Chapter 1

Introduction

In a packet-switched network, packets are buffered when they cannot be processed or transmitted at the rate they arrive. There are three main reasons that a router, with generic switching architecture as shown in Figure 1.1, needs buffers: to store packets at times of congestion, to store packets when there is internal contention, and for pipelining and synchronization purposes.

Congestion occurs when packets destined for a switch output arrive faster than the speed of the outgoing line. For example, packets might arrive continuously at two different inputs, all destined to the same output. If a switch output is constantly overloaded, its buffer will eventually overflow, no matter how large it is; it simply cannot transmit the packets as fast as they arrive. Short-term congestion is common, due to the statistical arrival time of packets. Long-term congestion is usually controlled by an external mechanism, such as the end-to-end congestion avoidance mechanisms of TCP, the XON/XOFF mechanisms of Ethernet, or by the end-host application.

Deciding how big to make the congestion buffers depends on the congestion control mechanism; if it responds quickly to reduce congestion, then the buffers can be small; otherwise, they have to be large.

Even when the external links are not congested, most packet switches can experience internal contention because of imperfections in their data paths and arbitration mechanisms. The amount of contention, and therefore the number of buffers

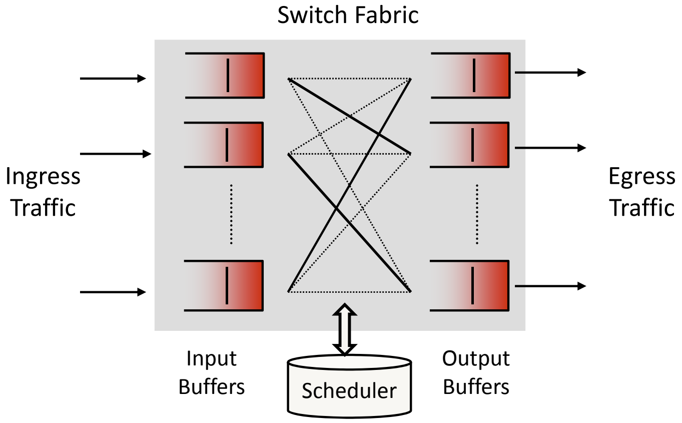

Figure 1.1: Input and output buffers in a CIOQ router. Input buffers store packets when there is internal contention. Output buffers store packets when output links are congested.

needed, is, in part, determined by the switch architecture. For example, output-queued switches have no internal contention and need no contention buffers. At the other extreme, input-queued switches can have lots of internal contention. For 100% throughput, these switches need large internal buffers (theoretically, of infinite depth) to hold packets during times of contention. Some architectures can precisely emulate output queueing [23, 21] through careful arbitration and a combination of input and output queues (CIOQ). These switches still need contention queues (at their inputs) to hold packets while the arbitration algorithm decides when to deliver each to its output queue. Most switches today use CIOQ, or multiple stages of CIOQ.

Packet switches also have staging buffers for pipelining and synchronization. Most designs have hundreds of pipeline stages, each with a small fixed-delay buffer to hold a fixed amount of data. Most designs also have multiple clock-domains, with packets crossing several domains between input and output; each transition requires a small fixed-size FIFO.

In this work, we will not be considering staging buffers; these buffers are of fixed size and delay determined by the router's internal design, not by the network.

1.1 Router buffer size

Network operators and router manufacturers commonly follow a rule-of-thumb to determine the required buffer size in routers. To achieve 100% utilization, the rule dictates that the buffer size must be greater than or equal to $RTT \times C$, also known as the *delay-bandwidth product*. Here, RTT is the average round-trip time of flows passing through the router, and C is the output link's bandwidth. This rule, as will be explained in Chapter 2, is based on the congestion control mechanism of TCP and the way transmission rate is cut off in response to packet drops in the network. The suggested buffer size is devised to ensure that buffers can stay in continual transmission, even when the sender's transmission rate is reduced. In high-speed backbone networks, this requirement could translate into the buffering of millions of packets in routers' linecards. For example, with an average two-way delay of 100ms, a 10Gb/s link requires 1Gb buffers to follow the rule-of-thumb. The buffer size has to grow linearly as the link speed increases.

Why does the buffer size matter? There are two main disadvantages in using million-packet buffers. First, large buffers can degrade network performance by adding extra delay to the travel time of packets. Second, larger buffers imply more architectural complexity, cost, power consumption and board space in routers' linecards. These issues are discussed in Sections 1.2 and 1.3.

The problem of finding the right buffer size in routers has recently been the subject of much discussion, which will be reviewed in Section 1.4. There is general agreement that while the delay-bandwidth-product rule is valid in specific cases (e.g., when one or a few long-lived TCP flows share a bottleneck link), it cannot be applied to determine the buffer size in all Internet routers.

In this dissertation, we consider routers in the backbone of the Internet. Backbone links typically carry tens of thousands of flows. Their traffic is multiplexed and aggregated from several access networks with different bandwidths that are typically much smaller than the core bandwidth. We discuss the conditions under which routers in backbone networks perform well with very small buffers. We will show that if the core traffic comes from slower access networks (which is the case in a typical network, as the traces collected from backbone links show), then buffering only a few tens of packets can result in high throughput.

1.2 Buffer size and network performance

In a packet-switched network, the end-to-end delay consists of three components: propagation delay, transmission delay, and queueing delay.

While propagation delay and transmission delay are independent of the buffer size, queueing delay varies widely depending on the number of packets in the buffers along the path. Large buffers can potentially result in large delay and delay variations (jitter), and negatively impact the users' perceived performance. Some examples of these problems include:

- Over-buffering increases the end-to-end delay in the presence of congestion. This is especially the case with TCP, as a single TCP flow in the absence of other constraints will completely fill the buffer of a bottleneck link, no matter how large the buffer is. In this case, large buffers cannot satisfy the low-latency requirements of real time applications like video games.

  Consider a 10Gb/s link shared by flows with 100ms average round-trip time. A buffer of size $RTT \times C = 1$Gb, if full, adds 100ms delay to the travel time of packets going through this link, making it twice as large. In online gaming, a latency difference of 50ms can be decisive. This means that a congested router with buffers of size $RTT \times C$ will be unusable for these applications, even though the loss rate of the router is very small because of the huge buffers used.

- Unlike in open-loop systems, larger buffers do not necessarily result in larger throughput (or equivalently smaller flow completion time) under the closed-loop rate control mechanism of TCP. In an open-loop system, the transmission rate is independent of the buffer size, hence the throughput is only a function of the buffer's drop rate. Under the closed-loop mechanism of TCP, the average throughput over a round-trip time RTT, is W/RTT. Both RTT and W vary as the buffer size changes. Larger buffer size means larger RTT and at the same time smaller drop rate, or equivalently larger window size. Whether we gain or lose throughput by increasing the buffer size depends on how RTT and W change versus the buffer size.

- Large delay and delay variations can negatively affect the feedback loop behavior. It has been shown that large delay makes TCP's congestion control algorithm unstable, and creates large oscillations in the window size and in the traffic rate [32, 41]. This in turn results in throughput loss in the system.

1.3 Buffer size and router design

Buffers in backbone routers are built from commercial memory devices such as dynamic RAM (DRAM) or static RAM (SRAM). SRAMs offer lower (faster) access time, but lower capacity than DRAMs.

With 100Gb/s linecards under development, it has become extremely challenging to design large and fast buffers for routers. The typical buffer size requirement, based on the delay-bandwidth product rule, is 250ms, which is equivalent to 25Gb at 100Gb/s speed. To handle minimum length (40B) packets, a 100Gb/s linecard's memory needs to be fast enough to support one read/write every 1.6ns.

The largest currently available commodity SRAM is approximately 72Mb and has an access time of 2ns [1]. The largest available commodity DRAM today has a capacity of 1Gb and an access time of 50ns [2].

To buffer 25Gb of data, a linecard would need about 350 SRAMs, making the board too large, expensive, and hot. If instead DRAMs are used, about 25 memory chips would be needed to meet the buffer size requirement. But the random access time of a DRAM chip could not satisfy the access time requirement of 1.6ns.

In practice, router line cards use multiple DRAM chips in parallel to obtain the aggregate memory bandwidth they need. This requires using a very wide DRAM bus with a large number of fast data pins. Such wide buses consume large amounts of board space, and the fast data pins consume too much power.

The problem of designing fast memories for routers only becomes more difficult as the line rate increases (usually at the rate of Moore's Law). As the line rate increases, the time it takes for packets to arrive decreases linearly. But the access time of commercial DRAMs decreases by only 1.1 times every 18 months [1][44].

However, memory dimension cannot become very small, since breaking memory into more and more banks results in an unacceptable overhead per memory bank.

There could be significant advantages in using smaller buffers. With buffers a few hundred times smaller, the memory could be placed directly on the chip that processes the packets (a network processor or an ASIC). In this case, very wide and fast access to a single memory would be possible, but the memory size would be limited by the chip size. The largest on-chip SRAM memories available today can buffer about 64-80Mb in a single chip [1]. If memories of this size are acceptable, then a single-chip packet processor would need no external memories.

If very small buffers could be made to work, it might even be possible to use integrated optical buffers in routers. Optical routers, if built, would provide almost unlimited capacity and very low power consumption.

The following section explains more about technological constraints and advances in building optical memories.

1.3.1 Optical buffers

Over the years, there has been much debate about whether it is possible (or sensible) to build all-optical datapaths for routers.

On the one hand, optics promises much higher capacities and potentially lower power consumption. Optical Packet switching decouples power and footprint from bit-rate by eliminating the optoelectronic interfaces. The results are higher capacity and reduced power consumption, and hence increased port density. Over time, this could lead to more compact, high-capacity routers.

[1] The access time of a DRAM is determined by the physical dimensions of the memory array, which do not change much from generation to generation. Recently, fast DRAMs such as Reduced-Latency DRAM (RLDRAM) have been developed for networking and caching applications. The shortest access time provided by RLDRAM today is 2.5ns at 288Mb density [3]. This architectures reduces the physical dimension of each array by splitting the memory into several banks.

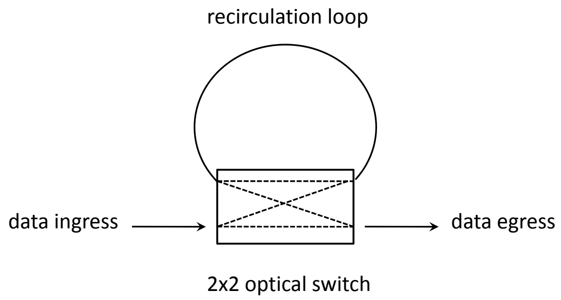

Figure 1.2: Schematic of a feed-back buffer. A 2 × 2 switch is combined with a waveguide loop to provide variable delay for an optical signal.

On the other hand, most router functions are still beyond optical processing, including header parsing, address lookup, contention resolution and arbitration, and large optical buffers. Optical packet switching technology is limited in part by the functionality of photonics and the maturity of photonic integration. Current photonic integration technology lags behind the equivalent electronic technology by several years [12].

To ease the task of building optical routers, alternative architectural approaches have been proposed. For example, label swapping simplifies header processing and address lookup [13, 16, 49], and some implementations transmit headers slower than the data so they can be processed electronically [35, 36]. Valiant load-balancing (VLB) has been proposed to avoid packet-by-packet switching at routers, which eliminates the need for arbitration [30].

Building random access optical buffers that can handle variable length packets is one of the greatest challenges in realizing optical routers. Storage of optical data is accomplished by delaying the optical signal either by increasing the length of the signal's path or by decreasing the speed of the light. In both cases, the delay must be dynamically controlled to offer variable storage times, i.e., to have a choice in when to read the data from the buffer. Delay paths provide variable storage time by traversing a variable number of short delay lines—either several concatenated delays (feed forward configuration) or by looping repeatedly through one delay (feedback

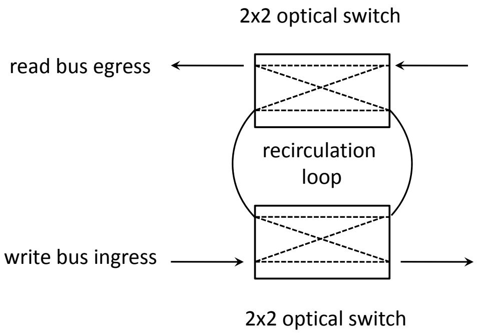

Figure 1.3: Physical implementation of speedup and simultaneous read/write.

configuration). Buffers that store optical data through slowing the speed of light do so by controlling resonances either in the material itself or in the physical structure of the waveguide.

Among the various optical buffering technologies, feedback buffers are beneficial for their low component count and small footprint [14]. The base memory element shown in Figure 1.2 can be built using two photonic chips and cascaded to form a practical optical buffer for many packets. The element is flexible in that it may be used as a recirculating feedback buffer or concatenated to form a feed forward buffer for arbitrary packet lengths. Feedback loops can store packets for a number of recirculations, whereas feed forward configurations require N loops to store a packet for N packet durations. In a feedback buffer, the length of the delay line determines the resolution of possible delays. These buffer elements can also enable easy implementation of simultaneous read/write as well as speedup. The design extension to enable a speedup of 2 and simultaneous read/write is shown in Figure 1.3.

Integrated feedback buffers as developed by Burmeister et al. [15] and Chi et al. [20] show promise of offering a practical solution to optical buffering. These recirculating buffers meet all the necessary requirements for buffering packets at high link bandwidth by providing low optical loss and fast switching time. The integrated optical buffer described in [15] achieves 64ns of packet storage, or 5 circulations, with 98% packet recovery at 40Gb/s link bandwidth.

1.4 Related work

Even though research on router buffer sizing has been done since early 1990s, this problem has only recently attracted wide interest, especially following the work of Appenzeller et al. in 2003 [9]. Since then, there has been much discussion and research on buffer sizing; it has been studied in different scenarios, and under different conditions. Some studies have concluded that the rule-of-thumb excessively overestimates the required size, while others have argued that even more buffering is required under certain conditions. Below is a brief summary of the related work.

Villamizar and Song [48] showed that a router's buffer size must be equal to the capacity of the router's network interface multiplied by the round-trip time of a typical flow that passes through the router. This result was based on experimental measurements of up to eight long-lived TCP flows on a 40Mb/s link.

Appenzeller et al. [9] suggest that the required buffer size can be scaled down by a factor of $\sqrt{N}$, where N is the number of long-lived TCP flows sharing the bottleneck link. These authors show that the buffer size can be reduced to $2T \times C/\sqrt{N}$ without compromising the throughput. For example, with 10,000 flows on the link, the required buffer size is reduced by two orders of magnitude. This follows from the observation that the buffer size is, in part, determined by the saw-tooth window size process of TCP flows. The bigger the saw-tooth, the larger must the buffers be to achieve full utilization. As the number of flows increases, the aggregate window size process (the sum of all the congestion window size processes for each flow) becomes smoother, following the Central Limit Theorem. This result relies on three assumptions: (1) flows are sufficiently independent of each other to be desynchronized (2) the buffer size is dominated by long-lived flows, and (3) there are no other significant, un-modeled reasons for buffering more packets.

In [25], Enachescu et al. show that the buffer size can be further reduced to as small as $O(\log W)$, at the expense of losing only a small fraction of the throughput $(10 - 15\%)$. The suggested buffer size is about $20 - 50$ packets, if the traffic is paced, either by implementing paced-TCP [7] at the source or by running the bottleneck link much faster than the access links. We will examine these assumptions more closely in Chapter 2. Similar results are shown independently by Raina and Wischik [41], who study the stability of closed-loop congestion control mechanisms under different buffer sizes. Using control theory and simulations, the authors show that a system is stable with tiny buffers.

Dhamdhere et al. study a particular network example in [24], and argue that when packet drop rate is considered, much larger buffers are needed, perhaps larger than the buffers in place today. In their work, they study a situation in which a large number of flows share a heavily congested low capacity bottleneck link towards the edge of the network, and show that one might get substantial packet drop rate even if buffers are set based on the rule-of-thumb. In [39], Prasad et al. argue that the output/input capacity ratio at a network link largely determines the required buffer size. If that ratio is larger than one, the loss rate drops exponentially with the buffer size and the optimal buffer size is close to zero. Otherwise, the loss rate follows a power-law reduction with the buffer size and significant buffering is needed.

1.5 Organization of thesis

The rest of this dissertation is organized as follows. Chapter 2 gives an overview of buffer sizing rules and analyzes the utilization of CIOQ routers with tiny buffers at input and output ports. Chapter 3 presents simulation results on the impact of using very small buffers in routers, and studies the effect of various traffic and network conditions on the required buffer size. Chapter 4 describes two sets of buffer sizing experiments, one run in a testbed and another in a real network. Chapter 5 considers a network with multiple routers and explains how traffic can be made buffer-friendly and smooth across the network. Chapter 6 discusses some issues in building optical buffers. Chapter 7 is the conclusion.