Example 4: A convoluted recurrence.

Our next example is especially important. In fact, it's the "classic example" of why generating functions are useful in the solution of recurrences.

locks.

Suppose we have $n+1$ variables $x_0, x_1, \ldots, x_n$ whose product is to be computed by doing n multiplications. How many ways C_n are there to insert parentheses into the product $x_0 \cdot x_1 \cdot \ldots \cdot x_n$ so that the order of multiplication is completely specified? For example, when $n = 2$ there are two ways, $x_0 \cdot (x_1 \cdot x_2)$ and $(x_0 \cdot x_1) \cdot x_2$. And when $n = 3$ there are five ways,

$$x_0 \cdot (x_1 \cdot (x_2 \cdot x_3)),\ x_0 \cdot ((x_1 \cdot x_2) \cdot x_3),\ (x_0 \cdot x_1) \cdot (x_2 \cdot x_3),$$
$$(x_0 \cdot (x_1 \cdot x_2)) \cdot x_3,\ ((x_0 \cdot x_1) \cdot x_2) \cdot x_3.$$

Thus $C_2 = 2$, $C_3 = 5$; we also have $C_1 = 1$ and $C_0 = 1$.

Let's use the four-step procedure of Section 7.3. What is a recurrence for the C's? The key observation is that there's exactly one '$\cdot$' operation outside all of the parentheses, when $n > 0$; this is the final multiplication that ties everything together. If this '$\cdot$' occurs between x_k and x_{k+1}, there are C_k ways to fully parenthesize $x_0 \cdot \ldots \cdot x_k$, and there are C_{n-k-1} ways to fully parenthesize $x_{k+1} \cdot \ldots \cdot x_n$; hence

$$C_n = C_0 C_{n-1} + C_1 C_{n-2} + \cdots + C_{n-1} C_0, \qquad \text{if } n > 0.$$

By now we recognize this expression as a convolution, and we know how to patch the formula so that it holds for all integers n:

$$C_n = \sum_k C_k C_{n-1-k} + [n=0]. \tag{7.66}$$

Step 1 is now complete. Step 2 tells us to multiply by z^n and sum:

$$\begin{aligned} C(z) &= \sum_n C_n z^n \\ &= \sum_{k,n} C_k C_{n-1-k} z^n + \sum_{n=0} z^n \\ &= \sum_k C_k z^k \sum_n C_{n-1-k} z^{n-k} + 1 \\ &= C(z) \cdot zC(z) + 1. \end{aligned}$$

The authors jest.

Lo and behold, the convolution has become a product, in the generating-function world. Life is full of surprises.

Step 3 is also easy. We solve for $C(z)$ by the quadratic formula:

$$C(z) = \frac{1 \pm \sqrt{1-4z}}{2z}.$$

But should we choose the $+$ sign or the $-$ sign? Both choices yield a function that satisfies $C(z) = zC(z)^2 + 1$, but only one of the choices is suitable for our problem. We might choose the $+$ sign on the grounds that positive thinking is best; but we soon discover that this choice gives $C(0) = \infty$, contrary to the facts. (The correct function $C(z)$ is supposed to have $C(0) = C_0 = 1$.) Therefore we conclude that

$$C(z) = \frac{1 - \sqrt{1-4z}}{2z}.$$

Finally, Step 4. What is $[z^n]\,C(z)$? The binomial theorem tells us that

$$\sqrt{1-4z} = \sum_{k\geq 0} \binom{1/2}{k} (-4z)^k = 1 + \sum_{k\geq 1} \frac{1}{2k} \binom{-1/2}{k-1} (-4z)^k;$$

hence, using (5.37),

$$\begin{aligned} \frac{1-\sqrt{1-4z}}{2z} &= \sum_{k\geq 1} \frac{1}{k} \binom{-1/2}{k-1} (-4z)^{k-1} \\ &= \sum_{n\geq 0} \binom{-1/2}{n} \frac{(-4z)^n}{n+1} = \sum_{n\geq 0} \binom{2n}{n} \frac{z^n}{n+1}. \end{aligned}$$

The number of ways to parenthesize, C_n, is $\binom{2n}{n}\frac{1}{n+1}$.

So the convoluted recurrence has led us to an oft-recurring convolution.

We anticipated this result in Chapter 5, when we introduced the sequence of *Catalan numbers* $\langle 1, 1, 2, 5, 14, \ldots \rangle = \langle C_n \rangle$. This sequence arises in dozens of problems that seem at first to be unrelated to each other [46], because many situations have a recursive structure that corresponds to the convolution recurrence (7.66).

For example, let's consider the following problem: How many sequences $\langle a_1, a_2 \ldots, a_{2n} \rangle$ of $+1$'s and -1's have the property that

$$a_1 + a_2 + \cdots + a_{2n} = 0$$

and have all their partial sums

$$a_1, \quad a_1 + a_2, \quad \ldots, \quad a_1 + a_2 + \cdots + a_{2n}$$

nonnegative? There must be n occurrences of $+1$ and n occurrences of -1. We can represent this problem graphically by plotting the sequence of partial

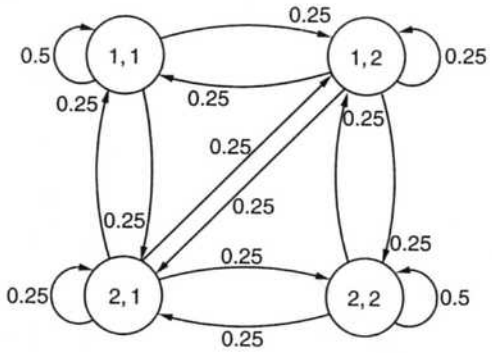

Figure 1.5 Markov chain representation of the state of the 2 × 2 input queued switch at the beginning of every slot. The numbers next to the arrows are the transition probabilities. The state of the Markov chain is $[d^{(1)}, d^{(2)}]$, where $d^{(i)}$ is the destination port of the packet at input i.

the Markov chain. The saturation throughputs for eight values of N are shown in Table 1.1 after obtaining them using this technique. As shown in the table, the throughput decreases as N increases. The question then is, does the saturation throughput converge to some value sufficiently greater than zero, or would it continue to decrease to zero with increasing N, thereby negating the technological advantages of the input-queued switch through a significant performance penalty for large N? The answer to this question is in the asymptotic analysis, discussed

N	Saturation throughput
1	1.0000
2	0.7500
3	0.6825
4	0.6553
5	0.6399
6	0.6302
7	0.6234
8	0.6184

Table 1.1 Saturation throughput of an $N \times N$ input-queued switch for $N = 1, 2, \cdots, 8$.

in the cases not covered here, the input queues will be stable if the arrival rate is less than the saturation throughput of the switch, although this has not been formally proved for any other case. Specifically, the case of FIFO service of the HOL queues has not been considered, nor has the case when more than one cell can be switched from each input or to an output. This leads us to make the following definition.

Definition 10.2

For a specified distribution of the cell destinations, we say that the arrival process satisfies the *saturation stability property* if the input queues are stable whenever the expectation of the number of arrivals in each slot to each input is less than the saturation throughput from that input. ■

We use this property in discussing the delays in the switches.

10.1.2 Saturation Throughput of an IQ Switch

In Chapter 1 we consider a Markov chain model of an input saturated $N \times N$ IQ switch for arbitrary finite N. Recall that the saturation throughput per port decreases as N increases (see Table 1.1). The numerical values suggest that there is probably a limiting value of the saturation throughput as $N \to \infty$.

Consider what happens in the saturated IQ switch in each slot. From each nonempty HOL queue, one cell is transmitted to the output, and a total of $D(t)$ cells are transmitted in slot t. We assume that the cells depart at the end of a slot, that fresh packets arrive at the beginning of a slot, and that the arrivals in a slot are available for departure in that slot. Let $Q(t)$ denote the total number of HOL packets in the input queues at the end of the slot, after the switching of the packets. This is shown in Figure 10.4. Because the input queues are saturated, these $D(t)$ cells are replaced by fresh cells in slot $(t + 1)$. Thus into the HOL queues of the outputs, $D(t)$ cells arrive in slot $(t+1)$, each independently choosing any of outputs $1, \ldots, N$ with probability $1/N$ (i.e., they are uniformly routed). Let $A_j(t + 1)$ be the number of new cells arriving into HOL_j in slot $t + 1$. $A_j(t + 1)$ has a binomial distribution with mean $D(t)/N$.

$$\Pr(A_j(t+1) = k) = \binom{D(t)}{k} \left(\frac{1}{N}\right)^k \left(1 - \frac{1}{N}\right)^{D(t)-k}$$

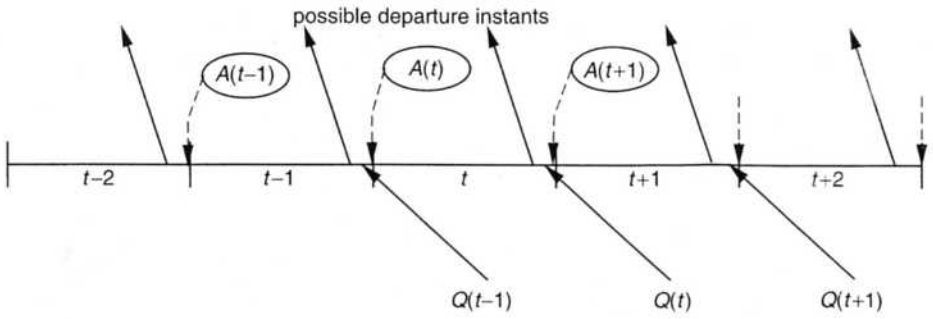

Figure 10.4 The arrival instants, the departure instants, and the instants at which the queue state is observed in each slot. Solid lines show possible departure instants, and dashed lines show arrival instants. *A(t)* is the number of new arrivals in slot *t*. Slots *t*−2,..., *t*+2 are shown.

$\sum_{j=1}^{N} Q_j(t)$ is the number of cells remaining in the inputs at the end of slot t, and we can write

$$D(t) = N - \sum_{1}^{N} Q_j(t) = \sum_{j=1}^{N} A_j(t+1) \tag{10.3}$$

As before, let $\gamma(N)$ denote the saturation throughput from each output port. Note that $\gamma(N)$ can also be interpreted as the probability of a cell departing from an output—say, output j. Under stationary conditions, $\gamma(N) = \mathsf{E}(D)/N$, where $\mathsf{E}(D)$ is the stationary average of the number of cells departing from the switch in a slot. Taking expectations in the first equality of Equation 10.3, dividing by N, and using the fact that all outputs are statistically identical, we get

$$\gamma(N) = 1 - \mathsf{E}(Q_j) \quad \text{or} \quad \mathsf{E}(Q_j) = 1 - \gamma(N) \tag{10.4}$$

Under input saturation, the average number of fresh cell arrivals to an HOL queue in a slot is equal to the saturation throughput from an input port and thus is $\gamma(N)$. To obtain $\gamma(N)$ as $N \to \infty$, we use the following lemma.

Lemma 10.2

Let $\gamma(\infty) = \lim_{N\to\infty} \gamma(N)$. In the steady state, if exactly one cell is switched from each nonempty HOL queue, then as $N \to \infty$, $A_j(t)$ has a Poisson distribution with mean $\gamma(\infty)$ and is independent of the number in the queue. ■

Let us now consider the evolution equation for $Q_j(t)$. From Figure 10.4, we can write the following.

$$\begin{aligned} Q_j(t+1) &= \max\left(0, Q_j(t) - 1 + A_j(t+1)\right) \\ &= Q_j(t) + A_j(t+1) - \Delta_{Q_j(t)+A_j(t+1)} \end{aligned} \tag{10.5}$$

where

$$\Delta_Q \triangleq \begin{cases} 1 & \text{if } Q > 0 \\ 0 & \text{if } Q = 0 \end{cases}$$

Taking expectations on both sides of Equation 10.5 we get

$$\mathsf{E}\big(Q_j(t+1)\big) = \mathsf{E}\big(Q_j(t)\big) - \mathsf{E}\Big(\Delta_{Q_j(t)+A_j(t+1)}\Big) + \mathsf{E}(A(t+1))$$

In steady state, the statistics of $Q_j(t+1)$ and $Q_j(t)$ will be identical, and $\mathsf{E}\big(Q_j(t+1)\big) = \mathsf{E}\big(Q_j(t)\big)$, leaving us $\mathsf{E}\Big(\Delta_{Q_j(t)+A_j(t+1)}\Big) = \mathsf{E}(A(t+1))$. Define p_0 to be the steady state probability that $(Q_j(t) + A_j(t+1)) = 0$.

Exercise 10.2

Show that in steady state, $\mathsf{E}\Big(\Delta_{Q_j(t)+A_j(t+1)}\Big) = 1 - p_0$ and $p_0 = 1 - \gamma(\infty)$.

Denote the moment-generating function of $Q_j(t)$ and $A_j(t)$ by $\mathcal{Q}_j(t,z)$ and $\mathcal{A}_j(z,t)$, respectively. They are defined by

$$\mathcal{Q}_j(t,z) := \mathsf{E}\Big(z^{Q_j(t)}\Big) \qquad \mathcal{A}_j(t,z) := \mathsf{E}\Big(z^{A_j(t)}\Big)$$

To obtain $\mathcal{Q}_j(t,z)$, we proceed as follows.

$$\begin{aligned}\mathcal{Q}_j(t+1,z) &:= \mathsf{E}\left(z^{Q_j(t+1)}\right)\\ &= \mathsf{E}\left(z^{Q_j(t)-\Delta_{Q_j(t)+A_j(t+1)}+A_j(t+1)}\right)\\ &= p_0 + \sum_{k=1}^{\infty} \mathsf{Pr}\left(Q_j(t)+A_j(t+1)=k\right) z^{k-1}\\ &= p_0 + \frac{1}{z}\left(\mathcal{Q}_j(t,z)\mathcal{A}_j(t+1,z) - p_0\right)\end{aligned}$$

We obtain the third equality by separating the term for the case of $Q_j(t) = A_j(t+1) = 0$. The last equality follows from the assumption that the number of new packets into an HOL queue is independent of the current occupancy. In steady state, $\mathcal{Q}_j(t,z) = \mathcal{Q}_j(t+1,z)$, and we can drop the dependence on t to write the moment-generating function as $\mathcal{Q}_j(z)$. Similarly for $\mathcal{A}_j(t+1,z)$. The last equation then simplifies to

$$\mathcal{Q}_j(z) = \frac{p_0(1-z)}{\mathcal{A}_j(z) - z} \tag{10.6}$$

Substituting $p_0 = 1 - \gamma(\infty)$ and noting that because the number of new arrivals to an HOL queue in a slot is from a Poisson distribution with mean $\gamma(\infty)$, $\mathcal{A}_j(z) = e^{\gamma(\infty)(z-1)}$, we get

$$\mathcal{Q}_j(z) = \frac{(1-\gamma(\infty))(1-z)}{e^{\gamma(\infty)(z-1)} - z}$$

Differentiating with respect to z, putting $z = 1$, and using Equation 10.4 as $N \to \infty$, we get

$$\mathsf{E}(Q_j) = \frac{\gamma(\infty)^2}{2(1-\gamma(\infty))} \tag{10.7}$$

Solving for $\gamma(\infty)$ from Equations 10.4 and 10.7, $\gamma(\infty) = 2 - \sqrt{2} \approx 0.586$.

For an $N \times N$ IQ switch under independent, uniform routing and FIFO buffering at each input, as $N \to \infty$, the saturation throughput is $2 - \sqrt{2} \approx 0.586$.

10.1.3 Discussion

From the foregoing results, we see that queue placement in a cell switch must trade off the simple implementation and lower throughput of the IQ switch and the complex, possibly infeasible implementation and higher throughput of the OQ switch. We can say that an $N \times N$ OQ switch needs $O(N)$ times more resources than an $N \times N$ IQ switch and provides approximately twice the capacity. Many proposals have been made to design a switch whose complexity is a constant times that of the IQ switch. These designs essentially take one of two approaches. The maximum number of cells that can be transmitted from an input or to an output port in a slot is increased. This method necessarily requires that queues be maintained at both the input and the output (CIOQ switches). The second approach is to change the FIFO scheduling at the input buffers. In the next section we consider these design choices for CIOQ switches.

10.2 Combined Input–Output Queueing

10.2.1 The Knockout Principle

A simple way of achieving the throughput characteristics of an OQ switch but with an implementation complexity that does not grow with N as compared with that of an IQ switch is to have the capacity to resolve a smaller number of output conflicts rather than N. Consider a switch that can handle up to L conflicts to an output port; that is, if, in a slot, n cells arrive for an output port, then if $n \leq L$, all n cells are switched to the respective outputs, whereas if $n > L$, then an arbitrary L of these are switched to the output. The *knockout* switch takes this approach and drops the cells that are not switched.

Consider an $N \times N$ knockout switch. Assume that the cell arrival to each input is a Bernoulli process with mean λ. Also assume uniform routing. In a slot, the number of arrivals for a tagged output will be a binomially distributed random variable with mean $\frac{\lambda}{N} \times N = \lambda$:

$$\mathsf{Pr}(k \text{ cells with destination } j) = \binom{N}{k} \left(\frac{\lambda}{N}\right)^k \left(1 - \frac{\lambda}{N}\right)^{N-k}$$

For large N we can approximate a binomial random variable by a Poisson random variable. With this approximation, the average number of dropped packets per