THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 57, No. 1, January 1978
Printed in U.S.A.

Reduction of Network States Under Symmetries

By V. E. BENEŠ

(Manuscript received May 24, 1977)

It is a folk-theorem of traffic theory that if all sources have the same stochastic behavior, then symmetries of a telephone connecting network can be used to lump together equivalent states and to reduce the number of equations to be solved for the state probabilities. The structural and algebraic bases of this idea, and its connections to stochastic models, are studied here by means of concepts from lattice theory, group theory, and combinatorics generally. It is shown that when offered traffic is homogeneous and routing is structurally consistent, the state equations for certain natural Markov processes (representing operating telephone networks) can be substantially simplified by restricting attention to "macrostates," defined as the structural equivalence classes of states, of which there are typically many fewer than of states. Reduced state equations are then obtained for general networks under simple Markovian traffic assumptions.

I. INTRODUCTION

Most telephone connecting networks are built in stages of identical units, arranged symmetrically in frames, and joined by symmetric cross-connect fields (Fig. 1). As a result, their structure has so much symmetry that it is possible quickly to identify at least some network states that are "essentially" equivalent in that their combinatorial structure is the same, and that they differ only in point of renaming terminals, switches, links, customers, etc. It has long been known in the informal lore of traffic theory that if the traffic sources at the terminals have the same stochastic behavior, then these symmetries of the network could be used as a basis for lumping together equivalent states and reducing the number of equations to be solved for the state probabilities.

Such a line of thought was first pursued in a formal way by S. P. Lloyd[1] in unpublished work dated September 19, 1955, and we follow it further here. But whereas Lloyd right away considered probability transition rate matrices (for Markov processes) which he assumed admitted symmetry operations, we shall instead first relate the relevant symmetries to the network graph and the semilattice of states, without reference to a probabilistic model, and only later consider a natural traffic model. Our approach remains combinatorial as long as possible, and allows us eventually to include the effects of routing decisions, to connect the reduction ideas with *optimal* routing, and to find that certain natural transition matrices *necessarily* admit the network symmetries. In particular, we show that a traffic model used in previous work[2] has a transition rate matrix that admits the symmetries of the network graph, provided only that the routing matrix used also admits these symmetries.

An additional practical incentive for the present study is the fact that for small networks, such as concentrators, it is often possible to press the advantages gained by reducing the states to their equivalence classes, and to solve completely the problem of optimal routing. One can then devise a circuit or a finite-state machine to mechanize the optimal routing policy, as has been done by A. F. Bulfer for the RTA concentrator structure.[3,4] Indeed, historically, it was our attempts to *prove* some of Bulfer's surprising empirical results that led to a realization that a thorough study of structural equivalence of states was valuable and necessary.[5,6]

II. SUMMARY

Various preliminaries are in Sections III and IV: a model for discussing connecting networks, with an account of the role of symmetry. Prior results of S. P. Lloyd on the use of groups to reduce state equations are described in Section V, and there is a heuristic discussion of some necessary conditions on such groups in Section VI: they should be groups of automorphisms of the semilattice of states that preserve the relation of having the same calls up. The symmetry group G_n of the network graph appears in Section VII; it is used in Section VIII to define structural equivalence of states and its associated group G_ν, and in Sections IX and X to define a natural homomorphic image G_η of G_n into G_ν which usually gives a more economical description of equivalence in terms of a group than does G_ν. The semilattice of states induces a partial ordering of the reduced states, described in Section XI.

The next four sections are devoted to detailed calculations of symmetry groups G_n, G_ν, and G_η for various well-known networks: individual 2×2 switches in Section XII, a random slip concentrator in Section XIII, a small three-stage network in Section XIV, and general crossbar networks in Section XV, including frames, cascades of stages, and the No. 5 crossbar type network. The final four sections consider a simple stochastic model for traffic in a network with routing decisions, and show how the reduced state equations are derived in this setting.

It can be concluded that symmetry groups afford a precise definition of structural equivalence for network states; this equivalence in turn allows a substantial reduction in the number of equilibrium equations for state probabilities in suitable stochastic models, thus extending the range of computable examples. For modest networks the reduction method can be used to perform optimal routing in explicit ways.[5,6]

III. PRELIMINARIES

We shall use a model for the structural and combinatorial aspects of a connecting network. This model arises by considering the network structure to be given by a graph G whose vertices are the terminals of the network, and whose edges represent crosspoints between terminals by pairs, with some of the terminals designated as inlets or outlets. Calls in the network are described by paths on G from an inlet to an outlet. Thus a connecting network ν is a quadruple $\nu = (G,I,\Omega,S)$ where G is a graph depicting network structure, I is the set of vertices of G which are inlets, Ω is the set of outlets, and S is the set of permitted or physically meaningful states. It is possible that $I = \Omega$ (one-sided network), that $I \cap \Omega = \phi$ (two-sided network), or that some intermediate condition obtain, depending on the "community of interest" aspects of the network ν. Variables w, x, y, and z at the end of the alphabet denote states, while u and v denote a typical inlet and a typical outlet, respectively.

A possible state x can be thought of as a set of disjoint chains on G, each joining I to Ω. Not every such set of chains need represent a state in S: wastefully circuitous chains may be excluded from S. The set S is partially ordered by inclusions $\leq$, where $x \leq y$ means that state x can be obtained from state y by removing zero or more calls. It is reasonable that if y is a state and x results from y by removal of some chains then x should be a state too; i.e., S should be closed under "hangups." It can be seen from this requirement that the set S of permitted states has the structure of a semilattice, that is, a partially ordered system whose order relation is definable in terms of a binary operation $\cap$ that is idempotent, commutative, and associative, by the formula $x \leq y$ iff $x = x \cap y$. Here for $x \cap y$ we can simply use literal set intersection: $x \cap y$ is exactly the state consisting of those calls and their respective routes which are common to x and y.

An *assignment* is a specification of what inlets are to be connected to what outlets. The set A of assignments can be represented as the set of all fixed-point-free correspondences from subsets of I to Ω. The assignments form a semilattice in the same way that the states do, and A is related to S as follows: call two states x,y in S equivalent as to assignment, written $x \sim y$, iff all and only those inlets $u \epsilon I$ are connected in x to outlets $v \epsilon \Omega$ which are connected to the same v in y, though possibly by different routes; the realizable assignments can then be identified with the equivalence classes of states under $\sim$, and there is a natural map $\gamma: S \rightarrow A$, the projection that carries each state x into the assignment $\gamma(x)$ it realizes, i.e., the equivalence class it belongs to under $\sim$.

With x and y states such that $x \geq y$, it is convenient to use x-y to mean the state resulting from x by removing from x all the calls in y. Similarly, with a and b assignments such that $a \geq b$, we use a-b to mean the assignment resulting from a by dropping all the connections intended in b. Note that here x-y, a-b have their usual set-theoretic meaning.

It can now be seen that the map γ is a semilattice homomorphism of S into A, with the properties:

$$x \geq y \Rightarrow \gamma(x) \geq \gamma(y)$$

$$x \geq y \Rightarrow \gamma(x - y) = \gamma(x) - \gamma(y)$$

$$\gamma(x \cap y) \leq \gamma(x) \cap \gamma(y)$$

$$\gamma(x) = \phi \Rightarrow x = 0 = \text{zero state, with no calls up}$$

Not every assignment need be realizable by some state of S. Indeed, it is common for practical networks to realize only a vanishing fraction of the possible assignments, and the networks that do realize every assignment, the so-called *rearrangeable* networks, have been the objects of substantial theoretical study. Thus the image set $\gamma(S)$ of realizable assignments is typically much smaller than the set A it is embedded in. A *unit* assignment is, naturally, one that assigns exactly one outlet to some inlet, and it corresponds to having just one call in progress. It is convenient to identify calls c and unit assignments, and to write $\gamma(x) \cup c$ for the larger assignment consisting of $\gamma(x)$ and the call c together, with the understanding of course that c is "new in x" in the sense that neither of its terminals is busy in x.

We denote by A_x the set of states that are immediately above x in the partial ordering $\leq$ of S, and by B_x the set of those that are immediately below. Thus

A_x = {states reachable from x by adding a call}

B_x = {states reachable from x by a hangup}

For c new in x, let $A_{cx} = A_x \cap \gamma^{-1}[\gamma(x) \cup c]$; A_{cx} is the subset of states of A_x that could result from x by putting up the call c, because $\gamma^{-1}\gamma(y)$ is precisely the equivalence class of y under $\sim$. If A_{cx} is empty then we say c is *blocked* in x: there is no $y \epsilon A_x$ which realizes the larger assignment $\gamma(x) \cup c$. It can be seen that with F_x the set of new calls of x that are not blocked, the family $\{A_{cx}, c \epsilon F_x\}$ forms the partition of A_x induced by the equivalence $\sim$.

IV. HOW DOES SYMMETRY HELP?

Symmetry in the structure of a connecting network, together with the theoretically convenient (but in practice false) assumption of homogeneous or interchangeable traffic sources, leads to simplifications in calculating state probabilities, loss, carried load, and other traffic quantities. In most cases the simplification occurs because network states that have the same structure to within renaming of customers, links, and switches are in a definite sense equivalent, and because if traffic sources are interchangeable, such equivalent states can turn out to have the same equilibrium or transition probabilities. Whether they do or not depends on the rest of the traffic model, especially on the rule used to make routing decisions: roughly speaking, if the rule is *consistent* in that it opts for analogous routes for analogous calls in equivalent states, then equivalent states will (or at least can) have the same probabilities. In such cases the state probabilities can be calculated from those of the equivalence classes, of which there are usually many fewer, by considering, in place of the original microscopic stochastic process on the set of states, a macroscopic one taking values on the equivalence classes.

Our problem in this paper is to make all these notions, especially that of "equivalent" states, as precise as possible in the general network setting. In view of the central role of symmetry it is natural to expect that the equivalence idea be expressed mathematically by means of group theory, and particularly, in terms of the symmetry group of the graph G depicting network structure. Applications to optimal routing in networks and concentrators will appear in later work.[5,6] These applications are considerably complicated by the following "problem of refusals": It turns out that analytical methods for finding optimal routing rules are greatly simplified if, as operator of the network, the telephone company is allowed the option of refusing to complete an unblocked call if it thinks that this denial of service will improve performance according to some criterion of interest; with this added option the task of finding out when to decline unblocked calls is part of the routing problem, a part which it turns out is usually much harder than actually choosing the best route if the call is to go in; however, it is often possible to solve the routing problem *up to refusals,* that is, to specify optimal routes for calls that might go in without ruling on whether they go in or not.

V. PRIOR RESULTS OF S. P. LLOYD

The relevance of group theory was well understood in 1955 by S. P. Lloyd, whose unpublished work[1] is now sketched. He used a standard method of identifying an equivalence relation on a set with a group of bijections of the set into itself. This method considers a group G_ν of correspondences of S onto itself, and describes "equivalent" states as follows: G_ν is said to be *transitive* on a subset X of S iff

(*i*) $x \epsilon X, g \epsilon G_\nu \Rightarrow gx \epsilon X$

(*ii*) $x,y \epsilon X \Rightarrow \exists g \epsilon G_\nu$ such that $gx = y$

X is then called a transitive set. With $|X|$ the cardinality of X, it can be shown that $|X|$ divides the order of G_ν, and that each member of X appears exactly

$$\frac{|G_\nu|}{|X|}$$

times in the array gx, $g \epsilon G_\nu$, where x is any element of X. For each $x \epsilon S$ define $\pi x = \{gx : g \epsilon G_\nu\}$. It can be seen that each πx is a transitive set, and that for any $x,y \epsilon S$ we have either $\pi x = \pi y$ or $\pi x \cap \pi y = \phi$. Thus the πx form a partition of S, and so G_ν induces a corresponding equivalence relation $\equiv$ according to $x \equiv y$ iff there is a $g \epsilon G_\nu$ such that $gx = y$. Conversely, given an equivalence relation $\equiv$ on S, the set of all bijections $g:S \leftrightarrow S$ with $gx \equiv x$ forms a group G_ν under composition which induces $\equiv$ in the sense above, and we have the following "summation formula:" With α an equivalence class of $\equiv$, x any member of α, and f a real valued function on S,

$$\sum_{y \epsilon \alpha} f(y) = \frac{|\alpha|}{|G_\nu|} \sum_{g \epsilon G_\nu} f(gx)$$

Now let x_t be a continuous-parameter Markov process taking values on S, with a stationary transition rate matrix $Q = (q_{xy})$, assumed to be ergodic. There is then a unique probability vector $p = \{p_x, x \epsilon S\}$ such that p solves the "statistical equilibrium" equations

$$\sum_{x \epsilon S} p_x q_{xy} = 0, \qquad y \epsilon S \tag{1}$$

p is the stationary probability distribution for x_t. The group G_ν and the relation $\equiv$ become relevant to eq. (1) when the matrix Q of rates is unaffected by the permutations (of its rows and columns) corresponding to $g \epsilon G_\nu$; indeed, if G_ν and $\equiv$ express what we mean by saying that equivalent states differ only in respect of renaming customers, switches, etc., this is the precise way that symmetry affects the problem of calculating state probabilities. This relevance is recognized in the following

Def. 1: *Q admits* G_ν iff $g \epsilon G_\nu$, $x,y \epsilon S \Rightarrow$

$$q_{xy} = q_{(gx)(gy)}$$

which leads quickly to Lloyd's basic 1955 results:

Theorem 1: Let G_ν be a group of bijections of S onto itself such that Q admits G_ν, and let $E = \{\pi x, x\epsilon S\}$ be the set of equivalence classes induced by G_ν. Then

(*i*) The projection map $\pi{:}x \rightarrow \pi x$ defines a "macroscopic" Markov process πx_t on E, with transition rates

$$q_{\alpha\beta} = \sum_{y\epsilon\beta} q_{xy} \text{ for } x\epsilon\alpha \text{ and } \alpha\epsilon E$$

and stationary probabilities $\{p_\alpha, \alpha\epsilon E\}$ satisfying

$$\sum_{\alpha\epsilon E} p_\alpha q_{\alpha\beta} = 0,\ \beta\epsilon E$$

(*ii*) For each x$\epsilon\alpha\epsilon$E,

$$p_x = \frac{1}{|\alpha|} p_\alpha$$

Thus if Q admits G_ν, then equivalent states have *the same* stationary probabilities, and these can be computed from a *reduced state equation* of lower order.

Proof of Theorem 1: Everything follows from the fact that if $\beta\epsilon E$, then

$$\sum_{y\epsilon\beta} q_{xy}$$

has the same value for all $x\epsilon\alpha\epsilon E$; so we prove this first. If $z\epsilon\alpha$ then by transitivity of α there is a $g\epsilon G_\nu$ with $x = gz$, so that

$$\sum_{y\epsilon\beta} q_{xy} = \sum_{y\epsilon\beta} q_{(gz)y} = \sum_{y\epsilon\beta} q_{(gz)(gy)} = \sum_{y\epsilon\beta} q_{zy}$$

The second equality arises from $g^{-1}\beta = \beta$, the third from Q's admitting G_ν. Since x and z were arbitrary elements of α, the result is proved. It implies, by results[7] of M. Rosenblatt and C. J. Burke, that πx_t is a Markov process with transition rates

$$q_{\alpha\beta} = \sum_{y\epsilon\beta} q_{xy}, \text{ any } x\epsilon\alpha$$

and πx_t is ergodic if x_t is. Hence it has a unique stationary probability distribution $\{p_\alpha, \alpha\epsilon E\}$ satisfying

$$\sum_{\alpha\epsilon E} p_\alpha q_{\alpha\beta} = 0,\ \beta\epsilon E$$

Thus (i) is proved; to prove (ii) we show first that

$$p_\alpha = \sum_{x \epsilon \alpha} p_x$$

where $\{p_x, x \epsilon S\}$ is the stationary distribution of x_t. We find

$$\sum_{\alpha \epsilon E} \sum_{x \epsilon \alpha} p_x q_{\alpha\beta} = \sum_{x \epsilon S} p_x \sum_{y \epsilon \beta} q_{xy} = \sum_{y \epsilon \beta} \sum_{x \epsilon S} p_x q_{xy} = 0$$

since the inner sum in the last term is always zero. Thus

$$\left\{ \sum_{x \epsilon \alpha} p_x,\ \alpha \epsilon E \right\}$$

satisfy the (reduced) equilibrium equations for πx_t, and so by the uniqueness of its (probability vector) solution,

$$p_\alpha = \sum_{x \epsilon \alpha} p_x$$

Now define q by $q_x = |\pi x|^{-1} p_{\pi x}$ and consider that

$$0 = \sum_{\alpha \epsilon E} p_\alpha q_{\alpha\beta} = \sum_{x \epsilon S} |\pi x|^{-1} p_{\pi x} q_{(\pi x)\beta}$$

$$= \sum_{x \epsilon S} q_x \sum_{y \epsilon \beta} q_{xy} = \sum_{y \epsilon \beta} \sum_{x \epsilon S} q_x q_{xy}$$

However if $z \epsilon \beta$, there is $g \epsilon G_\nu$ with $y = gz$, and (since q is constant on equivalence classes) $q_x = q_{(g^{-1}x)}$ so that

$$\sum_{x \epsilon S} q_x q_{xy} = \sum_{x \epsilon S} q_x q_{x(gz)} = \sum_{x \epsilon S} q_{(g^{-1}x)} q_{(g^{-1}x)z} = \sum_{x \epsilon S} q_x q_{xz}$$

Thus

$$\sum_{x \epsilon S} q_x q_{xy}$$

is constant over equivalence classes and so it must be zero. Hence q is a probability vector solution of the equilibrium equation for x_t, so $q = p$, i.e., $p_x = |\pi x|^{-1} p_{\pi x}$.

Lloyd's theorem accurately captured the relevance of the rate matrix Q's admitting the group G_ν, and he gave examples of the application of his result to small networks, such as individual switches and partial access concentrators, but he did not elaborate on the groups G_ν to be considered. However, in applying such a result to traffic in connecting networks we want to be sure that the groups G_ν we use reflect the intuitive notions of invariance of structure under renaming of terminals, switches, etc. The theorem thus leaves us with these important questions: What groups G_ν are appropriate or available for describing equivalence relations $\equiv$ useful in applications to traffic in networks? What traffic models give rise to rate matrices Q that admit these groups?

Furthermore, since we expect the applications we make of Lloyd's theorem to networks to depend on both network symmetry and customer interchangeability, not to mention routing, it would be well to have a formulation in which these items are clearly separated, as they are not in Theorem 1. What we need is a more specific definition of "equivalence" of states, one independent of probabilistic models, and peculiar to the network applications we intend, and one that reflects the idea of invariance of structure under renaming. These requirements will be met by constructing some groups that are appropriate from the symmetry group of the network graph G; Lloyd entertained[8] such an idea but did not describe it in Ref. 1. We shall first argue that certain natural necessary conditions, to be met by groups considered "appropriate," imply that they should be automorphism groups of $(S, \leq)$ whose elements preserve $\sim$; then we show how such groups arise directly from the symmetry group of the network graph.

VI. TWO INTUITIVELY NECESSARY CONDITIONS

We need properties and concepts that help make precise the notion of "structurally equivalent" states. Some of these will now be arrived at quickly and intuitively. Consider therefore two states x and y that differ only in point of renaming terminals and links, but otherwise have the same structure. In such a situation we expect to be able to make a correspondence between the calls in progress in x and those in y, because structural equivalence requires that each call c in progress in x have at least one analog in progress in y, playing the role of c in the structure of y. This being so, we see in the same way that the elements of B_x (obtainable from x by a hangup) have a natural correspondence to those of B_y, going beyond the fact that $|B_x| = |B_y| = |x| = |y|$: namely, to $z \epsilon B_x$ we assign a state in B_y obtained by hanging up an analog of the call hung up in x to get z, i.e., an analog of $\gamma(x - z)$.

An exactly similar situation holds for the sets A_x and A_y of states which are respectively above x and y; actually, more is true, since for every call c free and not blocked in x there will be an analog (possibly more than 1), call it c', free and not blocked in y, and to every way of putting up c in x, i.e., for every $z \epsilon A_{cx}$, there will be a corresponding way of putting up an analog c' in y, and thus a natural correspondence of A_{cx} with $A_{c'y}$. It is intuitively clear that $A_{c'y}$ cannot have either more or fewer states than A_{cx}, else there would have to be something structurally different about x and y to account for the discrepancy.

Let now g denote the partial natural correspondence we have so far, with the properties $gx = y$, $g(B_x) = B_{gx}$, and $g(A_{\gamma(z-x)x}) = A_{\gamma(gz-gx)gx}$

for $z \epsilon A_x$. As we indicated, not even this much of g is unique. Nevertheless we suggest that g can be extended in a similar way, and to be defined on all of S in such a way as to satisfy the commutation $g(B_x) = B_{gx}$ and the distribution $g(A_{\gamma(z-x)x}) = A_{\gamma(gz-gx)gx}$ for $z \epsilon A_x$. It is not hard to see that these conditions are the same as

$$x \leq y \Rightarrow gx \leq gy$$

$$x \sim y \Rightarrow gx \sim gy$$

The map g then takes any state into one equivalent in structure, in such a way as to preserve order in $(S, \leq)$ and also equivalence in the other sense of having the same people talking. We can expect that $g(S) = S$, and that g is one-to-one.

Now an isomorphism between two partially ordered systems (POS) is precisely a bijection that preserves order both ways; in our case the two POS coincide, so g is called an *automorphism* of $(S, \leq)$. Since equivalence is transitive, it follows that if there are automorphisms g, h such that $y = gx$, $z = hy$, then there must be one f such that $z = fx$, namely the composition $f = hg$. Hence one is led naturally to consider groups of automorphisms that preserve $\sim$. Thus while *any* equivalence relation on S can be described by a group of bijections on S, we claim that to adequately express what is meant by structure invariant under renaming, the groups of interest for a theory of networks should be automorphism groups of $(S, \leq)$. Thus the first part of this study of equivalence $\equiv$ between states is the search, for a general network ν, for a group G_η of automorphisms of $(S, \leq)$ that preserve $\sim$ such that the "usual" definition of equivalence via a group, viz.,

$$x \equiv y \text{ iff } \exists g \epsilon G_\eta \ni x = gy$$

agrees with what we mean by structural equivalence.

VII. SYMMETRIES OF THE NETWORK GRAPH

Structural equivalence of states rests ultimately on properties of the network graph G that are independent of and prior to the choice of inlets I, outlets Ω, and states S that complete the description of a network $\nu = (G,I,\Omega,S)$. In an informal way, one might say that the equivalence of two states under renaming of terminals and links really depends on what it means for the network to "look the same" to distinct terminals. As an example consider two arbitrary distinct inputs on the left side of the standard No. 5 crossbar type network in Fig. 1. It is obvious intuitively that if the frames and switches are identical, and the connections within and between frames correspond to complete bipartite graphs in the usual way, then the network "looks the same" to two such inlets. The same is true of any two interframe junctors, or of any two links from the same

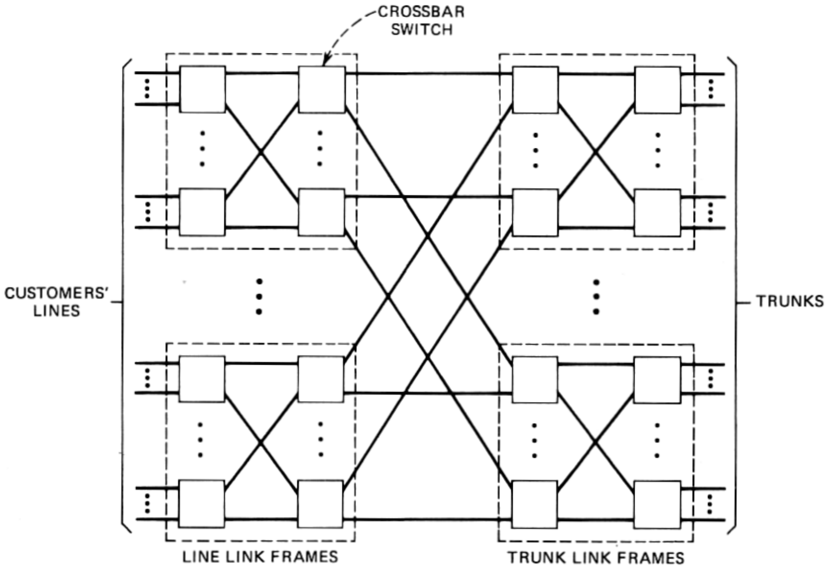


Fig. 1—Connecting network.

or from two distinct frames. We seek to clarify this informal notion of "looking the same," and to develop it into a precise definition of structural equivalence in terms of a natural symmetry group for the network graph.

Let us think of the terminals of ν as the vertices, and of the crosspoints as the edges, of the network graph G. It is clear that G is determined by giving a relation N on the set T of terminals such that for t,s in T, tNs iff there is a crosspoint or edge between t and s. N is the symmetric "nextness" or adjacency relation that completely depicts the network structure.

Now suppose that we rename the terminals in T according to some permutation τ. Most permutations would play havoc with the adjacency relation; that is, if t and s had a crosspoint between them, then τt and τs easily might not, and conversely. But there might be *some* permutations other than the identity which *preserved* N in the sense that for every $t,s \epsilon T$

$$tNs \text{ iff } (\tau t)N(\tau s)$$

In this case the permuted terminals have crosspoints between them in exactly the same pattern as the unpermuted. It is the existence of such an "N-preserving" permutation τ that we take as the precise meaning of "looking the same."

The network "looks the same" to two terminals t and s iff there is an N-preserving permutation τ of T into itself such that $s = \tau t$. It is apparent that these N-preserving permutations form a group, which we call the symmetry group of the network graph G. We remark that this group may be trivial (if there are no N-preserving permutations except the identity), and that in any case it in no way depends on what terminals have been designated as inlets or outlets, or on what ways of closing the crosspoints are to be allowed as physically meaningful states.

In most telephone connecting networks the set I of inlets and that Ω of outlets are fixed sets of terminals, and one is not interested in whether the network "looks the same" to an inlet as it does to an intermediate link or junctor; in most cases it will not, in any case, because of their different functions in operation. So it makes sense to restrict the N-preserving bijections we think of as renamings of switches and terminals to those which either preserve both I and Ω, or else map each onto the other. This restriction defines a subgroup G_n, called the symmetry group of G for I and Ω, which will be used to define structural equivalence of states.

VIII. SYMMETRIES OF THE SET OF STATES

The set S of states of a network $\nu = (G,I,\Omega,S)$ represents all the ways of closing the crosspoints which we regard as physically sensible. It is closed under hangups, that is, under removal of a chain from a state; it need not be closed under adding new chains, nor even under adding new chains which by themselves already represent a state with one call in progress. It is convenient, however, to require that sets of chains in a structural equivalence class either all belong to S, or that none of them does. This requirement of course implicitly assumes that we know what structural equivalence is before we choose states for S; it will be seen that the definition of equivalence below applies to *arbitrary* sets of chains on G, so the requirement can be met as we choose such sets to belong to S. Specifying the set S represents definite choice of the ways in which the network with graph G is to be used.

We shall now use the symmetry group G_n of the network to define what we mean by two states' differing only in point of renaming links, terminals, etc. Indeed, it can be seen intuitively that the symmetry group G_n of the network induces a natural equivalence relation on sets of chains on G, and thus on whatever such sets we choose as states: two sets of chains on G are equivalent if there is some group element $\tau \epsilon G_n$ such that a terminal t is busy in the first iff τt is busy in the second. The incorporation of a map $\tau \epsilon G_n$ in the definition ensures that *simultaneously* the network looks the same to a terminal t and to its analog τt. For states, then, we define structural equivalence $\equiv$ by

Def. 2: $x \equiv y$ iff $\exists \tau \epsilon G_n \ni t_1, \ldots, t_m$ are the terminals busy in x iff $\tau t_1, \ldots, \tau t_m$ are those busy in y.

It is apparent that since G_n is a group, $\equiv$ is an equivalence relation on S, partitioning S into a set E of equivalence classes $\alpha,\beta,\ldots$, and inducing a natural projection map $\pi: S \rightarrow E$ such that

$$\pi x = \{y{:}y \equiv x\} \epsilon E$$

The elements α of E will be the *reduced network states.* As in Section V, we see that there is an associated group G_ν that provides an alternative description of $\equiv$; G_ν is the group of all bijections $g: S \leftrightarrow S$ which map each equivalence class α into itself. In fact, G_ν is the largest strictly imprimitive group of bijections of S onto itself whose sets of imprimitivity are precisely the $\alpha \epsilon E$:

Def. 3: $G_\nu = \{g{:}S \leftrightarrow S \ni g(\alpha) = \alpha \text{ for } \alpha\epsilon E\}$

Remark 1: As in Section V, we have $x \equiv y$ iff $\exists g\epsilon G_\nu \ni gx = y$.

Remark 2: Although G_ν is a group of bijections which does characterize $\equiv$, it is typically *not economical.* It turns out that a much smaller subgroup of G_ν, defined directly from G_n, suffices to characterize $\equiv$ as in Remark 1. These subgroups appear in Section 10, and they are the $\sim$-preserving automorphism groups desired in Section VI.

IX. ACTIONS OF $\tau\epsilon G_n$ ON A AND S

An assignment a is a correspondence or injection from a subset of I into Ω. Thus an element $\tau\epsilon G_n$ acts in a natural way on an assignment $a\epsilon A$ to produce a new assignment τa according to the rule that u and v correspond in a iff τu and τv correspond in τa. Similarly an element $\tau\epsilon G_n$ acts in a natural way on a set X of chains on the graph G to produce a new set τX of chains consisting of the τ-images of chains in X, thus: $t_1, \ldots, t_l$ is to be a chain of X iff $\tau t_1, \ldots, \tau t_l$ is a chain of τX. In particular a $\tau\epsilon G_n$ acts on a state $x\epsilon S$ to produce a set of chains τx, and it is reasonable to assume, as we have done here, that the choice of S is consistent with the symmetry group G_n in that S is closed under the action of any $\tau\epsilon G_n{:}x\epsilon S,\ \tau\epsilon G_n \Rightarrow \tau x\epsilon S$. This will ensure that either all the sets of chains in a structural equivalence class are states, or none of them is.

Remark 3: Since τ may map Ω onto I its action on $a\epsilon A$ may reverse the "usual" order of the pairs $(u,v)\epsilon I \times \Omega$ to $(\tau u,\tau v)\epsilon\Omega \times I$. Therefore we do not distinguish between an assignment

$$a = \{(u,v)\epsilon I \times \Omega{:}(u,v)\epsilon a\}$$

from its inverse

$$a^{-1} = \{(v,u)\epsilon\Omega \times I{:}(u,v)\epsilon a\}$$

or else we specify that when $\tau I = \Omega$ then the action is defined by

$$\tau a = \{(\tau v,\tau u){:}(u,v)\epsilon a\}$$

X. HOMOMORPHISM OF G_n INTO G_ν

The network symmetry group G_n depends only on the basic network structure: the adjacency relation N of the graph G that depicts the network, and the choice of I and Ω, since $\tau \epsilon G_n$ are restricted to preserve inlets and outlets. The state symmetry group G_ν describing $\equiv$, however, also depends on what sets of chains on the graph G are chosen as physically interesting or important states. Since G_n is used in the definition of $\equiv$, and thus of G_ν, it is not surprising that G_n and G_ν happen to be algebraically related: there is natural homomorphic image G_η of G_n in G_ν, consisting entirely of $\sim$-preserving automorphisms of $(S, \leq)$, and describing the same equivalence relation $\equiv$. Thus for most purposes it is more convenient to use the "equivalent" subgroup G_η than the full symmetry group G_ν associated with $\equiv$ by the standard method. The big group G_ν induced by $\equiv$, incidentally, is not necessarily an automorphism group: consider, e.g., the 2×2 switch of Fig. 4 supra; clearly $5 \equiv 6$ and $1 \equiv 3$, but $1 \leq 5$ and $3 \nleq 6$. Thus the map g defined by the permutation (13) (24) (56) belongs to G_ν but is not an automorphism.

Theorem 2: The action of $\tau \epsilon G_n$ on states x defines a homomorphism $\eta: G_n \rightarrow G_\nu$ according to the rule that $\eta(\tau)x = \tau x$; the image group $G_\eta = \eta(G_n)$ is an automorphism group that preserves $\sim$.

Proof: For $\tau \epsilon G_n$, define $\eta(\tau) \epsilon G_\nu$ by the condition $\eta(\tau)x = \tau x$ that $t_1, \ldots, t_l$ is a chain of $x \epsilon S$ iff $\tau t_1, \ldots, \tau t_l$ is a chain of $\eta(\tau)x (= \tau x)$. The subset $\eta(G_n)$ is closed under composition, so it is a subgroup G_η of G_ν. If $x \leq y$, then $\eta(\tau)x \leq \eta(\tau)y$ for each $\tau \epsilon G_n$, so that each $\eta(\tau)$ is an automorphism of $(S, \leq)$. It is easily verified that $\eta(\tau_1\tau_2) = \eta(\tau_1)\eta(\tau_2)$; preservation of $\sim$ follows from the definition of action.

Theorem 3: If every terminal $t \epsilon T$ is busy in some state $x \epsilon S$, then the homomorphism η of Theorem 2 is an isomorphism, i.e., it is injective (one-to-one).

Proof: If $\eta(\tau_1) = \eta(\tau_2)$, let x be a state with one call up in which a terminal t is busy. Then x consists of a single chain $t_1, \ldots, t_l$ such that t is some t_i and $\tau_1 t_j = \tau_2 t_j$ for $1 \leq j \leq l$. Then $\tau_1 t = \tau_2 t$, and since t is arbitrary we have $\tau_1 = \tau_2$, and so η is injective.

Remark 4: The following conditions are all equivalent:

(*i*) $x \equiv y$
(*ii*) For some $\tau \epsilon G_n$, $x = \tau y = \eta(\tau)y$
(*iii*) For some $h \epsilon G_\eta$, $x = hy$
(*iv*) For some $g \epsilon G_\nu$, $x = gy$

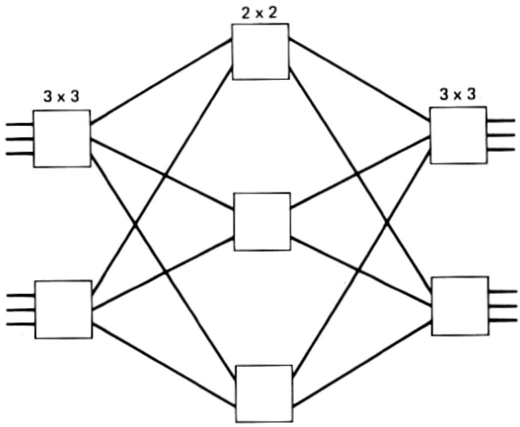


Fig. 2—Small 3-stage network.

XI. PARTIAL ORDERING OF REDUCED STATES

It is natural to try to partially order the set E of reduced states according to this idea: an equivalence class α is "above" another β if some element $x \epsilon \alpha$ can be reached from some member $y \epsilon \beta$ by adding more calls, i.e., if $x \geq y$ for some $x \epsilon \alpha$ and $y \epsilon \beta$. Formally, we make the

Def. 4: $\alpha \geq \beta$ iff $\exists x \epsilon \alpha, y \epsilon \beta \ni x \geq y$

For this to define a partial ordering it must be reflexive, antisymmetric, and transitive. The first is obvious; we prove the other two. Let then $\alpha \geq \beta$ and $\beta \geq \alpha$; to show $\alpha = \beta$ it is enough to show $\alpha \cap \beta \neq \phi$, because each is an equivalence class. There exist $x,z \epsilon \alpha$ and $y,w \epsilon \beta$ such that $x \geq y$ and $w \geq z$. Since $x \equiv z$ and $y \equiv w$, there exist *automorphisms* $\tau_1,\tau_2 \epsilon G_\eta$ such that $\tau_1 x = z$ and $\tau_2 y = w$. Hence $\tau_1 x \geq \tau_1 y$ and

$$\tau_2 y = w \geq z = \tau_1 x \geq \tau_1 y$$

so that $\tau_1^{-1}\tau_2 y \geq y$. But a state z that is above another $\tau_1 y$ and has the same number of calls in progress equals it: $z = \tau_1 y$. So $\alpha \cap \beta \neq \phi$.

To prove transitivity let $\alpha \geq \beta$ and $\beta \geq \gamma$, so that there exist $x \epsilon \alpha$, $y \epsilon \beta$, $z \epsilon \beta$, and $w \epsilon \gamma$ such that $x \geq y$, $y \equiv z$, and $z \geq w$. There is an automorphism $\tau \epsilon G_\eta$ such that $\tau y = z$, whence

$$\tau x \geq \tau y = z \geq w$$

Hence there is something in α which is above something in γ, i.e., $\alpha \geq \gamma$. Thus $\geq$ is a partial ordering on $E = \pi(S)$.

Remark 5: $x \geq y \Rightarrow \pi x \geq \pi y$

Remark 6: The partial order $\geq$ on E need not be a semilattice, and *a fortiori* the projection $\pi{:}S \rightarrow E$ need not be a semilattice homomorphism, as is $\gamma{:}S \rightarrow A$, if A is ordered by inclusion. Figures 2 and 3 provide a

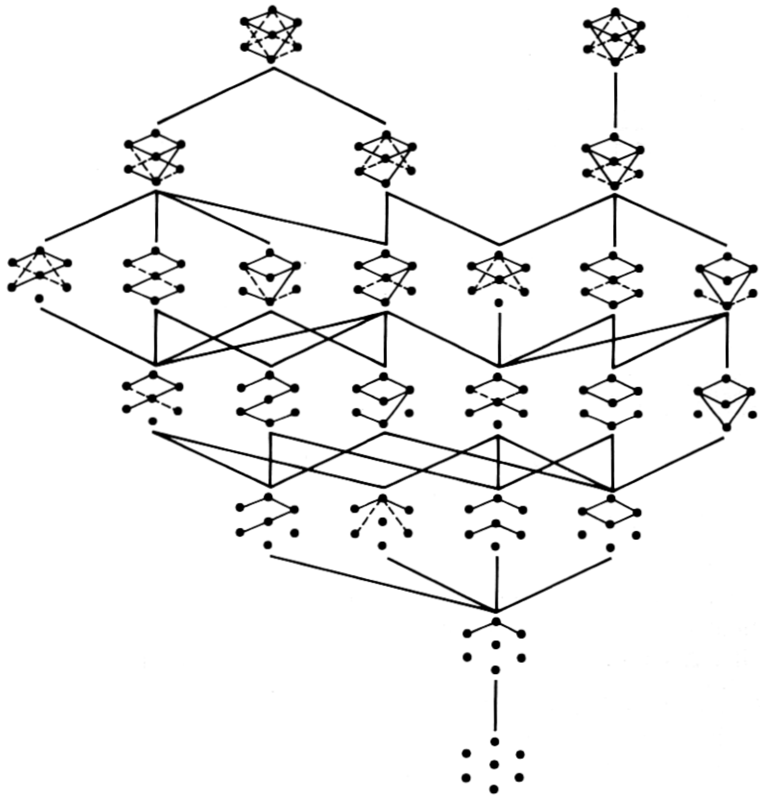


Fig. 3—Reduced state diagram for small 3-stage network of Fig. 2.

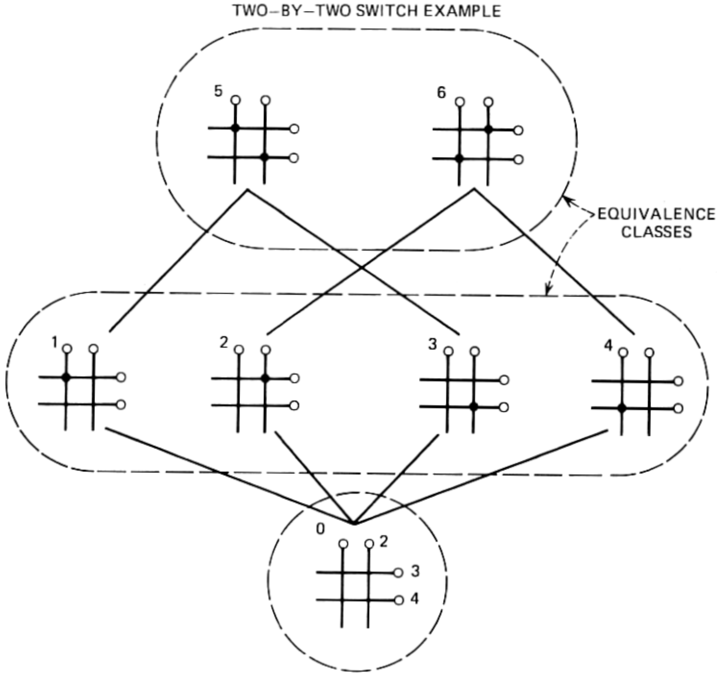
counterexample, as well as an illustration of the reduced states of a network.

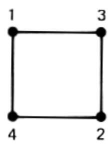
XII. 2 × 2 SWITCH

The next four sections are devoted to increasingly complex examples that will illustrate the notions we have introduced to make precise the idea of structural equivalence. Our first and simplest example is the 2 × 2 switch shown in Fig. 4. We arbitrarily label the inlets 1 and 2 and the outlets 3 and 4. So with $I = \{1,2\}$ and $\Omega = \{3,4\}$ the network graph G is a square with the terminals of I on diagonally opposite vertices, and similarly for Ω. The adjacency relation N consists of exactly the pairs (1,3), (1,4), (2,3), and (2,4) together with the results of interchanging the first and second members of these pairs so as to make N symmetric.

The maps of the terminal set $T = \{1,2,3,4\}$ into itself which preserve N and either preserve both I and Ω, or carry one onto the other, are exactly the permutations

identity, (12), (34), (12)(34), (13)(24), (23)(14), (1423), (1324)





	N	T 1	2	3	4
T	1	0	0	1	1
	2	0	0	1	1
	3	1	1	0	0
	4	1	1	0	0

Fig. 4—Network graph, adjacency matrix, and states for 2×2 switch.

These 4-permutations form the symmetry group G_n of the 2×2 switch for $I = \{1,2\}$, $\Omega = \{3,4\}$. This group has a nice geometric meaning in terms of the network graph, the square in Fig. 4: it consists precisely of all the rotations and reflections of the square into itself. The multiplication table for this group is given in Table I, along with "generators" $A = (1423)$ and $B = (12)$ which, under the relations $A^4 = B^2 =$ identity, $BA = A^3B$, identify the group as (an isomorph of) the dihedral group D_4 of order 8. Every terminal is busy in some state, so Theorem 3 applies and we need only calculate G_n to define $\equiv$, instead of passing through G_ν.

The symmetry group G_n describes the equivalence of states through the action table in Fig. 5. Here the columns are indexed by states $x \epsilon S$ and the rows by group elements $\tau \epsilon G_n$, and the τ,x entry is the state τx defined by the action of τ on x, i.e., by replacing every chain $(t_1,t_2) \epsilon x$ by $(\tau t_1, \tau t_2)$.

Table I — Multiplication table of G_n for 2×2 switch

g \ f	I	B (12)	A^2B (34)	A^2 (12)(34)	AB (13)(24)	A^3B (14)(23)	A (1423)	A^3 (1324)
I	I	(12)	(34)	(12)(34)	(13)(24)	(14)(23)	(1423)	(1324)
(12)	(12)	I	(12)(34)	(34)	(1324)	(1423)	(14)(23)	(13)(24)
(34)	(34)	(12)(34)	I	(12)	(1423)	(1324)	(13)(24)	(14)(23)
(12)(34)	(12)(34)	(34)	(12)	I	(14)(23)	(13)(24)	(1324)	(1423)
(13)(24)	(13)(24)	(1423)	(1324)	(14)(23)	I	(12)(34)	(12)	(34)
(14)(23)	(14)(23)	(1324)	(1423)	(13)(24)	(12)(34)	I	(34)	(12)
(1423)	(1423)	(13)(24)	(14)(23)	(1324)	(34)	(12)	(12)(34)	I
(1324)	(1324)	(14)(23)	(13)(24)	(1423)	(12)	(34)	I	(12)(34)

$\tau \in G_n$ \ $x \in S$	0	1	2	3	4	5	6	$\eta(\tau)$
I	0	1	2	3	4	5	6	I
(12)	0	2	1	4	3	6	5	(12) (34) (56)
(34)	0	4	3	2	1	6	5	(14) (23) (56)
(12) (34)	0	3	4	1	2	5	6	(13) (24)
(13) (24)	0	1	4	3	2	5	6	(24)
(14) (23)	0	3	2	1	4	5	6	(13)
(1423)	0	2	3	4	1	6	5	(1234) (56)
(1324)	0	4	1	2	3	6	5	(1432) (56)
	$\tau x \; (= \eta(\tau) x)$							

Fig. 5—Action and isomorphism for the 2×2 switch.

Each row of the table defines a bijection of $S = \{0, \ldots, 6\}$ into itself that corresponds to the row index $\tau \epsilon G_n$, and thus an isomorphism $\eta : G_n \rightarrow S_7$ depicted by

τ	id	(12)	(34)	(12)(34)	(13)(24)	(23)(14)	(1423)	(1324)
$\eta(\tau)$	id	(12)(34)(56)	(14)(23)(56)	(13)(24)	(24)	(13)	(1234)(56)	(1432)(56)

It can be verified that $G_\eta = \eta(G_n)$ is an automorphism group of $(S, \leq)$,

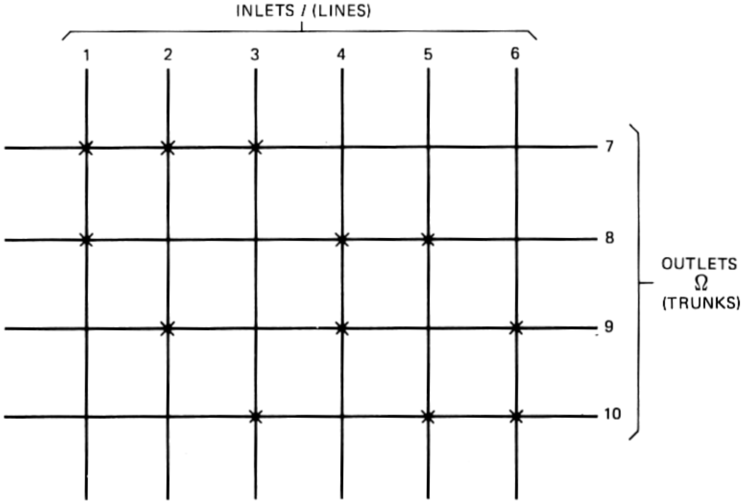


Fig. 6—Random slip concentrator.

isomorphic to D_4 and preserving $\sim$. It remains to check that $\equiv$ defined with the help of G_n is in fact the intuitive one that is indicated in Fig. 4: looking at the values of η in the two line table above we easily see that $5 \equiv 6$, $1 \equiv 2 \equiv 3 \equiv 4$, and no other distinct states are equivalent, so the equivalence classes are $\{0\}$, $\{1,2,3,4\}$, and $\{5,6\}$, as we had guessed. The group G_ν of *all* bijections which preserve these equivalence classes is isomorphic to $S_2 \times S_4$, a group of order 48; in contrast, the subgroup G_η is sufficient to characterize $\equiv$ and is only of order 8.

XIII. RANDOM SLIP CONCENTRATOR

The crosspoint assignment for a 6-line to 4-trunk random slip concentrator is shown in Fig. 6. The phrase "random slip" is old telephone terminology that clearly originated as a description of the even or regular way in which the incomplete access of lines to trunks is distributed over the network, something like the statisticians' balanced block designs. Figure 6 may also be depicted as the labelled tetrahedron of Fig. 7, with the interpretation that an edge (line) "has access" to the two vertices (trunks) that it connects. Figure 7 leads naturally to the network graph (Fig. 8) and the adjacency matrix (Table II): just add a vertex of degree 2 in the middle of each edge, with the same label as the edge. In this network every t is busy in some state, so Theorem 3 applies, $G_\eta \approx G_n$, and we need only calculate G_n. The reduced states are shown in a convenient representation as a partially ordered system in Fig. 9.

Thus we seek maps of $I = \{1,2, \ldots ,6\}$ and $\Omega = \{7,8,9,10\}$ into themselves which preserve adjacency. It can be seen from Figs. 6 and 7 that every

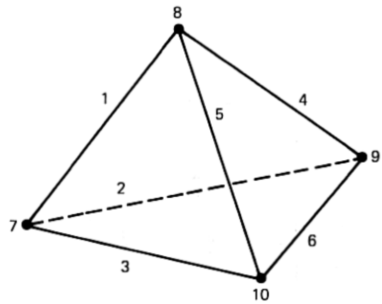


Fig. 7—Labeled tetrahedron.

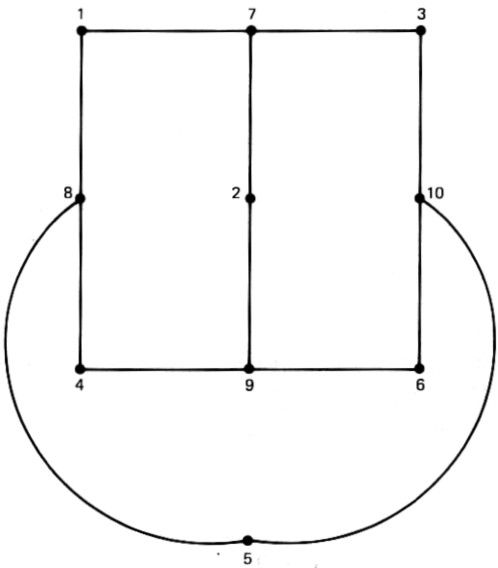


Fig. 8—Network graph for concentrator.

Table II — Adjacency matrix for concentrator

N	1	2	3	4	5	6	7	8	9	10
1							1	1	0	0
2							1	0	1	0
3			0				1	0	0	1
4							0	1	1	0
5							0	1	0	1
6							0	0	1	1
7	1	1	1	0	0	0				
8	1	0	0	1	1	0			0	
9	0	1	0	1	0	1				
10	0	0	1	0	1	1				

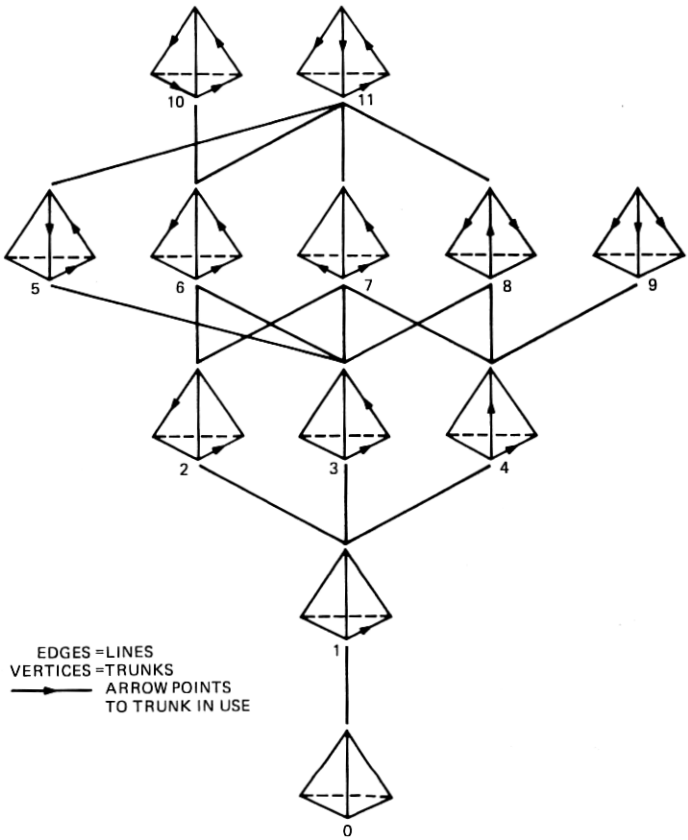


Fig. 9—Reduced states of random slip concentrator.

permutation of {7,8,9,10} *forces* a unique permutation of {1, . . . ,6} if adjacency is to be preserved; graphically, we see that all the maps we seek are rotations or reflections of the tetrahedron of Fig. 7 into itself, and that all permutations of {7,8,9,10} are allowed. More formally, every transposition of Ω is allowed and requires a unique pair of disjoint transpositions in I according to the table

(78) forces (24)(35)
(89) forces (12)(56)
(9 10) forces (23)(45)
(7 10) forces (15)(26)
(7 9) forces (14)(36)
(8 10) forces (13)(46)

Since the allowed maps form a group, and every permutation of Ω is a product of transpositions, every permutation of Ω forces a unique one

Table III — G_n for concentrator

id.	id.
(78)	(24)(35)
(89)	(12)(56)
(9 10)	(23)(45)
(7 10)	(15)(26)
(79)	(14)(36)
(8 10)	(13)(46)
(78)(9 10)	(25)(34)
(89)(7 10)	(16)(25)
(79)(8 10)	(16)(34)
(789)	(142)(356)
(798)	(124)(356)
(78 10)	(153)(246)
(7 10 8)	(135)(264)
(79 10)	(145)(263)
(7 10 9)	(154)(236)
(89 10)	(123)(465)
(8 10 9)	(132)(456)
(789 10)	(1463)(25)
(798 10)	(2453)(16)
(79 10 8)	(1265)(34)
(78 10 9)	(1562)(34)
(7 10 89)	(3542)(16)
(7 10 98)	(3641)(25)

of I, as claimed. It follows that G_n is isomorphic to S_4, and consists of the permutations shown in Table III. The "forced" permutations of I are of course a subgroup of S_6 isomorphic to S_4.

XIV. CLOS NETWORK EXAMPLE

The crosspoint structure and network graph for the simplest 3-stage Clos rearrangeable network are shown in Figs. 10–11. A planar form of the network graph is on Fig. 12. At the start of this paper we loosely described the invariances of structure to be studied as those associated with "renaming terminals, switches, and links." The little Clos network now under discussion gives us a specific example of what this means: it means, of course, permuting these entities while preserving the structure, e.g., interchanging switches in a stage while dragging along each switch's rat's

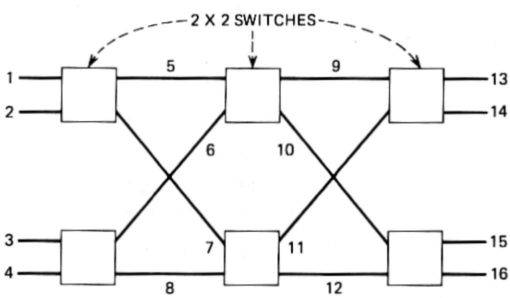


Fig. 10—Simple 3-stage Clos network.

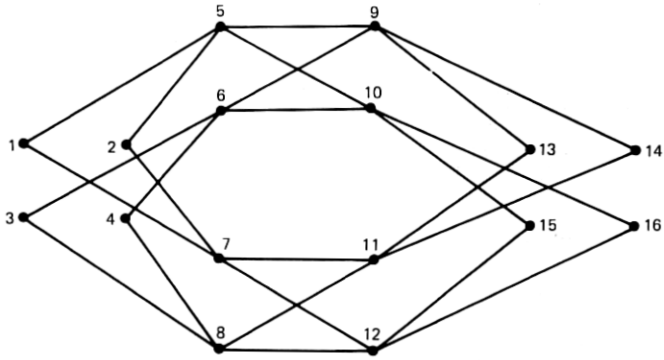


Fig. 11—Crosspoint structure and network graph for 3-stage Clos network made of 2 × 2 switches.

nest of links that connect to other stages. In particular, the organization of crosspoints into switches must also be preserved by these permutations of terminals, switches, and links. In the next section (XV) we shall describe how this constraint provides an approach to calculating the group G_n for general crossbar networks. For the present we show how the approach applies to the example under discussion.

Any 16-permutation that preserves the adjacency or crosspoint structure depicted in Fig. 11 must permute the inner terminals or links {5,6, . . . ,12} among themselves only, and it can do so in exactly the 16 ways shown in Table IV. It is readily seen that (*i*) these 16 ways correspond to interchanging switches, and rotating or reflecting the network,

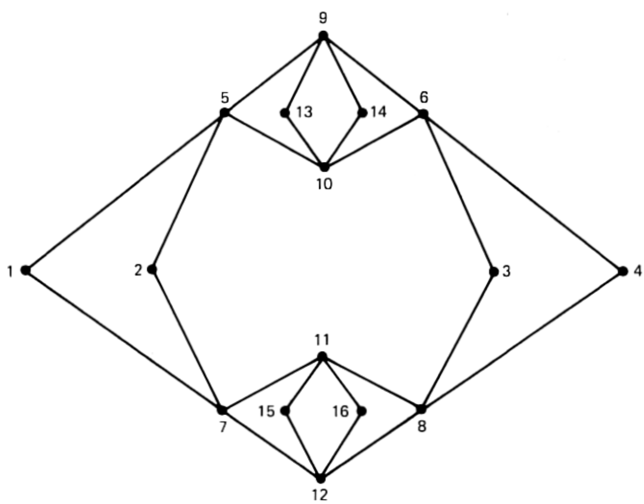


Fig. 12—Planar form of network graph of Fig. 11.

Table IV — Allowed permutations of inner terminals for 3-stage Clos network of 2 × 2 switches

No.	Map	Type	Action, if simple
1	id.	*ss*	move no switches
2	(57)(68)(9 11)(10 12)	*ss*	interchange m. sw.
3	(9 10)(11 12)	*sm*	interchange rt. sw.
4	(57)(68)(9 12)(10 11)	*sm*	interchange rt. and m. sw.
5	(56)(78)	*ms*	interchange l sw.
6	(58)(67)(9 11)(10 12)	*ms*	interchange l. and m. sw.
7	(56)(78)(9 10)(11 12)	*mm*	interchange rt. and l. sw.
8	(58)(67)(9 12)(10 11)	*mm*	rt., m., and l.
9	(59)(7 11)(6 10)(8 12)	*aa*	rotate about vertical axis
10	(5 11)(79)(6 12)(8 10)	*aa*	
11	(5 10 69)(7 12 8 11)	*ad*	
12	(5 12 6 11)(7 10 8 9)	*ad*	
13	(5 9 6 10)(7 11 8 12)	*da*	
14	(5 11 6 12)(7 9 8 10)	*da*	
15	(5 12)(6 11)(7 10)(89)	*dd*	
16	(5 10)(69)(7 12)(8 11)	*dd*	

and (*ii*) that they form a subgroup of S_{16}. Each of these ways can "go with" a number (here always 16) of permutations of outer terminals among *themselves,* to form an element of G_n. By writing down the possible ways this *matching* can be done (so as to preserve adjacency) we get a brute force way of calculating the group G_n for the 3-stage Clos network made of 2 × 2 switches.

The matching in question has a block structure: the permitted permutations of links can be partitioned in such a way that to each parititon element there corresponds a set of permitted permutations of outer terminals any one of which can "go with" each link permutation in the element. This block structure is indicated in Table IV by the type symbol; all allowed link permutations of the same type can match with all the same permutations of the outer terminals. Thus to present G_n

Table V — G_n for Clos network example

Type	*ss*	*sm*
	id.	(13 1)(14 16)
	(12)	(13 16)(14 15)
	(34)	(12)(13 15)(14 16)
	(13 14)	(12)(13 16)(14 15)
	(15 16)	(34)(13 15)(14 16)
	(12)(34)	(34)(13 16)(14 15)
	(12)(13 14)	(12)(34)(13 15)(14 16)
	(12)(15 16)	(12)(34)(13 16)(14 15)
	(34)(13 14)	(13 15 14 16)
	(34)(15 16)	(13 16 14 15)
	(13 14)(15 16)	(12)(13 15 14 16)
	(12)(34)(13 14)	(12)(13 16 14 15)
	(12)(34)(15 16)	(34)(13 15 14 16)
	(12)(13 14)(15 16)	(34)(13 16 14 15)
	(34)(13 14)(15 16)	(12)(34)(13 15 14 16)
	(12)(34)(13 14)(15 16)	(12)(34)(13 16 14 15)

Table V — (Continued)

Type	ms	mm
	(13)(24)	(1324)(13 15)(14 16)
	(14)(23)	(1324)(13 16)(14 15)
	(13)(24)(13 14)	(1423)(13 15)(14 16)
	(14)(23)(13 14)	(1423)(13 16)(14 15)
	(13)(24)(15 16)	(13)(24)(13 15)(14 16)
	(14)(23)(15 16)	(13)(24)(13 16)(14 15)
	(13)(24)(13 14)(15 16)	(14)(23)(13 15)(14 16)
	(14)(23)(13 14)(15 16)	(14)(23)(13 16)(14 15)
	(1324)	(1324)(13 15 14 16)
	(1324)(13 14)	(1423)(13 15 14 16)
	(1423)(13 14)	(1423)(13 16 14 15)
	(1324)(15 16)	(13)(24)(13 15 14 16)
	(1423)(15 16)	(13)(24)(13 16 14 15)
	(1324)(13 14)(15 16)	(14)(23)(13 15 14 16)
	(1423)(13 14)(15 16)	(14)(23)(13 16 14 15)

Type	aa	ad
	(1 13)(2 14)(3 15)(4 16)	(1 15 3 13)(2 16 4 14)
	(1 13)(2 14)(3 16)(4 15)	(1 16 3 13)(2 15 4 14)
	(1 14)(2 13)(3 15)(4 16)	(1 15 4 13)(2 16 3 14)
	(1 14)(2 13)(3 16)(4 15)	(1 16 4 13)(2 15 3 14)
	(1 13)(2 14)(3 15 4 16)	(1 15 3 14)(2 16 4 13)
	(1 13)(2 14)(3 16 4 15)	(1 16 3 14)(2 15 4 13)
	(1 14)(2 13)(3 15 4 16)	(1 15 4 14)(2 16 3 13)
	(1 14)(2 13)(3 16 4 15)	(1 16 4 14)(2 15 3 13)
	(1 13 2 14)(3 15)(4 16)	(1 15 3 13 2 16 4 14)
	(1 13 2 14)(3 16)(4 15)	(1 16 3 13 2 15 4 14)
	(1 14 2 13)(3 15)(4 16)	(1 15 4 13 2 16 3 14)
	(1 14 2 13)(3 16)(4 15)	(1 16 4 13 2 15 3 14)
	(1 13 2 14)(3 15 4 16)	(1 15 3 14 2 16 4 13)
	(1 13 2 14)(3 16 4 15)	(1 16 3 14 2 15 4 13)
	(1 14 2 13)(3 15 4 16)	(1 15 4 14 2 16 3 13)
	(1 14 2 13)(3 16 4 15)	(1 16 4 14 2 15 3 13)

Type		
	(1 13 4 15)(2 14 3 16)	(1 15)(2 16)(3 13)(4 14)
	(1 13 4 16)(2 14 3 15)	(1 15)(2 16)(3 14)(4 13)
	(1 13 3 15)(2 14 4 16)	(1 16)(2 15)(3 13)(4 14)
	(1 13 3 16)(2 14 4 15)	(1 16)(2 15)(3 14)(4 13)
	(1 14 4 15)(2 13 3 16)	(1 15)(2 16)(3 13 4 14)
	(1 14 4 16)(2 13 3 15)	(1 15)(2 16)(3 14 4 13)
	(1 14 3 15)(2 13 4 16)	(1 16)(2 15)(3 13 4 14)
	(1 14 3 16)(2 13 4 15)	(1 16)(2 15)(3 14 4 13)
	(1 13 3 15 2 14 4 16)	(1 15 2 16)(3 13)(4 14)
	(1 13 3 16 2 14 4 15)	(1 15 2 16)(3 14)(4 13)
	(1 13 4 15 2 14 3 16)	(1 16 2 15)(3 13)(4 14)
	(1 13 4 16 2 14 3 15)	(1 16 2 15)(3 14)(4 13)
	(1 14 3 15 2 13 4 16)	(1 15 2 16)(3 13 4 14)
	(1 14 3 16 2 13 4 15)	(1 15 2 16)(3 14 4 13)
	(1 14 4 15 2 13 3 16)	(1 16 2 15)(3 13 4 14)
	(1 14 4 16 2 13 3 15)	(1 16 2 15)(3 14 4 13)

it is enough to list the "outer" permutations that match each type of "inner"; this is done in Table V. The reduced states are in Fig. 13.

The multiplication table of the group of "inner" or switch permuta-

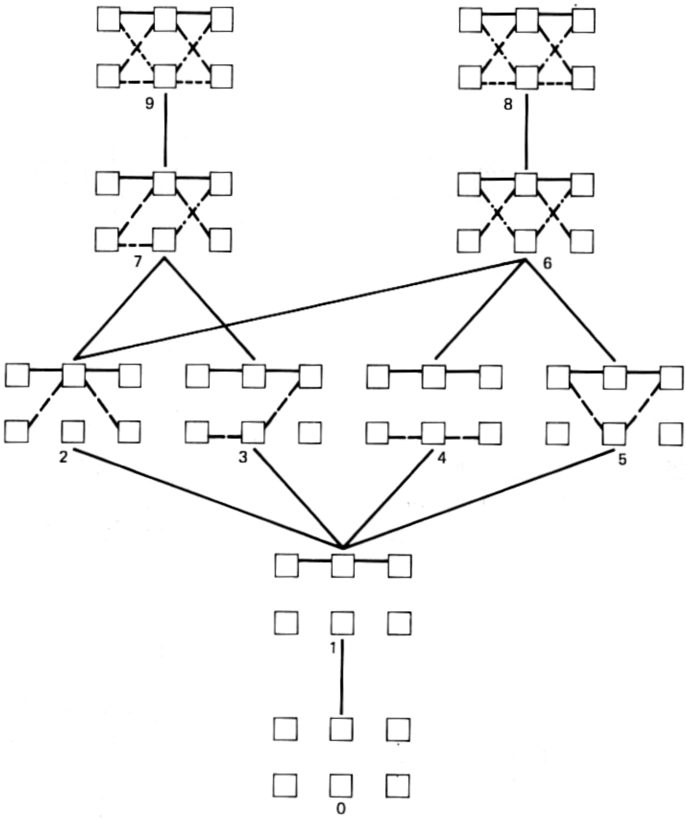


Fig. 13—Reduced states for 3-stage Clos network made of switches.

tions can be calculated from the cycle representations of Table IV; it is displayed in Table VI, from which it can be seen that elements $R_1 = 2$, $R_2 = 4$, and $R_3 = 15$ satisfy the relations

$$(R_1R_3)^2 = (R_2R_1)^4 = (R_2R_3)^4 = \text{identity}$$

which identify the group as (an isomorph of) $S_2 \times D_4$. This factorization is related to the fact that the type of a product is determined by the types of the factors in a way that is summarized in Table VII, in which the row is the type of the first factor, and the column that of the second. Indeed the type symbols themselves form an isomorph of D_4 under the "multiplication" Table VII. What is more, we can identify this group as the subgroup ($\cong D_4$) of the link or switch permutation group which restricts attention to the motion of the outer switches. Clearly the motion of the two middle switches is independent of that of the outer ones; this fact shows up in the feature that every type consists of just two permutations, and in the factor S_2 in the switch permutation group. The possible in-

Table VI — Multiplication table for switch permutation group

	id. 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
id.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
	2	1	4	3	6	5	8	7	10	9	12	11	14	13	16	15
	3	4	1	2	7	8	5	6	11	12	9	10	16	15	14	13
	4	3	2	1	8	7	6	5	12	11	10	9	15	16	13	14
	5	6	7	8	1	2	3	4	13	14	16	15	9	10	12	11
	6	5	8	7	2	1	4	3	14	13	15	16	10	9	11	12
	7	8	5	6	3	4	1	2	16	15	13	14	11	12	10	9
	8	7	6	5	4	3	2	1	15	16	14	13	12	11	9	10
	9	10	13	14	11	12	16	15	1	2	5	6	3	4	8	7
	10	9	14	13	12	11	15	16	2	1	6	5	4	3	7	8
	11	12	16	15	9	10	13	14	3	4	7	8	1	2	6	5
	12	11	15	16	10	9	14	13	4	3	8	7	2	1	5	6
	13	14	9	10	16	15	11	12	5	6	1	2	7	8	4	3
	14	13	10	9	15	16	12	11	6	5	2	1	8	7	3	4
	15	16	12	11	14	13	10	9	8	7	4	3	6	5	1	2
	16	15	11	12	13	14	9	10	7	8	3	4	5	·6	2	1
Type	*ss*	*ss*	*sm*	*sm*	*ms*	*ms*	*mm*	*mm*	*aa*	*aa*	*ad*	*ad*	*da*	*da*	*dd*	*dd*

terchanges of outer switches are summarized in Fig. 14 and they correspond to the type symbols as follows

switch interchange	1	*r*	*l*	*h*	*v*	*x*	*a*	*b*
type symbol	1	*sm*	*ms*	*aa*	*mm*	*dd*	*da*	*ad*

The switch interchange symbols of course also form a group $\cong D_4$ under the multiplication transferred from the type symbols; indeed $\{1,h,v,x\}$, $\{1,r,l,v\}$, and $\{1,a,b,v\}$ each forms a sub-"Vierergruppe":

XV. CALCULATION OF G_n FOR CROSSBAR NETWORKS

In connecting networks built out of stages of rectangular crossbar switches, a permutation of the terminals cannot preserve adjacency in the network graph unless it maps the terminals of each switch onto those of some switch, either itself or another one, so as to either preserve or

Table VII — "Multiplication" table for type symbols. Top row: generators identifying with D_4. Left column: identification with outer switch interchanges.

		A^3B *aa*	AB *dd*	A *da*	A^3 *ad*	A^2 *mm*	B *ms*	A^2B *sm*	1 *ss*
h	*aa*	*ss*	*mm*	*sm*	*ms*	*dd*	*ad*	*da*	*aa*
x	*dd*	*mm*	*ss*	*ms*	*sm*	*aa*	*da*	*ad*	*dd*
a	*da*	*ms*	*sm*	*mm*	*ss*	*ad*	*dd*	*aa*	*da*
b	*ad*	*sm*	*ms*	*ss*	*mm*	*da*	*aa*	*dd*	*ad*
v	*mm*	*dd*	*aa*	*ad*	*da*	*ss*	*sm*	*ms*	*mm*
l	*ms*	*da*	*ad*	*aa*	*dd*	*sm*	*ss*	*mm*	*ms*
r	*sm*	*ad*	*da*	*dd*	*aa*	*sm*	*mm*	*ss*	*sm*
1	*ss*	*aa*	*dd*	*da*	*ad*	*mm*	*ms*	*sm*	*ss*

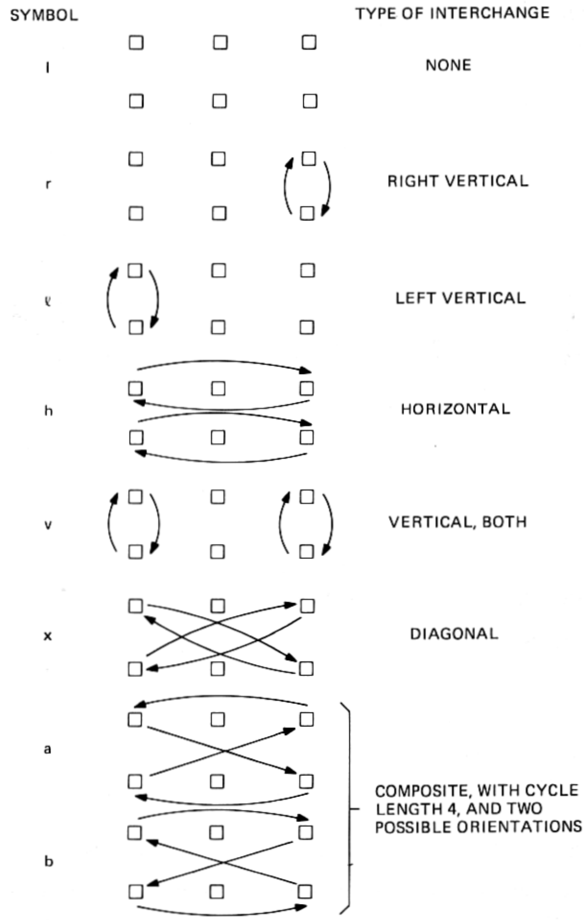


Fig. 14—Possible interchanges of outer switches.

interchange inlets and outlets. It follows that any such permutation determines a permutation of the switches, and that the set G_s of such induced "switch permutations" is again a group, which we call the switch permutation group. The task of calculating G_n for a crossbar network is substantially simplified by first finding G_s.

In the language of group theory, the symmetry group G_n for a crossbar network must be an *imprimitive* group, because it consists of permutations that either preserve switches or map them onto each other. Indeed it can be seen that the map ϕ which assigns to each $g \epsilon G_n$ the switch permutation that g induces is a homomorphism of G_n onto G_s. Once a permitted switch permutation $p \epsilon G_s$ is chosen, each element in $\phi^{-1}\{p\}$ is determined by choosing, for each permuted outer switch, a map which

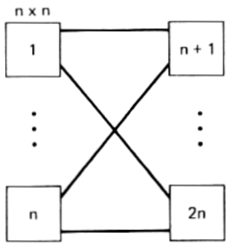


Fig. 15—Frame with "canonical" complete bipartite cross-connect field.

assigns the outer terminals of that switch to the outer terminals of its image under p. These maps are independent, so we have proved

Theorem 4: If ν is a crossbar network, then

$$G_n \approx G_s \times \prod_{0} S_{|0|}$$

where the product is over outer switches o, $|o|$ is the number of inlets (outlets) on o and S_k is the symmetric group on k objects.

For many crossbar networks made of stages it is rather straightforward to calculate the switch permutation group G_s, because G_s turns out to be isomorphic to a semidirect product of groups that are determined by the way the stages are joined together.

Example: *Frame*. It is clear that for a frame made of two identical stages, interconnected by the "canonical" (complete bipartite) cross-connect field as shown in Fig. 15, G_s will consist of all maps that take the inlet switches onto the outlet switches and vice versa, together with all maps that permute the inlet switches among themselves, and also the outlet switches among themselves. If each stage has n switches, then G_s is isomorphic to the largest imprimitive permutation group on $2n$ objects with two equinumerous sets of imprimitivity. If $1, \ldots, n$ are the inlet and $n + 1, \ldots, 2n$ are the outlet switches respectively, then

$$G_s = F_n \cup \sigma F_n$$

where F_n is the isomorph of $(S_n)^2$ in S_{2n} which permutes $1, \ldots, n$ among themselves and also $n + 1, \ldots, 2n$ among themselves and $\sigma = (1\ n + 1)(2\ n + 2)\cdots(n\ 2n)$. Each switch is $n \times n$, so

$$G_n \approx (F_n \cup \sigma F_n) \times (S_n)^{2n}$$

It can be verified that the union $F_n \cup \sigma F_n$ is in fact a group and a semidirect product. The order of G_s for the frame is $2(n!)^2$. For $n = 2$, $G_s \approx D_4$, the dihedral group of order 8.

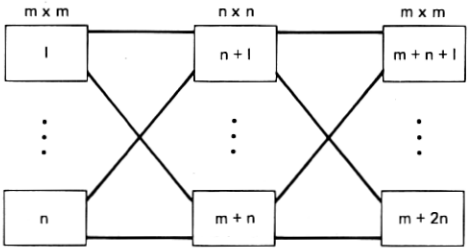


Fig. 16—Three-stage Clos network.

Example: *3-Stage Clos Network.* From Fig. 16 it is evident that G_s is the imprimitive group which permutes the middle switches, *and* either permutes the inlet switches and the outlet switches independently, or else maps inlet switches onto outlet and vice versa. If now K_n is the isomorph of $(S_n)^2$ in S_{m+2n} which permutes $1, \ldots, n$ and $m + n + 1, \ldots, m + 2n$ independently, and L_m is the isomorph of S_m in S_{m+2n} which permutes only $n + 1, \ldots, m + n$, then for the Clos network of Fig. 16,

$$G_s = (K_n \cup \tau K_n) \times L_m$$

where $\tau = (1\ 2n + 1) \cdots (n\ 3n)$, and $G_n \approx G_s \times (S_m)^{2n}$. For the case $m = n = 2$, treated in detail in Section XIV, we have $G_s \approx S_2 \times D_4$ and $G_n \approx (S_2)^5 \times D_4$. It is now easy to see that the $S_2 \times D_4$ structure of G_s arises from viewing the outer stages as a frame that yields D_4 as in the previous example, while the S_2 arises from permuting the middle switches.

Example: *Cascade of s stages with "complete bipartite" cross-connect.* The structure of this network is shown in Fig. 17. The parity of s plays a role here, as follows: If s is odd, then under a switch permutation the switches of the middle stage can only go into each other, and as we saw in the previous example, they contribute to G_s a group factor isomorphic

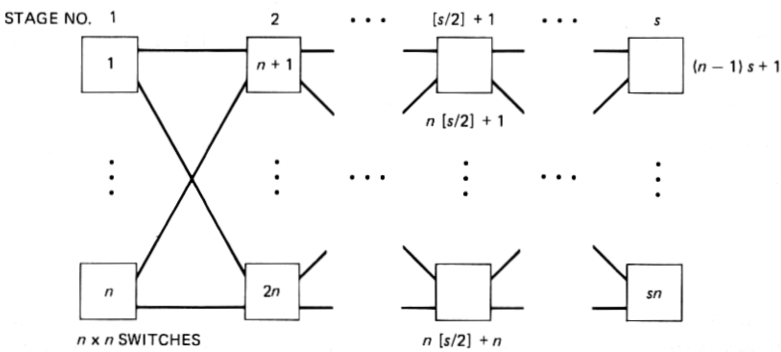


Fig. 17—Cascade of s stages with "complete bipartite" cross connect.

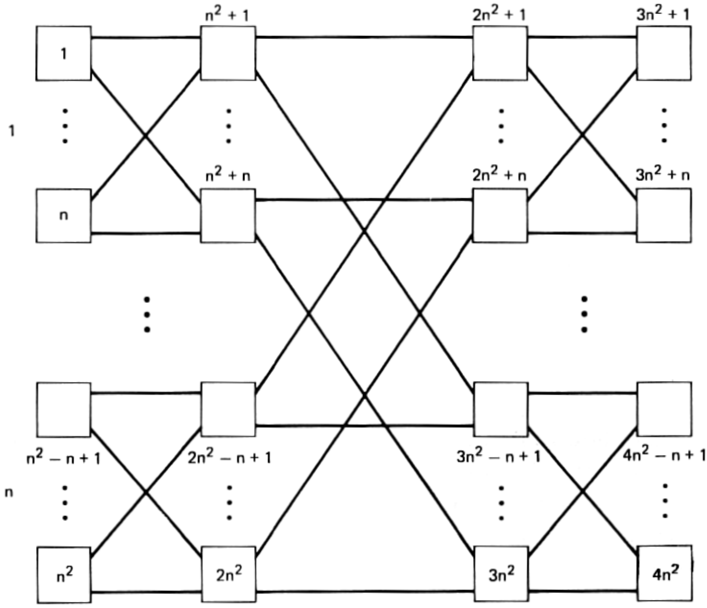


Fig. 18—No. 5 crossbar type network.

to S_n. Whatever the parity of s the noncentral switches contribute a factor isomorphic to an imprimitive group (*not* the largest) with $2[s/2]$ sets of imprimitivity; these are partitioned into pairs and the group elements either map each set onto itself, or each set onto its paired set, and never some of each. If the switches are numbered as in Fig. 17, this group is essentially the semidirect product

$$M_n \cup \theta M_n$$

where M_n is the largest subgroup of S_{sn} which fixes all the central switches and permutes vertically within every other stage, and

$$\theta = \prod_{k=0}^{[s/2]} (kn + 1\ (s - 1 - k)n + 1) \cdots (kn + n\ sn - kn)$$

Finally

$$G_s \approx (M_n \cup \theta M_n)(S_n)^{s-2[s/2]}$$

Example: *No. 5 Crossbar*. With the switches numbered as in Fig. 18, let P_n be the isomorph of $(S_n)^3$ in S_{4n^2} which for $k = 0,1,2,3$, permutes $kn^2 + 1, \ldots, (k + 1)n^2$ among themselves in such a way as to map frames onto frames, outer stages onto outer of the image frame, and inner stages onto inner stages of the image frame, and inner stages onto inner stages of the image frame. Let

$$\sigma = (1\ 3n^2 + 1)(2\ 3n^2 + 2)\cdots(n^2\ 4n^2) \times (n^2 + 1\ 2n^2 + 1)(n^2 + 2\ 2n^2 + 2)\cdots(2n^2\ 3n^2)$$

Then

$$G_s = P_n \cup \sigma P_n$$

$$G_n \cong (P_n \cup \sigma P_n) \times (S_n)^{2n^2}$$

The remainder of this paper derives the reduced state equations for the stochastic model of Ref. 2.

XVI. ROUTING OF CALLS

We shall use a routing matrix $R = (r_{xy})$ as a convenient formal description of how routes are chosen for calls. The class of routing matrices R can be described thus: for each $x \epsilon S$ let Π_x be the partition of A_x induced by the relation $\sim$ of "having the same calls up," or satisfying the same assignment of inlets to outlets; it can be seen that Π_x consists of exactly the sets A_{cx} for c free and not blocked in x; for $Y \epsilon \Pi_x$, r_{xy} for $y \epsilon Y$ is to be a probability distribution over Y, that is $r_{xy} \geq 0$ and $\sum_{y \epsilon Y} r_{xy} = 1$; r_{xy} is to be 0 in all other cases.

The interpretation of the routing matrix as a method of choice is to be this: any $Y \epsilon \Pi_x$ represents all the ways in which a particular call c (free and not blocked in x) *could* be completed when the network is in state x; for $y \epsilon Y$, r_{xy} is the chance (or fraction of times) that if call c arises in state x it will be completed by being routed in the network so as to take the system to state y. The distribution $\{r_{xy}, y \epsilon Y\}$ indicates how the calling-rate due to c is to be spread over the possible ways of putting up this call. Evidently, such a description of routing could be made time-dependent, and extended to cover refusal of unblocked calls as an option; we do not consider these possibilities here. The problem of choosing an optimal routing matrix R has been worked on at some length, in Refs. 9 and 10; its relations to state reduction are described in forthcoming sequels[5,6] to the present paper.

XVII. STOCHASTIC MODEL

We now recall[2] a stochastic model for the traffic offered to a network. A Markov stochastic process x_t taking values on S can be based on these simple probabilistic and operational assumptions:

(*i*) Holding times of calls are mutually independent variates, each with the negative exponential distribution of unit mean.

(*ii*) If u is an inlet idle in state $x \epsilon S$, and $v \neq u$ is any outlet, there is a conditional probability $\lambda h + o(h)$, $\lambda > 0$, as $h \to 0$, that u attempt a call to v in $(t, t + h)$ if $x_t = x$.

(*iii*) A routing matrix $R = (r_{xy})$ is used to choose routes, as follows: If $c = \{(u,v)\}$ is a call free and not blocked in x, then the fraction of times that the system pass from x to $y \epsilon A_{cx}$ if c arises when $x_t = x$ is just r_{xy}.

(*iv*) Blocked calls are declined, with no change of state.

It is convenient to collect these assumptions into a transition rate matrix $Q = (q_{xy})$, the generator of x_t; this matrix is given by

$$c_{xy} = \begin{cases} 1 \quad \text{if } y \epsilon B_x \\ \lambda r_{xy} \text{ if } y \epsilon A_x \\ -|x| - \lambda s(x) \text{ if } y = x, \text{ with } s(x) = |F_x| \\ \qquad 0 \quad \text{otherwise} \end{cases}$$

and the associated statistical equilibrium (or state) equations take the simple form

$$[|x| + \lambda s(x)]p_x = \sum_{y \epsilon A_x} p_y + \lambda \sum_{y \epsilon B_x} p_y r_{yx} x \epsilon S$$

where $\{p_x, x \epsilon S\}$ is the asymptotic distribution of x_t.

The remainder of this paper takes up the problem of correctly writing some *reduced* state equations for the stochastic model we have described.

XVIII. MULTIPLICITY

In an (unreduced) state x belonging to a structural equivalence class α there may be several calls in progress whose termination would carry the system into an equivalence class β. For example, in Fig. 13, for a state x in $\alpha = 7$ there are two calls whose ending would yield a state from 3, and there is one call leading to 2. Since we have assumed a unit hangup rate per call in progress, the downward transition rates between adjacent unreduced states is always unity; but as soon as we lump states into equivalence classes $\alpha, \beta, \ldots$ these rates in effect add up. A similar situation obtains for new calls: there may be several calls free and not blocked in a state $x \epsilon \alpha$ whose completion could (under a suitable choice of route) lead to a state in equivalence class β.

We call this phenomenon *multiplicity,* and we need an account of it in order to define transition rates between reduced states, and to correctly use[5] and understand the reduced state equations.

The "hangup" matrix $H = (h_{xy})$ associated with a network ν is a matrix of zeros and ones that tells which states can be reached from which by a hangup:

$$h_{xy} = \chi_{y \epsilon B_x} = \chi_{x \epsilon A_y}$$

It can be seen that H admits the automorphism subgroup G_η, i.e., that

$$h_{xy} = h_{(gx)(gy)} \qquad g \epsilon G_\eta$$

H is so-called because if we make the standard traffic assumption that all calls in progress terminate independently of any past history at unit rate, then H is precisely the part of the transition rate matrix due to "hangups."

The transition rate, due to hangups, from a state x to an equivalence class β is

$$\sum_{y \epsilon \beta} h_{xy}$$

We claim that this number is the same for all x in an equivalence class α, and that it is the hangup rate from α to β. There are two ways of proving this result, which will be put in the following form:

Lemma 1: The numbers $|B_x \cap \beta|$ and

$$\sum_{y \epsilon \beta} h_{xy}$$

are equal; they assume the same value for all x in an equivalence class α, and represent the hangup rate $h_{\alpha\beta}$ from α to β.

Proof: Equality is obvious. The numbers are zero unless α covers β in the partial ordering on E. Let $x \epsilon \alpha$, $z \epsilon \alpha$, $z = gx$ where g is an automorphism of $(S, \leq)$. If now $y \epsilon B_x \cap \beta$, then $gy \epsilon B_{gx} = B_z$, and $gy \epsilon \beta$ because β is an equivalence class. Conversely if $y \epsilon B_z \cap \beta$, then $g^{-1}y \epsilon B_x \cap \beta$, by the same argument, so $B_z \cap \beta = g(B_x \cap \beta)$. The result follows because g is a bijection.

Alternative proof: With $x \epsilon \alpha$, $z \epsilon \alpha$, $z = gx$ as above, consider that where w is any element of β

$$\begin{aligned}
|B_{gx} \cap \beta| &= \sum_{y \epsilon \beta} \chi_{y \epsilon B_{gx}} \\
&= \frac{|\beta|}{|G_\eta|} \sum_{f \epsilon G_\eta} \chi_{gfw \epsilon B_{gx}} = \frac{|\beta|}{|G_\eta|} \sum_{f \epsilon G_\eta} h_{(gx)(gfw)} \\
&= \frac{|\beta|}{|G_\eta|} \sum_{f \epsilon G_\eta} h_{x(fw)} = \sum_{y \epsilon \beta} h_{xy} = |B_x \cap \beta|
\end{aligned}$$

The multiplicity question is more complicated for new calls than it is for hangups; this is because only one state can arise by removing a call c in progress, whereas possibly one of several could arise by completing a call c which is new and is not blocked; as a result, the calculation of transition rates between reduced states depends on routing decisions as well as on multiplicity. For by itself multiplicity will only provide the calling rates which might take the state into a reduced state β: whether a particular transition occurs depends, however, on what state x the system is in and what routing policy is being followed. We shall see, though, that if the routing is *consistent* in the sense that it routes analogous calls analogously in equivalent states, then these calling rates will depend only on πx and β, i.e., on the reduced states in question. Mathematically this idea of consistency is expressed by the condition that the routing matrix $R = (r_{xy})$ *admit* the group G_η: if c is a call free and not blocked in x, and $\tau \epsilon G_\eta$, then in the terminology above, c and τc are analogous calls, x and τx are equivalent states, and $\tau(A_{cx}) = A_{(\tau c)(\tau x)}$, so consistency amounts to asking that we have $r_{xy} = r_{(\tau x)(\tau y)}$ for $y \epsilon A_{cx}$, i.e., that R admit G_η.

The matrix $N = (n_{xy})$ associated with a network ν is a matrix of zeros and ones that tells which states can be reached from which by adding a new call; it is obviously the transpose of the hangup matrix H:

$$n_{xy} = h_{yx} = \chi_{x \epsilon B_y} = \chi_{y \epsilon A_x}$$

Thus N admits the automorphism subgroup G_η, and we have this analog of Lemma 1, with a similar proof.

Lemma 2: The numbers $|A_x \cap \beta|$ and

$$\sum_{y \epsilon \beta} n_{xy} (= n_{\alpha\beta})$$

are equal; they assume the same value $n_{\alpha\beta}$ for all x in an equivalence class α, and they represent the number $n_{\alpha\beta}$ of calls free and not blocked in x which could be put up so as to lead to a state of β.

To calculate the actual transition rate from a state x into an equivalence class β, we must use the routing matrix R. Notice that R is just N with enough of the ones reduced (but still ≥ 0) so that r_{xy} summed over $y \epsilon A_{cx}$ is unity for each x and $c \epsilon F_x$.

For the traffic model assumed in Section XVII run according the routing matrix R the transition rate from a state x into an equivalence class β that intersects A_x is

$$\sum_{\substack{c \text{ free in } x \\ c \text{ not blocked}}} \sum_{y \epsilon \beta \cap A_{cx}} r_{xy} \left(= \sum_{y \epsilon \beta \cap A_x} r_{xy} \right)$$

Notice that replacement of each r_{xy} by 1 in this expression increases its value to precisely

$$n_{\alpha\beta} = |A_x \cap \beta| = |\{c : \beta \epsilon \pi (A_{cx})\}|$$

Lemma 3: If R admits G_η, then the numbers

$$\sum_{y \epsilon \beta \cap A_x} r_{xy} \ (= r_{\alpha\beta})$$

are the same for $x \epsilon \alpha$.

Proof: Like that of Lemmas 1 and 2.

We put

$A_\alpha = \{\beta \epsilon E$: β covers α in the partial order of E induced by $\leq\}$

$B_\alpha = \{\beta \epsilon E$: α covers β in the partial order of E induced by $\leq\}$

These are the reduced analogs of the sets A_x and B_x, useful for writing the state equations. It can be seen that

$$A_\alpha = \bigcup_{x \epsilon \alpha} \pi(A_x)$$

$$B_\alpha = \bigcup_{x \epsilon \alpha} \pi(B_x)$$

Let us set

$$q_{\alpha\beta} = \begin{cases} h_{\alpha\beta} & \beta \epsilon B_\alpha \\ \lambda r_{\alpha\beta} & \beta \epsilon A_\alpha \\ -[\lambda s(x) + |x|] & x \epsilon \alpha, \quad \alpha = \beta \\ 0 & \text{otherwise} \end{cases}$$

We can now informally describe the reduced equations as follows: when R admits G_η, they correspond to equilibrium equations for a Markov process πx_t on E whose transition rates up and down are $\lambda r_{\alpha\beta}$ and $h_{\alpha\beta}$ respectively, with the obvious necessary convention on the "diagonal." In terms of the notation introduced above, these reduced equations are

$$-p_\alpha q_{\alpha\alpha} = \sum_{\beta \epsilon A_\alpha} p_\beta h_{\beta\alpha} + \lambda \sum_{\beta \epsilon B_\alpha} p_\beta r_{\beta\alpha} \tag{2}$$

with $q_{\alpha\alpha} = -[\lambda s(x) + |x|]$ for any $x \epsilon \alpha$.

XIX. REDUCTION OF STATE EQUATIONS

We can now give precise explicit conditions under which the original "microscopic" equations of state can be exactly replaced by the less numerous "macroscopic" or reduced equations for the probabilities of equivalence classes. It will be shown that if the routing matrix admits G_η then the original and the reduced equations imply each other, and that the state probabilities $\{p_x, x \epsilon S\}$ are constant over equivalence classes α. Only the equilibrium case is considered.

Theorem 5: Let the routing matrix R admit G_η. Then

(*i*) The "microscopic" transition rate matrix

$$Q = H + \lambda R - \text{diag}[\lambda s(x) + |x|]_{x \epsilon S}$$

also admits G_η.

(*ii*) The projection map $\pi{:}x \to \pi x$ defines a "macroscopic" Markov process πx_t on the $\equiv$ – equivalence classes with transition rate matrix

$$q_{\alpha\beta} = \begin{cases} h_{\alpha\beta} + \lambda r_{\alpha\beta} & \alpha \neq \beta \\ q_{xx} \qquad \alpha = \beta, & x \epsilon \alpha \end{cases}$$

(*iii*) If $p_\alpha = \sum_{x\epsilon\alpha} p_x$, then the equilibrium equations $\sum_x p_x q_{xy}$ and $\sum_\alpha p_\alpha q_{\alpha\beta}$ imply each other.

(*iv*) If $\{p_\alpha, \alpha\epsilon\pi S\}$ solves the reduced equations, then $\{p_x, x\epsilon S\}$ defined by

$$p_x = \frac{p_\alpha}{|\alpha|}, \quad x\epsilon\alpha$$

solves the original equilibrium equations.

Proof: (*i*) is clear from the hypothesis that R admits G_η, from Lemmas 1 and 3, and from the fact that $s(x) = s(\tau x)$, $|x| = |\tau x|$ for $\tau\epsilon G_n$. To prove (*ii*) it is enough, by (Lloyd's) Theorem 1, to prove that

$$\sum_{y\epsilon\beta} q_{xy}$$

are constant for $x\epsilon\alpha$. For $\alpha = \beta$ this follows from invariance of $s(\cdot)$ and $|\cdot|$ under $\tau\epsilon G_n$. If β covers α in the induced partial order on πS, then

$$\sum_{y\epsilon\beta} q_{xy} = \lambda \sum_{y\epsilon\beta} r_{xy} = \lambda r_{\alpha\beta}$$

Here we have used Lemma 3. Similarly if α covers β, then for $x\epsilon\alpha$

$$\sum_{y\epsilon\beta} q_{xy} = \sum_{y\epsilon\beta} h_{xy} = |B_x \cap \beta| = h_{\alpha\beta}$$

by Lemma 1; thus (*ii*) is proved.

To prove the reduced equations from the original ones, we sum the latter over $x\epsilon\alpha$, to get

$$\sum_{x\epsilon\alpha} p_x[\lambda s(x) + |x|] = \sum_{x\epsilon\alpha}\sum_{y\epsilon A_x} p_y + \lambda \sum_{x\epsilon\alpha}\sum_{y\epsilon B_x} p_y r_{yx}$$

Since $s(\cdot)$ and $|\cdot|$ are constant over α, the left-hand side is just $-p_\alpha q_{\alpha\alpha}$. For the first term on the right, argue thus: since $h_{yx} = 0$ unless $y\epsilon A_x$ and

$$\sum_{\beta\epsilon A_\alpha}\sum_{y\epsilon\beta}$$

sums over (at least) all $y\epsilon A_x$ if $x\epsilon\alpha$, we have

$$\begin{aligned}\sum_{x\epsilon\alpha}\sum_{y\epsilon A_x} p_y &= \sum_{\beta\epsilon A_\alpha}\sum_{y\epsilon\beta} p_y \sum_{x\epsilon\alpha} h_{yx} \\ &= \sum_{\beta\epsilon A_\alpha}\sum_{y\epsilon\beta} p_y |B_y \cap \alpha| \\ &= \sum_{\beta\epsilon A_\alpha} p_\beta h_{\beta\alpha}\end{aligned}$$

using Lemma 1. Similarly, since $r_{yx} = 0$ unless $h_{xy} = 1$, we find

$$\sum_{x\epsilon\alpha} \sum_{y\epsilon B_x} p_y r_{yx} = \sum_{x\epsilon\alpha} \sum_{\beta\epsilon B_\alpha} \sum_{y\epsilon\beta} p_y r_{yx} h_{xy}$$

$$= \sum_{\beta\epsilon B_\alpha} \sum_{y\epsilon\beta} p_y \sum_{x\epsilon\alpha} r_{xy}$$

$$= \sum_{\beta\epsilon B_\alpha} p_\beta r_{\beta\alpha}.$$

This proves the reduced equations from the original; the original ones can be proved from the reduced by reversing the identities used above. Neither argument depends on the fact (*iv*), to be proved next, that p_x is constant over $x\epsilon\alpha$.

To prove (*iv*) we shall set $p_x = p_{\pi x}/|\pi x|$ in the unreduced equations and obtain an identity. Substitution in eq. (2) and multiplication by $|\alpha|$ gives, for $x\epsilon\alpha$,

$$p_\alpha[\lambda s(x) + |x|] = |\alpha| \sum_{y\epsilon A_x} \frac{p_{\pi y}}{|\pi y|} + \lambda |\alpha| \sum_{y\epsilon B_x} \frac{p_{\pi y}}{|\pi y|} r_{yx}$$

$$= |\alpha| \sum_{\beta\epsilon A_\alpha} p_\beta \frac{1}{|\beta|} \sum_{y\epsilon\beta} h_{yx}$$

$$+ \lambda |\alpha| \sum_{\beta\epsilon B_\alpha} p_\beta \frac{1}{|\beta|} \sum_{y\epsilon\beta} r_{yx}$$

$$= \sum_{\beta\epsilon A_\alpha} p_\beta \frac{1}{|\beta|} \sum_{y\epsilon\beta} \frac{|\alpha|}{|G_\eta|} \sum_{g\epsilon G_\eta} h_{(gy)x}$$

$$+ \lambda \sum_{\beta\epsilon B_\alpha} p_\beta \frac{1}{|\beta|} \sum_{y\epsilon\beta} \frac{|\alpha|}{|G_\eta|} \sum_{g\epsilon G_\eta} r_{x(gy)}$$

$$= \sum_{\beta\epsilon A_\alpha} p_\beta \frac{1}{|\beta|} \sum_{y\epsilon\beta} \frac{|\alpha|}{|G_\eta|} \sum_{g\epsilon G_\eta} h_{y(gx)}$$

$$+ \lambda \sum_{\beta\epsilon B_\alpha} p_\beta \frac{1}{|\beta|} \sum_{y\epsilon\beta} \frac{|\alpha|}{|G_\eta|} \sum_{g\epsilon G_\eta} r_{(gx)y}$$

$$= \sum_{\beta\epsilon A_\alpha} p_\beta \frac{1}{|\beta|} \sum_{y\epsilon\beta} \sum_{x\epsilon\alpha} h_{yx}$$

$$+ \lambda \sum_{\beta\epsilon B_\alpha} p_\beta \frac{1}{|\beta|} \sum_{y\epsilon\beta} \sum_{x\epsilon\alpha} r_{yx}$$

$$= \sum_{\beta\epsilon A_\alpha} p_\beta h_{\beta\alpha} + \lambda \sum_{\beta\epsilon B_\alpha} p_\beta r_{\beta\alpha}$$

Since the left side is $-p_\alpha q_{\alpha\alpha}$, the substitution has resulted in the reduced equations. Since the solution p of the original equations is unique, (*iv*) is established.

XX. ACKNOWLEDGMENTS

The author is indebted to S. P. Lloyd for discussion and access to personal notes, to R. L. Graham for helping to formulate ideas and produce counterexamples, and to N. J. A. Sloane for helping to identify some of the groups that arose.

REFERENCES

1. S. P. Lloyd, "Markov Chains Admitting Symmetry Operations," unpublished Bell Laboratories memorandum, September 19, 1955.
2. V. E. Beneš, "Markov Processes Representing Traffic in Connecting Networks," B.S.T.J., *42,* No. 6 (November 1963), pp. 2795–2837.
3. A. F. Bulfer, "Blocking and Routing in Two-Stage Concentrators," Conference Record, National Telecommunications Conference, New Orleans, Dec. 1–3, 1975./4w 4. A. F. Bulfer, U.S. Patent No. 3,935,394, January 27, 1976.
5. V. E. Beneš, "Optimal Routing in Networks Whose States are Reduced Under Symmetries," to appear.
6. V. E. Beneš, "Optimal Routing in Some Two-Stage Concentrators," to appear.
7. C. J. Burke and M. Rosenblatt, "A Markovian Function of a Markov Chain," Ann. Math. Stat., *29* (1958), pp. 1112–1122.
8. S. P. Lloyd, private communication.
9. V. E. Beneš, "Programming and Control Problems Arising from Optimal Routing in Telephone Networks," B.S.T.J., *45,* No. 9 (November 1966), pp. 1373–1438.
10. V. E. Beneš, Optimal routing in connecting networks over finite time intervals, B.S.T.J., *46,* No. 10 (December 1967), pp. 2341–2352.