#1

NEW APPROACHES TO THE ANALYSIS OF CONNECTING AND SORTING NETWORKS

MICHAEL J. MARCUS

LOAN COPY

TECHNICAL REPORT 486

MARCH 17, 1972

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
RESEARCH LABORATORY OF ELECTRONICS
CAMBRIDGE, MASSACHUSETTS 02139

The Research Laboratory of Electronics is an interdepartmental laboratory in which faculty members and graduate students from numerous academic departments conduct research.

The research reported in this document was made possible in part by support extended the Massachusetts Institute of Technology, Research Laboratory of Electronics, by the JOINT SERVICES ELECTRONICS PROGRAMS (U. S. Army, U. S. Navy, and U. S. Air Force) under Contract No. DAAB07-71-C-0300.

Requestors having DOD contracts or grants should apply for copies of technical reports to the Defense Documentation Center, Cameron Station, Alexandria, Virginia 22314; all others should apply to the Clearinghouse for Federal Scientific and Technical Information, Sills Building, 5285 Port Royal Road, Springfield, Virginia 22151.

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

RESEARCH LABORATORY OF ELECTRONICS

Technical Report 486 March 17, 1972

NEW APPROACHES TO THE ANALYSIS OF CONNECTING AND SORTING NETWORKS

Michael J. Marcus

Submitted to the Department of Electrical Engineering at the Massachusetts Institute of Technology, October 22, 1971, in partial fulfillment of the requirements for the degree of Doctor of Science.

(Manuscript received December 15, 1971)

ABSTRACT

This report is a study of the application of information theory techniques to the field of connecting networks. It shows that these techniques are useful in this new area, and demonstrates that two types of networks can be constructed with a complexity that is proportional to an informational lower bound which is due to Claude E. Shannon.

The concepts of information and entropy are extended to fit the context of connecting networks. These not only give insight into the problems of these networks, but are used directly to show that the basic sort-merge algorithm does not of itself imply that sorting networks are inefficient.

It is then shown that connecting networks with small blocking probability and time-slot interchangers can be implemented with a complexity that is proportional to the informational minimum. By defining an ensemble of connecting networks, analogous to ensembles used in coding theory, it is shown that connecting networks with n inputs and blocking probability ϵ can be constructed with $0(n \log n) + 0(n \log 1/\epsilon)$ contacts for all n and ϵ. Shannon's results show that the first term in this expression is the minimum possible order of growth for such networks. A similar result for time-slot interchangers is illustrated by giving an explicit construction.

TABLE OF CONTENTS

I. INTRODUCTION 1

1.1 Background of This Research 2

1.1.1 Network Complexity 2

1.1.2 Information Theory and Connecting Networks 2

1.1.3 Previous Results in Connecting Network 3

1.1.4 Sorting Networks 4

1.2 Summary of Report 4

II. NETWORKS AND INFORMATION 6

2.1 Networks and Their Tasks 6

2.2 Applications to Sorting Networks 8

III. SORTING NETWORKS 10

3.1 Odd-Even Sorting Network 10

3.1.1 Joint Entropy of a Column of Comparators 13

3.1.2 Conditional Entropy of Comparators 15

3.1.3 Entropy Accounting for the Case n = 4 16

3.1.4 Limiting Behavior 18

3.2 Bitonic Sorting Network 18

3.3 Summary 21

IV. CONNECTING NETWORKS WITH FINITE BLOCKING PROBABILITY 22

4.1 Background 22

4.2 Introduction to Ensembles of Connecting Networks 25

4.3 Analysis of an Ensemble of Homogeneously Structured Connecting Networks 28

4.4 Bounds for $\overline{P}_j$ 31

4.5 An Ensemble of Nonhomogeneously Structured Connecting Networks 38

4.6 Summary 43

V. TIME-DIVISION SWITCHING NETWORK 44

5.1 Introduction to Time-Division Switching 44

5.2 Efficient Implementation of Time-Slot Interchangers 45

5.3 Another Type of Time-Slot Interchanger 49

VI. CONCLUSION 51

6.1 Summary of Results 51

6.2 Problems for Future Work 51

Acknowledgment 52

References 53

I. INTRODUCTION

Information theory has provided many useful models for the transmission channels that are used in communications systems, yet there has never been an analogous theory for the switching systems that are used to connect transmission channels. Although such switching systems have been used almost from the beginning of telephony, they are not well understood theoretically. This report is the result of an effort to use the techniques of information theory to study connecting networks, a basic component of switching systems. In particular, it is shown that this approach can be used to determine the minimum amount of complexity for implementing certain types of connecting networks.

For most practical applications it is not possible to interconnect directly all pairs of locations between which communication is necessary. Switching systems are used to control the flow of information through a communication system. Although the distinctions between them are sometimes unclear, these switching systems fall into two classes. (i) Message-switching systems store information from the source at various intermediate points and send it over an available channel at a later time. Thus a complete path from input to output does not have to be available at one time in order to transmit the message. But this does imply that the message will usually be delayed en route by the switching system itself. A store and forward telegraph system is a good example. (ii) Line-switched systems always create a direct path between the source and the destination. The delay is solely a function of the transmission channels that are used. A telephone system is a good example.

This report is concerned only with line-switched systems. In particular, it deals with connecting networks; that is, with networks that actually create the path from a set of inputs to a set of outputs. The telephone system uses thousands of such networks, the most visible of which are in local telephone exchanges.

The main theme of this report is the application of information theory techniques to the field of connecting networks. Using these techniques, I shall discuss the minimum possible complexity for implementing four different types of networks: connecting networks with small blocking probability; nonblocking connecting networks; sorting networks (special connecting networks with decentralized control); and time-slot interchangers (a network used in switching time-multiplexed signals).

An important conclusion to be drawn is that the concepts of information, entropy, and ensembles of systems can be used to yield useful results about connecting networks. Among these results is an informational bound giving the minimum possible complexity for these types of networks.

It is then shown that for a connecting network with small blocking probability and for time-slot interchangers networks exist whose growth in complexity as the number of inputs is increased is proportional to the informational minimum. These results are obtained by using an ensemble of systems in the case of connecting networks with finite blocking and by giving an explicit construction in the case of time-slot interchangers.

1.1 BACKGROUND OF THIS RESEARCH

1.1.1 Network Complexity

The complexity of connecting networks can be viewed in three parts. First, a network is composed of contacts that form electrical paths in the network. Second, the network has associated with it a finite number of states each of which corresponds to a different set of contact closings. A useful measure of this kind of complexity is the logarithm to the base 2 of the number of states which corresponds, for standard relay networks, to the number of relay coils in the network. Third, a certain amount of computational complexity is required to calculate for each network the desired network state, given the desired permutation that the network will implement. This is the hardest kind of complexity to quantify because the choice of units is arbitrary. In giving examples of networks the unit that is used to measure computational complexity will be discussed for each case. In general, only orders of growth can be obtained and even these may vary if different units are used.

It appears that these three types of complexity are closely interrelated, in that a network designer can make tradeoffs between them to a limited degree. Little is known, however, about the fundamental limits imposed by this interrelationship.

1.1.2 Information Theory and Connecting Networks

In the past, several attempts have been made to relate information theory to connecting networks. In 1950, Shannon[1] showed that the minimum number of states which a connecting network must have could be considered an information theory problem. He showed that since a network needed n! states to perform all permutations, at least $\log_2 n!$ relay coils were needed for such a network. The Stirling bound[2] gives $n \log \frac{n}{e} + \frac{1}{2} \log 2\pi n < \log n! < n \log \frac{n}{e} + \frac{1}{2} \log 2\pi n + \frac{1}{8n}$, and this is approximately equal to requiring $n \log n$ coils. Shannon's result gives no bounds for the number of contacts required for such a network or for the computational complexity involved.

In 1953, Baker[3] discussed the relationship between Shannon's original work and connecting networks, but he gave no significant results. Syski,[4] in 1955, published an interesting study of the analogy between information theory and telephone traffic theory. In particular, he compared the type of stochastic processes involved in certain types of telephone systems with the type of processes which Shannon viewed as information sources. Unfortunately, there was no follow-up of this work.

More recently, Elias[5] calculated the average number of bits necessary to encode a permutation based on three representations: direct encoding, results of a sorting operation, and the state of a rearrangeable network which performs the permutation.

1.1.3 Previous Results for Connecting Networks

A straightforward implementation of a connecting network that will perform all permutations is a square array of n^2 contacts, where n is the number of inputs and outputs. Such a network is called an $n \times n$ crossbar switch, or square switch. It has n^2 coils, even though it has only $\log n! \sim n \log n$ meaningful states. The difference between the number of coils and the logarithm of the number of states arises because each contact has its own coil in such a network. In order to set up such a network for a permutation of all of its inputs, n steps of computation are needed. Each of these corresponds to setting a contact determined directly from the permutation. Each of these steps involves a one-out-of-n selection, so the computation can be viewed as being $O(n \log n)$.

Benes[6] has defined a class of networks called rearrangeable networks. These can perform all permutations with the restriction that changing part of the permutation may result in temporarily interrupting other connections in the network. Thus these networks are not practical for telephony. It can be shown,[6-10a] for values of n that are integral powers of two, that such networks require $n \log n - n + 1$ devices called β elements, each of which has one coil and four contacts. The number of coils used then is only slightly more than the informational minimum given by Shannon.

The computational complexity associated with these rearrangeable networks has not been studied much. Paull[11] has given a bound for the maximum number of paths which must be interrupted in one type of rearrangeable network to change an existing path. Nakamura[12] and Bassalygo et al.[13] have expanded this somewhat to show that fewer paths must be disturbed if the network has extra contacts.

Several algorithms[14] have been given for determining the state of the coils in a rearrangeable network, given the desired permutation. It appears that the complexity of these algorithms is $O(n(\log n)^2)$, although there has been no proof that this is the minimal order of growth.

Clos[15] has shown that it is possible to construct networks that are strictly nonblocking, like square networks, but have fewer than n^2 contacts. Clos gave an algorithm for constructing such networks, but he did not give a closed-form expression for the minimum number of contacts required as a function of n. Keister[10b] has recently found empirically that the minimal Clos networks for practical values of n have somewhat more than $n(\log n)^2$ contacts.

Cantor[16] has used a somewhat different approach to show that strictly nonblocking networks can be constructed with fewer than $8n(\log n)^2$ contacts. While these constructions may not be practical as compared with the Clos networks, their significance is that through them an explicit bound for their complexity is given which holds for all values of n.

No algorithms have been reported to set up nonsquare strictly nonblocking networks. It seems likely, however, that such algorithms would be simpler than the algorithms required for rearrangeable networks, since no paths must be disturbed and there are many possible routes for each path.

1.1.4 Sorting Networks

Sorting networks have received much attention in recent publications[17-21] and present many unsolved problems. Like connecting networks, sorting networks permute their inputs in connecting them to the outputs. Sorting networks, by definition, have no external control. They are built with binary comparators that can sense the signal associated with each line and order a pair of lines so that the signal on the first is always less than the signal on the second. The most convenient way of measuring the complexity of a sorting network is by the number of comparators it contains. Each comparator has two final states: input $1 \geq$ input 2, and input $1 <$ input 2. To permute the inputs, 4 contacts are required. Since each comparator makes one and only one binary decision, the computational complexity can be viewed as being directly proportional to the number of comparators in the network.

The best-known general algorithm for constructing sorting networks comes from Batcher[17] and leads to networks with $0(n(\log n)^2)$ contacts. It has recently been shown,[19] however, that this algorithm is not optimal for $n = 9, \ldots 16$ so it is not certain that Batcher's algorithm is best for arbitrary n greater than 16. Indeed, just as in the case of strictly nonblocking networks, it has never been shown that sorting networks cannot be constructed with $0(n \log n)$ comparators.

1.2 SUMMARY

In Section II some of the basic concepts of information theory are extended to connecting networks. The combinatorial information of a network, as well as the entropy associated with a set of network tasks, is defined. An example of the usefulness of these concepts is given by using them to prove that the basic sort-merge algorithm does not itself imply that sorting networks which use sort-merge are informationally inefficient.

Sorting networks are studied in detail in Section III. Two algorithms from Batcher[17] are examined closely, and it is found that an informational approach helps to identify the inefficiencies in these networks.

Section IV then discusses connecting networks with finite blocking probability. The main question here is how many contacts are needed to implement a network, given the number of inputs and the desired blocking probability. The most interesting result shows that for any number of inputs n and blocking probability P_f there exists at least one network with $0(n \log n) + 0(n \log 1/P_f)$ contacts which has the desired characteristics. Since at least $0(n \log n)$ contacts are required on informational grounds, this shows that the requirement of having an arbitrarily small blocking probability does not increase the order of growth of the number of crosspoints. This result is proved by defining an ensemble of systems and bounding the average blocking probability.

In Section V time-slot interchangers are studied. As in the case of connecting networks it is shown that these can be constructed with a complexity proportional to the informational minimum. An explicit construction is given for such networks.

II. NETWORKS AND INFORMATION

We shall now extend some of the basic concepts of information theory to the field of connecting networks in order to gain insight into this class of problems and to get useful results, which will then be explored in the rest of the report.

2.1 NETWORKS AND THEIR TASKS

We shall develop models of information and entropy for connecting networks and the set of tasks which they are given to perform. These networks basically permute the signals present on their input lines to produce the signals on their output lines. Many practical networks, however, especially those used in telephony, can perform an incomplete permutation, that is, only a subset of the inputs may be connected to the outputs.

A given network can implement all of the functions in the subset T_N, $T_N \subseteq T$. If a network is capable of performing the set T_N it must have exactly T_N states that can be distinguished externally. We shall call these the <u>external</u> <u>states</u>.

<u>DEFINITION</u> 1. The external combinatorial information of a network N is $\log_2 |T_N| = I_c(T_N)$. We call T_N the task set.

This definition is related to Kolmogorov's definition of combinatorial entropy,[22] and is the amount of information needed to describe the state of the network. No assumptions about probability distributions have been made explicitly thus far, but the combinatorial approach definition gives the same result as the usual probabilistic definition when the probabilities of all possible network states are equal.

In general there may be a nonuniform probability distribution associated with the set T_N which is called $P_N(t_i)$, $t_i \in T_N$

$$\sum_{t_i \in T_N} P_N(t_i) = 1. \tag{1}$$

The internal states of a network are the distinct states of the components of the network. In many cases there are distinct internal states that are not distinguishable externally. This is analogous to a similar situation with equivalent states which occurs in the decomposition of finite-state machines.

<u>DEFINITION</u> 2. The internal states of a network N form a set S_N. There is a mapping σ_T from the set S_N onto the set T_N.

If a network is to be implemented in the form of several interconnected subnetworks, the task set of the whole network T_N and the task sets of the subnetworks T_i, i = 1, k must satisfy the relationship

$$|T_N| \leqslant \prod_{i=1}^{k} |T_1|. \tag{2}$$

Equivalently,

$$\log |T_N| \leq \sum_{i=1}^{k} \log |T_i| \tag{3}$$

or

$$I_c(T_N) \leq \sum_{i=1}^{k} I_c(T_i). \tag{4}$$

Thus the information of the subnetworks must be at least that of the whole network. The equality holds if and only if there is a one-to-one correspondence between elements of T_N and k products of states of the subnetworks.

We can also make informational statements about networks when the task probabilities are not equal. Thus

DEFINITION 3. The task entropy of a network N is

$$H_t(N) = \sum_{t \in T} \Pr(t) \log [\Pr(t)]. \tag{5}$$

The task entropy of subnetworks can be defined in an identical way. We can also condition the task probabilities of the subnetworks so that we get the following definition.

DEFINITION 4. The conditional entropy of subnetwork i gives the states of subnetworks 1, 2, ..., k

$$H_t(N_i | N, \ldots N_k) = \sum_{j=1}^{n_i} P\left(t_j^i | N_1 \ldots N_k\right) \log P\left(t_j^i | N_1 \ldots N_k\right). \tag{6}$$

In the same manner that we could decompose network combinatorial information, we can also decompose network task entropy to give[7]

$$H_t(N) \leq \sum_{i=1}^{k} H_t(N_i). \tag{7}$$

The equality holds if and only if each of the subnetworks takes on all states with probabilities that are statistically independent of the states of the other subnetworks. We use a product decomposition of the joint distribution into a product of conditional probabilities, take the logarithms, and average[7] to get

$$H_t(N) = \sum_{i=1}^{k} H_t(N_i | N_1, N_2 \ldots N_{i-1}). \tag{8}$$

In many cases networks have more states and thus more devices than would be dictated by first-order, or unconditioned, informational bounds of Shannon.[1] This informational approach helps us identify those parts of such networks in which informational inefficiencies exist and helps find the reasons for such inefficiencies.

2.2 APPLICATIONS TO SORTING NETWORKS

As mentioned previously, little is known about the minimum number of binary comparators required to implement a sorting network. The most efficient known sorting networks are based on the concept of recursive merging. As illustrated in Fig. 1 this involves dividing the inputs into two groups of n/2 inputs which are then sorted with two n/2 input sorting networks; then these two-ordered lists are merged to form an ordered

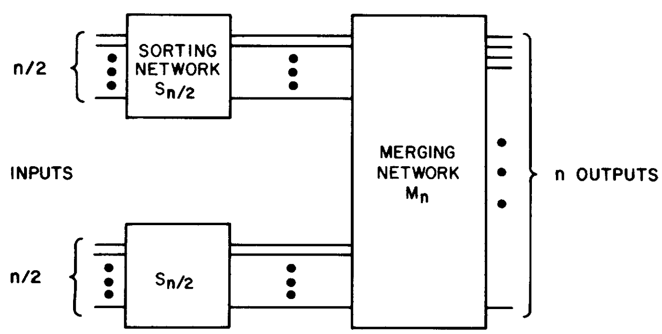


Fig. 1. Sort-merge algorithm for sorting networks.

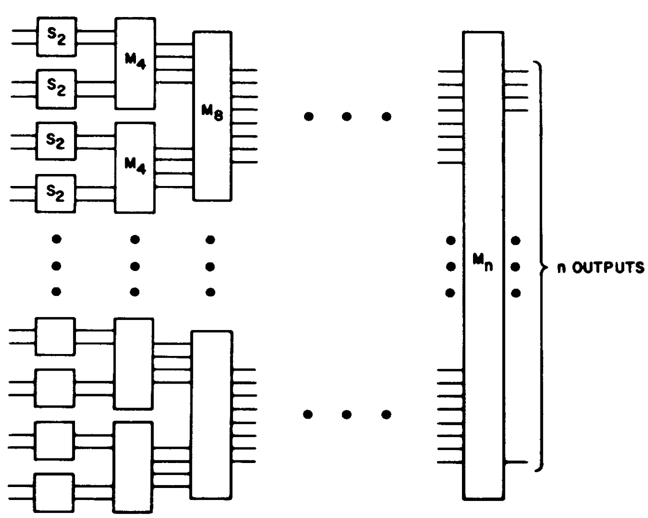


Fig. 2. n-Input sorting network using the sort-merge algorithm.

list of all n inputs. This algorithm is then applied to the smaller sorting networks recursively until the only remaining sorting networks are two input sorting networks which are simply binary comparators. The resulting network has the general form shown in Fig. 2.

Theorem 1

The sum of the combinatorial information of the components of a sorting network of size n using sorting by recursively merging is just $\log n!$

Lemma

The combinatorial information of a merging network with two sets of n/2 inputs and n outputs is $\log_2 \binom{n}{n/2}$.

Label the outputs according to which input list they came from. There are $\binom{n}{n/2}$ such labelings, each of which corresponds to a distinct state of the merging network. The information of the network is then $\log \binom{n}{n/2} = I(M_n)$.

Now

$$
\begin{aligned}
I(N) &= 2I(S_{n/2}) + I(M_n) \\
&= 2 \log n/2! + \log \binom{n}{n/2} \\
&= \log \frac{(n/2)!^2\, n!}{(n/2)!^2} \\
&= \log n! \qquad (9)
\end{aligned}
$$

Q. E. D.

Corollary 1. The component networks are independent.

Corollary 2. If the global network states are equiprobable, the states of each sub-network are also equiprobable (by identity of combinatorial and probabilistic information in the equiprobable case).

III. SORTING NETWORKS

We shall deal with the analysis of sorting networks, an area in which much remains to be learned. The concepts of information which we have derived will be applied to this type of network and will be helpful in explaining the inefficiencies in known algorithms. Two types of networks will be studied in detail.

A sorting network is a loop-free network composed of two input comparators. One output is the minimum of the input signals, the other is the maximum. A sorting network permutes the order of the signals presented on its inputs so that the signals on its outputs are in increasing order. Thus a sorting network is a special case of a connecting network in which the control of the network is very decentralized.

Software sorting algorithms are well known.[23] These are computer programs which order a list of numbers in the computer's memory. There are several algorithms which on the average order n numbers in approximately n log n comparisons. It is obvious that this is the minimal order of growth. The best known sorting networks, however, require $0(n(\log n)^2)$ comparators. The reason for this difference is that software sorting algorithms can be adaptive; that is, the numbers that they compare can be a complicated function of the results of previous comparisons. By definition, sorting networks must always make a fixed set of comparisons, enough for the worst case.

The literature of sorting networks has been comprehensively surveyed by Knuth.[19] The earliest reference to this type of problem was a patent by O'Connor and Nelson,[24] in 1957. Nelson and Bose[18] later showed that networks could be constructed with $0\left(n^{\log_2 3}\right)$ comparators. In 1968, Batcher[17] showed the two networks that are described in detail in sections 3.1 and 3.2. These both require $0(n(\log n)^2)$ comparators and are the best known general constructions. Knuth, on the other hand, reports that networks very different from those of Batcher have a few less comparators for n = 9, 10, ..., 16. Indeed, the apparent minimal networks for n = 9, 10, 12, 16 bear little apparent relationship to each other. Thus the general question of establishing the minimal order of growth remains in doubt.

3.1 ODD-EVEN SORTING NETWORK

The first of Batcher's two networks which will be discussed is usually called the odd-even network, and is the best known general algorithm for sorting networks. The algorithm defines a merging network recursively in terms of two smaller merging networks and some comparators. The merging networks so defined are then used to construct a sorting network.

The recursive definition of this network is shown in Fig. 3. Inputs and outputs are labeled with subscript letters. The inputs with odd subscripts $(a_1, a_3, \ldots a_{n-1}, b_1, b_3, \ldots b_{n-1})$ are directed to the top half of the network, $M^o_{n/2}$, while the remaining inputs $(a_2, a_4, \ldots, a_n, b_1, b_2, \ldots, b_n)$ are directed to $M^e_{n/2}$. By using this algorithm, a network for merging two lists of n elements can be implemented with $n(\log 2n-1) + 1$

comparators rather than the informational limit of $\log_2 \binom{2n}{n} \sim 2n$. Applying the second step of the recursion, as in Fig. 4, results in a sorting network requiring a total of $\frac{n}{4}(\log^2 n - \log n + 4) - 1$ comparators.

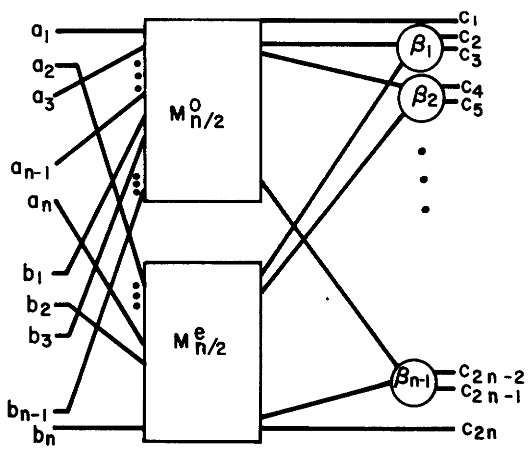


Fig. 3.
Batcher odd-even merging network.

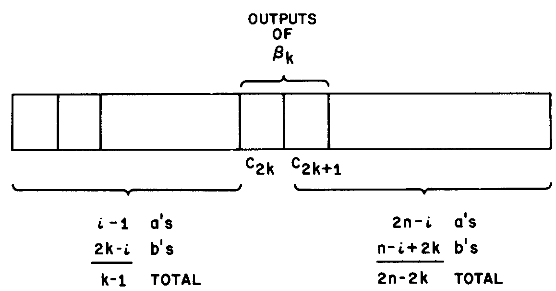


Fig. 4.
Schematic representation of the outputs of an odd-even merging network.

Several approaches can be used to study the cause of the inefficiency in the basic merging network. First, it is possible to calculate the state probabilities of the individual comparators to see if any of the comparators really had one bit of entropy. In order to do this it is assumed that all the $\binom{2n}{n}$ possible relative orderings of the two input lists were equally probable. The key to calculating the state probabilities of the comparators is to observe that the k^{th} comparator β_k is in the zero state if and only if output 2k of the merging network is from an input with an odd subscript.

Theorem 2

The first-order probability that β_k is in the zero state in a Batcher odd-even merging network with a total of 2n inputs, under the assumption only that the relative order of the inputs is random, is given by

$$P_k^n = \frac{\displaystyle\sum_{\substack{i=\max(1,\,2k-n)\\ i\ \text{odd}}}^{\min(2kn)} 2\binom{2k-1}{i-1}\binom{2n-2k}{n-i}}{\binom{2n}{n}}. \tag{10}$$

Proof: Figure 4 is a schematic representation of the 2n outputs of the merging network. Consider any comparator, β_k, which has outputs 2i and 2i+1. Assume that β_k is in the zero state and that output 2k is a_i, where i is odd. There are 2k-1 outputs before output 2k and 2n−2k outputs after it. Since output 2k is a_i and there is a total of n a's which must be distributed over the outputs, there must be i−1 a's before position 2k and n−i a's after this position. These a's may be arranged in any sequence with b's within these constraints. Therefore, there are $\binom{2k-1}{i-1}\binom{2n-2k}{n-i}$ ways of arranging these symbols, given values for i and k. Since the case where output 2k is b_i is symmetrical with the case above where it is a_i, we can multiply this expression by a factor of 2 and get the total number of possible orderings as $2\binom{2k-1}{i-1}\binom{2n-2k}{n-i}$.

The total number of ways in which the a's and b's can be arranged is $\binom{2n}{n}$. Therefore, the P_k^n can be given as

$$P_k^n = \sum_i \frac{2\binom{2k-1}{i-1}\binom{2n-2k}{n-i}}{\binom{2n}{n}}, \tag{11}$$

the sum being over all possible values of i.

There are 2n−2k positions to the right of position 2k in which n−i a's must be distributed. In order that there be room for all of these a's, the following inequality must be satisfied

$$\begin{aligned} n-i &\leq 2n-2k \\ -i &\leq n-2k \\ i &\leq 2k-n. \end{aligned} \tag{12}$$

Since i must be at least 1, these conditions give a lower limit in the summation of $i = \max(k, 2k-n)$.

Similarly, i can be no greater than 2k or n, so the upper limit for the summation is $i = \min(2k, n)$. Since it is assumed that i is odd, the summation is only over the odd values of i. Q. E. D.

A computer program was written to get numerical values for the P_k^n's. Table 1 shows the values for the case n = 8. Also shown is the entropy, $H\left(P_k^n\right)$ for each value. The general trend for the values is illustrated in Fig. 5. For large n, P is an even

Table 1. Numerical values for the P_k^n for n = 8.

k	P_k^8	H_k^8
1	.533	.996
2	.492	.999
3	.503	1.000
4	.497	1.000
5	.503	1.000
6	.492	.999
7	.533	.996

function about i = n/2. P alternates about the line P = 1/2 with $|P-1/2|$ reaching a minimum at i = n/2. As n increases, the maximum value for $|P-1/2|$ is reached at i = 1, and i = n rapidly approaches zero. Because of this rapid convergence to P = 1/2, the

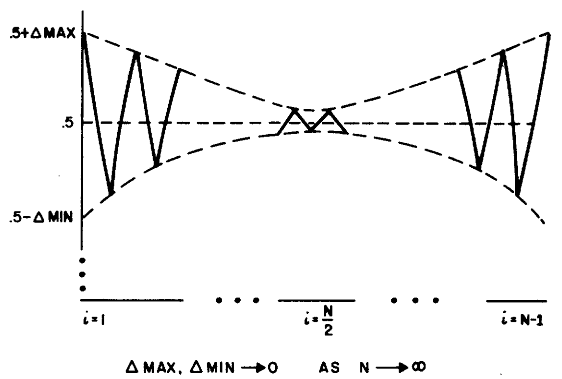


Fig. 5.
Variation of P_k^n.

entropy associated with each comparator also rapidly approaches 1 bit per comparison. Thus it can be said that the individual decisions made by the comparator are informationally efficient if all the inputs to the merging networks are equally likely and the comparators know only about their inputs.

3.1.1 Joint Entropy of a Column of Comparators

We have calculated the individual comparator entropy for an odd-even merging network. It is also possible to consider a column of comparators as a group and ask what the joint state probabilities are. Thus for the a, given n, there are n − 1 comparators in a column at the first stage of the recursion. These comparators have a total of 2^{n-1} states when taken as a group, all of which are reachable. As before, the total number of states the network has is $\binom{2n}{n}$. Since this is not divisible by 2^{n-1} for n > 2, it follows that the joint states of the comparators cannot be equally likely.

For the case n = 4 it can be shown that the number of ways in which each joint state of the comparators can be realized is listed as follows:

000 − 8	100 − 8
001 − 8	101 − 8
010 − 16	110 − 8
011 − 8	111 − 8

where the binary number is the state of the 3 comparators. The joint state probabilities can be calculated from these numbers by dividing them by $\binom{8}{4} = 70$. The entropy associated with this probability distribution is 2.936 bits, which is very close to the number of comparators involved.

These results can now be generalized. The most likely joint state is $0(10)^*$. For n = 4 this corresponds to 010, which can be realized with 2^n different states of the merging network. With this in mind we can state the following theorem.

Theorem 3

The joint entropy of a column of n comparators satisfies the bound

$$H_{jc} \geqslant n - \frac{\log n}{2} - .825. \tag{13}$$

Proof: The entropy is defined as

$$H_{jc} = \sum_j P_j \log P_j, \tag{14}$$

where the index j is over all 2^{n-1} joint states, and P_j is the probability of joint state j. Since the most probable state is $0(10)^*$, we can get the inequality

$$P_j \leqslant \frac{2^n}{\binom{2n}{n}} . \tag{15}$$

Also, since P_j is a probability distribution, it follows that

$$\sum_j P_j = 1. \tag{16}$$

Now we can get a lower bound for H_{jc}. Since the entropy function is convex $\cap$, a lower bound can be obtained for the total entropy by concentrating the nonzero terms of the probability distribution. In this case we have assumed that all of the nonzero terms have the maximum possible probability, $\frac{2^n}{\binom{2n}{n}}$, and all other terms are zero. There can be only $\frac{\binom{2n}{n}}{2^n}$ such terms if the distribution is to sum to one. Therefore, we can say

$$\begin{aligned} H_{jc} &\geqslant \frac{-\binom{2n}{n}}{2^n} P_{max} \log P_{max} \\ &= - \log \frac{2^n}{\binom{2n}{n}} \\ &= +\left[n - \log \binom{2n}{n} \right] \\ &= \log \binom{2n}{n} - n \\ &\geqslant 2n - \frac{\log 2n}{2} - .325 - n \end{aligned}$$

$$\geqslant n - \frac{\log 2n}{2} - .325$$

$$\geqslant n - \frac{\log n}{2} - .825 \tag{17}$$

Q.E.D.

The entropy per comparator can then be given as

$$H_i \geqslant \frac{n - \frac{\log n}{2} - .825}{n - 1}, \tag{18}$$

or as n gets large

$$H_i \rightarrow 1 \quad \text{as} \quad n \rightarrow \infty. \tag{19}$$

Thus for large n the individual comparators are informationally efficient even when the whole column is considered as a group. In other words, vertically adjacent comparators are essentially independent.

3.1.2 Conditional Entropy of Comparators

If both the inputs to a certain comparator are a's or b's, there is no uncertainty involved in making the decision about the state of the comparator, since the a's and b's are already in ascending order within themselves. To consider this effect we have calculated the conditional entropy of a column of comparators, given the state of the two half-size merging networks in the recursion. This quantity, $H_n\left(\beta\text{'s} \mid S\left(M^o_{n/2}, M^e_{n/2}\right)\right)$, can be defined as follows:

<u>DEFINITION</u> 5.

$$H_n\left(\beta\text{'s} \mid S\left(M^o_{n/2}, M^e_{n/2}\right)\right) = S_k \; \Pr[S_k(M^o, M^e)] \; H_n(\beta\text{'s} \mid S_k), \tag{20}$$

where $\Pr[S_k(M^o, M^e)]$ is the probability of state k of the two half-merging networks, and $H_n(\beta\text{'s} \mid S_k(M^o, M^e)$ is the entropy of the column of comparators when the half-merging networks are in state k. In the summation k is taken over all $\binom{n}{n/2}^2$ possible states of the half-merging networks.

The conditional entropy can be bounded as

$$H_n(\beta\text{'s} \mid S(M^o, M^e)) \leqslant n - \sum_{i=1}^{n-1} i p_i, \tag{21}$$

where p_i is the probability that i of the $n-1$ comparators have inputs which are both a's or both b's.

Expressions for p_i have been found. This technique, however, does not lend itself to closed-form solution, so another approximation was used to calculate the comparator entropy, given the states of the subnetworks.

The probability that the inputs to a comparator match can be calculated directly. Figure 6 shows the 2n outputs of the merging network and the two outputs associated with comparator j. Assume that these are both a's. There are n–2 remaining a's and 2n–2 remaining outputs. The total number of ways in which these can be arranged is $\binom{2n-2}{n-2}$. Therefore, the probability that both inputs to a comparator are a's is

$$\frac{\binom{2n-2}{n-2}}{\binom{2n}{n}}. \tag{22}$$

The case in which both inputs are b's is symmetrical so that the total probability of a match is

$$\frac{2\binom{2n-2}{n-2}}{\binom{2n}{n}}$$

$$= \frac{n^2 - n}{2n^2 - n}. \tag{23}$$

As n gets large, the probability that both inputs to a comparator match is 1/2 (in which case there is a trivial decision to make). In the other 1/2 cases the decision takes at most one bit. Thus, the entropy of each comparator must be less than 1/2.

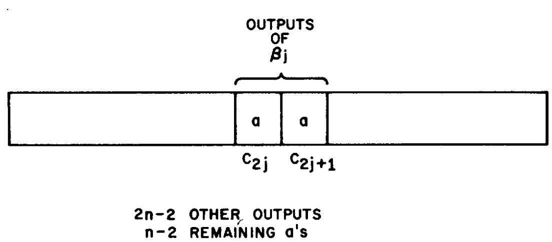


Fig. 6. Schematic representation of the outputs in a conditional entropy calculation.

3.1.3 Entropy Accounting for the Case n = 4

The case n = 4 has been exhaustively studied to account for all entropy involved in the network. There are $\binom{8}{4} = 70$ possible states for the whole network. Only the 35 states in which the smallest element in the list was an a were considered, as the other cases are symmetrical. In these 35 cases there were 3 possible states for the odd subnetwork: abab, aabb, and abba. For each of these states the possible states of the even subnetwork and the number of output sequences corresponding to each of them is shown in Table 2.

Table 2. Possible states of the odd and even subnetworks.

Odd Subnetwork State	Even Subnetwork State	Number of Sequences/70
abab	aabb	2
	abab	8
	abba	4
	baab	2
	baba	1
		17
aabb	aabb	2
	abab	2
	abba	1
		5
abba	abab	2
	abba	4
	baab	2
	baba	4
	bbaa	1
		13

Thus the probability of state abab for the odd subnetwork is 17/70, for state aabb it is 5/70, and for state abba it is 13/70. The 3 other states have the same probabilities so that entropy of the odd subnetwork is 2.432 bits.

Given that the odd subnetwork is in state abab, the probability distribution of the even subnetwork can be calculated quickly from Table 2 just by dividing the number of sequences for each state by the subtotal associated with that state of the odd subnetwork, in this case 17. The entropy of the resulting distribution is 1.972 bits. The entropies of the even subnetwork in the other two cases are 1.522 bits and 2.163 bits. The 3 cases, including the 3 symmetrical cases, have probabilities 17/35, 5/35, and 13/35, respectively. When these are used to weigh the entropies for the odd subnetwork a total entropy of 1.98 bits is found.

It was found by exhaustive study that in 16/70 cases all comparators had to make nontrivial decisions (i.e., none of the inputs matched), in 24/70 two comparators had to make nontrivial decisions, and in 30/70 cases only one comparator had to make a nontrivial decision.

Under the assumption of an upper bound of 1 bit/decision where the decision was nontrivial, this means an entropy of 1.71 bits associated with the comparators. These results are summarized as follows.

2.43 bits — odd subnetwork

1.98 bits — even subnetwork|odd subnetwork

1.71 bits — comparators odd and even subnetworks

6.12 bits — total.

From Len a 1, the lower bound for entropy in this network is $\log \binom{8}{4} = 70 = 6.13$, which agree ithin round-off error with the accounting procedure above.

3.1.4 Limiting Behavior

As n gets large the informational limit for the merging networks approaches 2n. The exhaustive accounting procedure is not easily extended to large n, but it seems appropriate to state what appears to be the trend. These 2n bits of entropy are somehow allocated among three places: the two subnetworks and the comparators. At this point it seems reasonable to guess that n bits of entropy are associated with the first subnetwork accounted for. Furthermore, it appears that n/2 bits are associated with the second subnetwork conditioned upon the state of the first subnetwork. Finally, the remaining n/2 bits should be associated with the comparators which is consistent with the upper bound found by considering the probability that the inputs to a comparator match.

3.2 BITONIC SORTING NETWORK

Batcher's second algorithm leads to the bitonic sorting network. While having the same order of growth of the number of comparators as the odd-even network, it differs in the lower order terms so that it always has somewhat more comparators. Its construction is very similar to the odd-even network, although the analysis is rather different. Stone[20] has shown that this algorithm can be used advantageously in a hardware sorting system in which several comparisons are made simultaneously.

Like the odd-even network, bitonic networks are based on sorting by merging. Where the odd-even merging network is defined recursively in terms of two merging networks, the merging network in a bitonic system is defined recursively in terms of smaller bitonic networks. Batcher[17] discusses the meaning of bitonic as follows.

> "We will call a sequence of numbers <u>bitonic</u> if it is the juxtaposition of two monotonic sequences, one ascending, the other descending. We also say it remains bitonic if it is split anywhere and the two parts interchanged. Since any two monotonic sequences can be put together to form a bitonic sequence, a network which rearranges a bitonic sequence into monotonic order, a bitonic sorter, can be used as a merging network."

Thus two ordered sequences can be merged by reversing the order of one and placing them on adjacent inputs so that the network receives as inputs a bitonic sequence. The recursive definition of the bitonic network is shown in Fig. 7. A merging network that merges two lists of size n is composed of n comparators and two n input bitonic networks which in turn are decomposed. Thus to merge two lists of n items requires $n \log n + n$ comparators. A sorting network is constructed from these bitonic networks using the sort-merge algorithm and requires a total of $n/4(\log^2 n + \log n)$ comparators. This is $\frac{n}{2}$ (log n) + n − 1 more than are required for the odd-even network with the same number of inputs. Basically, this is because each stage of the recursion of the odd-even network yields the minimum and the maximum of the inputs involved directly without using a comparator (see Fig. 3).

In order to investigate the inefficiencies of a bitonic network, the state probabilities of the comparators in the first column of a network were examined in detail. These are the same comparators that were shown in the recursive definition of the network (Fig. 7). The inputs to these comparators are items from two ordered lists: $(a_1, a_2, \ldots, a_n)$ and $(b_1, b_2, \ldots, b_n)$. For simplicity, these will be called the a's and the b's. Comparator i, labeled β_i, compares a_i with b_{n-i+1}. There are two possible outcomes and these correspond to the states of β_i. Either $a_i \leq b_{n-i+1}$ or $a_i > b_{n-i+1}$.

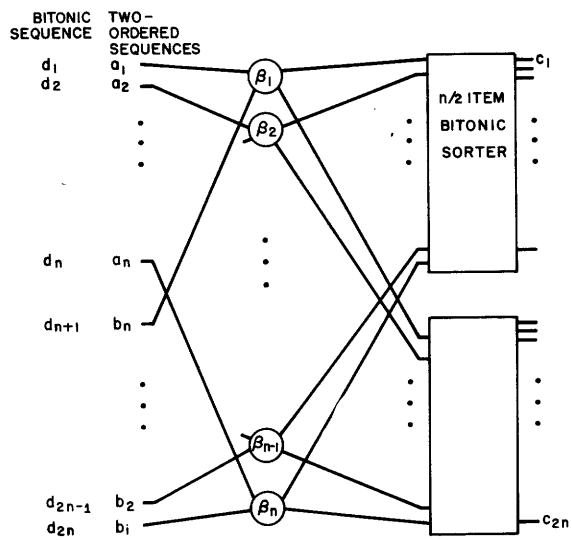


Fig. 7.

Batcher bitonic sorting network.

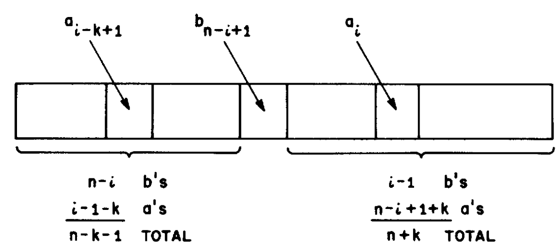


Fig. 8.

Schematic representation of the output of a bitonic network.

The probability $\Pr(a_i > b_{n-i+1})$ can be calculated explicitly by studying the ordered lists of all inputs. Such a typical list for the case $a_i > b_{n-i+1}$ is shown in Fig. 8. The element a_i is shown to the right of b_{n-i+1}, since it is assumed to be larger. There must be n−1 b's to the left of b_{n-i+1} and i−1 b's to the right of it. To the left of n_{n-i-1} there must be less than i−1 a's, since $a_i > b_{n-i+1}$; in particular, it is assumed that there are n − i + 1 + k b's to the right of b and i−1 b's to the right of b_{n-i+1}. The possible values for k are $0 < k < i - 1$.

The total number of sequences of a's and b's that have $a_i > b_{n-i+1}$ is then

$$\sum_{k=0}^{i-1} \binom{n-k-1}{i-1+k}\binom{n+k}{n-i+1+k}. \tag{24}$$

Since there are $\binom{2n}{n}$ ways in which n a's and n b's can be ordered, it follows that

$$\Pr(a_i > b_{n-i+1}) = \frac{\sum_{k=0}^{i-1} \binom{n-k-1}{i-1+k}\binom{n+k}{n-i+1+k}}{\binom{2n}{n}}, \tag{25}$$

under the assumption that all sequences are equally likely.

This expression has been evaluated numerically and the results for n = 8 are shown in Table 3. The comparators near the top and the bottom of the column are almost always in one state. The comparators near the middle have reasonable probabilities of being in either state. Table 2 also shows the binary entropy associated with each comparator and the sum of the entropies. This sum is an upper bound to the joint entropy of the whole column. In this case the entropy per comparator is only .32 bits. This trend is shown more dramatically when larger values of n are considered.

Table 3. Bitonic network state probabilities for n = 8.

Comparator Number	$\Pr(a_i > b_{n-i+1})$	H(Pr)
1	.000077	.0012
2	.0051	.045
3	.066	.350
4	.309	.892
5	.690	.892
6	.934	.350
7	.9449	.045
8	.9999	.0012
	Total	2.58
	Entropy/Comparator	$= \frac{2.58}{8}$
		= .32

Figure 9 shows the comparator probabilities and entropies for the case n = 32. The entropy of most comparators is less than 0.1, indeed the average entropy/comparator is only 0.16 bits. For large n the entropy per comparator decreases slowly as is shown in Fig. 10. It appears that the order of decrease is approximately 1/log n.

The conclusion drawn from these calculations is that most of the comparisons made in the bitonic network are very poor, in that there is little uncertainty associated with them as indicated by the low entropy. Some comparators do make reasonable decisions with a high amount of uncertainty, however, as is indicated by their entropy being close to 1.0.

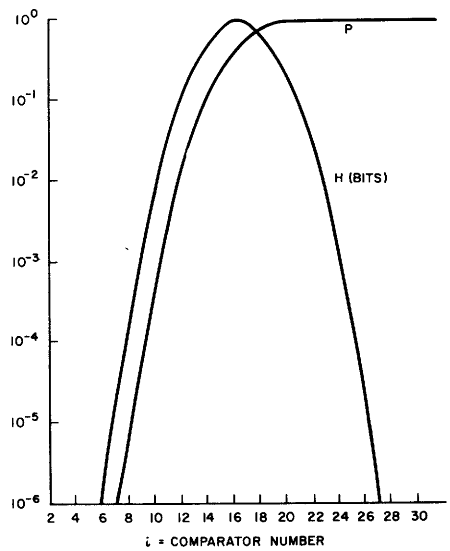


Fig. 9.
Comparator probability and entropy bitonic network for n = 32.

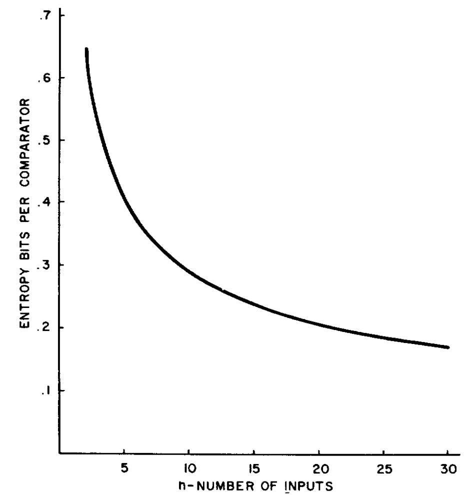


Fig. 10.
Entropy per comparator for bitonic networks.

3.3 SUMMARY

We have explored two sorting networks that require about the same number of comparators and hence are equally inefficient. The inefficiencies in each network can be attributed, however, to different causes.

In the case of the odd-even network the inefficiency is very subtle and was shown to be due to the lack of independence between the two n/2 input networks in the basic recursion. Since these networks are inefficient, in that they have some joint stages with low probability, the individual comparators in them must be inefficient no matter how they are interconnected.

The inefficiency in the bitonic network is easier to see. It is shown that it is due to poor comparisons made by individual comparators. Since these comparators have low state entropy, more comparators must be used so that the total entropy equals log n!.

Can sorting networks be built with less than $0(n(\log n)^2)$ comparators? This remains an unsolved question. Although the sort-merge algorithm is not inefficient in itself, it appears that it may be impossible to construct efficient merging networks with $0(n)$ comparators. This in itself does not rule out the possibility of efficient networks for some values of n which are not based on merging.

IV. CONNECTING NETWORKS WITH FINITE BLOCKING PROBABILITY

The connecting networks that are commonly used in telephone systems will now be discussed. Although many practical models have been given for such networks, little is known about bounds for the growth of their complexity. In order to get such bounds, the concept of ensembles of systems, which has been used successfully in the context of coding systems, is applied to connecting networks. Two types of ensembles of connecting networks are defined and their average blocking probability bounded. This is used in turn to give a bound on the growth of network complexity. It is shown that the requirement of finite blocking probability can be met with a complexity that has the same order as the informational minimum.

4.1 BACKGROUND

It has been said that the problem of telephone switching was first recognized when the third telephone was built. Since then many ingenious systems have been developed to solve practical problems. It was recognized at an early stage that the variable demands of telephone users made it desirable to engineer the amount of equipment to be installed to attain a suitable grade of service during "normal" use, rather than to install enough equipment to handle all possible calls simultaneously.

To approach this class of problems, traffic theory was developed. As Beneš[6] has written,

> "The first contribution to traffic theory appeared almost simultaneously in Europe and the United States during the early years of the 20th century. In America, G. T. Blood of the American Telephone and Telegraph Company had observed as early as 1898 a close agreement between the terms of a binomial expansion and results of observations on the distribution of busy calls. In 1903, M. C. Rorty used the normal approximation to the binomial distribution in a theoretical attack on trunking problems, and in 1908, E. C. Molina improved Rorty's work by his (or Poisson's) approximation to the binomial distribution."

Perhaps the greatest contributor to early traffic theory was A. K. Erlang who worked for the Copenhagen Telephone Company. During 1909-1918 he developed the first dynamic theory of telephone traffic which in many ways set the stage, in 1933, for the more elegant and more general theory of stochastic processes of A. N. Kolmogorov.

Early telephone switching systems were easily modeled as a series of independent obstacles through which a call had to pass. This was a result of the direct control of the equipment by the dialed digits. In 1938, the concept of common control was introduced with the Bell No. 1 Crossbar System. Rather than setting up part of a path as each digit was dialed, the digits were stored and the <u>common control</u> sought to set up a path when the whole number was dialed and available. This resulted in more efficient use of equipment, but it also required the development of a new class of models of system behavior. Now it was possible that links could be available in each stage of the network, but the call could not be completed because it was not possible to interconnect the idle links.

Approximations have been developed to estimate the blocking probabilities in crossbarlike networks. Most results in this field are reviewed in the comprehensive work of Syski.[25] None of these models is simple enough to be both really tractable and on firm mathematical ground. For example, Lee[26] and LeGall[27] have developed a model that is very simple to deal with and is reasonably accurate in many cases, but, Beneš[6] has recently shown that it is on very shaky theoretical grounds. The model of C. Jacobaeus[28] is more detailed, but is still based on assumed a priori distributions.

One of the most successful approaches to calculating blocking probabilities has been the NEASIM program of Grantges and Sinowitz.[29] This is a hybrid approach, combining both simulation and mathematical modeling. It does not give much insight into the general problem, however.

The study of nonblocking networks has been more successful than the study of networks with blocking. This appears to be because nonblocking networks can be discussed readily from a pure combinatorial viewpoint and definite statements can be made about their behavior.

The first nonblocking network, derived in 1951 by Clos, is based on the algorithm shown in Fig. 11. Clos showed that if $s \geq 2R-1$, the network would be strictly nonblocking; that is, any call between idle terminals could be completed regardless of the state of the network. This algorithm can then be applied recursively to yield nonblocking networks with more switches of a smaller size and more stages. In general, the number of crosspoints decreases as more and more stages are used. No closed-form solution exists for the optimum number of stages or the optimal factoring of n used in the recursion, however. The best guide to the synthesis of such networks is an unpublished work by Keister[10b] which used a computer program to find by exhaustive search the optimal networks, for n $16 < n < 50,000$.

More recently, Cantor[16] has used a somewhat different argument to show that the number of contacts $X(n)$ satisfies the following inequality $X(n) \leq 8\, n \log^2 n$. Then

$$X(n) = 0(n \log^2 n). \qquad (26)$$

Beneš[6] has derived a class of networks called <u>rearrangeably nonblocking</u>. These networks are nonblocking if existing calls can be rearranged or moved before new calls are added. This is not practical in conventional telephony so these systems have not been manufactured. The algorithm is basically the same as shown in Fig. 11, except now the condition is only $s \geq r$. In contrast to the case of strictly nonblocking networks, the minimum possible number of contacts can be easily found for this type of network. The minimum network has been shown by Beneš[7] to be a function of the prime factors of n. When n is an integral power of two the case is much simpler. The minimal network is then composed of 2×2 switches which Joel[9] has called β elements. The resulting networks are equivalent to those derived by the recursion shown in Fig. 12. These networks have $n \log n - n + 1$ β elements, which is the minimum possible order of growth.

In practical telephone systems, switching networks are designed with blocking probabilities of order .001-.01, depending upon the actual purpose of the network.

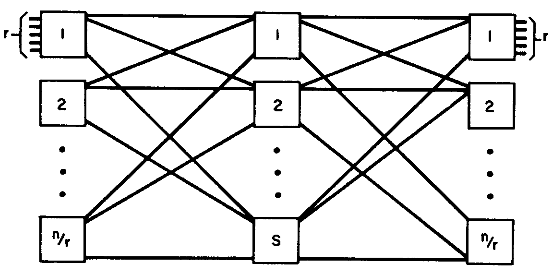


Fig. 11. Definition of Clos network.

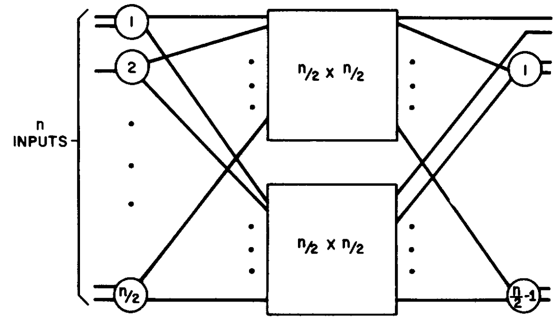


Fig. 12. Recursive definition of rearrangeable network of β elements.

The inputs to the network can be very low utilization lines with offered load of the order of .02 Erlang, or in some cases they could be long-distance trunks with offered load of the order of .5 Erlang/line. (The load expressed in Erlangs can be interpreted as the average number of calls a line or group of lines carries or is offered.) For example, a No. 1 crossbar district office with 80,000 crosspoints might have 1000 inputs and outputs and handle 708 Erlangs with a blocking probability of .003. In comparison, a Clos network could handle 1000 inputs and outputs carrying 1000 Erlangs with no traffic and have approximately 138,000 crosspoints. It is not clear that nonblocking networks offer advantages over conventional networks with practical values of n (less than 50,000). One reason for this uncertainty is that there have been no firm results for the order of growth of networks with blocking.

There have been two basic papers on limits for contact growth in connecting networks. Ikeno[30] has shown that networks can be built with X(n) contacts, where

$$X(n) < 10.9\ A \log A, \tag{27}$$

with A the total carried load in Erlangs, or

$$A = na(1-B), \tag{28}$$

where a is the offered load per input line, and B is the blocking probability. This result does not furnish a direct relationship between blocking probabilities and network growth because it only considers the total amount of traffic offered and may lead to networks having expandors or concentrators so that the inputs to the network may carry more or less traffic than the original lines in which we are interested.

A more direct approach to the problem of network size with blocking is the work of Lotze,[31] which has recently been extended by Feiner and Kappel.[32] (In the nomenclature, α is called "A", and only this case n = 1 is relevant to the present discussion.) They define an access factor α as

$$\alpha = \frac{k^s(1-a)^s}{n}, \tag{29}$$

where the networks are composed of $k \times k$ crossbar switches. Then the number of contacts can be minimized by taking derivatives and setting them equal to zero. Feiner and Kappel have shown empirically that this access factor is highly correlated with blocking for several different networks. The minimum number of contacts is

$$X(n) = 4e(n \log_e n + n \log_e \alpha). \tag{30}$$

The biggest problem with this approach is that the relationship between A and the blocking probability is not rigorously proved. Indeed, the authors give plots of X blocking obtained by simulation as a function of A for several networks, and sometimes the curves are convex $\cup$, while in other cases they are convex $\cap$.

A similar approach to minimization is possible with the probability graph model of LeGall[27] and Lee.[26] Even though this model has been helpful in engineering applications, there is no reason to believe that it will yield absolute bounds.

4.2 INTRODUCTION TO ENSEMBLES OF CONNECTING NETWORKS

An ensemble of connecting networks will now be defined. Such networks have a fixed number of inputs, stages, and crosspoint contacts, but different connection patterns. We shall then study the properties of this ensemble of networks and of a similar ensemble. A bound is computed for the average blocking probability for such an ensemble with some arbitrary number of inputs and crosspoints. This result is then turned around to give an upper bound for the minimum number required to achieve an arbitrary blocking probability, given the number of inputs.

The major block to finding limits for growth of contacts with inputs is that there are no mathematically tractable expressions for bounding network blocking. A similar problem arises in the context of error probabilities for error-correcting codes. In Shannon's classic paper[33] of 1948, he introduced a new approach to such problems, the average performance of an ensemble of systems.

One of Shannon's fundamental theorems is stated thus.

> "Let a discrete channel have a capacity C and a discrete source the entropy per second H. If $H < C$ there exists a coding system such that the output of the source can be transmitted over the channel with an arbitrarily small frequency of errors."

He then points out the key to proof of this and of others that followed.

> "The method of proving the first part of this Theorem is not by exhibiting a coding method having the desired properties, but by showing that such a code must exist in a certain group of codes. In fact we will average the frequency of errors over this group and show that this average can be made less than ϵ. If the average of a set of numbers is less than ϵ, there must exist at least one in this set less than ϵ. This will establish the desired result."

Wozencraft and Jacobs[34] have commented on the significance of this proof.

> "It may be surprising that one can bound the average probability for a collection of communication systems when one cannot calculate the probability of error of an individual system. Such was Shannon's insight."

The theorem is then proved by showing that the average error probability for memoryless channels over the ensemble of all codebook codes decreases exponentially as the block length increases, if the rate is less than the channel capacity.

The logic of this proof has been extended to more specific types of coding systems than just codebook codes. For example, a result of Elias[35] shows that for the ensemble of all convolutional encoding systems such as the one shown in Fig. 13, the average error probability of an ensemble of such systems when L bits are sent can be bounded by

$$\overline{P(E)} \leqslant L2\ \exp[-vk(R_o' - 1/v)], \tag{31}$$

where

v = number of modulo 2 adders used in the encoder

k = number of bits in the input shift register, also known as the "span length"

R_o' = binary symmetric channel error exponent
$= 1 - \log_2 \left[1 + 2\sqrt{p(1-p)}\right]$

p = channel error probability.

Such an encoder works as follows: Input bits are shifted from left to right in the k-bit shift register. Every time a new bit is inserted the v modulo 2 sums of the shift register contents are sent out. Decoder operation has been described by Wozencraft and Jacobs.[34]

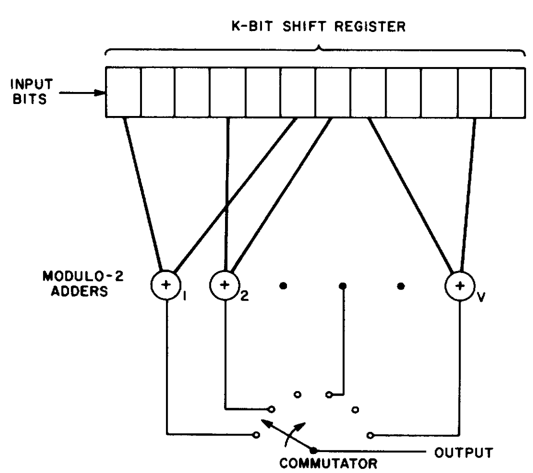


Fig. 13. Convolutional encoder.

Although such proofs do not tell how to build an encoder that will work as well as the bound, the knowledge of their existence has led to the discovery of encoders with this order of behavior. Indeed, from the Chebyshev inequality[36]

$$\Pr(t > \delta)\ \frac{\bar{t}}{\delta}, \tag{32}$$

we can see that most encoders must have error probabilities of this order if proper decoding is used.

The concept of using random structure for existence proofs can also be used in the context of switching networks. The objectives here will be to define an ensemble of networks with n inputs with a fixed number of crosspoints and to calculate the blocking probability over the ensemble. Then a bound for the rate of growth of number of contacts in such networks for a given n and P(B) can be given.

The first ensemble of networks that will be analyzed is an ensemble of homogeneously structured switching networks and a typical member network that is shown in Fig. 14. In this diagram, inputs, outputs, and intermediate points are shown as nodes. Contacts are shown as edges connecting nodes and are assumed to be normally open. These networks can be classified with four parameters:

The network has n inputs and ℓ stages.

k is called the expansion factor. All stages except stage 0 and stage $\ell-1$ have kn nodes. k does not have to be integral as long as the product kn is.

c is called the fanout. All nodes except those in stage 0 are connected to nodes in the next stage to the right by c contacts. The contacts between stage 1 and stage 0

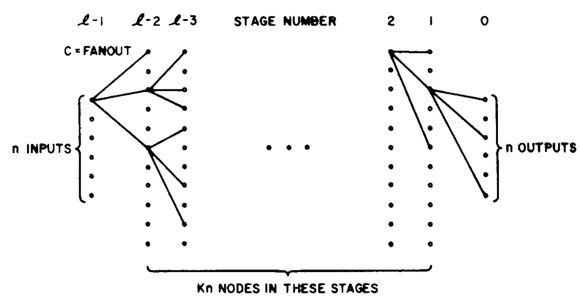


Fig. 14. Homogeneously structured connecting network.

are placed without replacement so that each node of stage 1 is connected to c distinct nodes in stage 0. The contacts in all other stages are placed with replacement so it is possible that two or more parallel contacts might join two nodes. This may seem somewhat inefficient, but it simplifies the mathematics. The blocking for such a network must be larger than if only placing contacts without replacment was used, and if $c \ll kn$ the fraction of the contacts involved in such parallel links is negligible.

Now an ensemble of such networks can be defined.

DEFINITION 6. $N(n, \ell, c, k)$ is the ensemble of all homogeneous randomly structured switching networks with the same value for parameters n, ℓ, c, and k as described above, each with equal probability.

A bound for the blocking probability of such networks averaged over the ensemble will be calculated in sections 4.3 and 4.4. In section 4.5 a related ensemble will be defined and similar results will be given which lead to a bound for the growth of networks with blocking. As for the ensemble of all block encoding systems and the ensemble of all convolutional encoders discussed above, it is impossible to calculate the blocking or a good bound to the blocking for any particular network in $N(n, \ell, c, k)$. It is relatively straightforward, however, to calculate a bound to the average blocking over the ensemble. This shows then that at least one network in the ensemble has a blocking less than the bound.

4.3 ANALYSIS OF AN ENSEMBLE OF HOMOGENEOUSLY STRUCTURED CONNECTING NETWORKS

A bound will be derived for the average blocking probability of the ensemble $N(n, \ell, c, k)$. The blocking probability is the probability that an input output pair cannot be connected for lack of a path, given that the end points are idle. In general, this probability is a function of the traffic offered to the network and the resulting probability distribution of network states. It is virtually impossible to calculate directly, or even bound, the network state distribution or the blocking probability for a particular network. It is possible, however, to bound the average blocking probability over the ensemble of networks in a relatively straightforward manner. This then shows that at least one network in the ensemble has the average blocking probability. In particular, the following theorem will be proved.

Theorem 4

For any blocking probability ϵ there is a c such that for any number n of inputs there exists at least one connecting network with n inputs and blocking probability ϵ which has $< cn \log n$ contacts.

For such networks the requirement of finite blocking probability ϵ results in networks whose number of contacts is a multiple of the informational minimum. It will be shown that the constant c is $0(n \log 1/\epsilon)$.

For any given offered traffic to the network (the sequence of paths which it is requested to make), there exists a stochastic process whose states are the number of calls in progress. With certain assumptions it can be shown that this process satisfies the Markov property, but this is not important here. In equilibrium such a process has a well-defined state distribution $\Pr(q)$, where q is the number of paths connected, $0 \leq q \leq n$. If the blocking probability of a network is known, given that q_o paths are connected, the overall blocking can be computed as follows:

$$\Pr(\text{blocking}) = \sum_{q_o=0}^{n-1} p_r(q_o) \Pr(\text{blocking} | q = q_o).$$

As we have stated, it is difficult to calculate $\Pr(q)$, but if $\Pr(\text{blocking } q = q_o)$ is upper-bounded for all q, then an upper bound for the overall blocking can be computed as follows:

$$\Pr(\text{blocking}) \leq \max\, [\Pr(\text{blocking} | q = q_o)].$$

Consequently, bounding the blocking probability of a network is reduced to bounding the probability of blocking, given that a certain number of paths are connected.

Now, consider the ensemble of all networks of the form $N(n, \ell, c, k)$ each of which is in all possible states in which q_o paths are connected. The next step is to calculate the blocking probability from an idle point in stage j to an idle point in stage 0 averaged over this ensemble. This probability is called $P_j(q_o)$, and can be shown to be given by

$$P_j(q_o) = \left[1 - \left(1 - \frac{q_o}{k_n}\right)\left(1 - P_{j-1}(q_o)\right)\right]^c . \tag{33}$$

The $\left(1 - \frac{q_o}{cn}\right)$ term is the probability that the point in stage j-1 that a contact goes to is idle, under the assumption that all routing patterns are equally likely. The $(1-P_{j-1}(q_o))$ term is the probability that, given a point in stage j-1 is reached, a connection can be made to the desired output point. The exponent c is used because there are c contacts leaving every node. Equation 33 is monotonic, increasing as a function of q_o, so it can be upper-bounded by setting $q = n - 1$.

The value for $P_1(q_o)$ is independent of q_o, since other paths do not interfere with the possibility of connecting from stage 1 to stage 0. Then

$$P_1 = 1 - \frac{c}{n}. \tag{34}$$

Now the problem of bounding the blocking probability of the ensemble is reduced to bounding the iterations obtained from (33) and (34).

For mathematical convenience, it is helpful to use the following bound:

$$\begin{aligned} 1 - \frac{n-1}{kn} &> 1 - \frac{n}{kn} \\ &> 1 - \frac{1}{k} \\ &> 1 - \frac{k-1}{k}. \end{aligned} \tag{35}$$

This has a major advantage: it gives a bound on which P_j is independent of n!

$$\left[1 - \left(\frac{k-1}{k}\right)(1-P_{j-1})\right]^c > \left[1 - \left(1 - \frac{n-1}{kn}\right)(1-P_{j-1})\right]^c . \tag{36}$$

So

$$P_j < \left[1 - \left(\frac{k-1}{k}\right)(1-P_{j-1})\right]^c \tag{37}$$

approaching equality as $1/kn \to 0$.

Figure 15 shows graphically the meaning of this relationship. Here the bound on P_j

is plotted against P_{j-1}. The curve passes through the points $(0, (1/k)^c)$ and $(1, 1)$. At $P_{j-1} = 0$ the slope is $c(k-1)/k^c$, and at $P_{j-1} = 1$ the slope is $c(k-1)/k^c$. As c and k become large the curve becomes shaped like a backwards "L".

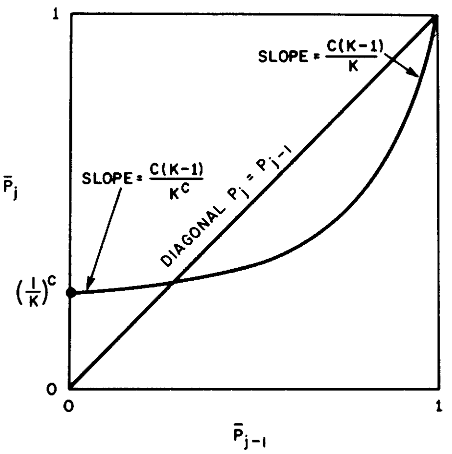


Fig. 15. Probability of connection from stage J as a function of probability from stage J-1.

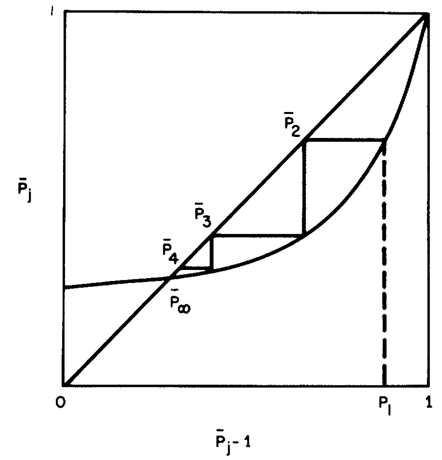


Fig. 16. Construction of values for $\overline{P}_j$, given P_1.

The manner in which the bound on P_j changes as j increases can be shown graphically by finding P_1 on the curve and then constructing steps between the curve and the 45° line connecting (0, 0) and (1, 1). All the derivatives of Eq. 37 are non-negative, so if the slope at (1, 1) is greater than unity it can be shown that the curve crosses the diagonal once and only once. This construction is shown in Fig. 16 where the decrease of P_j as j increases is apparent. The blocking does not decrease without limit for, as $j \to \infty$, $P_j \to P_\infty$, where $P_\infty > 0$. The value of P_∞ can be found by solving the following equation for the intersection between the curve and the diagonal,

$$P_\infty = \left[1 - \left(\frac{k-1}{k}\right)(1-P_\infty)\right]^c. \tag{38}$$

No closed-form solution exists for P_∞ if $c > 3$, but if $c\left(\frac{k-1}{k}\right) < 1$, then the only solution is $P_\infty = 1$. This has a useful interpretation. The factor $c\left(\frac{k-1}{k}\right)$ is the expected number of idle nodes in the next stage that a node is connected to, thus the "effective fanout." Therefore the condition necessary for the blocking to decrease as more stages are added to a network is that the effective fanout be greater than unity.

No closed-form general solution exists for P_∞ for $c > 3$. But from Fig. 15 it is clear that $P_\infty > (1/k)^c$. Also, from substitution it can be shown that $P_\infty < \exp\left[-c\left(\frac{k-1}{k}\right)\right]$. Therefore,

$$\left(\frac{1}{k}\right)^c < P_\infty < \exp\left[-c\left(\frac{k-1}{k}\right)\right]. \tag{39}$$

A useful interpretation of part of this bound is that the probability that all c contacts from an input go to busy nodes in the next stage is

$$\left(\frac{n-1}{kn}\right)^c$$

$$\cong \left(\frac{1}{k}\right)^c. \tag{40}$$

Since all connections from each input must go through such a bottleneck, the overall blocking cannot be decreased below this amount, regardless of how many stages are used.

4.4 BOUNDS FOR $\overline{P}_j$

The expression for $\overline{P}_j$ (Eq. 37) does not lend itself directly to bounds as j increases. An upper bound for $\overline{P}_j$ will be found using a two-piece linear bound for the defining equation. The linear bound that is used is shown in Fig. 17. The two lines that are used connect $(0, (1/k)^c)$ and $(\theta, P(\theta))$, and $(\theta, P(\theta))$ and $(1, 1)$, where $P(\theta)$ means the point on the curve corresponding to $P_{j-1} = \theta$. Since these lines lie above the curve, any value for P_j obtained by using them is an upper bound to the actual value for P_j. In the following derivation the primed variables such as P_i' will mean values obtained from using the linear bound. Thus $P_i' \geqslant P_i$. Now

$$P_j' = P_1 - (P_1-P_2') - (P_2'-P_3') - \dots - (P_{j-1}'-P_j'). \tag{41}$$

It can be shown that

$$m_1 = \frac{P_i' - P_{i+1}'}{P_{i-1}' - P_i'}. \tag{42}$$

Therefore

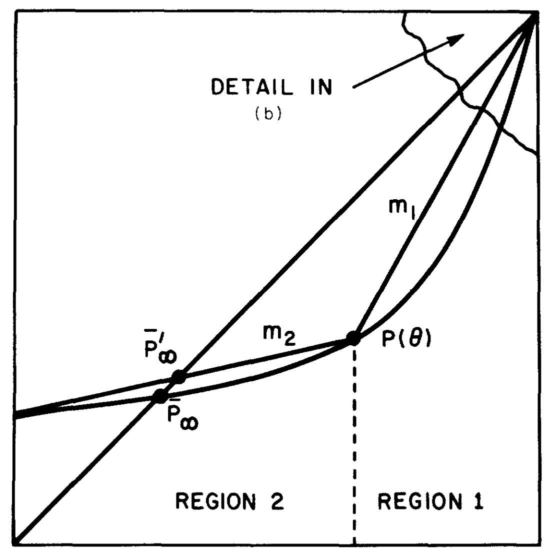
$$P_j < P_1 - (P_2'-P_1) \sum_{i=0}^{j=2} m_1^i \tag{43}$$

for $P_j > P(\theta)$. Now $P_1 = 1 - \frac{c}{n}$, as was stated previously in Eq. 34.

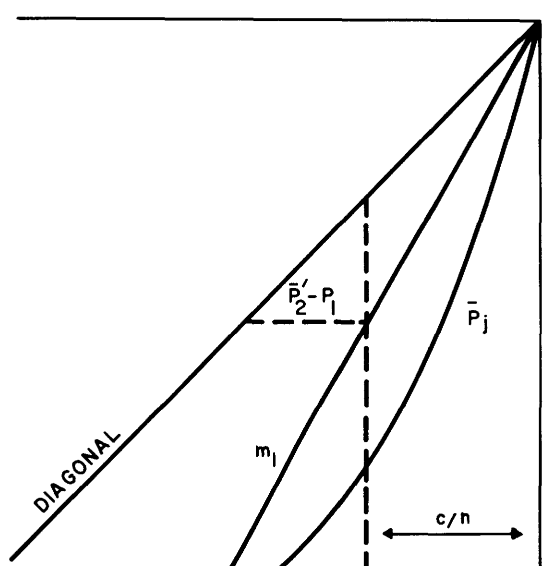
From Fig. 17, it can be seen that

$$P_2' - P_1 = \frac{c}{n} m - \frac{c}{n}$$

$$= \frac{c}{n}(m-1). \tag{44}$$



(a)



(b)

Fig. 17. (a) Linear bound for $\overline{P}_j$.
(b) Detail of the corner of (a).

Then

$$P_j < P_1 - \frac{c}{n}(m-1) \sum_{i=0}^{j=2} m^i$$

$$< 1 - \frac{c}{n} - \frac{c}{n}(m-1)\frac{m^{j-1} - 1}{m - 1}$$

$$< 1 - \frac{c}{n} - \frac{c}{n}(m^{j-1}-1)$$

$$< 1 - \frac{c}{n} m^{j-1}.$$

Now

$$m_1 = \frac{1 - P(\theta)}{1 - \theta}. \tag{45}$$

At this point it is convenient to set $\theta = 1/2$. Any value for θ such that $P_\infty < \theta < 1$ will suffice, indeed some may even yield tighter bounds, but $\theta = 1/2$ appears to give results with an interesting order of growth, other values of θ have not affected this order of growth, and $1/2$ is a convenient value to calculate with. Then

$$m_1 = \frac{1 - \left[1 - \left(\frac{k-1}{k}\right)\left(1 - \frac{1}{2}\right)\right]^c}{1 - \frac{1}{2}}$$

$$= \frac{1 - \left[1 - \frac{k-1}{2k}\right]^c}{\frac{1}{2}}$$

$$= 2\left\{1 - \left(\frac{2k-k+1}{2k}\right)^c\right\}$$

$$= 2\left[1 - \left(\frac{k+1}{2k}\right)^c\right], \tag{46}$$

Substituting this in Eq. 45 yields

$$P_j < 1 - \frac{c}{n}\left[2\left(1 - \left(\frac{k+1}{2k}\right)^c\right)\right]^{j-1}. \tag{47}$$

To find out how large j_1 must be to get to the breakpoint in the bound, substitute $\theta = 1/2$ for P_j to obtain

$$\frac{1}{2} = 1 - \frac{c}{n}(m_1)^{j_1 - 1}$$

$$\frac{1}{2} = \frac{c}{n}(m_1)^{j_1 - 1}$$

$$\log \frac{n}{2c} = (j_1 - 1) \log m_1$$

$$j_1 - 1 = \frac{\log n/2c}{\log m_1}$$

$$j_1 = \frac{\log n/2c}{\log m_1} + 1$$

$$= \frac{\log n/2c}{\log \left[2\left(1 - \left(\frac{k+1}{2k}\right)^c\right)\right]} + 1. \tag{48}$$

Since j_1 must be integral, it follows that

$$j_1 < \frac{\log n/2c}{\log \left[2\left(1 - \left(\frac{k+1}{2k}\right)^c\right)\right]} + 2. \tag{49}$$

The significance of this equation is that for arbitrary k and c the blocking probability can be reduced to θ, in this case $\theta = 1/2$, with a number of stages which increases only logarithmically with n.

A similar approach has been used to study the decrease of P_j for $P_j < P(\theta)$. This analysis will be directed to the question: How many stages are required to reduce the blocking from $P(1/2)$ to some desired P_f? In order to do this, P' will be set arbitrarily to $P_f/2$ and the value of j_2 will be found that yields $P_{j_1+j_2} = P_j$. Other choices for P_f lead to similar results.

First, a bound for m_2 must be found.

$$m_2 = \frac{P(1/2) - P(0)}{1/2 - 0}$$

$$= \frac{\left(\frac{k+1}{2k}\right)^c - \left(\frac{1}{k}\right)^c}{1/2}$$

$$< 2\left(\frac{k+1}{2k}\right)^c. \tag{50}$$

Then P' is the intersection of the diagonal and the line with slope m_2.

$$P'_\infty = (1/k)^c + m_2 P'_\infty$$

$$P'_\infty(1-m_2) = (1/k)^c$$

$$P'_\infty = \frac{(1/k)^c}{1 - m_2}$$

$$< \frac{(1/k)^c}{1 - 2\left(\frac{k+1}{2k}\right)^c} \qquad (51)$$

If k and c are limited to $c \geq 3$, $k \geq 2$ to simplify bounding, this yields

$$P'_\infty < \frac{(1/k)^c}{1 - 2\left(\frac{3}{4}\right)^3}$$

$$< 6.4(1/k)^c. \qquad (52)$$

Letting $P'_\infty = P_f/2$, we obtain

$$P_f/2 = 6.4(1/k)^c$$

$$P_f = 12.8(1/k)^c$$

$$\log P_f = c \log k + \log 12.8$$

$$c \log k = \log 12.8 - \log P_f$$

$$c = \frac{\log 12.8 + \log 1/P_f}{\log k} \qquad (53)$$

Now an expression similar to Eq. 48 can be derived for Region 2. Assume that $P_1 = \theta = 1/2$ which is convenient, since the desired result is the number of additional stages required to reduce the blocking from 1/2 to P_f. Then as before

$$P_j < P'_j = P_1 - (P_1 - P'_2) - \ldots - (P_{j-1} - P'_j)$$

$$< P_1 - (P_1 - P'_2) \sum_{i=0}^{j-2} m_2^i.$$

Now

$$P'_\infty = P_1 - (P'_2 - P_1) \sum_{i=0}^{\infty} m_2^i,$$

so

$$P_1 = P_\infty + (P'_2 - P_1) \sum_{i=0}^{\infty} m_2^i. \tag{54}$$

Substituting Eq. 54 in Eq. 45 gives

$$P'_j = P'_\infty + (P_1 - P'_2) \sum_{i=0}^{\infty} m_2^i - (P_1 - P'_2) \sum_{i=0}^{j-2} m_2^i$$

$$= P'_\infty + (P_1 - P'_2) \sum_{i=j-3}^{\infty} m_2^i$$

$$= P'_\infty + (P_1 - P'_2) \frac{m_2^{j-3}}{1 - m_2}. \tag{55}$$

Similarly to Eq. 44,

$$P_1 - P'_2 = \frac{1}{2} - \left(\frac{m_2}{2} + \left(\frac{1}{k}\right)^c\right)$$

$$= \frac{1}{2}(1 - m_2) - (1/k)^c$$

$$< \frac{1}{2}(1 - m_2), \tag{56}$$

so

$$P'_j < P'_\infty + \frac{1}{2}(1 - m_2) \frac{m_2^{j-3}}{1 - m_2}$$

$$< P'_\infty + \frac{1}{2} m_2^{j-3}. \tag{57}$$

The first term in this expression has already been set equal to $P_f/2$; therefore, to get the sum to be P_f, the first term is set equal to the second term.

$$6.4(1/k)^c = \frac{1}{2} m_2^{j_2-3}$$

$$\log 6.4 - c \log k = (j_2-3) \log m_2 - \log 2$$

$$\log 6.4 - c \log k + \log 2 = (j_2-3) \log m_2$$

$$j_2 < \frac{\log 6.4 - c \log k + \log 2}{\log m_2} + 3$$

$$< \frac{\log 6.4 - c \log k + \log 2}{\log \left(\frac{k+1}{2k}\right)^c} + 3$$

$$< \frac{\log 6.4 - c \log k + \log 2}{c \log \frac{k+1}{2k}} + 3. \tag{58}$$

Since it was previously assumed that $c \geqslant 3$ and $k \geqslant 2$, we have

$$j_2 < \frac{\log 6.4 - 2 \log 2}{3 \log \frac{3}{4}} + 3$$

$$< \frac{1.85 - 2(.69)}{-3(.29)} + 3$$

$$< .54 + 3.$$

Because k must be integral this yields

$$j_2 < 4. \tag{59}$$

This means that if k and c are chosen in accordance with Eq. 43, only four stages are required to reduce the blocking from 1/2 to P_f, regardless of the value chosen for P_f. Thus the value of P_f affects only the number of contacts by controlling the values of k and c, not the number of stages.

The total number of contacts now required for the network is

$$x = n(j_1+j_2) \, ck$$

$$= n \frac{\log n/2c}{\log \left[2\left(1 - \left(\frac{k+1}{k}\right)^c\right)\right]} + 4ck, \tag{60}$$

where

$$c = \frac{\log 12.8 + \log 1/P_f}{\log k}. \tag{61}$$

Thus

$$x = 0(n \log n + \log 1/P_f), \tag{62}$$

which proves Theorem 4.

4.5 AN ENSEMBLE OF NONHOMOGENEOUSLY STRUCTURED CONNECTING NETWORKS

The homogeneously structured connecting networks discussed above have constant values of c and k throughout the network. It is possible to obtain different orders of the growth of contacts in a network by using values of these parameters which change from stage to stage. Chung L. Liu[37] has pointed out, however, that for the type of network discussed thus far $P_{\ell-1} < P_j k_j n$, where k_j is the expansion factor in stage j. This follows from the fact that an input must have access to at most all $k_j n$ nodes in stage j, each of which has blocking to the outputs of P_j. This means that the first few stages act as a bottleneck and dominate blocking. Thus such networks do not have a lower order of growth than was found previously. This bound does not affect the networks with constant c and k, since in such networks $P_1^{kn} < P_\infty$, so the overall blocking is greater than the limit that was implied by the bottleneck.

Lower orders of growth of contacts can be reached by some modification of the network definition. In the rest of this section a somewhat different ensemble will be defined that will prove the following theorem.

Theorem 5

There is a constant c_1 such that given any probability of blocking $\epsilon > 0$, there exists a value n_o such that for any $n > n_o$ a network can be constructed with $< c_1 n \log n$ contacts and blocking probability $< \epsilon$. Conversely, a network cannot achieve blocking probability $\epsilon < 1$ for large n with less than $c_2 n \log n$ contacts.

In particular, it will be shown that networks exist for any n and ϵ which have $0(n \log n) + 0(n \log 1/\epsilon)$ contacts. Alternatively, this result can be viewed as meaning that for any ϵ there exists a class of networks such that for large n the ratio of the number of contacts in the network to log n! approaches a constant. The converse is that it is impossible to achieve a lower order of growth.

It should be observed that this bound is not useful for values of P_f less than e^{-n}, since the second term then becomes $0(n^2)$ and networks with no blocking can be built with only n^2 contacts, that is, the square crosspoint nets. Also, the proof using randomly

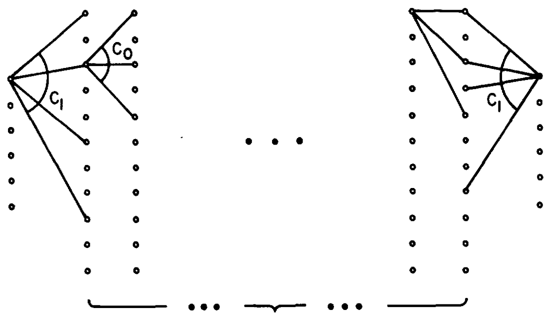


Fig. 18. Nonhomogeneously structured connecting network.

structured networks breaks down at this point. Because of the nonblocking square networks the bound still holds. Actually, such low values of blocking are not of practical interest because, even for n = 5000, e^{-n} is so small that blocking would never occur during the physical life of the system. The type of network discussed here is called a nonhomogeneously structured connecting network and is illustrated in Fig. 18. It is fundamentally the same as the networks already discussed with the following changes: The contacts between the two end stages and the stages adjacent to them are placed differently from the other contacts. In particular, every node in stage $\ell-1$ has c_1 contacts connecting it to nodes in stage 1-2. Every node in stage 0 has c_1 contacts connecting it to nodes in stage 1. Thus the contacts are placed from left to right everywhere in the network, except between stage 0 and 1. The rest of the network has fanout c_o placed as before.

DEFINITION 7. $N'(n, \ell, c_o, c_1, k)$ is the ensemble of all nonhomogeneous randomly structured switching networks with parameters n, ℓ, c_o, c_1, and k as described above.

It has been shown that the end-to-end blocking of the center part of the network can be P_o if the network has $0(n \log n \log 1/P_o)$ contacts. Now, if each of the overall inputs and outputs of the network can reach m inputs and outputs of the center network, the overall blocking can be bounded by $P_f < (P_o)^m$. This can be shown as follows. Pick one of the m available inputs and one of the m available outputs. By definition, the probability of blocking between these points is P_o. If the connection between these points is blocked, the conditional probability of blocking between any other pair of points is less than P_o. The conditional blocking can be viewed as the blocking in an ensemble of homogeneous networks with at most n-2 calls and kn-1 nodes in each stage. The original center network had at most n-1 calls in kn nodes. Since

$$\frac{n-1}{kn} > \frac{n-2}{kn-1} \qquad \text{for } k > 1, \tag{63}$$

the blocking in this new network must be less than the blocking in the center network. Similarly, it can be shown that the probability of blocking between the i^{th} input-output pair, given that i-1 input-output pairs are blocked, is also less than P_o. Equation 63 then follows from the fact that there are m input-output pairs that can be used.

Actually, this is a very conservative argument. There are m^2 possible paths which could be used. It seems likely that the probability of each of these being usable must be of the order of P_o, so that the overall probability would be of the order of $(P_o)^{m^2}$. This cannot be readily shown, however, and the bound of Eq. 63 is sufficient for the proof of Theorem 5. Then, by the union bound, it follows that

$$P_f \leq P_{beg} + (P_o)^m + P_{end}, \tag{64}$$

where

P_f = overall blocking probability

P_{beg} = Pr[an input can reach less than m idle center inputs]

P_{end} = Pr[an output can reach less than n idle center outputs].

Now, given certain value of P_f, we find values of m network parameters which make each term in (64) $\leq P_f/3$.

$$P_{beg} = P_o^m = P_{end} = P_f/3$$

$$\therefore\ m > \frac{\log P_f/3}{\log P_o}. \tag{65}$$

Since the case $\theta = 1/2$ has been studied, it is convenient to set $P_o = 1/2$. The derivation of the number of stages necessary to achieve this value is identical to the previous derivation, except that in this case $P_1 = 1 - \frac{c}{kn}$. Thus Eq. 45 becomes

$$P_j < 1 - \frac{c}{kn} m^{j-1}, \tag{66}$$

and by following the same steps it can be found that

$$j_1 = \frac{\log kn/2c}{\log \left[2\left(1 - \left(\frac{k+1}{2k}\right)\right)^c\right]} + 2 \tag{67}$$

which is similar to Eq. 49.

Finding the appropriate value for c_1 is more complicated. Since fewer than n of the kn inputs to the center network are used at any one time,

$$\Pr[\text{a contact leads to an unavailable input}] \leq \frac{n}{kn}$$

$$\leq \frac{1}{k}.$$

Hence

$$\Pr[\text{a contact leads to an available input}] > \frac{k-1}{k}. \tag{68}$$

Thus a given contact counts toward the desired total of m with a probability greater than $(k-1)/k$.

The Chernoff bound[36] can be used to bound the value of c_1 which is necessary to ensure that with probability $1-P_f/3$ at least m contacts lead from each given input to usable inputs to the center network. (This can also be modeled as the first passage time in a pure birth process, but such an approach does not lead to a convenient bound for c_1.) The problem is similar to asking how many times a coin must be flipped so that the probability of having had less than r heads is less than some ϵ.

A useful form of the Chernoff bound is

$$\mathrm{Pf}[\mu < m] < e^{-sm}\, g_\mu(s) \qquad \text{for all } s < 0, \tag{69}$$

where μ is the number of contacts going to available inputs, and $g_\mu(s)$ is the s transform of the probability density function of μ. Now μ can be viewed as the sum of c zero-one independent random variables which can take on the value one with probability $(k-1)/k$. Then

$$g_\mu(s) = [\Pr(0)\, e^{0\cdot s} + \Pr(1)\, e^{1\cdot s}]^c$$

$$= \left[\frac{1}{k} + \frac{k-1}{k}\, e^s\right]^c. \tag{70}$$

Therefore

$$\Pr[\mu < m] \leq e^{-sm}\left[\frac{1}{k} + \frac{k-1}{k}\, e^s\right]^c \qquad \text{for } s < 0. \tag{71}$$

Let

$$P = e^{-sm}\left[\frac{1}{k} + \frac{k-1}{k}\, e^s\right]^c. \tag{72}$$

Now the object is to solve for c_1 as a function of P and m

$$\log P = -sm + c_1 \log\left[\frac{1}{k} + \frac{k-1}{k}\, e^s\right]$$

$$c_1 < \frac{\log P + sm}{\log\left[\frac{1}{k} + \frac{k-1}{k}\, e^s\right]} \tag{73}$$

and thus finding the minimal value for c_1, or equivalently by minimizing Eq. 73.

Neither approach leads to a closed-form solution. It can be shown, however, that

$$s = \log \frac{1}{k}$$

$$= -\log k \tag{75}$$

is a good approximation for the optimal value of s, especially for large m. The fact that this is only an approximation is not significant, since the Chernoff bound is good for <u>all</u> negative values of s.

Now substituting this value of s in Eq. 75 gives

$$c_1 < \frac{\log P + m \log \frac{1}{k}}{\log \left[\frac{1}{k} + \frac{k-1}{k}\left(\frac{1}{k}\right)\right]}$$

$$< \frac{\log P + m \log 1/k}{\log 2/k}$$

$$< \frac{\log P}{\log 2/k} + m\left(\frac{\log k}{\log k - \log 2}\right)$$

$$< \frac{\log 1/P}{\log k/2} + m\left(\frac{\log k}{\log k - \log 2}\right). \tag{76}$$

This seems intuitively correct, since it approaches m for large P and large m, and slowly becomes infinite for very small P.

The total number of contacts required for the network in Fig. 18 is

$$n[2c_1 + jc_o k]. \tag{77}$$

Letting the overall blocking probability P_f be distributed as in Eq. 64 yields

$$x(n) < n\left[\frac{\log 3/P_f}{\log k/2} + \frac{\log 3/P_f}{\log 2}\left(\frac{\log k}{\log k - \log 2}\right)\right]$$
$$+ n\left[\frac{\log kn/2c_o}{\log\left[2\left(1 - \left(\frac{k+1}{k}\right)\right)^c\right]} + 2\right] \tag{78}$$

which can then be minimized as a function of k and c. As before, the important point is the order of growth, which in this case is

$$x(n) = 0(n \log n) + 0(n \log 1/P_f) \tag{79}$$

which proves Theorem 5.

4.6 SUMMARY

Theorem 4 shows that the requirement of small blocking could be met with a number of contacts which is the informational minimum multiplied by a constant that is a function of the desired blocking probability. Theorem 5 shows that the requirement of some arbitrarily small blocking probability only requires an additive increase in the number of contacts required so that for a large number of inputs and fixed blocking probability the requirement of the informational bound predominates. For this second class of networks the term containing the variation with blocking was shown to be $0(n \log 1/\epsilon)$. The results of Lotze[31] suggest that a lower order of growth is possible for this term. However, it appears that the bottleneck phenomenon mentioned in section 4.5 may limit the average blocking probability of an ensemble to the order of growth which is shown. It may be possible to show lower orders with different approaches.

Perhaps as significant as these results are the methods which are used. Apparently, this is the first use of ensembles to show results concerning connecting networks. It is hoped that this technique, which has long been used in coding theory, will now be applied to other connecting-network problems.

V. TIME-DIVISION SWITCHING NETWORK

We have shown that connecting networks with finite blocking can be built with a complexity that has the same order of growth as the theoretical minimum. We shall now study a related type of network that is used to switch time-multiplexed signals. It is shown that a time-slot interchanger, a unit which permutes the time order of a sequence of signals, can be constructed with a complexity which is proportional to m log m, where m is the number of signals multiplexed on a line. As in the case discussed in Section IV, this is the same order of growth as the theoretical minimum. The results in Section IV were obtained from an existence proof based on the average blocking of an ensemble, whereas those that will now be discussed are based on an explicit construction which is given.

5.1 INTRODUCTION TO TIME-DIVISION SWITCHING

In all networks discussed thus far, each input carried signals from only one source. With current technology, it is often desirable to multiplex several sources together on one line and to build switching and transmission systems that can handle directly such multiplexed lines. Such multiplexing can be done by multiplying each signal by a waveform from an orthogonal set $\{\phi_i(t)\}$ of waveforms and then by adding the products as shown in Fig. 19. Demultiplexing can be accomplished with a similar process.

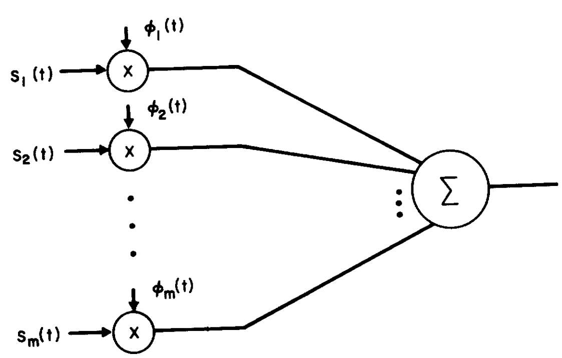


Fig. 19.
Orthogonal multiplex system.

Almost all intercity telecommunication in the United States is now handled by frequency-multiplexed transmission systems. Because of the expense of variable frequency shifting equipment, direct switching of these frequency-multiplexed signals has never been practical; consequently, signals are always demultiplexed to baseband before they are switched.

Recent technology has led to a rapid increase in the use of time-multiplexing for transmission equipment. Time-division switching has been used mainly in small, specialized systems. It is reasonable to expect, however, that the use of time-division multiplexed transmission will increase rapidly as large-scale integration costs decrease and that it will become a major factor in the national network.

In general a time-division switching network has several time-multiplexed inputs and outputs. The task of the network is to transfer signals from each time slot on each input line to an appropriate time slot on the desired output line. Thus signals must be switched in both space and time. It has been shown[35] that networks that do such switching have at least one equivalent pure space-division network with the same traffic properties.

Inose[38] has published a survey of time-division switching techniques which shows that time-slot interchangers (TSI), devices that permute the relative order of signals, are basic components in a wide variety of systems. The rest of this section will be devoted to discussing efficient techniques for the implementation of TSI's.

5.2 EFFICIENT IMPLEMENTATION OF TIME-SLOT INTERCHANGERS

Time-slot interchangers (TSI's) are basic components in many different types of time-division switching systems. A TSI is equivalent[39] to an $m \times m$ square switch with m^2 crosspoints. The usual structure for a TSI is an m-bit shift register with a tap at each stage and a local memory with m words of log m bits each. Each word corresponds to a time slot, and its contents is used to select the proper tap to be used for the signal present in that time slot. In particular, the contents contains the difference, modulo m, of the original time-slot number and the desired time-slot number. The TSI then delays each incoming signal for an appropriate amount of time.

A serious problem with such a scheme is that direct implementation requires a gate with a fanout of m which is impractical for reasonable values of m and high speeds. Alternatively, a TSI could be built as shown in Fig. 20, in which case a binary tree with log m layers and $m-1$ gates would be used. It will now be shown that TSI's can be constructed using only $0(\log m)$ gates at the expense of using more shift registers. Since shift registers are less expensive than gates in some technologies, this can be a desirable tradeoff.

The first type of TSI that will be discussed is based upon a rearrangeable network of β elements which was discussed in section 4.1. It is necessary to redraw Fig. 12 slightly to get a modified form of the recursive definition. This change is shown in Fig. 21 and does not change the properties of the network, since it involves only changing the order of the inputs to the nonblocking center networks, but it does simplify later steps.

The resulting network for $n=8$ is shown in Fig. 22. In order to draw a time-division equivalent a decision must be made about how to transform links in the space-division network into corresponding time slots in the time-division network. The transformation used is shown in Fig. 23 for the network of Fig. 22. The vertical dimension shows the correspondence between time position and space position in the original network. For reference, the "switches" are given the same numbers as in Fig. 22 to which Fig. 23 is isomorphic, as can be shown by simply redrawing the figure. Note that in a given column the top output of each β element is delayed, or slipped, by d units, while the bottom output of each element is either not delayed, or delayed by 2d. Delaying every-

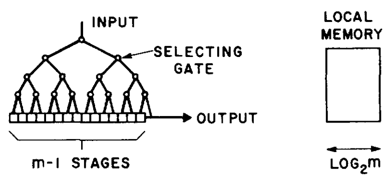


Fig. 20.
Direct implementation of a TSI.

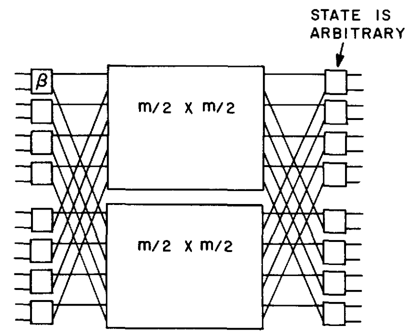


Fig. 21.
Recursive derivation of a rearrangeable network of β elements.

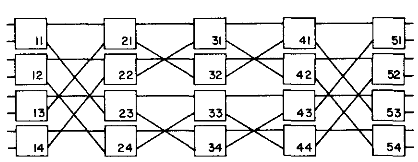


Fig. 22.
Rearrangeable network of β elements (m=8).

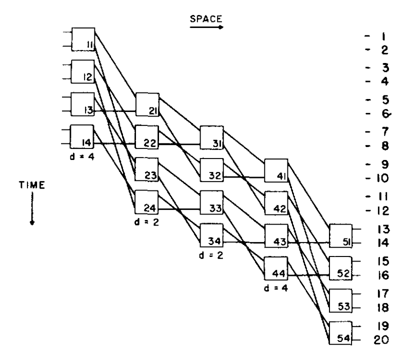


Fig. 23.
Space-time transformation of Fig. 22.

thing is necessary at each stage because otherwise a slip going "up" would correspond to advancing a signal in time (an upward slip could be implemented by delaying signals by almost a frame, but this uses more delay elements), which is an unrealizable situation.

The TSI is built with several units, S_i, in series as shown in Fig. 24. The general structure of S_i is shown in Fig. 25. Successive pairs of numbers are placed in two 1-bit registers according to information stored in the local memory. This operation requires m/2 bits of control information. After the two registers are filled, their contents are shifted to the right to make way for more signals. The output of the top register is always delayed by d while the output of the bottom register is either delayed by 2d or not delayed at all. This corresponds directly to the structure of Fig. 23. The formulas for d are

$$d = 2^{\log_2 m - i} \qquad i = 1, \ldots, \log_2 m - 1$$

$$d = 2^{i - \log_2 m + 1} \qquad i = \log_2 m, \ldots 2 \log_2 m - 1.$$

A switch determines whether or not the output of the bottom register is delayed by 2d. This switch can be controlled by a systemwide square wave with period d for each stage so it requires no local memory.

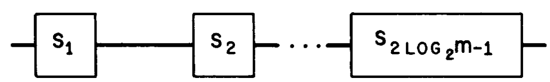


Fig. 24. General form of TSI based on a rearrangeable network of β elements (m=8).

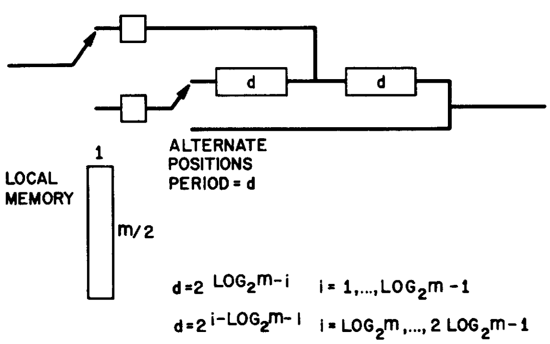


Fig. 25.
Construction of S_i.

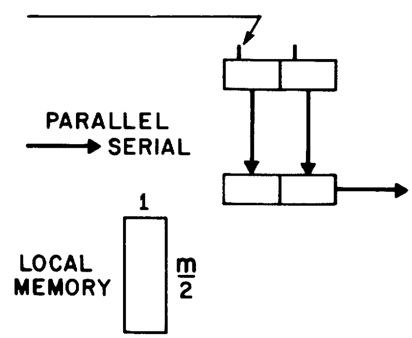


Fig. 26.
Alternate construction for $S_{2 \log_2 m-1}$.

An alternate form can be used for the last stage. This is shown in Fig. 26, and eliminates some of the delays which would serve no useful purpose in the last stage.

The number of components needed for this type of TSI is shown in the second row in Table 4 and in Table 5 for the case m = 128. The number of switches has been reduced greatly at the expense of using more bits of delay. The maximum number of layers of switches which a signal must propagate through asynchronously has been reduced from log m to one, while the maximum fanout has remained two. The minimum delay for a signal to be switched has been made nonzero by this approach, but this is not important in most applications because it is comparable to transmission and coding delays.

The contents of the local memories of the TSI is no longer as easily derived from the desired permutation. A processor is necessary to calculate the local memory state contents, given the desired permutation. It appears that this computation has a complexity which is proportional to $m(\log m)^2$ (see section 1.2). If a system has several

Table 4. Comparison of methods.

TSI Type	Local Memory Bits	Delay Bits	Number of Variable Switches	Number of Alternating Switches	Minimum Delay	Computational Complexity
Direct (Fig. 20)	$m \log_2 m$	$m - 1$	$m - 1$	0	0	$0(m)$
Rearrangeable β Element (Fig. 24)	$(m \log_2 m - m/2)$	$(4 \log_2 m + 4m - 4)$	$(2 \log_2 m - 1)$	$(2 \log_2 m - 2)$	$2m - 4$	$0(m(\log m)^2)$
Sort-Merge (Fig. 29)	$m \log_2 m$	$4m - 8$	$\log_2 m$	$\log_2 m - 1$	m	$0(m \log m)$

Table 5. Comparison of methods ($m = 128$).

TSI Type	Local Memory Bits	Delay Bits	Number of Variable Switches	Number of Alternating Switches	Minimum Delay
Direct (Fig. 20)	896	127	127	0	0
Rearrangeable β Element (Fig. 24)	832	536	13	12	252
Sort-Merge (Fig. 29)	832	504	7	6	128

TSI's, this processor could be shared, since it is only used when the desired permutation is to be changed.

5.3 ANOTHER TYPE OF TIME-SLOT INTERCHANGER

Sorting is similar to switching signals, in that the result is a permutation of the inputs. We shall now derive a TSI that is based on a well-known software sorting algorithm, and has a complexity growth similar to that of the TSI derived above. The significance of this TSI is that it requires slightly fewer components and it illustrates the applicability of sorting techniques to a class of connecting networks.

The contents of the local memories for this TSI are the binary decisions obtained in the process of sorting the input destination list using the tape-merge algorithm. This algorithm is illustrated in Fig. 27. In this case the input is a list of 16 numbers. Pairs of numbers are <u>sorted</u> to give 8 lists a_i of 2 numbers each. These lists are then <u>merged</u> pairwise to form 4 b_i lists of 4 numbers. These lists are again merged pairwise and so on until only one ordered list of all of the inputs is formed, thereby completing the algorithm.

Figure 28 shows one implementation of a TSI based on the tape-merge algorithm. In the first stage pairs of signals are switched as their destinations would be sorted. Alternate pairs go into the bottom shift registers, so at the end of a frame the top shift register contains the pairs a_1, a_3, and so on, while the bottom shift register contains a_2, a_4, and so on. This "sorting" requires m/2 bits of information in a local memory. After the first-stage shift register is full, its ordered lists are "merged" by a switch at its output. Signals going to low number time slots are dumped first at this merging point. The merging switch is controlled by an m-bit local memory which tells it which shift register to accept a signal from. Then the merged lists of size 4 are put into a pair of shift registers in the second stage. The first set of 4 which goes into the top register contains b_1 and b_3, while the bottom one contains b_2 and b_4. The switch that selects the register to be used can be controlled by a system clock, since it is always the same pattern. Thus no local memory is needed. This sequence of operations is then repeated until one merged list of the signals is formed.

A practical TSI based on tape-merge with pipelining is shown in Fig. 29. Here stage 1 comprises four 2-bit registers. Pairs of input signals are 'sorted' into r_1. When r_1 is full its contents is parallel-transferred to r_2. When both r_1 and r_2 are full the contents of r_1 is transferred to r_4, and the contents of r_2 is transferred to r_3. As before this operation requires m/2 bits of control memory. The contents of r_3 and r_4 are merged under the control of an m-bit local memory. The resulting sequence is sent to stage 2 where alternating sequences of 4 signals go into the top and bottom registers. When both registers are full their contents are parallel-transferred to outer registers. These outer registers are then merged under the control of an m-bit memory and so on. A total of $\log_2 m - 1$ merge stages is needed. The total amount of memory used is $m \times (\log_2 m-1) + m/2$ or $m \log_2 m - m/2$.

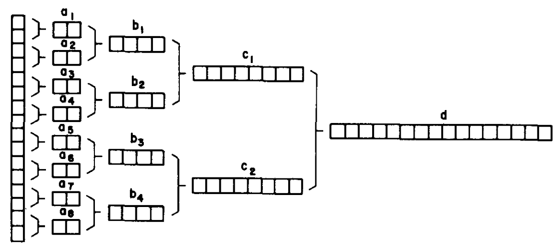


Fig. 27. Example of binary sort-merge (N=16).

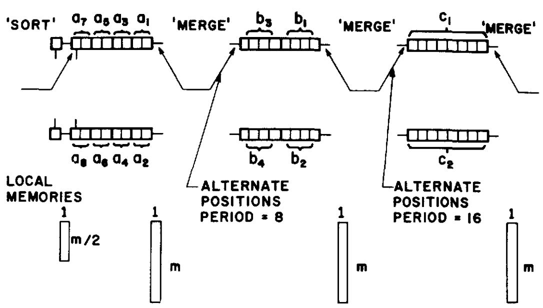


Fig. 28.

Basic TSI controlled by the tape-merge algorithm.

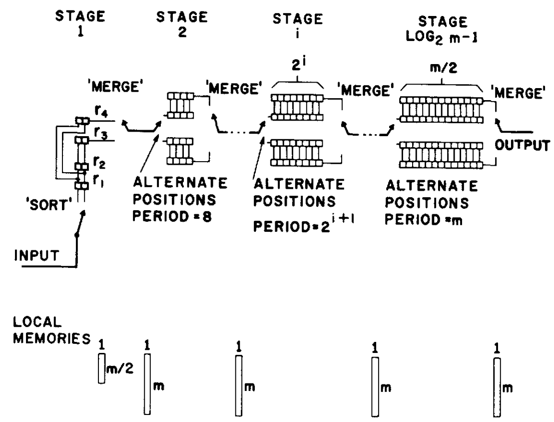


Fig. 29

TSI based on the tape-merge algorithm with pipelining.

The third rows of Tables 4 and 5 indicate the complexity of this type of TSI as compared with the two previously discussed types. The amount of local memory needed is the same as for the element TSI, but fewer switches and bits of delay are needed. Since there are fewer bits of delay, the minimum delay is also decreased, but the delays involved are insignificant from a systems viewpoint. The amount of computation involved is 0(n log n) because it can be shown that this is the maximum number of comparisons that must be made to sort the destination list for the inputs. The average number of computations necessary is also of the same order (10).

In summary, two new types of TSI's have been derived whose complexity is proportional to the informational minimum. These TSI's have less local memory than the direct implementation shown in Fig. 29 and also avoid the problem with propagation delay that the direct approach has in trees of gates. Also, both have a lower order of growth of the number of switches used than the direct approach. In those technologies such as magnetic bubbles or charge-coupled logic where bits of delay are very inexpensive these new approaches are economically desirable.

VI. CONCLUSION

6.1 SUMMARY OF RESULTS

In this work the techniques of information theory have been applied to connecting network problems. Not only does this approach give insight into the problems associated with connecting networks but it also yields several new results.

There are three significant results in this research.

1. The basic sort-merge structure does not of itself limit the efficiency of sorting networks (section 2.2).

2. There exist connecting networks with blocking probability whose order of growth is $0(n \log n) + 0(n \log 1/\epsilon)$ for arbitrary n and ϵ. In Theorem 4 it is stated that this result is analogous to C. E. Shannon's coding theorem for memoryless channels (section 4.3).

3. Time-Slot interchangers can be built with an order of complexity which grows in proportion to the informational minimum (section 5.2).

The approach that was used is perhaps as significant as the results themselves. For example, the first result is obtained by extending the concepts of information and entropy to network structures and to the tasks that they are asked to perform. The second result comes from defining an ensemble of networks and calculating the average performance, an approach that has been used previously in coding theory.

Two basic conclusions can be reached: (i) connecting networks can be built with complexities proportional to the informational minimum, and (ii) the techniques of information theory have been useful in dealing with these problems and may be serviceable for still unsolved problems.

6.2 PROBLEMS FOR FUTURE WORK

The field of connecting networks is full of unsolved problems. Even the types of networks used in telecommunication are not well understood theoretically.

One of the most obvious problems is the minimum order of growth that is possible for sorting networks. It is clear that such networks require $n \log n$ comparators, but the best known general constructions require $0(n(\log n)^2)$ comparators. Perhaps a key to the difficulty of this problem is that D. E. Knuth assigns it complexity "M50", his most complex classification. A novel approach to this problem, suggested by Peter Elias, would be to use an ensemble of networks of comparators and calculate the probability that a pair of outputs is in correct order. This would give bounds for networks which sort with high probability.

Strictly nonblocking networks pose a similar problem concerning the minimum possible order of growth of complexity. A novel approach would be to view each state of the network as a code word with each bit corresponding to a contact in the network. The strictly nonblocking condition then means that incremental changes in the desired permutation must have equivalent code words whose Hamming distance is the number of stages of the network. Then the problem becomes one of the code-word distance.

It appears that the minimum computational complexity required to calculate a function from the set of all permutations to the set of contact states of a network that implements that permutation is $0(n(\log n)^2)$. This suspicion is based on the number of contacts apparently required for a sorting network and the complexity of existing algorithms for determining the state of a rearrangeable network. While this may not be correct, it is surely an interesting problem for investigation.

More work remains to be done in the area of bounding the growth of network complexity. The results in Section IV give the order of growth, but the coefficient of the leading term must still be found. Similarly, the work of D. G. Cantor[16] could be extended to find a smaller coefficient for the number of contacts required in a strictly nonblocking network.

Acknowledgment

I would like to thank Professor Peter Elias for his guidance throughout my thesis research. His confidence in me and insight into this work were invaluable resources.

Professor C. L. Liu and Professor Jerome Saltzer served as thesis readers and I gratefully acknowledge their helpful comments.

I wish to thank the Electronic Systems Laboratory, M. I. T., and Project MAC for providing research assistantships during my first year of graduate work. The Research Laboratory of Electronics, M.I.T., provided a pleasant atmosphere for this research.

Support for this work was received from Bell Telephone Laboratories, Incorporated. In particular, I am grateful to W. H. Ninke, H. S. McDonald, and E. E. David, Jr., for arranging it and showing a continued interest in this area of research.

References

1. C. E. Shannon, "Memory Requirements in a Telephone Exchange," Bell System Tech. J. 29, 343-349 (1950).

2. W. Feller, An Introduction to Probability Theory and Its Applications, Vol. I (John Wiley and Sons, Inc., New York, 3rd ed., 1968).

3. G. T. Baker, "An Application of the Theory of Information," A.T.E. J., Vol. 9, No. 3, pp. 158-162, 1953.

4. R. Syski, "Analogies between Congestion and Communications Theory," A.T.E. J., Vol. 11, No. 4, pp. 221-243, 1955.

5. P. Elias, "Permutations, the Merge-Sort Algorithm, and Connecting Networks" (submitted to Information and Control).

6. V. E. Beneš, Mathematical Theory of Connecting Networks and Telephone Traffic (Academic Press, Inc., New York, 1965).

7. V. E. Beneš, "Square Switch Distribution Network Employing Minimum Number of Crosspoints," U.S. Patent 3,358,269 (1967).

8. L. J. Goldstein and S. W. Leibholz, "On the Synthesis of Signal Switching Networks with Transient Blocking," IEEE Trans. on Electronic Computers, Vol. EC-16, No. 5, pp. 637-641, 1967.

9. A. E. Joel, Jr., "On Permutation Switching Networks," Bell System Tech. J. 47, 813-822 (1968).

10a. T. Kamae, "On a Rearrangeable Three-Stage Switching Network," Electronics and Communications in Japan, No. 51-A, pp. 6-17, 1968.

10b. W. Keister (unpublished work, 1967).

11. M. C. Paull, "Reswitching of Connecting Networks," Bell System Tech. J. 43, 833-855 (1962).

12. G. Nakamura, "The Number of Reswitching for a Certain Three-Stage Connecting Network," J. Inst. Elect. Commun. Engrs., Japan, Vol. 50, No. 8, pp. 1419-1425, August 1967.

13. L. A. Bassalygo, I. I. Grushko, and V. I. Neyman, "The Number of Reswitchings for a 3-Stage Connecting Network," paper presented at the 6th International Teletraffic Congress, 1970.

14. A. Waksman, "A Permutation Network," J. Assoc. Computing Machinery, Vol. 15, pp. 159-163, 1968.

15. C. Clos, "A Study of Non-blocking Switching Networks," Bell System Tech. J. 32, 406-424 (1953).

16. D. G. Cantor, "On Non-blocking Switching Networks" [to appear in Networks (a new journal not yet published)].

17. K. E. Batcher, "Sorting Networks and Their Applications," American Federation of Information Processing Societies Conference, Vol. 32, 1968, Proc. Spring Joint Computer Conference, Atlantic City, New Jersey, April 30-May 2, 1968, pp. 307-314.

18. R.C. Bose and R.J. Nelson, "A Sorting Problem," J.Assoc.Computing Machinery, Vol. 9, pp. 282-296, 1962.

19. D. E. Knuth, The Art of Computer Programming, Vol. III (to appear as Section 5.3.4).

20. H. S. Stone, "Parallel Processing with the Perfect Shuffle," IEEE Trans. on Computers, Vol. C-20, No. 2, pp. 153-160, 1970.

21. S. Y. Levy and M. C. Paull, "An Algebra with Application to Sorting Algorithms," Proc. 1969 Princeton Conference on Information Science and Systems, pp. 286-251.

22. A. N. Kolmogorov, "Three Approaches to the Quantative Definition of Information," Problems Inform. Transm., Vol. 1, pp. 286-251, 1965.

23. H. Lorin, "A Guided Bibliography to Sorting," IBM Syst. J., Vol. 3, No. 3, pp. 244-254 (1971).

24. D. G. O'Connor and R. J. Nelson, "Sorting System with Two-line Sorting Switch," U.S. Patent 3,311,892 (September 30, 1963). [Continuation of Application Serial No. 641502 (February 21, 1957).]

25. R. Syski, Introduction to Congestive Theory in Telephone Switching (London, 1960).

26. C. Y. Lee, "Analysis of Switching Networks," Bell System Tech. J. 34, 1287 (1955).

27. P. LeGall, "Méthodes de Calcul de l'Encombrement dans le Système Téléphonique Automatique à Marquage," Ann. Telecom., Vol. 12, pp. 374-386, 1957.

28. C. Jacobaeus, "A Study on Congestion in Link Systems," Ericsson Technics, No. 48, pp. 1-68, 1950.

29. R. F. Grantjes and N. R. Sinowitz, "NEASIM – A General Purpose Computer Simulator for Load-Loss Analysis of Multistage Central Office Switching Networks," Bell System Tech. J. 43, 965-1004 (1964).

30. N. Ikeno, "A Limit on Crosspoint Number," Proc. 1959 International Conference on Circuit and Information Theory (IRE Trans. on Information Theory, Supplement, March 1959, pp. 187-196).

31. A. Lotze, "Optimum Link System," Proc. 5th International Teletraffic Conference, 1967, pp. 242-251.

32. A. Feiner and J. G. Kappel, "A Method of Deriving Efficient Switching Network Configurations," Proc. 1970 National Electronics Conference, pp. 818-823.

33. C. E. Shannon and W. Weaver, The Mathematical Theory of Communication (The University of Illinois Press, Urbana, 1949).

34. J. M. Wozencraft and I. M. Jacobs, Principles of Communication Engineering (John Wiley and Sons, Inc., New York, London, Sydney, 1965).

35. P. Elias, "Coding for Noisy Channels," IRE Convention Record, Vol. 3, Part 4, pp. 37-46, 1955.

36. R. G. Gallager, Information Theory and Reliable Communication (John Wiley and Sons, Inc., New York, 1968).

37. C. L. Liu, Private communication, M.I.T., 1971.

38. H. Inose, "Survey of PCM Switch Frame Types," paper presented at the 6th International Teletraffic Conference, 1970.

39. M. J. Marcus, "Space-Time Equivalents in Connecting Networks," Paper No. 70-CP-368-COM, Proc. 1970 International Conference on Communications, San Francisco, California, June 8-10, 1970, Sec. 35, pp. 25-31.

JOINT SERVICES ELECTRONICS PROGRAM
REPORTS DISTRIBUTION LIST

Department of Defense

Assistant Director (Research)
Office of Director of Defense, Research
& Engineering
The Pentagon, Rm 3C128
Washington, D.C. 20301

Technical Library
DDR&E
Room 3C-122, The Pentagon
Washington, D.C. 20301

Director For Materials Sciences
Advanced Research Projects Agency
1400 Wilson Boulevard
Arlington, Virginia 22209

Chief, R&D Division (340)
Defense Communications Agency
Washington, D.C. 20305

Defense Documentation Center
Attn: DDC-TCA
Cameron Station
Alexandria, Virginia 22314

Dr. Alvin D. Schnitzler
Institute For Defense Analyses
Science and Technology Division
400 Army-Navy Drive
Arlington, Virginia 22202

LTC Norman D. Jorstad
Weapons Systems Evaluation Group
400 Army-Navy Drive
Arlington, Virginia 22202

Central Intelligence Agency
Attn: CRS/ADD/PUBLICATIONS
Washington, D.C. 20505

Department of the Air Force

Hq USAF (AF/RDPE)
Washington, D.C. 20330

Hq USAF/RDPS
Washington, D.C. 20330

Hq USAF (RDSD)
The Pentagon
Washington, D.C. 20330

Colonel E. P. Gaines, Jr.
ESD (MCD)
L. G. Hanscom Field
Bedford, Massachusetts 01730

Dr L. A. Wood, Director
Electronic and Solid State Sciences
Air Force Office of Scientific Research
1400 Wilson Boulevard
Arlington, Virginia 22209

Dr Harvey E. Savely, Director
Life Sciences
Air Force Office of Scientific Research
1400 Wilson Boulevard
Arlington, Virginia 22209

AFSC (CCJ/Mr I. R. Mirman)
Andrews Air Force Base,
Washington, D.C. 20331

Rome Air Development Center
Attn: Documents Library (TDLD)
Griffiss Air Force Base, New York 13440

Mr H. E. Webb, Jr. (ISCP)
Rome Air Development Center
Griffiss Air Force Base, New York 13440

Dr L. M. Hollingsworth
AFCRL (CA)
L. G. Hanscom Field
Bedford, Massachusetts 01730

Hq ESD (TRI)
L. G. Hanscom Field
Bedford, Massachusetts 01730

Professor R. E. Fontana, Head
Dept of Electrical Engineering
Air Force Institute of Technology
Wright-Patterson Air Force Base,
Ohio 45433

Dr H. V. Noble, AFAL/TE
Chief, Electronics Technology Division
Air Force Avionics Laboratory
Wright-Patterson Air Force Base,
Ohio 45433

Director
Air Force Avionics Laboratory
Wright-Patterson Air Force Base,
Ohio 45433

AFAL/TEA (Mr R. D. Larson)
Wright-Patterson Air Force Base,
Ohio 45433

Director of Faculty Research
Department of the Air Force
U.S. Air Force Academy
Colorado 80840

Mr Jules I. Wittebort
Chief, Electronics Branch
Manufacturing Technology Division
AFAL/LTE
Wright-Patterson Air Force Base,
Ohio 45433

Academy Library (DFSLB)
USAF Academy, Colorado 80840

Director of Aerospace Mechanics Sciences
Frank J. Seiler Research
Laboratory (OAR)
USAF Academy, Colorado 80840

Major Richard J. Gowen
Tenure Associate Professor
Dept of Electrical Engineering
USAF Academy, Colorado 80840

Director, USAF PROJECT RAND
Via: Air Force Liaison Office
The RAND Corporation
Attn: Library D
1700 Main Street
Santa Monica, California 90406

AUL3T-9663
Maxwell Air Force Base, Alabama 36112

AFETR Technical Library
(MU-135)
Patrick Air Force Base, Florida 32925

ADTC (SSLT)
Eglin Air Force Base, Florida 32542

Hq AMD (RDR/Lt Col Godden)
Brooks Air Force Base, Texas 78235

USAFSAM (RAT)
Brooks Air Force Base, Texas 78235

Commanding General
Attn: STEWS-AD-L, Technical Library
White Sands Missile Range,
New Mexico 88002

European Office of Aerospace Research
Technical Information Office
Box 14, FPO New York 09510

VELA Seismological Center
312 Montgomery Street
Alexandria, Virginia 22314

Capt C. E. Baum
AFWL (SRE)
Kirtland Air Force Base,
New Mexico 87117

Biomedical Engineering Branch (SCB)
USAF School of Aerospace Medicine (AFSC)
Department of the Air Force
Brooks Air Force Base, Texas 78235

Department of the Army

Director
Physical & Engineering Sciences Division
3045 Columbia Pike
Arlington, Virginia 22204

Commanding General
U.S. Army Security Agency
Attn: IARD-T
Arlington Hall Station
Arlington, Virginia 22212

Commanding General
U.S. Army Materiel Command
Attn: AMCRD-TP (Dr Zarwyn)
Washington, D.C. 20315

Director
U.S. Army Advanced Materiel
Concepts Agency
2461 Eisenhower Avenue
Alexandria, Virginia 22314

Commanding General
USACDC Institute of Land Combat
Attn: Technical Library, Rm 636
2461 Eisenhower Avenue
Alexandria, Virginia 22314

Mr H. T. Darracott (AMXAM-FT)
U.S. Army Advanced Materiel
Concepts Agency
2461 Eisenhower Avenue
Alexandria, Virginia 22314

Dr Berthold Altmann (AMXDO-TI)
Harry Diamond Laboratories
Connecticut Avenue and
Van Ness Street N. W.
Washington, D.C. 20438

Commanding Officer (AMXRD-BAD)
U.S. Army Ballistic Research Laboratory
Aberdeen Proving Ground
Aberdeen, Maryland 21005

U.S. Army Munitions Command
Attn: Science & Technology Information
Branch, Bldg 59
Picatinny Arsenal, SMUPA-RT-S
Dover, New Jersey 07801

Dr Herman Robl
Deputy Chief Scientist
U.S. Army Research Office (Durham)
Box CM, Duke Station
Durham, North Carolina 27706

Richard O. Ulsh (CRDARD-IP)
U.S. Army Research Office (Durham)
Box CM, Duke Station
Durham, North Carolina 27706

Technical Director (SMUFA-A2000-107-1)
Frankford Arsenal
Philadelphia, Pennsylvania 19137

Redstone Scientific Information Center
Attn: Chief, Document Section
U.S. Army Missile Command
Redstone Arsenal, Alabama 35809

Commanding General
U.S. Army Missile Command
Attn: AMSMI-RR
Redstone Arsenal, Alabama 35809

Commanding General
U.S. Army Strategic Communications
Command
Attn: SCC-ATS (Mr Peter B. Pichetto)
Fort Huachuca, Arizona 85613

Dr Homer F. Priest
Chief, Materials Sciences Division,
Bldg 292
Army Materials & Mechanics Research
Center
Watertown, Massachusetts 02172

Commandant
U.S. Army Air Defense School
Attn: Missile Science Division, C&S Dept
P. O. Box 9390
Fort Bliss, Texas 79916

Commandant
U.S. Army Command and General
Staff College
Attn: Acquisitions, Lib Div
Fort Leavenworth, Kansas 66027

Dr Hans K. Ziegler
Army Member TAC/JSEP (AMSEL-XL-D)
U.S. Army Electronics Command
Fort Monmouth, New Jersey 07703

Mr I. A. Balton, TAC/JSEP
AMSEL-TL-DC (Room 4C-116)
U.S. Army Electronics Command
Fort Monmouth, New Jersey 07703

Director (NV-D)
Night Vision Laboratory, USAECOM
Fort Belvoir, Virginia 22060

Commanding Officer
Atmospheric Sciences Laboratory
U.S. Army Electronics Command
White Sands Missile Range,
New Mexico 88002

Atmospheric Sciences Laboratory
U.S. Army Electronics Command
Attn: AMSEL-BL-DD (Mr Marvin Diamond)
White Sands Missile Range,
New Mexico 88002

Chief
Missile Electronic Warfare Tech
Area (AMSEL-WL-M)
Electronic Warfare Laboratory,
USAECOM
White Sands Missile Range,
New Mexico 88002

Project Manager NAVCON
Attn: AMCPM-NC, Bldg 439
(H. H. Bahr)
Fort Monmouth, New Jersey 07703

Mr A. D. Bedrosian, Rm 26-131
U.S. Army Scientific Liaison Office
Massachusetts Institute of Technology
77 Massachusetts Avenue
Cambridge, Massachusetts 02139

Commanding General
U.S. Army Electronics Command
Fort Monmouth, New Jersey 07703
Attn: AMSEL-RD-O (Dr W. S. McAfee)
GG-DD
XL-D
XL-DT
XL-G (Dr S. Kronenberg)
XL-H (Dr R. G. Buser)
BL-FM-P
CT-D
CT-R
CT-S
CT-L
CT-O
CT-I
CT-A
NL-D (Dr H. Bennett)
NL-A
NL-C
NL-P
NL-P-2
NL-R
NL-S
KL-D
KL-I
KL-E
KL-S
KL-SM
KL-T
VL-D
VL-F
WL-D

Department of the Navy

Director, Electronics Programs
Attn: Code 427
Office of Naval Research
800 North Quincy Street
Arlington, Virginia 22217

Mr Gordon D. Goldstein, Code 437
Information Systems Program
Office of Naval Research
800 North Quincy Street
Arlington, Virginia 22217

Commander
Naval Security Group Command
Naval Security Group Headquarters
Attn: Technical Library (G43)
3801 Nebraska Avenue, N. W.
Washington, D.C. 20390

Director
Naval Research Laboratory
Washington, D.C. 20390
Attn: Code 5200
Mr A. Brodzinsky, Supt,
Electronics Div

Director
Naval Research Laboratory
Attn: Dr H. Rabin, Code 7000
Washington, D.C. 20390

Code 8050
Maury Center Library
Naval Research Laboratory
Washington, D.C. 20390

Dr G. M. R. Winkler
Director, Time Service Division
U.S. Naval Observatory
Washington, D.C. 20390

Naval Air Systems Command
AIR 310 - Research Administrator
Room 424 JP-1
Washington, D.C. 20360

Dr A. L. Slafkosky
Scientific Advisor
Commandant of the Marine Corps (Code AX)
Washington, D.C. 20380

Naval Ship Systems Command
Ship 035
Washington, D.C. 20360

U.S. Naval Weapons Laboratory
Dahlgren, Virginia 22448

Naval Electronic Systems Command
Attn: Code 0311, Rm 7W12, NC #1
Department of the Navy
Washington, D.C. 20360

Commander
U.S. Naval Ordnance Laboratory
Attn: Librarian
White Oak, Maryland 20910

Director
Office of Naval Research
Boston Branch
495 Summer Street
Boston, Massachusetts 02210

Commander (ADL)
Naval Air Development Center
Attn: NADC Library
Johnsville, Warminster,
Pennsylvania, 18974

Commanding Officer
Naval Missile Center
Attn: 5632.2, Technical Library
Point Mugu, California 93042

W. A. Eberspacher, Associate Head
Systems Integration Division, Code 5340A
U.S. Naval Missile Center
Point Mugu, California 93041

Commander
Naval Electronics Laboratory Center
Attn: Library
San Diego, California 92152

Deputy Director and Chief Scientist
Office of Naval Research Branch Office
1031 East Green Street
Pasadena, California 91101

Library (Code 2124)
Naval Postgraduate School
Monterey, California93940

Officer in Charge, New London Lab.
Naval Underwater Systems Center
Attn: Technical Library
New London, Connecticut 06320

Commanding Officer
Naval Avionics Facility
Attn: D/035 Technical Library
Indianapolis, Indiana 46241

Director
Naval Research Laboratory
Attn: Library, Code 2629 (ONRL)
Washington, D.C. 20390

Commanding Officer
Naval Training Device Center
Attn: Technical Library
Orlando, Florida 32813

U.S. Naval Oceanographic Office
Attn: M. Rogofsky, Librarian (Code 1640)
Washington, D.C. 20390

Other Government Agencies

Dr H. Harrison
AEC/NASA Space Nuclear Systems Office
AEC Headquarters (F-309)
Washington, D.C. 20545

NASA Lewis Research Center
Attn: Library
21000 Brookpark Road
Cleveland, Ohio 44135

Los Alamos Scientific Laboratory
Attn: Reports Library
P. O. Box 1663
Los Alamos, New Mexico 87544

Mr M. Zane Thornton
Deputy Director
Center for Computer Sciences and Technology
National Bureau of Standards
U.S. Department of Commerce
Washington D.C. 20234

U.S. Postal Service
Library – Room 6012
12th & Pennsylvania Ave., N. W.
Washington, D.C. 20260

Non-Government Agencies

Director
Research Laboratory of Electronics
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

Mr Jerome Fox, Research Coordinator
Polytechnic Institute of Brooklyn
333 Jay Street
Brooklyn, New York 11201

Director
Columbia Radiation Laboratory
Columbia University
538 West 120th Street
New York, New York 10027

Director
Coordinate Science Laboratory
University of Illinois
Urbana, Illinois 61801

Director
Stanford Electronics Laboratory
Stanford University
Stanford, California 94305

Director
Microwave Laboratory
Stanford University
Stanford, California 94305

Director
Electronics Research Laboratory
University of California
Berkeley, California 94720

Director
Electronics Sciences Laboratory
University of Southern California
Los Angeles, California 90007

Director
Electronics Research Center
The University of Texas at Austin
Engineering-Science Bldg 110
Austin, Texas 78712

Division of Engineering and
Applied Physics
210 Pierce Hall
Harvard University
Cambridge, Massachusetts 02138

Dr G. J. Murphy
The Technological Institute
Northwestern University
Evanston, Illinois 60201

Dr John C. Hancock, Head
School of Electrical Engineering
Purdue University
Lafayette, Indiana 47907

Dept of Electrical Engineering
Texas Technological University
Lubbock, Texas 79409

Aerospace Corporation
P. O. Box 95085
Attn: Library Acquisitions Group
Los Angeles, California 90045

Airborne Instruments Laboratory
Deerpark, New York 11729

The University of Arizona
Department of Electrical Engineering
Tucson, Arizona 85721

Chairman, Electrical Engineering
Arizona State University
Tempe, Arizona 85281

Engineering and Mathematical
Sciences Library
University of California at Los Angeles
405 Hilgred Avenue
Los Angeles, California 90024

Sciences-Engineering Library
University of California
Santa Barbara, California 93106

Professor Nicholas George
California Institute of Technology
Pasadena, California 91109

Aeronautics Library
Graduate Aeronautical Laboratories
California Institute of Technology
1201 E. California Boulevard
Pasadena, California 91109

Hunt Library
Carnegie-Mellon University
Schenley Park
Pittsburgh, Pennsylvania 15213

Dr A. G. Jordan
Head of Dept of Electrical Engineering
Carnegie-Mellon University
Pittsburg, Pennsylvania 15213

Case Western Reserve University
Engineering Division
University Circle
Cleveland, Ohio 44106

Hollander Associates
Attn: Librarian
P. O. Box 2276
Fullerton, California 92633

Walter H. Veazie, Head
Electronic Properties Information Center
Mail Station E-148
Hughes Aircraft Company
Culver City, California 90230

Illinois Institute of Technology
Department of Electrical Engineering
Chicago, Illinois 60616

Government Documents Department
University of Iowa Libraries
Iowa City, Iowa 52240

The Johns Hopkins University
Applied Physics Laboratory
Attn: Document Librarian
8621 Georgia Avenue
Silver Spring, Maryland 20910

Lehigh University
Department of Electrical Engineering
Bethlehem, Pennsylvania 18015

Mr E. K. Peterson
Lenkurt Electric Co. Inc.
1105 Country Road
San Carlos, California 94070

MIT Lincoln Laboratory
Attn: Library A-082
P. O. Box 73
Lexington, Massachusetts 02173

Miss R. Joyce Harman
Project MAC, Room 810
545 Main Street
Cambridge, Massachusetts 02139

Professor R. H. Rediker
Electrical Engineering, Professor
Massachusetts Institute of Technology
Building 13-3050
Cambridge, Massachusetts 02139

Professor Joseph E. Rowe
Chairman, Dept of Electrical Engineering
The University of Michigan
Ann Arbor, Michigan 48104

New York University
Engineering Library
Bronx, New York 10453

Professor James A. Cadzow
Department of Electrical Engineering
State University of New York at Buffalo
Buffalo, New York 14214

Department of Electrical Engineering
Clippinger Laboratory
Ohio University
Athens, Ohio 45701

Raytheon Company
Research Division Library
28 Seyon Street
Waltham, Massachusetts 02154

Rice University
Department of Electrical Engineering
Houston, Texas 77001

Dr Leo Young, Program Manager
Stanford Research Institute
Menlo Park, California 94025

GTE Waltham Research Lab Library
Attn: Documents Librarian
Esther McLaughlin
40 Sylvan Road
Waltham, Massachusetts 02154

Dr W. R. LePage, Chairman
Department of Electrical Engineering
Syracuse University
Syracuse, New York 13210

Dr F. R. Charvat
Union Carbide Corporation
Materials Systems Division
Crystal Products Department
8888 Balboa Avenue
P. O. Box 23017
San Diego, California 92123

Utah State University
Department of Electrical Engineering
Logan, Utah 84321

Research Laboratories for the
Engineering Sciences
School of Engineering and Applied Science
University of Virginia
Charlottesville, Virginia 22903

Yale University
Engineering and Applied Science Library
15 Prospect Street
New Haven, Connecticut 06520

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)* Research Laboratory of Electronics Massachusetts Institute of Technology Cambridge, Massachusetts 02139	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP None

3. REPORT TITLE
New Approaches to the Analysis of Connecting and Sorting Networks

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*
Technical report

5. AUTHOR(S) *(First name, middle initial, last name)*
Michael J. Marcus

6. REPORT DATE March 17, 1972	7a. TOTAL NO. OF PAGES 68	7b. NO. OF REFS 39

8a. CONTRACT OR GRANT NO. DAAB07-71-C-0300 b. PROJECT NO. C8-1-01504-01-C8-CA c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) Technical Report 486 9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)* None

10. DISTRIBUTION STATEMENT
This document has been approved for public release and sale; its distribution is unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY Joint Services Electronics Programs Through U.S. Army Electronics Command

13. ABSTRACT

This report is a study of the application of information theory techniques to the field of connecting networks. It shows that these techniques are useful in this new area, and demonstrates that two types of networks can be constructed with a complexity that is proportional to an informational lower bound which is due to Claude E. Shannon.

The concepts of information and entropy are extended to fit the context of connecting networks. These not only give insight into the problems of these networks, but are used directly to show that the basic sort-merge algorithm does not of itself imply that sorting networks are inefficient.

It is then shown that connecting networks with small blocking probability and time-slot interchangers can be implemented with a complexity that is proportional to the informational minimum. By defining an ensemble of connecting networks, analogous to ensembles used in coding theory, it is shown that connecting networks with n inputs and blocking probability ϵ can be constructed with $0(n \log n) + 0(n \log 1/\epsilon)$ contacts for all n and ϵ. Shannon's results show that the first term in this expression is the minimum possible order of growth for such networks. A similar result for time-slot interchangers is illustrated by giving an explicit construction.

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Information Theory						
Telephony						
Telephone Switching						
Teletraffic Theory						
Time-Division Switching						