Connections with Multiple Congested Gateways in Packet-Switched Networks
Part 2: Two-way Traffic

Sally Floyd*
Lawrence Berkeley Laboratory
1 Cyclotron Road
Berkeley, CA 94720
floyd@ee.lbl.gov

December 13, 1991

Abstract

In a previous paper [F91] we explored the bias in TCP/IP networks against connections with multiple congested gateways, for networks with one-way traffic. Using simulations and a heuristic analysis, we showed that in a network with the window modification algorithm in 4.3 tahoe BSD TCP and with Random Drop or Drop Tail gateways, a longer connection with multiple congested gateways can receive unacceptably low throughput. However, we showed that in a network with no bias against connections with longer roundtrip times and with no bias against bursty traffic, a connection with multiple congested gateways can receive an acceptable level of throughput.

In this paper we show that the the addition of two-way traffic to our simulations results in compressed ACKs at the gateways, resulting in much more bursty traffic in the network. We use Priority gateways to isolate the effects of compressed ACKs in networks with two-way traffic; these Priority gateways give priority to small packets such as ACK packets, thereby allowing two-way traffic without compressed ACKs. In our simulations with two-way traffic and multiple congested gateways with FIFO service and Drop Tail or Random Drop packet-drop algorithms, the throughput for the connection with multiple congested gateways suffers significantly. This loss of throughput can be reduced either by the use of Priority gateways, which reduces the burstiness of the traffic, or by the use of Random Early Detection (RED) gateways, which are designed to avoid a bias against bursty traffic.

*This work was supported by the Director, Office of Energy Research, Scientific Computing Staff, of the U.S. Department of Energy under Contract No. DE-AC03-76SF00098.

1 Introduction

In this paper we investigate the throughput of connections in TCP/IP networks with two-way traffic and multiple congested gateways.

2 Simulator algorithms

Our simulator is briefly described in [F91], along with the algorithms used in these simulations.

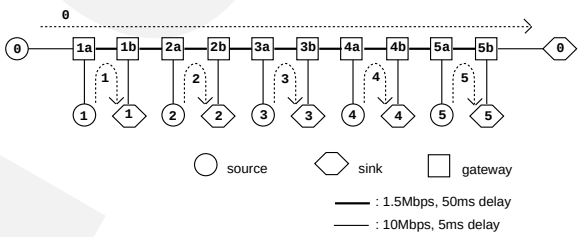

Figure 1: Simulation network with 5 congested gateways.

Figure 1 shows a simulation network for one-way traffic, with 6 connections, 10 gateways, and 5 congested gateways. The congested gateways in Figure 1 are gateways 1a, 2a, 3a, 4a, and 5a. The dotted lines show the connection paths; source i sends to sink i. Each connection has a maximum window just large enough so that, even when that connection is the only active connection, the network still occasionally drops packets.

For one-way traffic we use a family of simulation networks similar to Figure 1, where the number of congested gateways n ranges from 1 to 10. Figure 1 only shows the network for $n = 5$ congested gateways. For

Draft

a simulation network with n congested gateways for $n \geq 1$ there are $n + 1$ connections and $2n$ gateways. Connection 0 passes through multiple congested gateways, and connections 1 through n each pass through one congested gateway. Connection 0 is roughly $2n - 1$ times longer than the other n connections. In these simulations all connection paths have the same maximum bandwidth.

Definitions: Reduce-to-One, Increase-by-One. Some of the simulations in the paper use the Reduce-to-One window decrease algorithm and the Increase-by-One window increase algorithm from 4.3 tahoe BSD TCP [J88]. Our simulator does not use the 4.3-tahoe TCP code directly but we believe it is functionally identical. Briefly, there are two phases to the window-adjustment algorithm. In the slow-start phase the window is doubled each roundtrip time until it reaches a certain threshold. Reaching the threshold causes a transition to the congestion-avoidance phase where the window is increased by roughly one packet each roundtrip time. In this paper we call the increment algorithm used in the congestion-avoidance phase the Increase-by-One algorithm. Packet loss (a dropped packet) is treated as a "congestion experienced" signal. The source uses timeouts or "fast retransmit" to discover the loss (if four ACK packets acknowledging the same data packet are received, the source decides a packet has been dropped) and reacts by setting the transition threshold to half the current window, decreasing the window to one packet and entering the slow-start phase. In this paper we call this the Reduce-to-One algorithm. □

In order to achieve the highest throughput for longer connections, some of these simulations use the Fast Recovery algorithm in 4.3 reno BSD TCP, along with Selective Acknowledgements (or SACKs). In this paper the Fast Recovery algorithm implemented in 4.3 reno BSD TCP is called the Reduce-by-Half algorithm.

Definitions: Reduce-by-Half window decreases and Selective Acknowledgements. With the Reduce-by-Half window decrease algorithm, when a packet loss is detected by the "fast retransmission" algorithm the connection reduces its window by half. For the purposes of this paper, the important feature of the Reduce-by-Half window decrease algorithm is that with the "fast retransmission" algorithm, the source retransmits a packet and reduces the window by half, rather than reducing its window to one packet. For the simulations in this paper, we use Selective Acknowledgement sinks [JB88] with the Reduce-by-Half algorithm. With Selective Acknowledgement sinks, each ACK acknowledges not only the last sequential packet received for that connection, but also acknowledges all other (non-sequential) packets received. □

Some of the simulations in this section use the TCP Increase-by-One window-increase algorithm for the congestion-avoidance phase of the window-increase algorithm. As shown in [FJ91a], this algorithm has a bias against connections with longer roundtrip times. In order to eliminate this bias, some of our simulations use the Constant-Rate algorithm instead in the congestion-avoidance phase. We are *not* proposing the Constant-Rate algorithm for current networks; we simply are using the Constant-Rate algorithm to explore throughput in networks with a window-increase algorithm with no bias in favor of shorter-roundtrip-time connections.

Definitions: Constant-Rate window increases. In the Constant-Rate window-increase algorithm, in the congestion-avoidance phase each connection increases its window by roughly $a * r^2$ packets each roundtrip time, for some fixed constant a, and for r the calculated average roundtrip time. Using this algorithm, each connection increases its window by a pkts/sec in each second. For the simulations in this paper, we use $a = 4$. □

In this paper we examine networks with Drop Tail, Random Drop, and RED gateways. As shown in [FJ91a], simulations and measurement studies with Drop Tail gateways are vulnerable to traffic phase effects; small changes in network parameters can result in large changes in the performance of the network. In order to avoid these phase effects in networks with Drop Tail gateways, in this paper the simulator adds a small random component to the roundtrip time for each packet in simulations using Drop Tail gateways. (This is discussed in [FJ91a].) Normally, our simulator charges zero seconds for the time required to process packets at the nodes. In this paper each source node adds a random time uniformly distributed in [0, b], for $b \approx 5.3$ ms. the bottleneck service time of the network, to the time required by the source node to process each ACK packet in the simulations with Drop Tail gateways. This is not intended to model any assumptions about realistic network behavior, but to eliminate problems with traffic phase effects. With this added random component the simulations with Drop Tail gateways give similar results to the simulations with Random Drop gateways.

To avoid the bias against bursty traffic common to Random Drop and to Drop Tail gateways, we also examine performance in simulations with Random Early Detection (RED) gateways, a modified version of Random Drop gateways that detect incipient congestion. RED gateways maintain an upper bound on the average queue size at the gateway.

Definitions: RED gateways. With our implementation of RED gateways [FJ91c], the gateway computes the average size for each queue using an exponential weighted moving average. When the average queue size exceeds a certain threshold, indicating incipient

Draft

congestion, the gateway randomly chooses a packet to drop and increases the threshold. As more packets arrive at the gateway, the threshold slowly decreases to its previous value. The gateway chooses a packet to drop by choosing a random number n in the interval 1 to $range$, where $range$ is a variable parameter of the gateway. The gateway drops the nth packet to arrive at the gateway. RED gateways are described in more detail in a paper currently in progress [FJ91c]. □

One advantage of RED gateways is that, unlike Drop Tail and Random Drop gateways, RED gateways do not have a bias against bursty traffic. The bias of Drop Tail and of Random Drop gateways against bursty traffic and the correction of this bias in RED gateways are described in [FJ91a]. With Drop Tail or Random Drop gateways, the more bursty the traffic, the more likely it is that the queue will overflow and the Drop Tail or Random Drop gateway will drop a packet. This is because a burst of packets results in a temporary increase in the queue size at the gateway. With RED gateways the detection of congestion depends on the average queue size, not on the maximum queue size. Thus with RED gateways bursty traffic is less likely to result in the detection of congestion. With a RED gateway even when bursty traffic results in the detection of congestion at the gateway, the mechanism for dropping packets ensures that the bursty connection does not have a disproportionate probability of having a packet dropped. □

For the RED simulations in this paper the maximum queue size is 60 packets, and the RED gateways drop packets when the average queue size is between 10 and 20 packets. (The range from 10 to 20 packets for the average queue size for RED gateways is somewhat arbitrary; the optimum average queue size is still a question for further research.) For the two-way traffic simulations with Random Drop and Drop Tail gateways, the maximum queue size is 20 packets (I have to rerun these simulations for a maximum queue of 60 packets). The maximum windows for connections 1 to n are set sufficiently large to force occasional packet drops even in the absence of traffic from connection 0.

3 Simulations with two-way traffic

The simulations in Section 2 were of a network with one-way traffic. In this section, we explore simulations with two-way traffic. As discussed in [WRM91] and [ZC91], two-way traffic introduces the added complication of ACK-compression, caused by small ACK packets being queued at a congested gateway with no interleaving larger data packets. This ACK-compression increases the burstiness of the traffic, causing substantial problems for networks with Random Drop (or Drop Tail) gateways. We show that with Random Drop gateways, the addition of two-way traffic causes a substantial loss in throughput for the longer connection. With RED gateways, with their increased ability to accomodate bursty traffic, the addition of two-way traffic causes only a small change in the behavior of the network. We use gateways with Priority service, which give priority to small packets, to compare the throughput in networks with two-way traffic with and without compressed ACKs.

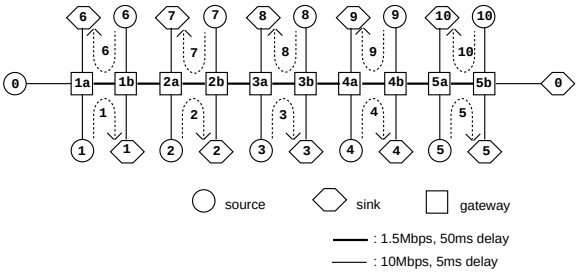

Figure 2: Simulation network with two-way traffic.

Figure 2 shows the simulation network with two-way traffic and with n congested forward gateways, for $n = 5$. Connection 0 goes from source 0 to sink 0. Gateways $1a$ to $5a$ are congested in connection 0's forward direction, and gateways $1b$ to $5b$ are congested in the reverse direction. With this two-way traffic, connection 0 suffers from significant ACK-compression. All of the connections in this network can experience compressed ACKs, but because connection 0 passes through multiple congested gateways, the problem of compressed ACKs is more severe for connection 0. All of the simulations in this section use Constant-Rate window increases and Reduce-by-Half decreases.

Draft

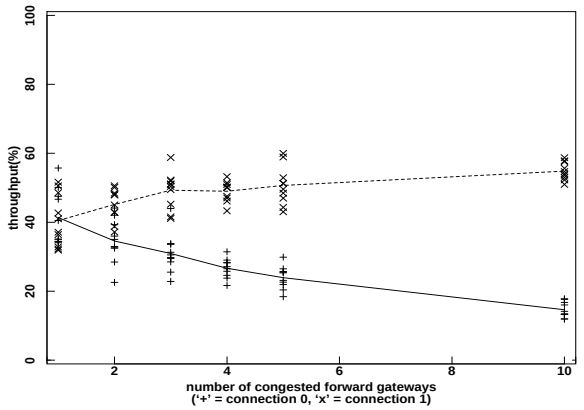

Figure 3: RED gateways, two-way traffic, FIFO service.

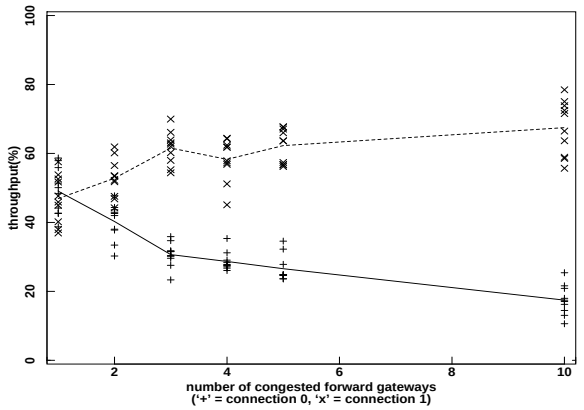

Figure 4: RED gateways, two-way traffic, Priority service.

Simulations with two-way traffic and RED gateways, with Constant-Rate increases and Reduce-by-Half decreases.

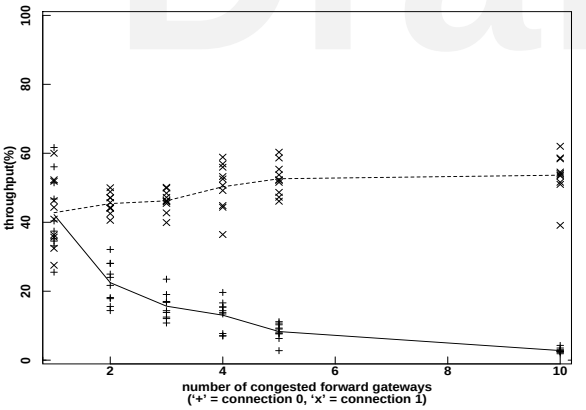
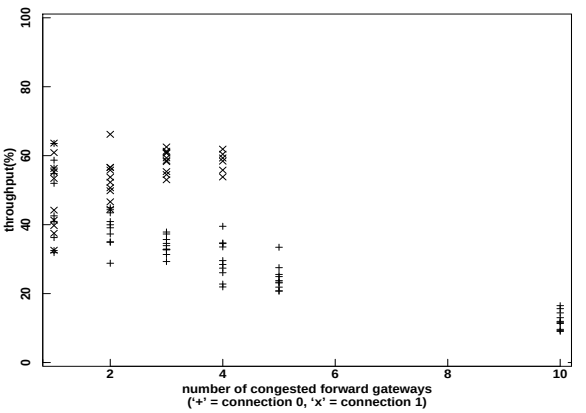
Figure 5: Random-Drop gateways, two-way traffic, FIFO service.