1988

Estimating Lower Bounds on Buffer Sizes for a Packet Switch

Douglas E. Comer
Purdue University, comer@cs.purdue.edu

Rajendra Yavatkar

Report Number:
88-778

ESTIMATING LOWER BOUNDS ON BUFFER SIZES FOR A PACKET SWITCH

Douglas Comer
Rajendra Yavatkar

CSD-TR-778
June 1988
(Revised December 1988)

ESTIMATING LOWER BOUNDS ON BUFFER SIZES FOR A PACKET SWITCH

Douglas Comer
Rajendra Yavatkar

CSD-TR-778
June 6, 1988

Abstract

In a wide area network that uses store–and–forward technology, a packet switch buffers incoming data while it is processing a packet. When a packet switch observes that the rate of incoming packets exceeds (or is likely to exceed) its processing capability, it sends a *source quench* message to the source(s) of traffic to reduce the flow of incoming traffic. During the time, a quench message propagates back to the source, the packet switch should have enough buffering capacity to buffer the additional traffic. A good packet switch design requires sufficient buffering capacity per input link to avoid loss of packets due to overrunning. In this paper, we discuss a method for estimating lower bound on buffer size for a given link speed. We also make certain observations about the processing requirements to allow scalability to accommodate very high speed links (up to 100 Mbps).

1 Introduction and Motivation

A packet switched long haul network typically consists of a set of packet switches interconnected by point–to–point links. Such a network uses store–and forward technique for transporting packets across the network. A packet traveling from its source to destination gets forwarded from a switch to another until it reaches its destination. An intermediate packet switch accepts a packet on an incoming link, processes it to determine its destination, and then forwards it onto an appropriate outgoing link. If packets arrive when it is processing a packet, the switch enqueues them for processing. The queue is normally a first–in–first–out (FIFO) buffer with a finite number of buffers. If the packets arrive too fast, the queue becomes full and the packet switch drops additional packets that arrive. Thus, the buffer space provides a speed match between host and the network when a sudden burst of traffic appears over an incoming link.

When a packet switch detects that the rate of incoming traffic exceeds (or approaches the rate at which it can process the packets), it sends a *source quench* message to the source of the traffic as a hint to reduce the packet rate to match its processing capability. The source quench message propagates from one switch to another until it reaches the source which then responds to that message appropriately. During the time the quench message travels back to the source and the source reacts to it, the packets continue to pour in. To avoid overrunning a packet switch, it is important that the FIFO buffer size is sufficient to allow buffering data during the interval between generation of a quench message and its reception at the source of the traffic (we refer to this interval as *Quench Latency*). Moreover, more than one source may be sending traffic through a switch and, therefore, quench messages must reach all the sources that cause congestion.

As part of the MultiSwitch project [CSY88], we are designing a multiprocessor–based packet switch for use in a wide area network with very high–speed links (speeds varying from T1–speed to 100 Mbps). Our goal is to design a packet switch that can be scaled to accommodate links up to 100 Mbps. Our design will consist of one or more high–speed FIFO buffers per incoming link to buffer the incoming traffic. As part of our design, we want to estimate the lower bound on FIFO buffer size given the speed of a link. The lower bound on the buffer size is needed to determine the minimum amount of buffering required to accept incoming packets during the quench latency.

Next section describes the problem in more detail along with our assumptions. We will then discuss the method of estimating the buffer size and the estimates made. The last section contains the observations made regarding the impact of processing speed on the buffer sizes.

2 Assumptions

As stated earlier, we want to estimate the buffer size per incoming link for a packet switch that will be used in a network of store–and–forward switches interconnected by point–to–point links in arbitrary configuration. In such a network, the *diameter* of the network is the maximum number of hops required to travel from any source to any destination. For worst case analysis, we will consider two packet switches A and B that are at the two ends of the diameter of the network as shown in Figure 1.

We will assume that the switch A sends traffic through B and that traffic is the source of congestion at B. When switch B detects that rate of incoming traffic exceeds its capacity, it will generate a *source quench* message destined for

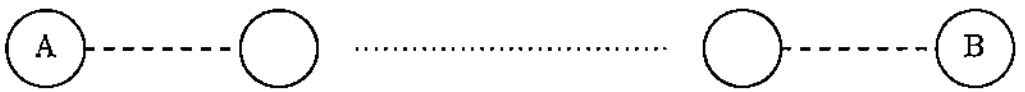

Figure 1: Two Packet Switches A and B a diameter apart

A. Given a link speed, our goal is to estimate the minimum amount of buffer space needed at B to buffer the data that can pour in during the quench latency.

To simplify our calculations, we make the following assumptions:

- We assume the diameter of the network to be 16.
- We assume that the links connecting switches use synchronous hardware. As a consequence, the number of bytes transmitted over a link per second is equal to its speed (in bits per second or *bps*) divided by 8.
- Because the quench messages are important to the correct functioning of the network, we assume that they are accorded highest priority. This assumption is consistent with the link level protocol used in the MultiSwitch Project [Yav88]. As a result, a packet switch will not enqueue quench messages, but, instead, will process (or forward) them immediately. Thus, as a quench message travels from one switch to another, the amount of time it spends at a packet switch is equal to the time required to switch a quench packet from an incoming link to an outgoing link.
- The control information in a quench message is typically restricted to a few bytes and we assume that the size of a quench packet will be 50 bytes.
- Given the various kinds of processing architectures available, the amount of time needed to switch a packet depends on the kind of processor–memory–network interface used. To simplify our calculations, we will assume that

it takes 1 millisecond to switch a quench packet from an incoming link to an outgoing link. This time period includes the time spent in receiving and transmitting the packet to and from the network interface. This is a reasonable estimate (in fact, an upper limit) given the current state–of–the–art of the processor–network interface[1]. It is also based on a transputer board we are using in the prototype.

3 Estimates

Let T be the total amount of time needed for a quench message to reach A from B, T_x be the total amount of time spent in transmitting a quench packet over the links, and let T_s be the total amount of time spent in switching a quench packet from one switch to another until it reaches A. Now, referring to Figure 1:

$$T = T_x + T_s$$

where

$$T_x = (15 * 8 * 50)/speed\_in\_bps * 1000$$

milliseconds (ms) because there are 15 intermediate links, and

$$T_s = 16 * time\_to\_switch\_per\_packet\_switch = 16$$

milliseconds. T_s here also includes the time spent in switches A and B.

Figure 2 plots the estimates of T for various link speeds. Figure 3 lists the lower bounds on buffer sizes for some common link speeds.

[1]To be able to receive traffic over a 100Mbps line, our network interface will have to process a 50–byte packet in 10 microseconds

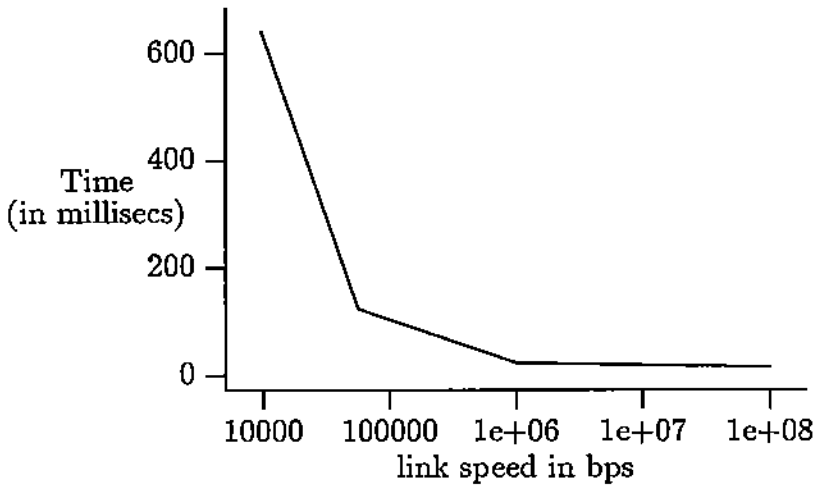

Figure 2: Estimate of amount of time to reach A from B

Link Speed in bps	Tx msecs	Ts msecs	Buffer Size in bytes
9600	625.0	16	769.2
56K	107.14	16	862
1M	6.0	16	2750
DS3 (45M)	0.133	16	90.75K
100M	0.06	16	200K

Figure 3: Lower bounds on buffer sizes for different link speeds

4 Observations

Figure 3 shows that at low speeds, the time to transmit packets over the links dominates the quench latency period. But, at higher speeds, the switching time is the major contribution towards the quench latency. Thus, a switching time of 1 msec requires a large amount of buffer space per link at 100 Mbps. In [Nag85], Nagle points out the disadvantages of having a large amount of buffer space at a packet switch. Because the datagram packets have a finite lifetime based on their *time–to–live* field, increased queuing delays due to larger buffer space cause more packets to be discarded. Also, increased round trip times interact unfavorably with higher level transport protocols such as TCP resulting in lower throughput.

Decreasing the switching time by using faster processors and better architectures will be useful to some extent. With the current state–of–the–art in hardware technology, we can build a processor–network interface that will need 200 microseconds for switching a packet from an incoming link to an outgoing link. Using a Motorola 68020 processor with 16 Mhz clock rate, a high–speed serial port, and high–speed data transfers between serial–access buffers and RAM, it will take 50 microseconds each to receive and transmit a packet to the network. If we assume that 150 instructions will be executed in processing a packet (that is, deciding the outgoing link to take), the total amount of switching time will be 200 microseconds.

At such a switching time, a DS3-speed (45 Mbps) link will need about 19K bytes of buffer space whereas a 100 Mbps link requires about 40K bytes. In order to reduce the switching times further, we need innovative host–to–network interfaces like the one described in [KC88]. Assigning highest priority to quench messages is just the first step in reducing switching time. Using *source routing*

for a quench message and processing it in the hardware without transferring such a packet to the host memory are other optimizations for achieving further reductions.

In conclusion, our discussion shows that achieving lowest possible switching times for quench messages is extremely important in the design of a packet switch so that reasonable amount of buffer space can be used to buffer incoming data at very high speeds.

References

[CSY88] D. Comer, J. Steele, and R. Yavatkar. *An Overview of MultiSwitch Project.* Technical Report in preparation, Computer Science Department, Purdue University, May 1988.

[KC88] H. Kanakia and D.R. Cheriton. The VMP Network Adaptor Board (NAB): High–Performance Network Communication for Multiprocessors. In *SIGCOMM '88 Symposium*, ACM, August 1988. To appear.

[Nag85] J. Nagle. On Packet Switches With Infinite Storage. ARPANET Working Group Requests For Comments, December 1985. RFC 970.

[Yav88] R. Yavatkar. *An Architecture for a High–Speed Packet Switched Network.* Technical Report, Dept. of Computer Science, Purdue University, West Lafayette, IN 47907, May 1988. Proposal for Ph.D. Dissertation.