Hybrid Technique for Simulating High Bandwidth Delay Computer Networks

Jong-Suk Ahn, Peter B. Danzig, Deborah Estrin, Brenda Timmerman
Computer Science Department
University of Southern California
Los Angeles, CA 90089-0781
danzig@usc.edu

1 Introduction

Researchers evaluate and contrast new network routing, admission control, congestion control and flow control algorithms through simulation. Analytically derived arguments justifiably lack credibility because, in the attempt to model the underlying physical system, the analyst is forced to make compromising approximations. However, unlike analytical techniques like Jackson Queueing Networks, simulations require significant computation and a simulation's state can consume a great deal of memory.

This paper describes a technique that we developed to reduce the memory consumption of communication network simulators. Reduced memory makes simulations of larger and higher bandwidth-delay networks possible, but introduces an adjustable degree of approximation in the simulation. The higher the memory savings, the less accurate the computed measures. We call our technique *Flowsim*. The paper motivates the need to simulate computer networks rather than model them analytically, motivates why a simulator's state can grow quickly, and explains why analytical techniques have failed to model modern communication networks.

2 Representation

Packet network simulators, like all discrete event simulators, are built around an event list. When a node forwards a packet along a link towards a neighboring node, it actually inserts an event into the event list with time set to the sum of the link's propagation delay, the packet's transmission time, and the current time. It includes the transmission time because few packet switches employ cut-through routing and hence cannot switch a packet until its last bit arrives. The event list invokes the destination node when simulated time catches up with the event. The destination node retrieves the pointer to the packet header and delivers, buffers, or forwards the packet.

1993 ACM SIGMETRICS-5/93/CA,USA

Train Descriptor Fields
`int packet_count`
`time lead_time`
`time tail_time`
`time link_idle_time`
`long sequence_number`

Figure 1: Fields of a train descriptor.

Jain noted that subsequent packets on a local area network tend to arrive from the same source and are headed to the same destination [6]. He called such a burst a *packet train*. He said that if the spacing between two packets exceeds some *inter-train gap*, that these packets belong to separate *trains*. In subsequent years, several protocol stack implementations were optimized to exploit this phenomena [2, 5]. Other investigators found that Jain's observation held for wide-area networks, although not as prominently as it does for local area networks [4, 3], and researchers already anticipate the declining occurrence of packet trains, at least as described by Jain [7].

Motivated by packet trains, Flowsim represents closely spaced packets with even fewer fields. Each link in Flowsim places a conversation's packets on a linked list of packet trains called a *flow descriptor*. Each conversation traversing a link has its own flow descriptor. When a node originates or forwards a packet onto a link, the simulator appends it to the corresponding flow descriptor.

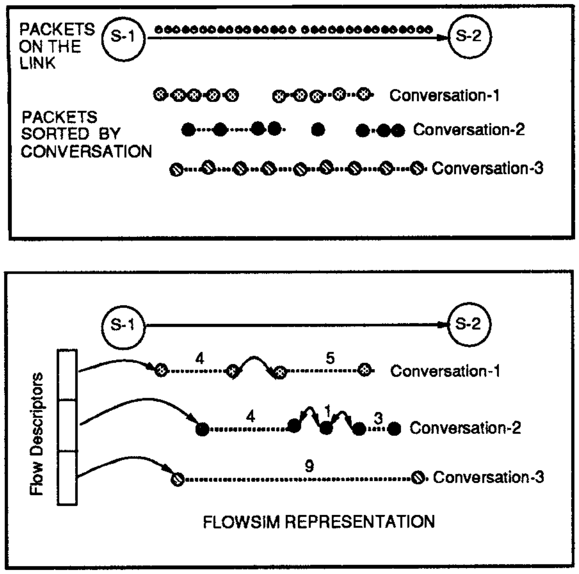

Figure 2: Flowsim's packet representation.

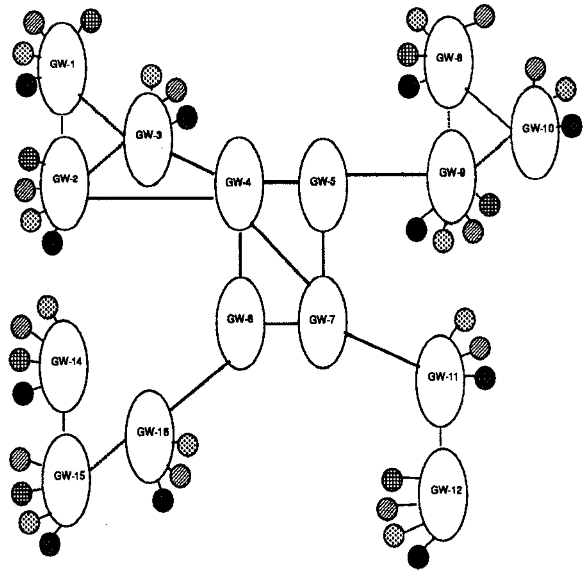

Figure 3: An example topology.

Each packet train is represented by a flow descriptor. A flow descriptor appends a packet to the "closest" packet train if the packet's arrival time at the next switch is within the inter-train gap of the train's youngest packet, `tail_time`. If the flow descriptor appends the packet to the train, it increments the train's `packet_count` and extends its `tail_time` accordingly. If it cannot append the packet to the train, it creates a new train for the packet and chains it appropriately. Other reasons for creating a new train depend on the specific protocols being simulated. For example, if packets carry sequence numbers, then a packet with a non-sequential sequence number starts a new train.

The upper half of Figure 2 illustrates a physical link carrying packets from three conversations. The lower half of the figure illustrates Flowsim's internal representation of the same. Line segments represent packet trains; chained packet trains belong to the same flow descriptor and hence conversation. Each train's `packet_count` is shown, as well as each train's first and last packet.

3 Summary

Flow descriptors and packet trains save memory because they efficiently represent groups of packets. Packet trains save CPU time because we modified the simulator to process entire trains of packets in a single operation. In simulations of the network in Figure 3, Flowsim consumes ten times less CPU and memory than our pure packet simulator. The full paper [1] describing Flowsim is available from anonymous FTP:

jerico.usc.edu:pub/danzig/flowsim.ps.Z.

References

[1] Jong-Suk Ahn, Peter B. Danzig, Deborah Estrin, and Brenda Timmerman. Hybrid technique for simulating high bandwidth dely computer networks. *USC CS Technical Report USC-CS-92-528.*

[2] John B. Carter and Willy Zwaenepoel. Optimistic implementation of bulk data tranfer protocols. *1989 ACM SIGMETRICS Conference*, pages 61–69, May 23-26, 1989.

[3] Deborah Estrin and Danny Mitzel. An assessment of state and lookup overhead in routers. *IEEE Infocom '92*, pages 2332–2342, May 1992.

[4] Steven A. Heimlich. Traffic characterization of the NSFnet national backbone. *Proceedings USENIX*, January 1989.

[5] Van Jacobson. Compressing TCP/IP headers for low-speed serial links. Technical Report RFC1144, LBL, February 1990.

[6] Raj Jain and Shawn A. Routhier. Packet trains–measurement and a new model for computer network traffic. *IEEE JSAC*, pages 986–995, September, 1986.

[7] Paul E. McKenney and Ken F. Dove. Efficient demultiplexing of incoming TCP packets. *ACM SIGCOMM 92 Conference*, pages 269–279, August 1992.