigated in [4] showing the sensitive dependence of the delay-throughput relation on self-similarity.

This paper studies the problem of provisioning quality of service (QoS) to real-time applications in the presence of self-similarity, first, with respect to network design, and second, with respect to congestion control. In the context of network design, two of the principal design variables are bandwidth and buffer capacity. The marginal utility of buffer capacity with respect to QoS provision is low due to the amplified queueing behavior exhibited by self-similar traffic: packet loss rate can only be reduced at a disproportionate increase in queueing delay. Figure 1 (left) shows the delay-packet loss trade-off curve as a function of self-similarity (α is a parameter determining the *degree* of self-similarity, being "highly" self-similar if α is close to 1 and being weakly so if $\alpha \approx 2$). For a fixed packet loss rate, we see a superlinear increase in delay as self-similarity is increased. Bandwidth, on the other hand, exhibits a much more robust performance gain, decreasing both packet loss rate and queueing delay as long as bottleneck links do not themselves form an extensive multi-stage network. Figure 1 (right) shows queueing delay as a function of self-similarity for several values of bottleneck bandwidth. It is interesting to note that under highly self-similar traffic conditions ($\alpha = 1.05$), the marginal benefit of increasing bandwidth is significantly greater vis-a-vis the case when traffic is less self-similar ($\alpha = 1.95$). The requirement on the bottleneck links not forming an extensive internetwork stems from the fact that for a given buffer size, the decrease in queueing delay may be accompanied by an increase in packet loss rate even if bandwidth is uniformly increased across all links.

In the context of congestion control, the fundamental limit to traffic shaping imposed by the stringent QoS requirements of real-time traffic implies that a reservation- or priority-based packet scheduling scheme is needed to guarantee QoS with high probability. Since, other things being equal, this entails a severe reduction in network resource utilization due to the scale-invariant burstiness of self-similar traffic, the central problem reduces to one of utilizing left-over bandwidth effectively by provisioning best-effort service to non-stringent-QoS traffic. A distributed, adaptive scheme based on noncooperative computational markets and its efficacy at finding efficient allocations was shown in [7, 2].

This paper presents an *application-sensitive* congestion control protocol in the context of best-effort service provision that seeks to exploit the structure of long-range dependent traffic. First, for real-time applications engaging in interactive conversations consisting of the exchange of "short" packets, virtually no flow control is exerted beyond guaranteeing reliable transport, if needed, using positive acknowledgements. Such conversations (e.g., generated by Internet game servers and their client applications such as the Internet Go Server), even if numerous, do not significantly contribute to the aggregate traffic being self-similar, and

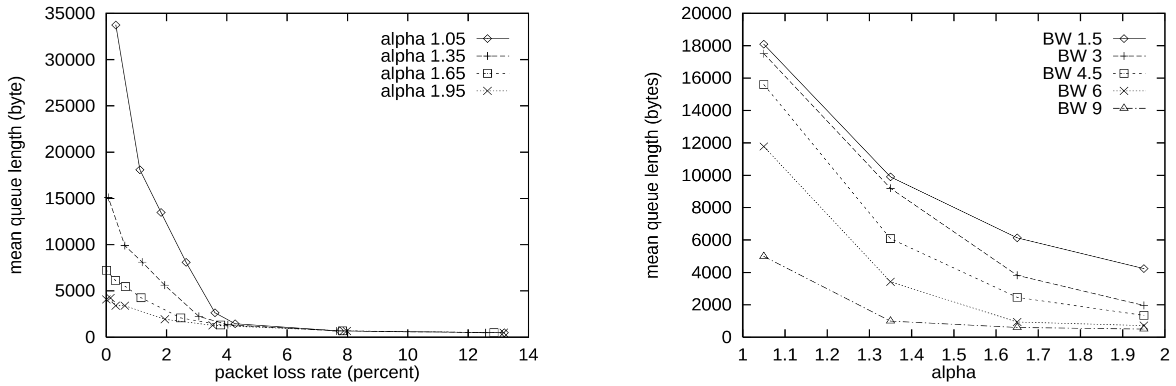

Figure 1: Marginal effect of network resources under self-similar traffic conditions: Buffer capacity on delay-packet loss rate (left); bandwidth on queueing delay (right).

unnecessary overhead is created by flow-controlling them. To achieve dependable QoS, however, forward error correction (FEC) [1] is exercised and duplicate packets are sent out where frequency and inter-packet spacing determine the level of QoS achieved. More generally, error-correcting codes can be employed to reduce data redundancy at the expense of increased processing delay. FEC is applied using a *redundancy schedule* which consists of a probability distribution π over the negative reals, called the *redundancy profile*, and a scaling factor $\kappa \geq 0$, called the *redundancy factor*. π determines how far into the future and with what relative frequency redundant information is transmitted; it is influenced, in part, by real-time constraints such as cut-off deadline τ, a threshold which determines the usefulness of the arriving data. In general (soft real-time constraints), τ itself is a function of time. Figure 2 shows two redundancy profiles π_1, π_2 both with finite support s. The optimal shape of π will be a function of the underlying traffic characteristics, possessing a heavier tail as the self-similarity of background traffic is increased. s can be adjusted such that the difference $\tau - s$ is increased for lengthy end-to-end connections where the shape of π is attenuated due to queueing effects. The redundancy factor κ is used to determine the data overhead associated with FEC; i.e., κ redundant packets are transmitted where the packets are distributed in time according to π. Thus, if $\kappa = 0$, then no duplicate packets are transmitted whatsoever. κ is adjusted as a function of packet loss rate $c = \ell/\kappa$ ($\kappa > 0$, $\ell \leq \kappa$) where ℓ is the number of packets dropped belonging to the same redundancy group, measured and fed back by the receiver:

$$\frac{d\kappa}{dt} = F(c).$$

Here $dF/dc > 0$; several specific schemes are currently being investigated including a control law given by

$$d\kappa/dt = \varepsilon(c - \gamma), \tag{1}$$

where $0 \leq \gamma < 1$ is a reliability threshold and $\varepsilon > 0$ is an adjustment factor. The analysis of this system is related to the stability condition associated with a static outflux-dynamic influx congestion system [6]. For ε sufficiently small (keeping in mind that, in practice, $\kappa \in \mathbb{Z}_+$), the optimal operating of (1) is stable.

For real-time conversations involving video and audio, a single traffic stream can have a non-negligible effect on the characteristics of aggregate traffic and amplifying reliability by a large factor $\kappa > 1$ can significantly impact the packet loss rate c itself. Employing error-correcting codes and block-encoding only decreases the overhead by a constant factor. Thus, for real-time traffic involving "large" transfers where, by definition, the arrival process is externally imposed, the problem reduces to one of dynamically adjusting the redundancy factor κ such that reliability—as captured by c—is minimized. Since

$$c = c(\kappa), \qquad dc/d\kappa > 0,$$

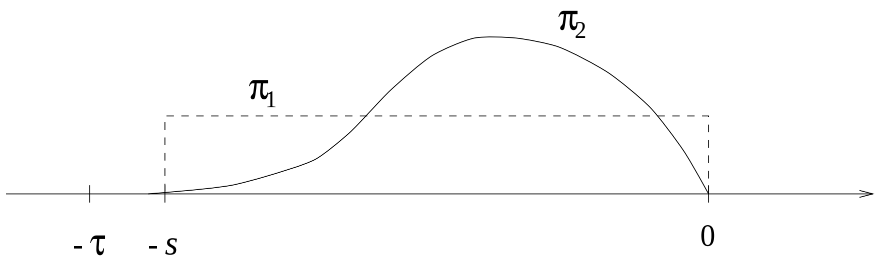

Figure 2: Redundancy schedule with two different redundancy profiles π_1, π_2, and cut-off deadline τ.

the control problem associated with (1) becomes nonlinear and achieving the optimal operating point is nontrivial. In fact, this corresponds to the generic congestion-susceptible system modeled via unimodal load-throughput functions [6], whose optimal operating point is asymptotically unstable. Thus, there are intrinsic difficulties associated with achieving real-time QoS control via FEC. We employ a variant of the Warp Control protocol suite [6] to seek effective, practical solutions. Again, of special interest is the optimal shape of π as a function of self-similarity.

With respect to the large bandwidth-delay product of B-ISDN systems and the penalty incurred by feedback control algorithms, a trade-off relation exists between γ (the target packet loss rate) and the data overhead (and reduced effective utilization) that a small γ in (1) entails. In other words, the smaller γ, the more leeway there is to adjust to sudden increases in resource contention, and employing feedback control remains feasible. These trade-offs, however, are in some sense fundamental and unavoidable when provisioning service to QoS-sensitive real-time traffic.

References

[1] Ernst Biersack. Performance evaluation of forward error correction in ATM networks. In *Proc. ACM SIGCOMM '92*, pages 248–257, 1992.

[2] A. Heddaya and K. Park. Parallel computing on high-speed wide-area networks: a pricing policy for its communication needs. In *Proc. 3rd IEEE Workshop on the Architecture and Implementation of High Performance Communication Subsystems*, pages 188–191, 1995.

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

[4] K. Park, G. Kim, and M. Crovella. On the effect of traffic self-similarity on network performance, 1996. Submitted for publication.

[5] K. Park, G. Kim, and M. Crovella. On the relationship between file sizes, transport protocols, and self-similar network traffic. *To appear in Proc. IEEE International Conference on Network Protocols*, October 1996.

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

[7] Kihong Park. Self-organized multi-class QoS provision for ABR traffic in ATM networks. In *Proc. 15th IEEE International Phoenix Conference on Computers and Communications*, pages 446–453, 1996.

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