Chapter 4 Interconnection Network Problems

Due Wed. 4/14/04

1. Show a design of a 3 stage network with N=9 inputs/outputs comprised entirely of 3x1 muxes such that all 9! complete connection mappings of inputs to outputs can be implemented, but sometimes rearrangement of existing paths is necessary to make a new connection path. What size switch modules can be used? Illustrate the design. Show a blocked state for a new connection path request from an unused input to an unused output using a minimal number of existing connection paths.

2. Suppose the network in Prob. 1 has on the average 6 of the 9 inputs connected to outputs. What is the average probability that a new connection path request from an unused input to an unused output is blocked by existing connection paths?

3. Repeat Prob. 1 but now the design must have nonblocking capability. What sizes of muxes must be used.. illustrate the network. What is total crosspoint count of the nonblocking design?

4. Suppose that N=512. Show a minimal crosspoint nonblocking 3 stage design. What is the number of crosspoints?

5. Suppose that N=64 and a multistage network must be completely built with 2x2 switch modules. It is acceptable that the network can block new connection path requests and when this occurred it is acceptable that existing connection paths are arranged to provide for this new connection path. What is a minimal number of 2x2’s that can be used for such a design? How many stages of 2x2’s must be used?

6. In a 64x64 3 stage network comprised entirely of 8x8 switch modules, if a blocking state existed for a connection path request from a unused input to an unused output, what would be the most existing connection paths that would have to be rearranged to make the new connection path.

The following material on interconnection networks is an extension of the discussions in Chapter 5 of the CNF text regarding switching network architectures.
10–4 Digital Switching Networks

We have alluded a number of times to the switching of calls at a given office or exchange from some input trunk group or set of lines to a trunk at a prescribed output group. Examples appear in Fig. 10–14. The same process of directing
incoming messages that arrive on a given link to the appropriate outgoing link was encountered in the packet-switched environment discussed in previous chapters. The process was described in detail in Chapter 6 in our description of routing in networks.

No specific switch is required in a packet-switched node since all packets are queued for outbound transmission. In the case of circuit switching a dedicated connection between input and output lines or trunks at an exchange must be made, however, and a switch is necessary. Historically, so-called space-division switches have been generally used. More commonly nowadays, time-division switches are being implemented. In large switches both time and space switching are used. We shall discuss both techniques in this section and show how they are analogous to one another. We begin by focusing on the process of space-division switching, then show how the same process can be carried out through time-division techniques. The words switch and switching network are used interchangeably to describe the device that carries out the switching.

Consider first the case of switching any one of \( N \) inputs to one of \( N \) outputs. An example, for \( N = 6 \), appears in Fig. 10–19. The switch in this case corresponds to an \( N \times N \) (square) array. Semiconductor switches or metallic contacts located at each of the crosspoints where input and output wires cross may be enabled, allowing any one of the input wires to be uniquely connected to any one of the outputs and thus establishing the connection. For this square array a connection between input and output is always possible (providing the output to which a connection is to be made is not already connected, i.e., busy). A switch of this type is called a nonblocking switch. A measure of its complexity is the number of crosspoints needed, generally \( N^2 \), or \( N^2 - N \), if the inputs and

![Figure 10–19 Square switching array, \( N = 6 \)]
outputs refer to the same set of terminals to be connected together. (In this latter case, the terminal connected to input line \( i \) is also connected to output line \( i \), \( 1 \leq i \leq N \). A terminal can then both generate and receive a call.)

More generally, an \( N \times K \) matrix may be used. (The notation used here is different from that of Fig. 10–13, but there should be no confusion since the calculation of blocking probability due to lack of an output trunk, as carried out in the previous section with reference to Fig. 10–13, will be consistently decoupled from the blocking properties of the switch, under consideration here.) It is apparent that if \( K \geq N \) the switch is again nonblocking. For \( K < N \), however, blocking by the switch is possible. An example, for \( N = 8 \) and \( K = 4 \), appears in Fig. 10–20. Four connections—1 to 2, 2 to 1, 3 to 3, and 4 to 4—are shown established. (Note that this implies that the outputs are different from the inputs.) Input lines 5–8 are thus blocked by the switch: Connections cannot be made to any of the output lines.

As the number of users or lines connected increases, the size and complexity of the switch increase correspondingly. As just noted, the complexity of a space switch is generally measured by the number of crosspoints required. For example, if 100,000 trunks are to be interconnected (this is roughly the number of trunk terminations in the AT&T No. 4 ESS tandem switch [BSTJ 1977]) and a square array is used, \( N^2 = 10^{10} \) crosspoints are required, obviously a tremendously large number, even in this era of VLSI. Can one reduce this number and

---

**Figure 10–20** Switching array, \( 8 \times 4 \)
yet retain either a nonblocking switch or one with a very small probability of switch blocking? The common way is to go to multiple switching stages. Consider a three-stage space-division example. (Time-division switches will be considered shortly.)

Let the number of input lines or inlets \( N \) equal the number of outputs for simplicity. Group the \( N \) inlets (outlets) into \( N/n \) switch arrays, each such array comprising an \( n \times k \) rectangular matrix. Figure 10-21(a) shows an \( n \times k \) switch.

Figure 10-21 A three-stage switching network
a. Typical \( n \times k \) switch and its symbolic representation
b. Three-stage switch
array and the simplified symbolic representation for it. Using this symbolic representation, Fig. 10–21(b) portrays the complete three-stage switch. There are $N/n$ input and $N/n$ output arrays, comprising stages 1 and 3, respectively. The second stage is made up of $k$ square arrays, as shown, each with $N/n$ inputs and outputs. Each of the output lines of a stage-1 array is connected to a different stage-2 array, in the same position as its array location in stage 1. Thus each of the output lines of the first array in stage 1 is connected to input line 1 of its corresponding array in stage 2, each output line of the second array in stage 1 is connected to its corresponding input line 2 in stage 2, and so forth. Similar connections are made between stage-2 and stage-3 arrays.

The total number of crosspoints in this case is

$$C = 2 \frac{N}{n} (nk) + k \left( \frac{N}{n} \right)^2$$

$$= 2Nk + k \left( \frac{N}{n} \right)^2$$  \hspace{1cm} (10–48)

Through proper choice of the parameters $n$ and $k$ the complexity of the switch can be reduced to a value considerably less than $N^2$, the crosspoint count of the single-stage square array.

A multiple-stage switch of this type (with extensions to four, five, or more stages) may exhibit nonblocking or blocking behavior, as in the single-stage case. We first consider the nonblocking case and show that the switch complexity (as measured by the number of crosspoints) can be reduced below the single-stage case. By allowing some blocking, however, even more reduction in size (and hence cost) is obtained. Most large modern switches operate as blocking networks, but with very small blocking probabilities. They are thus often called "essentially nonblocking" switches. (The switch blocking probability, as will be shown in the next section, is typically much less than the link-blocking probability calculated in the previous section. Circuit-switched blocking behavior may then, to a good approximation, be modeled by the link-blocking probability calculated by the Engset or Erlang formulas.)

C. Clos of Bell Laboratories, in a seminal paper published in the *Bell System Technical Journal* in 1953, derived the requirement for a three-stage nonblocking switch [CLOS]. This is given simply, in the notation of Fig. 10–21, by the condition $k = 2n - 1$. This relation is easily demonstrated as follows: Say that a call coming in at a particular stage-1 array is to be connected to an outlet in a given stage-3 array. Say $(n - 1)$ inputs in the same stage 1 array are already

connected (busy), as are \((n - 1)\) outputs in the desired stage-3 array. In the worst case let the \((n - 1)\) busy inputs and \((n - 1)\) busy outputs use different arrays in stage 2. To make a connection the desired call must be routed through an *unused* array in stage 2. Otherwise blocking will occur. (Recall, from Fig. 10–21, that each of the \(k\) lines connected from an array in stages 1 and 3 to stage 2 uses a different array in stage 2.) For an array in stage 2 to be available, then, we must have

\[
k = (n - 1) + (n - 1) + 1 = 2n - 1
\]

(10–49)

This condition is thus sufficient to guarantee an available path through the switch.

An example, with \(n = 5\) and \(k = 9\), appears in Fig. 10–22. A call arrives on input line 5 of the array shown in stage 1. All other inputs to this array are busy at this time. They are assumed to occupy lines 1–4 at the output of the array, as
shown. (Any other four output lines could be used just as well.) This call is to be
directed to the array indicated in stage 3. Lines 2–5 at the output of this array
are shown as busy at the time. They must therefore be connected to four of the
arrays in stage 2. These are shown as arrays 5–8 in Fig. 10–22. (In this worst-
冷冻 case analysis, with arrays 1–4 of stage 2 connected to the array of stage 1, any
four of the five remaining arrays in stage 2 could have been chosen.) It makes no
difference to which arrays in stage 1 these calls are connected. The fact that
these calls are in progress eliminates the use of their lines for the new call to be
set up. It is apparent that the new call can still be connected through to the
output, however, using the remaining stage 2 array, number 9 as shown. Had
there been only eight stage-2 arrays \( k = 2n - 2 \), this would not have been
possible.

For the nonblocking rule of Eq. (10–49), the number of crosspoints from
Eq. (10–48) becomes

\[
C_{\text{nonblocking}} = 2N(2n - 1) + (2n - 1) \left( \frac{N}{n} \right)^2
\]

(10–50)

This function exhibits a minimum with respect to \( n \), the number of inputs in
each of the stage-1 arrays. Specifically, it is readily shown that for \( N \gg 1 \), the
optimum \( n \) is approximately

\[
n_{\text{opt.}} \approx \sqrt{\frac{N}{2}}
\]

(10–51)

with the corresponding number of crosspoints given by

\[
C_{\text{nonblocking opt.}} = 4\sqrt{2} \cdot N^{3/2}
\]

(10–52)

This is obviously an improvement over the \( N^2 \) behavior noted earlier. For
example, take \( N = 100,000 \). Then the number of crosspoints required for the
nonblocking three-stage switch is about \( 1.7 \times 10^8 \), a considerable reduction
from \( N^2 = 10^{10} \), the number required for the square single-stage switch. For
\( N = 10,000 \), Eq. (10–52) gives \( 5.6 \times 10^6 \), compared with \( 10^8 \) for the single-
stage case.

There is one problem, however, with the optimum value of \( n \) of Eq.
(10–51). It does not guarantee that \( N/n \) is an integer, an obvious requirement
from Fig. 10–21. As an example, for \( N = 100,000 \), \( n = 222 \) from Eq. (10–51).
But \( N/n \) is not an integer in this case. The actual choice of \( n \) turns out to be
noncritical, however. Thus, for this same example, if \( n = 200 \), \( N/n = 500 \),
\( k = 2n - 1 = 399 \), and \( C = 1.8 \times 10^8 \), not much of a change from the minimum
value \( 1.7 \times 10^8 \). Similarly, for \( n = 250 \), \( N/n = 400 \), and \( C = 1.8 \times 10^8 \) also.

For a system this size \( (N = 10^5) \), even \( C = 10^8 \) (and the equivalent switch
cost and complexity for time-division systems) is too large a number. The num-
bers can be reduced by allowing some blocking, as noted earlier. For this pur-
pose, however, one must carry out a probabilistic analysis of the switch to
determine the switch-blocking probability as a function of the switch param-
eters. This we shall do after we study time-division switch structures.

First note, however, that the nonblocking condition for the three-stage
switch is readily extended to larger numbers of cascaded (tandem) stages by
using building blocks of three-stage switches. For example, let each of the square
arrays in stage 2 of the three-stage switch of Fig. 10–21 actually be a
three-stage nonblocking switch itself. The resultant switching network is a five-
stage one and obviously has the same nonblocking property [INOS, p. 94].
Further extensions may be made as well.

10–4–1 Time-division Switching

Our introductory discussion of switching networks has focused thus far on
space-division switching. These switches are equally applicable to analog and
digital message transmission. We now turn our attention to the more modern
time-division switches, which are applicable to digital switching only. These
switches are completely analogous to the space-division ones, with nonblocking
and blocking analyses carried out precisely the same way. As already noted,
large modern switching systems most frequently consist of cascaded (tandem)
combinations of the two types of switches.

A number of recent books describe digital switches, with particular empha-
sis on time-division switches, in much more detail than we do here. Reference
will be made to these books from time to time. They include the works by H.
Inose [INOS] and Collins and Pedersen [COLI.] already cited, a textbook by
John C. Bellamy [BELL], and a book edited by John C. McDonald [McDO]. This
last work is a particularly up-to-date compendium of tutorial chapters on various
aspects of digital switching. It includes discussions of hardware and software
issues in detail, as well as descriptions of a number of practical switching systems.

To carry out time switching, all calls or messages to be switched must first be
switched into recurrent time samples, with a group of successive samples appearing
on one physical line constituting a frame. The most common example in
telephony involves the pulse-code modulation (PCM) transmission of voice sam-

York, 1983.
ples [SCHW 1980a, Chap. 3]. Each voice signal to be converted to digital format is sampled at a rate of 8000 samples/sec, or once every 125 μsec. Frames are thus 125-μsec long. These samples are in turn normally digitized (quantized) to 8 bits each, so that a typical PCM voice channel requires 64 kbps of transmission capacity. Individual voice channels are in turn combined or multiplexed into one time stream. A recent CCITT Recommendation replaces PCM transmission with adaptive delta modulation at 32 kbps, almost doubling the number of voice channels that can be transmitted.

Two multiplexing formats are used worldwide. The North American standard, developed originally by AT&T in the United States and adopted by Canada and Japan as well, combines 24 8-bit voice channels into one time stream operating at 1.544 Mbps. (One framing bit is added at the beginning of each frame. The aggregate bit rate is thus \((24 \times 8 + 1) = 193\) bits per 125-μsec frame.) This is called the T1 system. A typical T1 frame appears in Fig. 10–23 [SCHW 1980a, pp. 138–140]. Each 8-bit word in a channel is also referred to as a time slot in the time-multiplexed signal stream. Although the T1 format was originally developed for telephony, with voice transmission as its primary objective, the 24-channel frame can carry any other kind of signal as well. Fifty-six-kbps data signals fall naturally, into this format, for example. (The eighth or least significant bit in each T1 voice channel is used every six frames for signaling purposes. In the case of 56-kbps data, then, one would simply ignore the eighth bit.) Details of the interfacing of T1 trunks with digital switching systems appear in [McDO]. The international standard, used by countries outside North America and Japan, multiplexes or combines 30 64-kbps voice channels plus two signaling channels for an aggregate bit rate of 2.048 Mbps. This system, too, can also be used for data (nonvoice) transmission.

Higher multiplexing rates are possible as well. For example, the AT&T No. 4 ESS combines five T1 streams for an aggregate of 120 8-bit channels. Eight additional time slots are added, so that a frame to be switched consists of 128 time slots [BSTJ 1977, p. 1024]. See [McDO] for details of both digital and analog interfaces with digital switching systems.

Assuming that message or signal segments each occupy a time slot in a frame, how is time switching carried out? The simplest procedure is the obvious one. Each frame, as it arrives over an incoming port at the switch, is written into a memory. Switching is then accomplished by simply reading out the individual words in any desired (switched) order. Such a device is called a time-slot interchanger (TSI) [McDO, Chap. 5], [INOS, Chaps. 3 and 5]. An example appears in Fig. 10–24. The frame consists of five time slots or channels, of which only two,
X and Y, are assumed to be active and communicating with one another. On the input side, user X’s data occupies channel 1; user Y’s data occupies channel 3. After storing each frame in memory, Y’s word is read out in, or transferred to, X’s time slot; X’s word is read out in Y’s time slot. The dashed lines in Fig. 10-24 indicate that there is a minimum of a single-frame delay introduced in interchanging the data words. More complex operations are possible as well.

Just as space switches have a crosspoint cost limit, time-slot interchangers have limitations on the number of channels per frame that may be multiplexed.
Multiplexing more channels requires more memory and becomes more costly as the number of channels to be stored increases. A more critical limit is determined by the access time required to read into and out of memory [BELL, p. 246]. Let $t_c$ be the memory cycle time in microseconds required to both write into and read out a channel word (sample) from memory. Take a 125-µsec frame time as the most common example. The maximum number of channels that can be supported is then [BELL], [TAWA]

$$N = \frac{125}{2t_c} \quad (10-53)$$

For 500-nsec logic, this says that the maximum number is 125. For 50-nsec logic, this increases to 1250, and so on. Recall that the AT&T No. 4 ESS uses 120-channel time interchangers, with 128 time slots, in its switching network [BSTJ 1977]. A switching system handling just this number of trunks (recall that the complete No. 4 ESS handles on the order of 100,000 trunks) would require $N^2 = 14,400$ crosspoints if a single-stage space switch were used. If a three-stage nonblocking space switch were used instead, $C = 7400$ crosspoints would be needed. The advantage of using time-switched technology is thus apparent. Note that the memory required is 8N bits, with 8 bits assumed per word. For $N = 120 - 128$ channels the memory required is only on the order of 1 Kbit. (Additional memory is required to control the memory addressing. Since each address is $\log_2 N$-bits long, this additional memory requires $N \log_2 N$ bits. For $N = 128$ channels, this only adds another 7N or 840 bits.)

Since memory cycle time limits the number of channels that may be switched using time-slot interchangers, larger switching systems have incorporated tandem time and space switching in their switching networks. Examples are the AT&T No. 4 ESS [BSTJ 1977] and the NTT (Japanese) DTS-11 [TAWA].

Consider in particular a time-slot interchanger followed by a space switch, followed in turn by an output TSI. The combination is designated a T-S-T switching network. The operation of this composite switch is completely analogous to the three-stage space switch of Fig. 10-21. (That switching network may be labeled an S-S-S switch.) To see this, consider the problem of switching $N$ channels (time slots). Instead of doing this with one very fast TSI, group these $N$ channels into $N/n$ TSIs, each containing $n$ time slots (channels) at its input. Let the number of output channels in a frame time be $k > n$. (Note that we are using the same notation as in the all-space-division case to preserve the analogy.) A typical frame at the input and output of a TSI appears in Fig. 10-25. Connect

these $N/n$ TSIs to a single $\frac{N}{n} \times \frac{N}{n}$ space switch as shown in Fig. 10–26. The $N/n$ outputs of the space switch are in turn each connected to a time slot interchanger in a third T stage, with $k$ time slots/frame at its input and $n$ slots at its output.

The crosspoint settings of the space switch are changed each of the $k$ time intervals corresponding to the $k$ time slots of the TSIs. The resultant space switch is termed a time-multiplexed switch. A simple analysis now indicates that if $k = 2n - 1$, one again has a nonblocking switching network, just as in the all-space network of Fig. 10–21. The T-S-T switch of Fig. 10–26 is thus completely analogous to the S-S-S switch of Fig. 10–21, as already noted. To demonstrate the nonblocking relationship, let $(n - 1)$ of the $n$ input time slots in one

![Figure 10–26](image-url)
of the input TSIs of Fig. 10–26 be busy. They in turn use \((n - 1)\) of the \(k\) different connections in time of the space switch. Let a call arrive on the remaining time slot of the same input TSI and let it be destined for one of the \(N/n\) output TSIs. Say that \((n - 1)\) output time slots of this output TSI are already occupied, or busy with calls. They also correspond to \((n - 1)\) of the \(k\) different connections in time of the space switch. In a worst-case analysis let the two sets of \((n - 1)\) space-switch crosspoint connections (those corresponding to the input TSI and those corresponding to the output TSI) be disjoint. These conditions are displayed in Fig. 10–27. In part (a) of the figure the \(k\)-slot time frame at the output of the stage-1 TSI, with \((n - 1)\) slots occupied (busy), is sketched schematically. Part (b) of Fig. 10–27 shows the \(k\)-slot time frame at the input of the stage-3 TSI, with its time axis aligned with that of the input (stage 1) TSI. \((n - 1)\) time slots are also assumed to be busy with calls, but are shown occupying a different or disjoint set than those of the input TSI. For a connection to be made between the two TSIs there must be at least one remaining free time slot in each of the two TSIs occurring at the same time. One such slot is shown appearing at the end of the frame. Since the space switch connecting the TSIs is square, any two TSIs can be connected through it. The two time slots must occur at the same time, however, as shown in Fig. 10–27. Under the conditions of Fig. 10–27, the new call can be connected, using the free slot occurring at the same time — the square switch allows the two TSIs to be connected. As previously then, the

---

**Figure 10–27** Nonblocking condition, time slot interchange, T-S-T switch  
a. Input TSI, stage 1  
b. Output TSI, stage 3
condition for this to happen, i.e., the call not to be blocked, is
\[ k = (n - 1) + (n - 1) + 1 = 2n - 1 \]
time slots, ensuring a nonblocking T-S-T switch.

The \( \frac{N}{n} \times \frac{N}{n} \) space arrays of Fig. 10-21 have thus been replaced by one
\( \frac{N}{n} \times \frac{N}{n} \) space array whose connections are independently changed or switched
\( k \) times during a frame.

Consider some examples. First let \( N = 960 \) input channels be connected to
960 output channels. For a single-stage space switch this would require on the
order of \( 10^6 \) crosspoints. In the T-S-T case, let \( n = 120 \) input time slots be used
in the input TSIs, \( N/n = 8 \) input TSIs are thus required, with the same number
at the output. For nonblocking operation \( k = 2n - 1 = 239 \) different connec-
tions of the space switch are required. (This might pose some problems because
of the arbitrariness of the number! It might be better to use \( k = 2n = 240 \) if
possible.) Since \( N/n = 8 \), an \( 8 \times 8 \) space switch is called for. This is obviously a
tremendous reduction in size over the single-stage \( 960 \times 960 \) space switch. If a
three-stage space switch were used, the number of crosspoints required would
be \( \mathcal{C} \simeq 168,000 \) crosspoints.

Consider another, much larger, example. Let \( N = 96,000 \) channels or
trunks. Again let \( n = 120 \) time slots. Then \( k = 239 \) again to ensure nonblocking
behavior. An \( 800 \times 800 \) space switch is now needed, switching at the 239-slot
rate. This T-S-T switch, combining 800 input and 800 output TSIs plus the
800 \( \times \) 800 time-switched space switch, is completely comparable to a three-
stage all-space (S-S-S) switch that requires on the order of \( 1.7 \times 10^8 \) crosspoints.

Further reduction in switch complexity and cost is possible using an "essentially
nonblocking" configuration. Many larger switching systems use this ap-
proach. The AT&T No. 4 ESS, a six-stage switch of the type T-S-S-S-S-S-S, is one
example. In the next subsection we discuss multistage switches of the blocking
type.

10-4-2 Blocking Probability Analysis of
Multistage Switches: Lee Approximation

The determination of blocking probability in a multistage switch is inher-
tently complex. Given a specific switching network, with the number of inputs
and outputs as well as the probability of line (trunk) occupancy specified, the
problem is to calculate the probability of not finding a free path (route) through
the switch between a given input-output pair. This task is quite difficult: There
are many possible paths to consider in a typical large switch, leading to combina-
torial problems. More significantly, however, dependencies between blocking probabilities on different links along the path make the problem almost intractable. Approximations are thus used commonly for the calculations.

The simplest approximation, due to C. Y. Lee [LEE], assumes that the probabilities of finding individual links along a path busy are independent. If these probabilities are known, it then becomes a straightforward although possibly complex task to calculate the overall blocking probability. The problem is similar to that encountered in network reliability or survivability studies where, given the probability of failure of a link (equivalent to busy here), one calculates the overall reliability of a network [WILK]. Another example is that of calculating the end-to-end blocking probability in a circuit-switched network if the individual-link blocking probabilities are known. (See, for example, [LINP].) C. Y. Lee’s approach in calculating the switch blocking probability is to use probability graphs in carrying out the calculations.

Consider a three-stage network as an example. As noted in our discussion of T-S-T nonblocking switch networks, there is a complete analogy between the T-S-T case and the all-space-division S-S-S network. The analogy carries over to the blocking switch case as well [COLL, pp. 22–39]. We thus need only consider one of the networks, either T-S-T or S-S-S. With appropriate identification of parameters in the final expression for blocking probability, one can use the same expression for both networks. The same analogy extends to other multistage configurations as well [COLL]. We focus here on the S-S-S case for simplicity. Figure 10–28 shows a portion of a three-stage space switch, indicating two typical channels that are to be connected together, one at the input and the other at the output. For a blocking switch we must, of course, have $k < 2n - 1$. In addition, let $k > n$. This ensures that the input and output arrays, or the TSIs in the T-S-T case, are inherently nonblocking. This would generally be the case for a tandem switch used to route calls from one set of incoming trunks to another set of outgoing trunks.

The case of $k < n$, or concentration, is precisely that discussed in the traffic analysis of the previous section. The input arrays themselves introduce blocking in this case. The overall blocking probability of the concentration of the input array and the arrays of the second (middle) stage becomes much more complicated to calculate. If the utilization per input channel is low, the blocking pro-


bility of the input array can be neglected, and the resultant system modeled the way we are proceeding. This procedure thus provides an approximate model for an end-office (local) switch or a PBX, with local users concentrated and switched over a relatively small number of trunks. Bellamy has used this model, including the assumption of low utilization per input, to provide some comparative results for end-office switches or PBXs with concentration [BELL, Table 5.2]. A more accurate approach for this case of concentration would follow the model of Nesenbergs and Linfield [NESE].

Returning now to the case of \( k > n \) in Fig. 10–28, let the probability that a typical input channel is busy be \( a \), just the binomial parameter \( \lambda/(\lambda + \mu) \) introduced in the last section (see Eq. (10–47)) in discussing the probabilities of state of the structures of Figs. 10–13 and 10–15 for the case \( M = N \). In saying that the parameter \( a \) is the same for each input channel, we are implicitly assuming an homogeneous system. This means then that the probability that a typical output channel in Fig. 10–28 is busy is \( a \) as well. We further assume that the incoming traffic is distributed uniformly over the \( k \) interstage links. The probability that an interstage link is busy is thus \( p = an/k \).

Note that in the T-S-T case, the \( k \) interstage links correspond to \( k \) output slot times. It is apparent that in the homogeneous traffic case the traffic carried by the \( n \) input slots may also be taken as distributed uniformly over the \( k \) output slots, providing the same probability \( p \) that a TSI output slot (comparable to an interstage link in Fig. 10–28) is busy (occupied).

**Figure 10–28** Calculation of blocking probability, three-stage switch
A complete path from input to output through one of the \( k \) nonblocking middle arrays in the S-S-S case of Fig. 10–28 traverses two links, as shown. Alternatively, in the T-S-T case, a connection from input TSI to output TSI requires matching time slots at the input and output of the middle space array. With the assumption of independent probabilities on each of the links in the switch network, the probability of blocking is the probability that no free path from input channel to output channel is available. (In the T-S-T case this corresponds to the probability that two time slots, one at an input TSI and the other at an output TSI, do not match up.) It is apparent that this probability is given by

\[
P_B = (1 - (1 - p)^2)^k \quad p = an/k \tag{10–54}
\]

Note that Eq. (10–54) does not equal zero for \( k = 2n - 1 \), the Clos nonblocking requirement, indicating the effect of the approximation.

As an example of the application of Eq. (10–54), let \( n = 120 \) and \( k = 128 \). These numbers appear in the No. 4 ESS system, although modified somewhat (see [BSTJ 1977] and our later discussion). Let \( a = 0.7 \) be the inlet channel utilization. Then the blocking probability for this case is

\[
P_B = (0.882)^{128} \approx 10^{-7}
\]

For \( a = 0.9 \) the blocking probability rises to 0.042, a considerable change. This indicates the sensitivity of the result to input utilization.

What does one gain by going to this blocking configuration? In the S-S-S case it reduces the number of crosspoints needed, as noted earlier. By going to the T-S-T configuration one trades additional crosspoints for switching in time. As an example, consider the case of \( N = 96,000 \) total input lines used earlier. The optimum nonblocking three-stage space switch would require \( 1.7 \times 10^8 \) crosspoints. The corresponding value of \( n \), the number of input channels per input array, is then \( n = 219 \), clearly an impractical number. The choice of \( n = 200 \) with \( k = 399 \) would be much better, except that the optimum number of crosspoints is still much too large. For \( n = 120 \) and \( k = 128 \), the example just worked out, one gets \( C = 1.06 \times 10^8 \), a reduction by almost a factor of two but clearly still too high. Now consider the T-S-T version. With \( n = 120 \) and \( k = 128 \), the center space switch is of size \( 800 \times 800 \), a clear reduction in crosspoints, at the added cost of requiring 800 TSIs at both the input and the output, with both the TSIs and the space switch operating at \( k = 128 \) slots or switch changes/sec. Further reduction in the effective number of crosspoints or, equivalently, the ability to operate at higher utilization/input channel, may be obtained by going to multiple space stages. As already noted, the No. 4 ESS switch is of the T-S-S-S-S-T type [BSTJ 1977].

The blocking probability of these more complex multistage switches may be calculated approximately by using the Lee graph approach. In particular, the analysis of the T-S-S-S-T switch using this approach is quite straightforward and is an extension of the three-stage analysis that leads to Eq. (10–54).
Consider the T-S-S-S-T switch of Fig. 10–29 as an example [COLL, p. 36]. Note that the notation used in our previous T-S-T discussion has been changed to accommodate this larger system. (We adopt here the notation of Collins and Pedersen. See [COLL, p. 36].) Thus $c$ represents the number of input time slots or channels in the stage-1 TSIs, with $\ell$ the corresponding number of output time slots. Alternatively, each input port handles $c$ trunks (channels). There are a total of $N$ TSIs in the first, as well as in the last, stages of the switch; the total number of trunks at input or output is thus $J = Nc$.

It is left to the reader to show, using the Lcc method of analysis, that the approximate expression for blocking probability is given by

$$P_b = (1 - q_1^2 [1 - (1 - q_2^b)]^\ell)^c \tag{10-55}$$

where $q_1 = 1 - a/\alpha$, $q_2 = 1 - a/\alpha\beta$, and $\alpha = \ell/c$ is the time stage expansion ratio (comparable to $k/n$ in the T-S-T case), $\beta = k/n$ is the space stage expansion ratio, and $a$ is again the probability that an input channel is busy.

Collins and Pedersen have carried out a number of typical design examples for this switch architecture as well as some others [COLL, Chap. 3]. In particular, consider a large switch example where the switch has a total of $J = Nc = 116,736$ input channels or trunks [COLL, design (3), p. 50]. $N = 256$ ports are available, with $c = 456$ input time slots (channels)/port, and $\ell = 512$ time slots at the output of each TSI. The first-stage space switches are constrained to be $16 \times 16$ switches (i.e., $n = k = 16$). It is then left to the reader to show that
C = 12,288 crosspoints are required. For an input channel (time-slot) loading of \( a = 0.9 \), the (approximate) blocking probability of the switch is readily calculated to be \( P_B = 6.4 \times 10^{-5} \). Since the time slots are each 8 bits long, recurring every 125 \( \mu \)sec, the TSI access time is 122 nsec. The space switch, which changes its configuration every slot interval, must switch within less than one bit interval, or within 30.5 nsec. It is also left as a problem to the reader to compare this design with one that uses the T-S-T configuration.