CHAPTER 10

Queueing in Packet Switches

Chapter 9 introduces the performance and architectural issues in the design of packet switches. We are now familiar with the concept of a cell switch and the placement of the queues relative to the switch fabric. Specifically, we have discussed the input-queueing (IQ), output-queueing (OQ), and combined input-output-queueing (CIOQ) options for packet queue placement in cell switches. This chapter first analyses the switching capacity and delay performance of IQ, OQ and CIOQ switches assuming that the queues are all FIFO. We then consider alternatives to FIFO scheduling and discuss the emulation of an OQ switch by an IQ switch and an appropriate switch scheduler.

Much of this chapter assumes cell switches. Recall that a cell switching fabric operates in a time-slotted manner, where the slot duration is equal to the cell transmission time. In our analysis, we assume that cell arrivals occur at the beginning of a slot and that cell departures are completed at the end of the slot. Thus, an arrival in a slot will be available for transmission in the same slot. Also, in this chapter, we consider only nonblocking switches and do not worry about how such a switch may be constructed.

10.1 FIFO Queueing at Output and Input

Let us first recollect what we discuss in Chapters 1 and 9: In a slot, if k active inputs have cells for the same output—say, output port j—and if there are to be no queue at the inputs and no cells are to be dropped at the inputs, the switch should be capable of switching all the k cells to output j. Of the k cells that reach the output, only one may be transmitted on the output link, and the other $k - 1$ must be put in a queue at the output port. It could be that at the output there are packets that are waiting from the previous slots, in which case all the k must be buffered. To handle the worst case situation, the switch should be capable of switching up to N cells to their respective outputs in one slot time if cells are not to be dropped at the input (i.e., the switch should operate at N times the line rate). Observe that the cells destined for different outputs do not interfere with each other at the inputs and are not delayed at the input. Thus the switch is work-conserving in the sense that no output link is idle while there is a cell to be transmitted on that link in the switch. This is true because every arriving cell is sent to the queue on the output link in the same slot that it arrived. Thus the OQ switch can achieve 100% throughput on the output links.

The queueing abstraction for an output port—say, j—of this switch is a single-server, discrete time queue with fixed-length service times and an input process that is a superposition of the arrival processes from each of the inputs to output j.

Now consider the IQ switch with one FIFO queue at each input. The switch can transfer at most one packet from an input, and at most one packet to an output in a slot. If, at the beginning of a slot, more than one head-of-line (HOL) cells from the input queues have the same destination, then only one of them is switched and transmitted on the output link in the slot. The other HOL cells continue to be queued at their inputs. If any of these inputs contains a non-HOL packet whose destination is free, it is not switched, because the queue is FIFO and the packet at the head of the queue is blocked. Thus, packets in an IQ switch can experience *head-of-line blocking*, in which a blocked HOL cell blocks the cells behind it in the input FIFO queue even though the destination ports of these other cells are free and are idling. Thus the IQ switch with FIFO discipline is *non-work-conserving*, in the sense that there may be cells queued in the switch that are to be transmitted on an output port but cannot be, and the output port idles. Because the IQ switch is non-work-conserving, its capacity is less than one cell per port per slot.

Given that the OQ switch has greater capacity than the IQ switch, why should we be interested in the IQ switch? To answer this, consider the construction complexity of both architectures. Because the queues are maintained at the inputs and because only one cell need be transmitted in the event of a destination conflict, the switch can operate at the same rate as the input and output links. This means that in an IQ switch each input should be capable of sending at most one cell in a slot, and each output should be capable of receiving at most one cell in a slot. Furthermore, the maximum transfer rates from memory (the rate in bits per second at which data can be read from or written to) used for the input queue should be twice that of the link rate. This is because, in a slot, at most one write operation (corresponding to an arriving cell) and one read operation (corresponding to reading the packet from the input queue and switching it to the output) are performed. However, in the case of the OQ switch, to handle the worst case situation the switch should operate at N times the line rate; that is, it should be capable of transferring up to N cells from the inputs to an output. Furthermore, it should allow a cell to be transmitted on the output link. This means that the memory used for the output queue should be capable of a memory transfer rate of $N + 1$ times the line rate for the OQ switch.

Exercise 10.1

a. What is the memory transfer rate required for an $N \times N$ IQ and OQ switch for $N = 16$, 32, and 64? Assume 64-bit cells and 10-Gbps line rates. Also assume that the internal organization of the switch uses a header of 32 bits per cell.

b. Find out about available memory technologies and their access times. Obtain information about the cost of, say, 1 MB of SRAM memory, and plot the access time versus cost function for this. Extrapolate and guess the cost of memory.

c. Repeat for 16- and 32-bit cells.

Although the switch speedup of N times the line rate may not be technologically infeasible, clearly the memory transfer rates required make it infeasible at high line rates. Thus, in terms of construction complexity, the IQ switch is probably the only technologically feasible option in the core of the Internet, where the number of ports required on the switches and the line rates are both very high. However, as we remarked earlier, the non-work-conserving property of the IQ switch means that its capacity is less than 100%. The question then is, how much less than 100% is the maximum achievable throughput of an IQ switch? We answer this question next.

We first consider an IQ switch with saturated inputs. Input saturation means that the input is always active and has a cell to transmit to an output in every slot; in other words, there is always a cell behind the HOL cell to take its place when the HOL cell departs from the input queue. If all the inputs are saturated, the rate at which cells depart from the switch is called the *saturation throughput* of the switch. There are two reasons for considering saturation throughput. First, the analysis is comparatively easy. Second, the results of this analysis give us insight into the capacity of the IQ switch. In fact, for a special case, we show that the saturation throughput is a lower bound for the capacity by showing that if the arrival rate is less than the saturation throughput, the input queues are stable. We then present an approximation argument to derive the saturation throughput and hence the capacity of the IQ switch.

10.1.1 Saturation Throughput and Capacity (⋆)

Consider a saturated $N \times N$ cell switch with uniform routing (i.e., the destination of each cell is independently and randomly chosen from among the N outputs). Assume that the input queues are FIFO and that only the packets at the head of the queues at the beginning of a slot can be switched to the output in the slot. Now consider the HOL cells destined for a tagged output (defined soon)—say, output O_j. Conceptually, we can view these cells as being in a queue to get to O_j. Of course there is no such physical queue. Call this the HOL queue for output j, denoted by HOL_j for $j = 1, \ldots, N$. Let $Q_j^{(H)}(t)$ be the number of cells in HOL_j at the beginning of slot t. If the inputs are saturated, $0 \leq Q_j^{(H)}(t) \leq N$ and $\sum_{j=1}^{N} Q_j^{(H)}(t) = N$.

An example of the evolution of the HOL queues for a 2×2 switch is shown in Figure 10.1. In this example, if the inputs are saturated, we can say that cell a "came back" to HOL_1 as cell b at the beginning of slot 2. Similarly, cell e "went" as cell f to HOL_2 at the beginning of slot 3. We can say that in the HOL queue after a cell finishes service, it goes into any of the N HOL queues, with probability equal to that of a cell having that queue as its destination. This is exactly like a *closed queueing network*: a network of queues in which the total number of

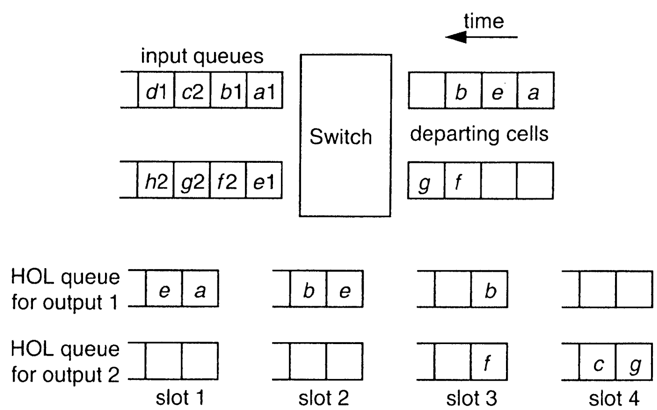

Figure 10.1 Evolution of the HOL queues in a 2 × 2 switch over four consecutive slots. The bottom panel shows occupancy of the HOL queues at the beginning of the slots. Letters are used to name a cell, and numbers indicate their destinations.

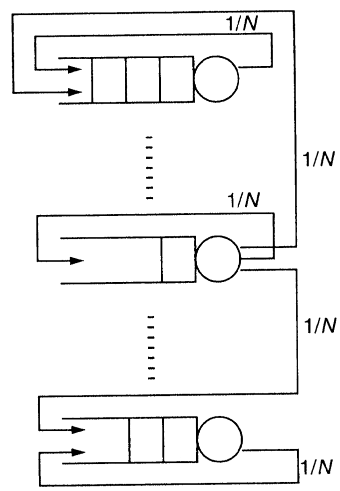

Figure 10.2 The closed queueing network representation of the HOL queues.

customers remains constant. In such a network, customers finishing service at one queue join another queue. There are no departures from the network nor external arrivals to it. Thus, for saturated inputs we can represent the set of HOL queues as a closed queueing network, shown in Figure 10.2. This means that the problem of finding the saturation throughput of the switch is the same as that of finding the throughput from any of the queues of such a closed queueing network. The closed queueing network is synchronous, and an exact analysis is intractable except for small N. We use this closed queueing network model in our analysis of the saturated IQ switch.

First, let us see what happens when an increasing number of inputs are saturated in an $N \times N$ switch. Consider input I_i, with n of the inputs (including I_i) being saturated. Define $\gamma^{(i)}(n)$ as the throughput from input I_i when n inputs are saturated and when the other $(N - n)$ inputs do not have any cell arrivals. In the closed queueing network, this corresponds to having n cells and N queues. We show that as n increases toward N, the throughput from input I_i decreases monotonically.

Lemma 10.1

For $1 \leq n \leq N$, $\gamma^{(i)}(n) \geq \gamma^{(i)}(n+1)$.

Proof: Consider a closed queueing network with n cells and N queues (like that shown in Figure 10.2) and cell C_i from input I_i. The throughput from I_i is the number of times C_i is switched to its desired output per unit time. (This in turn is the reciprocal of the average time for C_i to be switched once to its output.) The total throughput is $n\gamma^{(i)}(n)$. Now add one more cell, but give it the lowest priority. Clearly, the addition of this new cell does not affect the throughput of the n cells that are already in the system, and their throughput is still $\gamma^{(i)}(n)$. Because the cells are indistinguishable from each other in their routing behavior, the total throughput with $(n+1)$ cells is $(n+1)\gamma^{(i)}(n+1)$, and this is higher than that with n cells. The increase in the throughput is caused by the contribution of the new cell, and because its priority is the lowest, its throughput is less than or equal to that of the others. That is,

$$(n+1)\gamma^{(i)}(n+1) - n\gamma^{(i)}(n) \le \gamma^{(i)}(n)$$

$$(n+1)\gamma^{(i)}(n+1) \le (n+1)\gamma^{(i)}(n)$$

■

We remark here that the preceding argument is based on random selection of the cells from the HOL queues. If the HOL queue contains only the low-priority cell, that cell is selected. If there are others cells, any one of the others is randomly selected. Because the cells are indistinguishable, the total throughput is the sum of the throughputs of all the cells.

Lemma 10.1 suggests that an arrival rate less than $\gamma^{(i)}(N)$ should be a sufficient condition for the stability of the input queue at I_i and defines the per-port capacity of the IQ switch when all the inputs have the same packet arrival rate and when each packet chooses its destination independently and randomly. Before we show that this is indeed the case, we present a few definitions.

Definition 10.1

In a slotted service system of N queues, let $Q_i(t)$ be the number of cells in queue i at the beginning of slot t. Let $\mathrm{Q}(t) := [Q_1(t), \ldots, Q_N(t)]$ denote the queue-length vector at the beginning of slot t. The queueing system is considered stable if the distribution of $\mathrm{Q}(t)$ as $t \to \infty$ exists and is proper:

$$\lim_{t\to\infty} \Pr(\mathrm{Q}(t) \le \mathrm{m}) = \mathrm{Q}_{\mathrm{m}} \text{ and } \lim_{\mathrm{m}\to\infty} \mathrm{Q}_{\mathrm{m}} = 1$$

Here $\mathbf{m}$ is an N-dimensional vector and $Q_{\mathbf{m}}$ is the limiting distribution. Furthermore, $Q(t)$ is said to be *substable* if the following is true:

$$\lim_{\mathbf{m}\to\infty} \liminf_{t\to\infty} \Pr(Q(t) \leq \mathbf{m}) = 1$$ ■

If $\{Q(t)\}$ is substable, it means that $Q(t)$ is finite with probability 1; that is, there is no "escape of probability mass to infinity as $t \to \infty$." However, the probabilities need not converge to a single probability distribution. A substable, aperiodic, irreducible, discrete time Markov chain is stable.

Because all inputs have identical statistical behavior, we choose to concentrate on one input, called the *tagged input*, and characterize its behavior. Let input I_i be the tagged input of an $N \times N$ switch with the other $(N-1)$ inputs being saturated. Let the cell arrivals to this tagged input be from a Bernoulli process of rate λ. Assume that whenever the queue at I_i becomes empty, a dummy cell with a uniformly assigned destination is placed in the queue. If a new cell arrives before the dummy cell departs, it takes the place of the dummy cell and also adopts its destination. Thus all the inputs are saturated, and the closed queueing network model of Figure 10.2 with N queues and N cells can be used. Now consider the instants, $\{t_m, m \geq 1\}$, at which a cell departs from input queue I_i. Let $Q(t_m)$ be the number of cells in input queue I_i at instant t_m. Define $S_m = t_m - t_{m-1}$, and let A_m be the number of cell arrivals to input queue I_i in S_m. These are shown in Figure 10.3. The evolution of input queue I_i, embedded at these instants, can be written as

$$Q(t_{m+1}) = \max\{(Q(t_m) + A_{m+1} - 1), 0\} \tag{10.1}$$

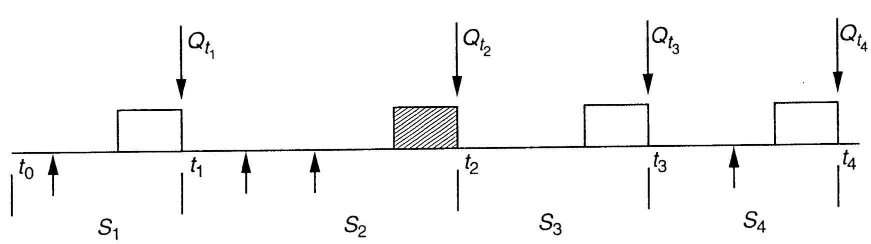

Figure 10.3 Queue evolution in the tagged input illustrating $\{t_m\}$, $\{S_m\}$, and $\{Q(t_m)\}$. The shaded cell is a dummy cell and the others are real cells. The upward arrows indicate packet arrivals.

In Section 5.6.2 we discuss how, for a stationary process $\{A_m\}$, $Q(t_m)$ as defined here almost surely converges in distribution to a proper random variable if $\sum_{r=1}^{m}(A_r - 1) \to -\infty$. Thus, intuitively, the input queue is stable if the sequence $\{A_m\}$ is stationary and if $\mathsf{E}(A_1) < 1$. The latter condition implies that the average number of cell arrivals between the times that a cell is being serviced by the output is less than 1.

In fact, it is sufficient to have $\{A_m\}$ to be asymptotically stationary; that is, the finite dimensional random vectors $(A_{J+r+m_1}, A_{J+r+m_2}, \ldots, A_{J+r+m_k})$ should not depend on r for arbitrary k, and $m_1, m_2, \ldots, m_k$ as $J \to \infty$. If the service from the HOL queues is in random order, then it can be shown that the sequence $\{A_m\}$ is indeed asymptotically stationary. We skip that proof here. We are now ready to derive the stability condition for the tagged input queue and the system of N input queues of the switch to be stable.

Theorem 10.1

(i) The tagged input queue is stable if $\lambda < \gamma(N)$.

(ii) The system of N queues is stable if $\lambda_i < \gamma(N)$ for $i = 1, 2, \ldots, N$.

Proof:

(i) For the tagged input queue evolving according to Equation 10.1, the following are almost surely true:

$$\lim_{m\to\infty} \frac{\sum_{r=1}^{m} A_r}{\sum_{r=1}^{m} S_r} = \lambda \qquad \lim_{m\to\infty} \frac{\sum_{r=1}^{m} 1}{\sum_{r=1}^{m} S_r} = \gamma(N)$$

The left expression is the arrival rate of packets to input I_i. The right expression is the definition of the saturation throughput of input I_i. This implies that if $\lambda < \gamma(N)$, then almost surely

$$\lim_{m\to\infty} \frac{\sum_{r=1}^{m}(A_r - 1)}{\sum_{r=1}^{m} S_r} < 0$$

and hence, almost surely $\sum_{r=1}^{\infty}(A_r - 1) = -\infty$. This shows that the process $Q(t)$ is stable in the sense of Definition 10.1. Because $Q(t_m)$ converges in distribution to a proper random variable and because $\{A_m\}$ is a stationary random process with a proper marginal distribution, it follows that $Q(t)$ is substable if $\lambda < \gamma(N)$. Note that we cannot conclude that $Q(t)$ is stable because we are only upper-bounding it by a stable process. Let $X^{(j)}(t)$ be the output port of the HOL cell in input I_j. Note that $\{(Q(t), X^{(1)}(t), \ldots, X^{(N)}(t)),\ t > 0\}$ is a multidimensional, irreducible, and aperiodic discrete time Markov chain. Hence substability implies stability of $Q(t)$, and a sufficient condition for the stability of the tagged input queue is $\lambda < \gamma(N)$.

(ii) Now consider the system of N input queues described by the N-dimensional vector process $\{\mathbf{Q}(t) = (Q^{(1)}(t), \ldots, Q^{(N)}(t)),\ t > 0\}$. Let λ_i be the Bernoulli cell arrival rate to input I_i. From part (i), $\lambda_i < \gamma(N)$ is sufficient for queue I_i to be stable. When $\lambda_i < \gamma(N)$,

$$\lim_{m\to\infty} \lim_{t\to\infty} \Pr\left(Q^{(i)}(t) \le m_i\right) = 1, \quad 1 \le i \le N \tag{10.2}$$

Define $\mathbf{m} = (m_1, m_2, \ldots, m_N)$ and we have

$$\begin{aligned} 1 &\ge \lim_{\mathbf{m}\to\infty} \lim_{t\to\infty} \Pr\left(Q^{(i)}(t) \le m_i, i = 1, 2, \ldots, N\right) \\ &\ge 1 - \sum_{i=1}^{N} \lim_{m_i\to\infty} \lim_{t\to\infty} \Pr\left(Q^{(i)}(t) > m_i, i = 1, 2, \ldots, N\right) \\ &= 1 \end{aligned}$$

The second inequality follows from De Morgan's theorem, and the union bound and the last equality follow from Equation 10.2. This means that

$$\lim_{\mathbf{m}\to\infty} \lim_{t\to\infty} \Pr(\mathbf{Q}(t) \le \mathbf{m}) = 1$$

Thus $\mathbf{Q}(t)$ is substable when $\lambda_i < \gamma(N)$ for $i = 1, 2, \ldots, \gamma(N)$. If $X^{(i)}(t)$ is as defined earlier, then $\{(\mathbf{Q}(t), X^{(1)}(t), X^{(2)}(t), \ldots, X^{(N)(t)}, t > 0)\}$ is an aperiodic, irreducible Markov chain, and part (ii) of the theorem is proved. ■

We reiterate that packet arrival rate for which the input queues will remain stable, as derived earlier, is for Bernoulli arrivals at the inputs with uniform routing and random order of service from the HOL queues. It is generally believed that