Proceedings of the 1984
Winter Simulation Conference
S. Sheppard, U. Pooch, D. Pegden (eds.)

TOPOLOGICAL OPTIMIZATION OF AN INTEGRATED CIRCUIT/PACKET-SWITCHED COMPUTER NETWORK

Mark J. Kiemele
United States Air Force
Udo W. Pooch
Texas A&M University

ABSTRACT

This paper presents a methodology which can be used to optimize the topology of an integrated circuit/packet-switched computer-communication network. This special kind of network possesses a circuit-switched backbone with various packet-switched local access networks feeding into the communications subnet. An iterative, heuristic approach is used to generate a sequence of suboptimal solutions in lieu of one optimal solution. Application of the methodology shows that it is a flexible tool that can be used to optimize the design of integrated networks.

INTRODUCTION

The explosive evolution of the computer and communications technologies has resulted in a marriage which is rapidly forming the basis of our information society. A computer-communication network may be described as an interconnected group of independent computer systems which communicate with one another [8]. The reasons for communicating are varied. They may include the sharing of resources, such as programs, data, hardware, or software; or it may be that a computer system is used solely as a communications processor for information transfer and/ or controlling terminals.

Sometimes a subtle distinction is made between a computer-communication network and a computer network. The difference between the two, according to Elovitz and Heitmeyer [18], is that in a 'computer-communication', the user is responsible for managing the computer resources; while in a 'computer network', the resources are managed automatically by a network operating system. In this paper, the two terms will not be distinguished and will be used interchangeably, along with the term 'network'.

The emphasis of this paper is on the topology of a network. Hence, the graph-theoretic aspects of a network are of major importance. In this regard, a computer network consists of a communications subnet (or backbone) together with the facilities needed to gain access to the subnet. The backbone of the network is comprised of communications processors (nodes) and trunk lines (links) which interconnect the nodes. The terms 'vertices' and 'nodes' are synonymous, and the terms 'link', 'arc', and 'edge' all may be used to denote a connector between two vertices in a graph or a transmission medium connecting two nodes in a network.

In this paper we present a methodology which can be used to design a special kind of computer network. We provide some necessary background on the particular computer network being considered. We motivate an integrated circuit/packet-switched computer network and describe briefly a simulation model used to obtain performance data from such a network. Finally, address the topological design issues of an integrated network, and section we address present the results of applying the developed design methodology to various sets of integrated network specifications.

AN INTEGRATED CIRCUIT/PACKET-SWITCHED COMPUTER NETWORK

As used here, the term "integrated circuit/packet-switched computer network" denotes a distributed computer network possessing a circuit-switched backbone or subnet with numerous packet-switched local access networks feeding into the communications subnet.

The Case for an Integrated Network

Current military communication systems are generally designed to handle either voice calls or data transactions but not both. Such deployed systems use separate facilities for the two classes of traffic, thereby magnifying both the manpower and maintenance problems that already exist. The grade of service for these systems is usually satisfactory, but crisis situations can and do force traffic flows that exceed system capabilities [3]. Recent Defense Communication Agency (DCA) studies have shown the Department of Defense's (DOD's) intent to implement an all-digital, integrated network that would be operational early in the next decade [3,11, 49, 54]. Such a network would transmit voice and multiple classes of data (e.g., interactive, bulk, facsimile) simultaneously on a common transmission medium. The feasibility of such networks has been demonstrated by Dysart et al. [17] who contend that "the future for fully digital integrated voice and data transmission is very promising". The concept of integrating voice and data rests on the fact that speech can be digitized and thus can be handled under packet switching schemes.

Other recent studies have also addressed the problem of transmitting voice and data in the same computer-communication network [12, 19, 20, 23, 32, 35, 41, 47, 50, 51, 52, 53, 55, 57]. Gruber [34] aptly summarizes these studies by stating that "the motivations for considering mixed voice and data traffic...include: the advent of new voice related applications with the technology now existing to economically support them, and...economy and flexibility. Perhaps the ultimate objectives of integration are...to realize the economics of equipment commonality, large-scale integration, higher resource utilization, and combined network operations, maintenance, and administrative policies".

The scenario to accomplish such integration has also been investigated [3, 11, 12, 17, 23, 37, 38, 44, 45, 52, 53, 57, 59, 60]. Despite the many tradeoffs between packet switching and circuit switching,: the consensus is that circuit switching delays have been improved to the point where both circuit switching and packet switching can be employed advantageously in the same network [6]. Hsieh et al. [37] and Rudin [53] emphasize that the integrated circuit/packet-switched network will become more prevelant in the future, eventually replacing circuit-switched or packet-switched systems.

Integrated Network Performance

Performance evaluation is a concern of all network designers. In networks, the complexity of the system design and the interrelationships of system components in a myriad of combinations makes any evaluation study a difficult task. Equally difficult is the task of generating accurate performance data upon which a performance evaluation can be based. To facilitate both performance data generation and evaluation, an integrated network simulation model has been developed. The simulator is a modified version of a network simulation model developed by Clabaugh [10]. The following paragraphs give a brief description of the simulator.

As implemented in the simulator, an integrated circuit/packet-switched network consists of the following major components (Figure 1):

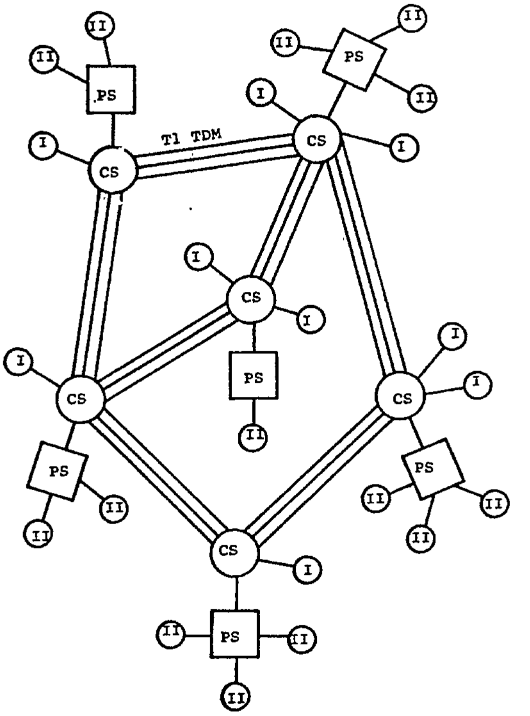


Figure 1. Integrated Network Configuration

A. Backbone Circuit Switch (CS) Nodes
B. Peripheral Packet Switch (PS) Nodes
C. Invariant Network Synchronous Time-Division-Multiplexed (TDM) Frame Switching Super-structure
D. Digital Network Using T1 Carriers and Digital Switching Nodes
E. Variable Subscriber Data Rates
F. Two Classes of Subscriber Traffic

1. Class I: Real-time traffic that once started cannot be interrupted (voice, video, facsimile, and sensor).
2. Class II: The general class of packet data, such as interactive, bulk, and narrative/message.

The backbone CS nodes and peripheral PS nodes form the nucleus of a distributed computer-communication network in which the transmission of data and voice between any two nodes on the subnet is accomplished by sharing the capacity of the T1 link. A Slotted Envelope Network (SENET) self synchronizing concept [13] is used to achieve simultaneous transfer of voice and data on the carrier. This concept treats the available bandwidth on a digital link as a resource for which all forms of communication must compete. Using SENET, the T1 link is synchronously clocked into frames of a fixed time duration, b, which are assumed invariant throughout the network. Each frame is partitioned into several data slots (channels) for which the various traffic types compete (Figure 2).

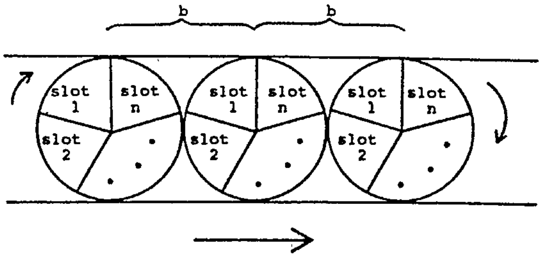


Figure 2. SENET Frame Clocking

Voice (Class I) subscribers are terminated directly at circuit switches to avoid packetizing and any unnecessary routing overhead through packet switches. Similar to a telephone dial-up process, a Class I transaction results in a physical end-to-end connection for the duration of the call or a system loss (blocking) occurs. Although not a design requirement, packet switches are co-located with the circuit switches. All data (Class II) subscribers are terminated at packet switches. While the Class II subscriber packet switch interface depends upon the individual terminal hardware configuration, the transmission of data between the packet and circuit switches is accomplished using TDM. The packet switches are primarily responsible for managing the movement of packets between input terminals and the circuit switches; placing traffic in queues according to a regional routing policy; and performing connection initiation, circuit disconnect, and coordination with other packet switches, depending on network loading.

The regional routing doctrine for each packet switch, coupled with virtual switch connections, reduces overhead and the traffic congestion problem. As traffic is entered into a packet switch from subscriber terminals, it is queued for the relevant destination packet switch. A circuit switch connection is then initiated/terminated by the packet switch on behalf of this traffic.

Progressive alternate routing is used on the backbone of the network. With this method, each circuit switch node has a primary and an alternate path. If blocking occurs at some node during connection initiation, the alternate route is tried for route completion. If this connection fails, the transaction is either queued at the packet node or considered a system loss at the circuit node, depending on its class.

The network simulation model requires 16 input parameters. Among these are number of nodes, number of links, number of time slots for each link, and the arrival rates at each packet and circuit switch. Due to the large number of system input parameters, the simulator is capable of generating a wide variety of performance measures. These include mean packet delay, average link utilization, packet throughput, average queue length, and fraction of calls blocked.

THE TOPOLOGICAL DESIGN PROBLEM

The topological design problem for an integrated computer network can be stated as follows [2, 9, 28, 58]:

Given:	packet switch and circuit switch node locations; data traffic requirement between packet node pairs; voice traffic requirement between circuit node pairs; cost/capacity matrix
Minimize:	cost of the integrated network
Subject to:	reliability constraint packet delay or throughput constraint voice call blocking constraint link utilization constraint
Variables:	link placement link capacity

Other common formulations of the design problem are the following:

1) minimize mean packet delay given a cost constraint, and

2) maximize throughput given cost and delay constraints.

However, it has been shown [27, 28] that all of these formulations are closely related and that the solution techniques that apply to the stated formulation above also apply to the other formulations.

The topological design of an integrated network means assigning the links and link capacities that interconnect the circuit-switched, backbone nodes. The nodes, locations, or sites are the sources and sinks of the information flow. The data traffic matrix specifies how many packets per second must be sent between nodes i and j. Similarly, the voice traffic matrix tells how many calls per minute are initiated at node i and directed to node j. The cost/capacity matrix gives the cost per unit distance for each of the various speed transmission links available, as well as the fixed charge for each line type. There are generally only a discrete number of link capacities (speeds) available, e.g.,50 Kbits/sec, 500 Kbits/sec, 1 Mbits/sec.

The specification of constraints establishes a grade of service for the network. The reliability constraint is usually given in terms of k-connectivity. When k=2, a most common situation [21], the biconnectivity constraint indicates that there must be at least two node-disjoint paths between every pair of nodes. Mean packet delay is a common delay constraint, and it is usually given in terms of "not to exceed a specified number of milliseconds or seconds". Call blocking and link utilization are usually given in percent or a decimal fraction between 0 and 1.

AN ITERATIVE APPROACH TO NETWORK DESIGN

The goal of any topological design procedure is to achieve a specified performance at minimum cost. The design problem as stated above can be formulated as an integer programming problem, but the number of constraint equations quickly becomes unmanageable for even small problems. In fact, the optimal topological design solution for networks with greater than ten nodes is believed to be computationally prohibitive [9, 27]. This is indeed plausible since Garey and Johnson [26] have shown the network reliability problem, which is a subproblem of the topology design problem, to be NP-hard. A viable alternative to finding the optimal solution is to use a computationally efficient heuristic to generate suboptimal solutions. This is the approach taken in this study.

The technique is to generate a starting topology and evaluate this topology using the simulation model described in above. The network topology is then perturbed in a manner determined by a heuristic. The heuristic uses the performance data obtained from the simulation to move the topology in the direction of satisfying the set of constraints. The perturbed topology is once again evaluated via simulation and the heuristic reapplied. The performance feedback mechanism, or heuristic, is applied repeatedly after each simulation until all of the performance constraints are satisfied, if possible. Once all of the constraints are satisfied, a "feasible" solution has been obtained. The model continues to try to improve on the feasible solution by repeated use of the heuristic until it can no longer do so, at which time the best feasible solution is considered to be a "local optimum". The iterative approach is depicted in Figure 3 where the heuristic is shown as a performance feedback mechanism tying the three main modules of the model in a loop.

The entire procedure can be repeated with other starting topologies, thereby generating a sequence of local optima. The situation of having many suboptimal solutions rather than the optimal solution is not all that bleak. Many factors usually enter into a design decision and modeling and analysis may be just one of them. The existence of several appropriate solutions could very well increase the flexibility of incorporating other nontechnical factors (e.g., political considerations) into the design decision.

STARTING TOPOLOGY GENERATION

The input to the starting topology generation process is the set of node locations, the traffic requirements, and the cost/capacity information for the set of leased lines available. The output of this process is a set of links with one of a discrete set of link capacities assigned to each existent link. That is, an integer matrix C can be used to describe a topology. If $c_{ij} = 0$, then there is no link between nodes i and j. Positive integers for c_{ij} indicate the link type or capacity between nodes i and j. Matrix C is referred to as the topology connectivity matrix and is symmetric with respect to the main diagonal, always a set of zeros. The process of generating a starting between nodes 3 and 6, resulting in a connected topology. Application of the algorithm guarantees a connected, but not necessarily k-connected network.

Link Capacity Assignment

The selection of link capacities from a finite set of options is one of the most difficult of all network design problems [22, 28, 31]. Furthermore, because digitized voice and packet data are being superimposed on a common carrier, the capacity assignment problem is even more complex in an integrated network than in either the circuit-switched or packet-switched network. There are no closed form solutions even in the case where link cost is a linear function of channel capacity [31], hardly a realistic situation [15, 17, 28, 36, 58].

Complicating matters is the fact that the capacity assignment problem is intimately related to the routing problem. In order to properly assign link capacities, an estimate of the traffic load on each link is needed. But the traffic on a link is highly dependent on the routing scheme used. Extensive research has been conducted on the design and analysis of routing algorithms, and the literature abounds with routing classification schemes [4, 24, 29, 33, 42, 43]. One simple classification of routing schemes is found in Tanenbaum [58] who categorizes routing algorithms as either static (fixed) or dynamic (adaptive). Dynamic algorithms base their routing decisions on the current load, so consequently it is difficult to estimate traffic loads when adaptive routing is used in a network. The possibility of designing a network using one routing algorithm and operating the resulting network with another routing algorithm is real and could result in poor performance. Fortunately, it has been shown [7, 27] that the performance of fixed multiple-path routing approximates that of optimal adaptive routing under stable conditions, i.e., where the traffic load in a network is not concentrated between a small percentage of the nodes.

The approach taken to assign link capacities in an integrated network is based on the shortest distance criterion, a commonly used strategy in the literature [14]. An outline of the procedure is as follows:

1) For each pair of nodes, A and B, find the shortest path between nodes A and B and label the path as $X_1X_2...X_n$, where $A=X_1$, $B=X_n$, and $X_2, X_3, ..., X_{n-1}$ are the intermediate nodes on the shortest path between A and B. Floyd's algorithm [16, 46] is used to determine the shortest path between each pair of nodes.
2) For a given node pair A and B, add the packet traffic load of (A, B) or (B, A), whichever is greater, to each link on the path from A to B. Similarly, add the voice traffic load of (A, B) or (B, A), whichever is greater, to each link on the path. A voice digitization conversion rate (e.g., 32 Kbits/sec) is used to transform voice arrival rate units to link capacity units (bps).
3) Repeat step 2 for each node pair.
4) For each link in the network, assign the smallest discrete link capacity that is greater than or equal to the estimated integrated traffic load determined in step 2. If there is no such option available, then assign the largest available link capacity.

Suppose it is assumed that each of the 10 circuit switch nodes in Figure 4 has an average voice call arrival rate of 4 calls per minute and that associated with each circuit switch node is a packet switch node having a data packet arrival rate of 400 packets (1000 bits/packet) per second. Application of the above heuristic under the assumed uniformly distributed network traffic load results in the starting topology shown in Table 1. The topology is given as a connectivity matrix. This table also shows the cost and capacity information for the five options as well as the coordinates of the randomized nodes in the backbone. Under these conditions the cost of this starting topology is $2112.39 (a monthly charge).

Table 1. Starting Topology Connectivity Matrix

```
LINE CAPACITY AND COST INFORMATION:
LINE   CAPACITY   COST($) PER   FIXED
TYPE    (BPS)     UNIT LENGTH  COST($)
  1     800000.          1.00    50.00
  2    1000000.          2.00    50.00
  3    1200000.          4.00   100.00
  4    1600000.          8.00   150.00
  5    2000000.         16.00   200.00

THE RANDOMIZED NODES:
   I       X(I)       Y(I)
   1      25.50      13.80
   2      23.10      10.90
   3      19.40      10.50
   4      13.40       6.10
   5       1.10      11.10
   6      15.10      13.90
   7      14.30       4.20
   8       2.80      17.20
   9       2.40       8.30
  10      24.30      12.40

THE CONECT MATRIX NOW LOOKS LIKE:
    1  2  3  4  5  6  7  8  9 10
 1  0  1  0  0  0  0  0  0  0  4
 2  1  0  2  0  0  0  0  0  0  0
 3  0  2  0  0  0  5  0  0  0  5
 4  0  0  0  0  0  4  3  0  0  0
 5  0  0  0  0  0  0  0  5  4  0
 6  0  0  5  4  0  0  0  5  0  0
 7  0  0  0  3  0  0  0  0  1  0
 8  0  0  0  0  5  5  0  0  0  0
 9  0  0  0  0  4  0  1  0  0  0
10  4  0  5  0  0  0  0  0  0  0

THE COST($) OF THIS TOPOLOGY IS:    2112.39
```

NETWORK TOPOLOGY OPTIMIZATION

Once a starting topology has been generated and the corresponding network performance and cost ascertained, it remains to determine whether or not modifications to the topology can enhance performance or decrease the cost or both. The procedure used to modify a topology with the expectation of improving performance and/or decreasing cost stands as the crux of any iterative approach to network design. Most of the techniques seen in the literature are geared to localized transformations or minor modifications that hopefully progress in a stepwise fashion to a local minimum.

Perturbation Techniques

Several sources in the literature [1, 9, 14, 25, 28, 30, 39, 40, 58] provide comprehensive surveys of heuristic algorithms that have been or are being used in network design. Three of these heuristics stand out as milestone approaches to perturbing a topology, that is, modifying the number of links and/or the capacity of links. These three approaches, all of which have been applied almost strictly to either packet switching or circuit switching networks, are

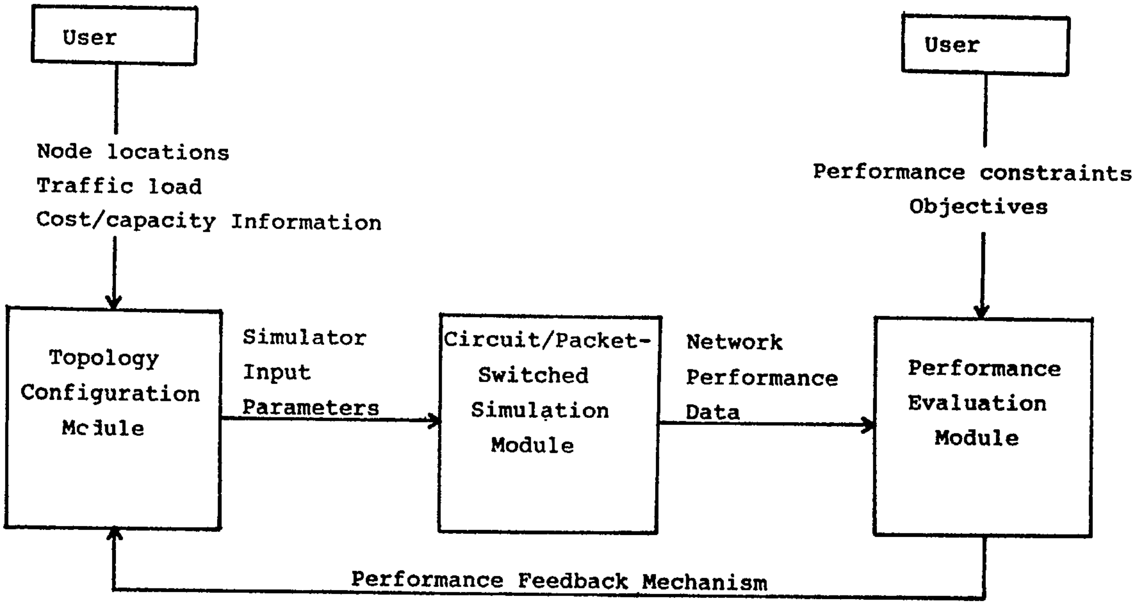

User
Node locations
Traffic load
Cost/capacity Information
User
Performance constraints
Objectives
Topology Configuration Module
Simulator Input Parameters
Circuit/Packet-Switched Simulation Module
Network Performance Data
Performance Evaluation Module
Performance Feedback Mechanism

Figure 3. Iterative Approach

topology consists of two major steps. Each of these steps is now considered in detail.

Link Assignment

If the link assignment process were such that the resulting initial topology were feasible, many of the subsequent problems involved in heuristic selection could be avoided. The technique implemented in the model is not quite so ambitious and does not guarantee a feasible starting topology. Instead, the approach is aimed at efficiently generating a network that is reasonably close to feasibility but at the same time has a relatively low cost. The algorithm used is a modified version of a heuristic due to Steiglitz et al. [56]. The heuristic is based on Whitney's theorem [6] which essentially states that if a network topology is k-connected, then every node in the network must have degree of at least k. The degree of a node is the number of links or arcs incident upon that node. Links in a network topology are undirected edges in a graph whose vertices correspond to the nodes in the backbone of the network. Although the condition that each node be of degree k is a necessary condition in a k-connected network, this condition is not sufficient [61].

The approach taken is called a link deficit approach. The difference between the required number (k) of links needed at a node and the actual degree of that node is called the link deficit for that node [58]. The algorithm can be described as follows:

1) The nodes are randomly numbered. It is the randomization of nodes that renders the algorithm nondeterministic and which allows many starting topologies to be generated from the same input data.
2) Determine the node with the highest link deficit. Call it A. Ties are broken by the ordering of nodes.
3) Determine the node with the highest link deficit that is not already linked to node A. Call it B. Ties are broken by using minimum distance from A as a criterion or by the ordering of nodes in case the distances are the same.
4) Add link AB to the network and repeat steps 2-4 until all nodes have degree of at least k.
5) If the network is connected (i.e., every node is capable of communicating with every other node), then stop.
6) Otherwise, determine the shortest link that spans two different connected components and insert this link. Go to step 5.

Figure 4 shows the result of applying steps 1-4 above to 10 nodes in the CYBERNET network under the assumption that k=2. The numbers on each of the links indicate the order in which the links were added to the network. At this point the network is still not connected. Execution of steps 5 and 6 adds a link

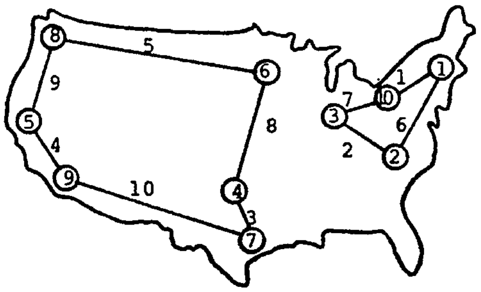


Figure 4. Link Deficit Approach to Assigning Links

described here to serve as a background for the approach taken in this paper.

Branch Exchange Method [9, 28, 30, 56, 58] The branch exchange (BXC) method seeks an improved design by adding links which are adjacent to deleted links. Two links are said to be adjacent if they share a common node. Possible criteria for choosing the links to be deleted or added are link utilization rates, cost, and estimated traffic loads between node pairs. Figure 5 illustrates two possible exchanges from a given starting topology. Figure 5 (a) is the starting network with links AD and EF assumed to be the links selected for deletion. Figures 5 (b) and (c) show two possible topologies that could result from the deletion of links AD and EF.

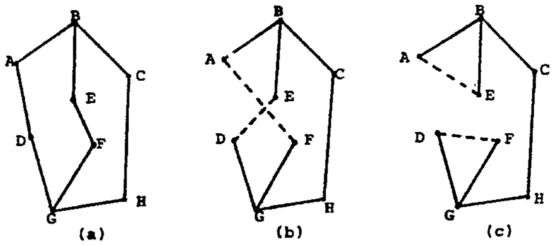


Figure 5. Branch Exchange Heuristic: (a) starting topology, (b) and (c) resultant topologies

Concave Branch Elimination Method [9, 27, 28] The concave branch elimination (CBE) method developed by Gerla [27] starts with a fully connected network and eliminates links until a local minimum is reached. A fully connected topology is one in which each node is connected directly to every other node. That is, if a network has N nodes, then the fully connected topology has N(N-1)/2 links. The scope of applicability of the CBE method is limited, however, to cases where the discrete costs can be reasonably approximated by concave functions and the packet queueing delay can be adequately described by the Pollaczek-Khintchine formula [9, 14]. This formula is a concise analytical expression which gives the average queueing delay for a single-server system having Poisson arrivals and an arbitrary distribution of service times [14]. An example of link costs that can be approximated by concave functions is given in Figure 6 [28].

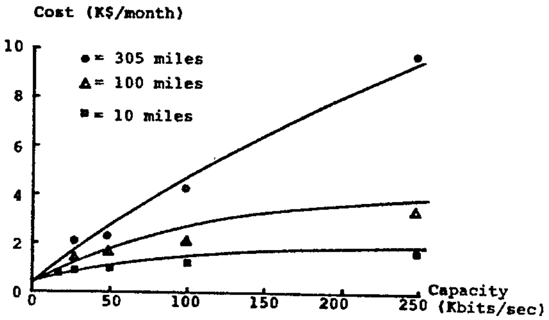


Figure 6. Concave Approximations to Link Costs for Various Link Lengths

The CBE method is computationally a more efficient design procedure than the BXC method [9], but its applicability is more restricted. The CBE heuristic is mentioned here not only because it represents a significant departure from the BXC process but also because its concept of approximating discrete costs with concave functions is used to determine bounds on heuristic performance. A complete discussion on the techniques of developing lower bounds on the cost of the optimal solution is presented in Gerla et al. [28, 30]. The existence of such bounds allows the appraisal of a heuristic algorithm's accuracy.

Cut-Saturation Method [2, 9, 28, 58] Both the BXC and CBE methods possess inherent deficiencies. The BXC heuristic requires an exhaustive search of all local transformations and is very time consuming when more than 20 nodes are involved. The CBE algorithm, although it can efficiently eliminate uneconomical links, does not have a link insertion capability. So once a link is deleted there is no possibility of recovering that link. As a result of these deficiencies, a new method, the cut-saturation (CS) algorithm, evolved.

The cut-saturation method can be viewed as an improved BXC algorithm. Rather than performing all possible link exchanges as the BXC method does, the CS algorithm selects only those exchanges that are likely to improve delay and cost. A saturated cut (or cutset) in a network is defined [28] to be the minimum set of most utilized links that, if removed, leaves the network partitioned into two disjoint components of nodes. If the links in the network shown in Figure 7 are numbered in the order from most utilized link to least utilized link, then the cutset is seen to be the set {1, 3, 5}.

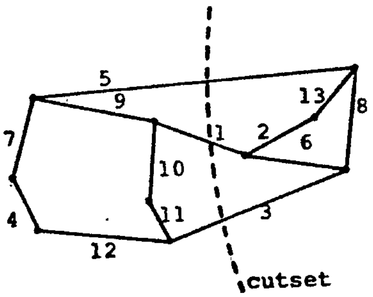


Figure 7. Example of a Saturated Cut

Tanenbaum [58] gives a scheme to find the cutset. First, rank order the links from most utilized to least utilized. Then remove from the ordered list one link at a time until the network is split into two disjoint connected components. In Figure 7, this occurs after links 1, 2, 3, 4 and 5 have been removed. To find the minimum cut put each of these links back into the network in turn. If putting a link back into the network does not reconnect the network, then that link is not part of the cutset. Such is the case in Figure 7 for links 2 and 4. Hence, the cutset does not include these links.

The saturated cut in a network functions somewhat like a bridge between the two components. In fact, the capacity of the cutset bounds the throughput that a network could possibly realize. Hence, it seems reasonable that if a link is to be added to improve throughput or delay, that link ought to span the cutset by joining the two disjoint components. Various criteria exist as to which nodes in the two components should be connected. A commonly used criterion is to add the link having the lowest cost, which also may be the shortest link, depending on the tariff structure. Similarly, link deletions should occur only within each of the individual components. Link utilization and cost usually determine the link to be deleted. While variations of the cut-saturation algorithm exist [9], the substance of these algorithms remains as that described above.

A comparative analysis of the three heuristics presented here has shown that the CS algorithm gives better results and is computationally more efficient than either the BXC or CBE methods [9, 28, 30]. Additionally, a comparison of CS solutions to theoretical lower bounds shows that the CS algorithm can produce near-optimal solutions [30].

A Two-Phase Approach to Network Optimization

The selection of an optimization heuristic can depend on a variety of factors. Among them are the cost-capacity relationship, the topological constraints involved, the desired accuracy, the degree of human interaction, and the type of network being designed. The three perturbation techniques described above have resulted from and been applied primarily to packet-switched networks. The design and analysis of integrated or hybrid networks is in its infant stage, and the literature is almost void of any kind of technique that is specially suited for optimizing integrated networks. Gitman et al. [31] state that there are basically two approaches to the integrated network design problem:

1) Solve the link/capacity problem for the voice traffic and data traffic separately, or

2) Solve the link/capacity problem for the combined voice and data traffic.

Gitman's approach has been to use option 1 together with a CBE heuristic [31]. Option 2 is the approach taken in this research.

A two-phase approach to the optimization of integrated networks has been adopted in this research. Phase 1 concerns itself with finding a feasible solution, while phase 2 attempts to improve performance and cost while maintaining feasibility. In phase 1, the topology is modified in a manner designed to satisfy the reliability, blocking, and delay constraints. This involves using a heuristic that will in general add capacity to the network and consequently increase the cost. Phase 2 attempts to modify the topology in a manner that will decrease the cost of the network while still satisfying the three constraints. This phase involves the use of a heuristic that will in general decrease the total capacity of the network. Decreasing capacity means decreasing the cost and increasing the utilization, since capacity and utilization are inversely related. The entire process can be depicted as a cubic polynomial as shown in Figure 8, where positive slopes are associated with phase 1 and negative slopes with phase 2. A description of the heuristic used in each phase is now presented.

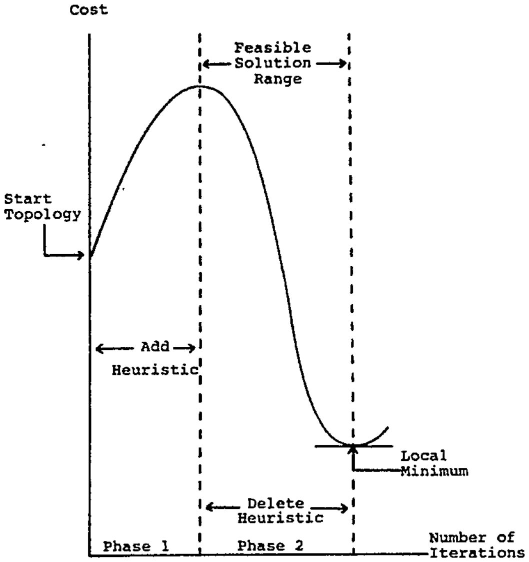


Figure 8. Two-Phase Approach to Topology Optimization

An Integrated Cut-Saturation Add Heuristic The add heuristic used in the topology optimization model is based on the cut-saturation algorithm. If a link is to be added, the nodes which form the link are determined by assigning two weights to each node on the backbone of the network. One weight is for the voice call blocking activity at the node and the other weight is for the data packet delay incurred at that node. The weights are the number of standard deviations, or z-values, that the individual node statistic (blocking or delay) deviates from the network mean statistic. If neither the delay nor blocking constraint is satisfied, then the final node weight is found by summing the two individual weights. If exactly one of these constraints is not satisfied, then only the weight of that particular statistic is taken as the final node weight. The link added is the one for which the combined weights across the cut is greatest.

The add heuristic may on any one iteration add more capacity to one or more links, add one or two new links, or it may do both. An outline of the heuristic's action is as follows:

Let MIN = minimum average link utilization constraint,
DELTA = a decimal fraction (e.g., .10) which is an indication of the heuristic's step size,
UTIL(i) = the actual utilization of link i,
NCAPS = number of discrete capacity options,
C(i) = current capacity of link i, where C(i) is an integer variable that can take on the values 0, 1, 2,...,NCAPS,
D,B,R, = indicator variables for the delay, blocking, and reliability constraints, respectively, where a 1 means the constraint is satisfied and a 0 means it is not satisfied, and

ADD = an indicator variable, where if ADD = 1 a link will have to be added (initially, ADD = 0).

1) If D = 1 and B = 1, go to step (4).
2) For each link i such that UTIL(i) > MIN, calculate N(i) = $\lceil$(UTIL(i) - MIN)/DELTA$\rceil$, where $\lceil$X$\rceil$ denotes the smallest integer greater than or equal to X. N(i) $\geq$ 0 for each i. If N(i) = 0 for all i, then set ADD = 1.
3) For each link i, modify C(i) to C(i) = C(i) + N(i). If C(i) > NCAPS, then set C(i) = NCAPS and ADD = 1.
4) If R = 1 and ADD = 0, then no link will have to be added, so stop.
5) If ADD = 1, then calculate the cutset and determine the components on either side of the cut.
6) If R = 1 and ADD = 1, then add the link determined by the weighting scheme described above and stop.
7) If R = 0 and ADD = 0, then a link need be added only to satisfy the reliability criterion. Biconnectedness (k = 2) is the assumed reliability constraint in the model, and an algorithm to determine the biconnected components has been implemented [1]. The link chosen to be added is the least costly link that spans two of the biconnected components. Add the link and stop.
8) If R = 0 and ADD = 1, then at least one link is added. The link chosen is the link with highest weighting across the cut that also spans two of the biconnected components, if such a link exists. If no such link exists, then two links are added. Specifically, the links selected are the ones that would be selected in steps (6) and (7). Add the link(s) and stop.

<u>A Biconnectivity-Preserving Delete Heuristic</u> Phase 2 of the optimization process inherits a feasible topology and seeks to reduce the cost of the network while preserving feasibility. The cost reduction, with a corresponding increase in utilization, is attained by decreasing the total capacity of the network. A biconnectivity-preserving delete heuristic is used in this model to systematically move toward a local optimum. Topological perturbations in phase 2 are always performed on a feasible topology, and deleting a link that would reduce connectivity to something less than biconnectivity (the reliability criterion) is prohibited. The heuristic is applied to the "best" feasible topology obtained up to that point in the iterative process. This necessitates the storing of the "best" topology, as well as keeping a record of unsuccessful perturbations on this topology. A limit is placed on the number of consecutive perturbation failures allowed, and when this limit is reached, the "best" topology is taken as a local optimum.

The heuristic selects a link and determines whether or not the link is to be deleted. If it is not to be deleted, it then determines by how much capacity the link should be reduced. A description of the heuristic's action is now presented. The notation is the same as in the previous section.

1) Find the least utilized link of those still qualified for investigation. Call it z.
2) If C(z) = 1, go to step (5).
3) If UTIL(z) $\leq$ MIN, then calculate N = $\lceil$(MIN - UTIL(z))/DELTA$\rceil$.
   If UTIL(z) > MIN, then set N = 1.
4) Reduce the capacity of link z by N units or to 1, whichever is greatest. That is, set C(z) = Maximum {C(z) - N, 1}. Remove z from the qualified list and stop.
5) Check to see if link z can be deleted without violating the reliability constraint. If it can, then delete z, remove z from the qualified list and stop. If it cannot be deleted, then remove it from the qualified list and return to step (1).

<u>Model Description</u>

A high-level logic flow of the entire design optimization process implemented in this dynamic topological model (CIRPAC) of a circuit/packet-switched computer network is shown in Figure 9. The upper loop corresponds to phase 1 and the lower loops make up phase 2 of the optimization approach described above. In order to determine network feasibility, the performance characteristics of the network must be obtained. CIRPAC does this by using the simulator described in earlier.

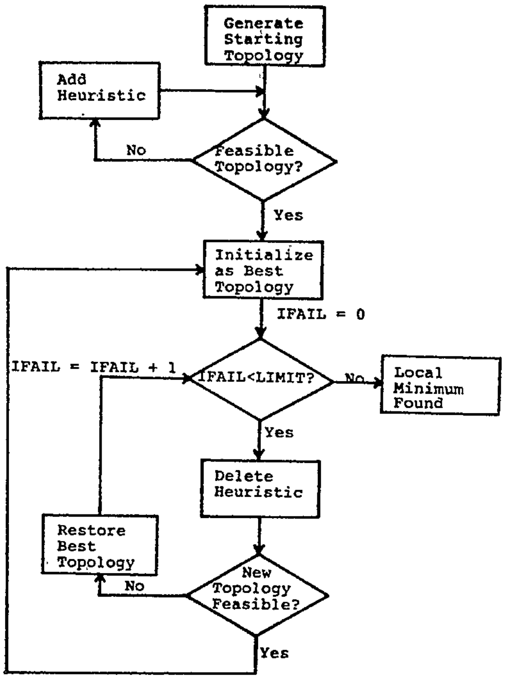


Figure 9. Optimization Logic Flow Diagram

Operational characteristics of the model have been collected for three different sizes of integrated networks. A FORTRAN H Extended (optimization level 2) compilation of the source code requires approximately 20 seconds of CPU time on an Amdahl 470/V6. Table II summarizes the time and memory requirements of the model for three different network sizes when an identical traffic load was imposed on each of the three networks. The CPU time required is given in CPU seconds for each minute of simulated time. Memory space is the compiled object program length in bytes when the dimensioning in the model matches the network size exactly. In each case the packet switch queues contained space for 300 data transactions.

		AT LOCAL MINIMUM					
Case	Start	Best	≤ 1.0 sec	≤ .10	> .60	Throughput	No. of
No.	Cost	Cost	Delay	Blocking	Utilization	(Packets/Sec)	Iterations
1	2125.02	1925.17	.982	.083	.707	2670	15
2	2472.66	2560.29	.838	.072	.675	3000	19
3	2112.39	1882.81	.963	.085	.740	2730	14
4	2268.81	2150.61	.996	.048	.718	2756	11
5	2774.31	2160.32	.865	.080	.667	2797	14
6	2335.16	2016.40	.999	.088	.706	2639	14
7	2322.21	2386.25	.961	.080	.716	2830	14
8	2790.64	2710.77	.848	.069	.715	2837	16
9	2361.61	2175.87	.984	.083	.698	2785	16
10	2120.75	2307.08	.977	.061	.716	2811	11

Table III. 10 Local Minima for 20-Node Network

On the other hand, if the throughput/cost tradeoff is examined, the plot in Figure 13 results. Case 3 is again a dominator, but this time it dominates only cases 1 and 6. Additionally, case 2 dominates case 8. If all of the non-dominated points are connected, the plot is capable of telling how much more throughput can be obtained for a given increase in cost. Similar investigations with other combinations of network performance measures are possible and provide the network designer with a degree of flexibility, even though the global optimum is not known.

52-Node Integrated Network

CIRPAC has also been used to optimize a 52-node integrated circuit/packet-switched network. In this case only one local optimum has been generated. The following paragraphs describe the starting topology configuration and track the optimization process until a local minimum is reached.

The 52 nodes in the network to be optimized consists of 26 circuit switch backbone nodes and 26 peripheral packet switch nodes. As in the 20-node network, the circuit and packet switches are in a one-to-one correspondence, with each packet switch mapping onto a unique circuit switch. The 26 backbone node locations correspond to a 26-node substructure of the ARPANET. These 26 locations are commonly used in the literature [5, 9, 22, 28] when the design and analysis of packet-switched networks is addressed. Figure 14 shows the backbone of the starting topology that was generated by CIRPAC. The integers on the links correspond to line type, and the underlying assumptions (e.g., workload and constraints) are the same as that for the 20-node network discussed in the previous section. The starting topology contains 27 links and does not satisfy the biconnectivity constraint.

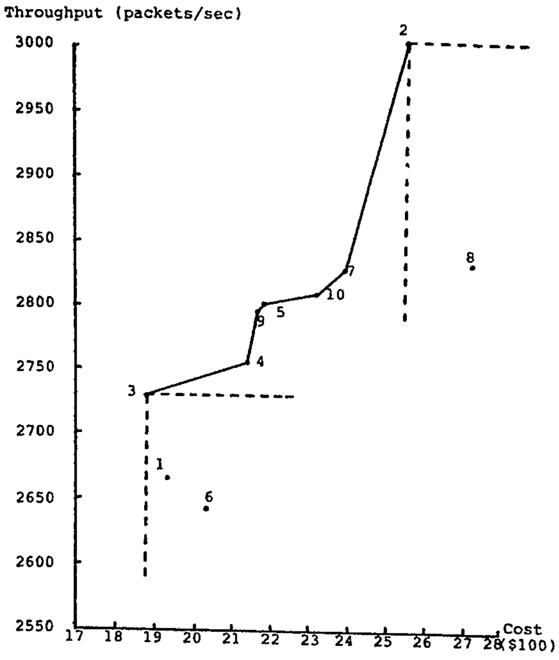


Figure 13. Throughput/Cost Tradeoff

Table II. Model Timing and Memory Requirements

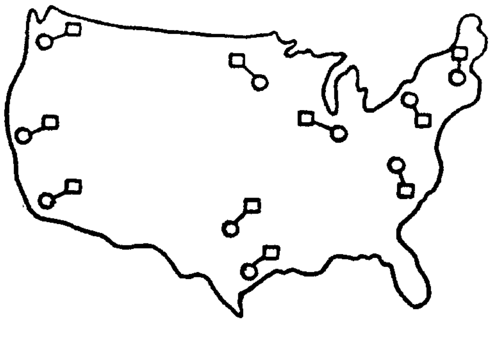
Network Size	CPU Time/Min. of Simulation	Space
10-Node	5 sec	150K bytes
20-Node	9 sec	270K bytes
52-Node	30 sec	730K bytes

APPLICATION OF THE METHODOLOGY

This section presents the results of applying CIRPAC to several integrated networks. Integrated circuit/packet-switched networks of 20 and 52 nodes, respectively, have been investigated. Unfortunately, empirical data for integrated networks is at this time scarce. However, even though this design tool is intended for integrated networks, the model has also been applied to a 26-node packet-switched network that is commonly used in the literature for comparative purposes [5, 9, 22, 28].

20-Node Integrated Network

CIRPAC has been used to obtain 10 local minima for a 20-node integrated circuit/packet-switched network. The network analyzed consists of 10 backbone circuit switches and 10 peripheral packet switches where each circuit switch represents the subnet entry point for packets from exactly one packet switch. The locations of the backbone nodes are the 10 major locations of the CYBERNET and are shown as the circular nodes in Figure 10. The square nodes are the co-located packet switches. The assumed workload consists of a voice call arrival rate of 4 calls/minute at each circuit switch and a data packet arrival rate of 400 packets/ second at each packet switch. The traffic is assumed to be symmetric and uniformly distributed between node pairs. The service time for calls is assumed to be exponentially distributed with a mean service time of 180 seconds, and the capacity/cost information is as shown in Figure 10.



LINE CAPACITY AND COST INFORMATION:

LINE TYPE	CAPACITY (BPS)	COST($) PER UNIT LENGTH	FIXED COST($)
1	800000.	1.00	50.00
2	1000000.	2.00	50.00
3	1200000.	4.00	100.00
4	1600000.	8.00	150.00
5	2000000.	16.00	200.00

Figure 10. 20-Node Configuration

Using 10 different starting topology generator seeds, CIRPAC produced 10 different local minima for the network input data given above. The results of these 10 cases are summarized in tabular form in Table III. The "start cost" is the cost of the starting topology and the "best cost" is the cost associated with the local minimum. The delay, blocking, and utilization constraints were as shown in the table. Throughput is defined to be the average number of packets/second traveling out of each node. The total number of iterations is the number of iterations needed to reach the local minimum. Biconnectivity is the assumed reliability constraint.

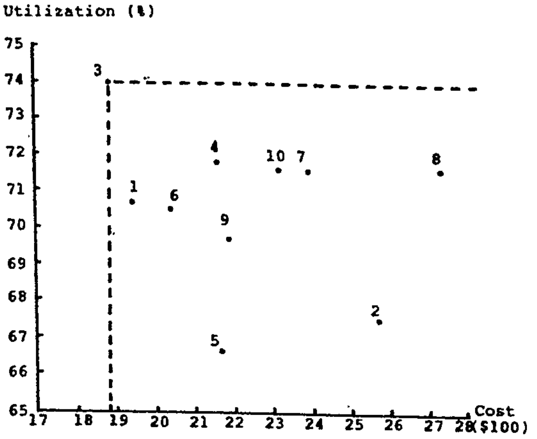


Figure 11. Domination of the Utilization/Cost Space by Case 3

The local minima obtained can be used to examine integrated network design tradeoffs. For example, if each of the local minima is plotted on a utilization versus cost coordinate systems, as shown in Figure 11, it is seen that topology number 3 dominates all of the other nine topologies. That is, each of the other minima has a higher cost and a lower utilization rate than case 3. The topology associated with local minimum number 3 is shown in Figure 12. The integers on each link indicate the link type for that connection. Solid lines represent links that were part of the starting topology while dashed lines indicate links that were added during the optimization process.

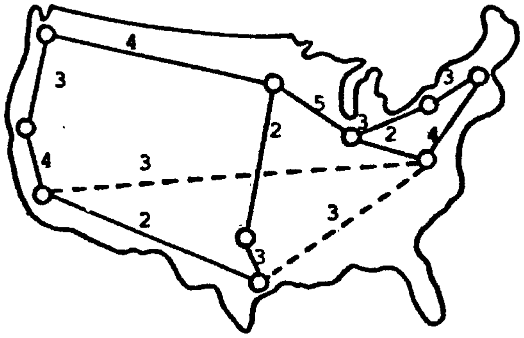


Figure 12. Topology for Local Minimum #3

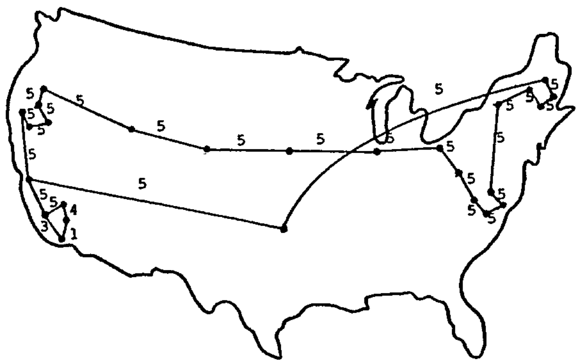


Figure 14. Starting Topology for 52-Node Network

The process of optimizing this network with CIRPAC required a total of 45 iterations. Eight iterations were required to reach feasibility, with eight links being added in phase 1. Phase 2 required 37 iterations and two links were deleted in this phase.

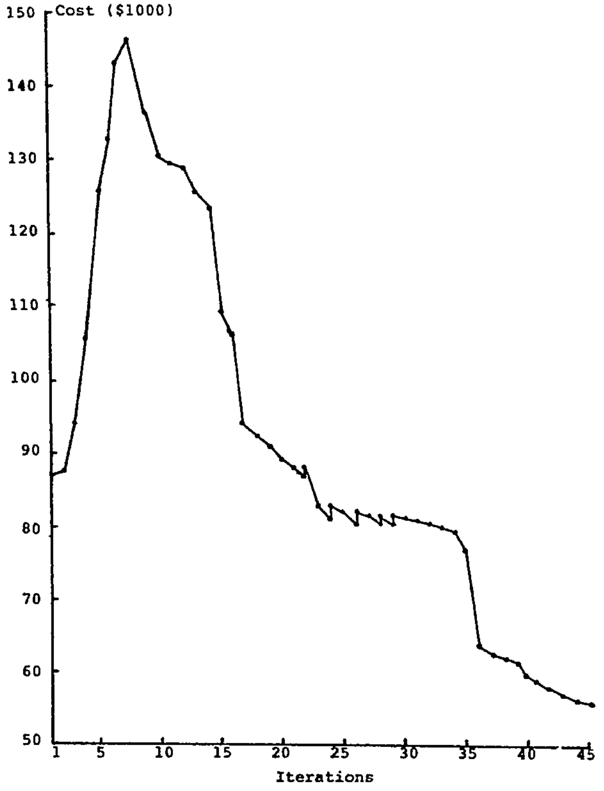


Figure 15. 52-Node Optimization: Cost Trace

Figure 15 shows a trace of the cost function from the starting topology to the local minimum obtained at iteration 45. The "hiccups" shown at iterations 22, 24, 26, 28 and 29 represent perturbation failures. That is, the modification to the network produced an infeasible topology, so the procedure reverted back to the "best" topology prior to the next iteration. In all five failures, the constraint failing to be satisfied was the call blocking constraint. The topology representing the local minimum is shown in Figure 16. It has 33 links; 25 of the original links remain (solid lines), and 8 new links have been added (dashed lines).

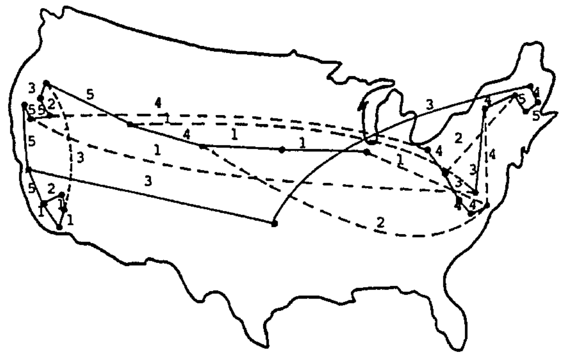


Figure 16. Local Minimum Topology for 52-Node Network

26-Node Packet-Switched Network

The absence of empirical data for integrated networks led to the application of CIRPAC to a 26-node ARPA-like network. CIRPAC was neither designed nor developed for packet-switched networks. Hence, any comparisons made between CIRPAC's heuristics and heuristics intended for strictly packet-switched networks have not been validated and should be interpreted with caution.

The 26 node locations for this design are the backbone locations shown in Figure 16. These locations are now packet switches, not circuit switches. The design specifications for the 26-node ARPA-like network are as follows [9]:

1) mean packet delay < .2 sec
2) data packet size = 460 bits
3) node processing delay = 0 sec for each node
4) propagation delay = 8.05μ sec/mile
5) traffic load ranges from 350 to 750 Kbits/sec and is symmetric and uniformly distributed between node pairs
6) capacity/cost options:

Line Type	Line Speed (BPS)	Cost($) Per Month/Mile	Fixed Cost ($) Per Month
1	9600	.40	1300
2	19200	2.50	1700
3	50000	5.00	1700

The input parameters to CIRPAC have been adjusted to accommodate these design criteria as best possible. Certain approximations have been made. The node processing delay (circuit switching delay) in CIRPAC must be used to approximate the propagation delay of the packet-switched network. This is accomplished by assuming an average distance traveled by each packet (2500 miles) as well as an average number of hops (4) on each routing path. Also, a mean packet delay of no more than .5 seconds with 1000-bit packets is used [48]. The voice call arrival rate is set to zero since only data packets are traversing this network.

CIRPAC has been used to generate two local minima. One is for a traffic requirement of 350 Kbits/sec and the other is for a workload of 700 Kbits/sec. Figure 17 illustrates the throughput/cost space for solutions obtained with CIRPAC and other heuristics operating on the 26-node network. The lower bound for optimal solutions shown on the graph are each dominating solutions. That is, they each represent the lowest

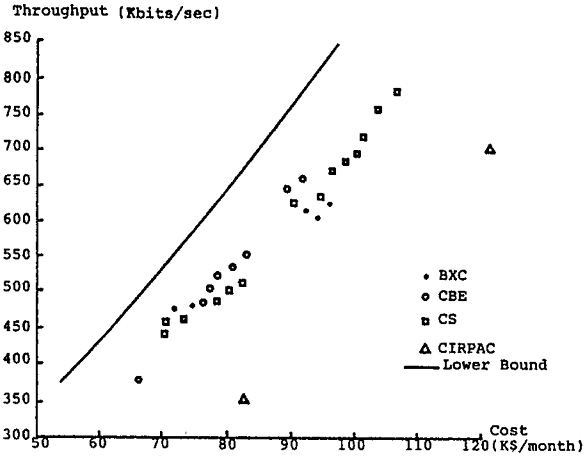


Figure 17. Heuristic Solutions for 26-Node Design

cost solution of several local optima obtained at each traffic requirement level shown.

SUMMARY

In this paper we have presented a dynamic topological configuration methodology which is capable of optimizing the topology of an integrated circuit/ packet-switched computer-communication network. Network topology as defined herein was taken to mean the manner in which the nodes of a network are interconnected and how much carrying capacity each connecting link should have.

Initial investigations were directed at identifying what an integrated circuit/packet-switched computer network is. It was shown that such networks are capable of carrying voice and data information simultaneously on their trunk lines. The existence of integrated networks in the near future appears to be a certainty since network designers and managers alike seek to make more efficient use of the information-carrying media available to them. The more recent innovations in the development of very high capacity transmission media, e.g., fiber optics, point to a faster transition to integrated networks than previously anticipated. The integration concept was shown to be implemented in a simulation model capable of generating accurate network performance data.

The exact solution to the topological optimization problem was seen to be intractable for even small networks. This research addressed the topology design problem using an interactive, heuristic approach whereby many suboptimal solutions (local minima) are efficiently generated in lieu of one optimal solution. The iterative scheme integrates the simulator as a performance generation device in the middle of a performance feedback loop. The loop consists of three processes that are repeated in turn: topology generation, network performance generation, and performance evaluation. The heuristics, part of the performance evaluation process, determine the direction in which the topology is to be modified. An integrated cut-saturation add heuristic is used to increase total network capacity until a feasible topology is obtained. Then a delete heuristic which preserves biconnectivity is used to reduce cost and increase link utilization. The methodology developed has been designed to be independent of the device which generates network performance data. Hence, if analytical means of providing adequate performance data for an integrated network should become available, only the interface modules would require change.

The methodology has been applied to design problems of varying size. Results indicate that local optima can be obtained in a reasonable number of iterations even with a relatively small step size. It has been shown how the existence of several local minima can be used to actually increase the flexibility that the decision maker has in making design decisions. The model itself allows for human intervention. Starting topologies may be supplied rather than automatically generated. The number of iterations allowed is easily controlled and the stopping criteria are easily modified. Links and capacities may be forced into the topology if so desired. The methodology represents a viable approach to the design problem, and this research has demonstrated that the model is a flexible tool that can be used to optimize the design of integrated networks.

REFERENCES

1. Aho, A.V., Hopcroft, J.E., and Ullman, J.D., *The Design and Analysis of Computer Algorithms*, Addison- Wesley Publishing Co., Reading, MA, 1974.

2. Ahuja, V., *Design and Analysis of Computer Communication Networks*, McGraw-Hill, Inc., New York, NY, 1982.

3. Bially, T., and McLaughlin, A.J., "Voice communications in integrated digital voice and data networks",*IEEE Trans. on Comm. Com-28,9* (Sept. 1980) 1478-1488.

4. Boehm, B.W., and Mobley, R.L., "Adaptive routing techniques for distributed communications systems" *IEEE Trans. on Comm. Techn. Com-17*, 3 (June 1969), 340-349.

5. Boorstyn,R.R., and Frank, H., "Large-scale network topological optimization", *IEEE Trans. on Comm. Com-25*, 1 (Jan. 1977), 29-47.

6. Branscomb, L.M., "Trends and developments in computer/telecommunications technologies", *Proc. of the Organization for Economic Cooperation and Development Conference*, Paris, France (Feb. 4-6, 1975), 57-77.

7. Cantor, D.G., and Gerla, M., "Optimal routing in a packet switched computer network", *IEEE Trans. on Comput. C-23*, 10 (Oct. 1974), 1062-1069.

8. Chou, W. (Ed.). *Computer Communications, Volume I: Principles*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1983.

9. Chou, W., and Sapir, D., "A generalized cut-saturation algorithm for distributed computer communications network optimization", *IEEE 1982 Int. Conf. on Comm. (ICC-82)*, Philadelphia, PA, (June 13-17, 1982), 4C.2.1-4C.2.6.

10. Clabaugh, C.A., "Analysis of flow behavior within an integrated computer-communication network", Ph.D. dissertation, Texas A&M University, (May 1979).

11. Coviello, G.J., and Lyons, R.E., "Conceptual approaches to switching in future military networks", *IEEE Trans. on Comm. Com-28*, 9 (Sep. 1980), 1491-1498.

12. Coviello, G.J., and Rosner, R.D., "Cost considerations for a large data network", *1974 Int. Conf. on Computer Comm. (ICCC-74)*, Stockholm, Sweden (Aug. 12-14, 1974), 715-720.

13. Coviello, G.J., and Vena, D.A., "Integration of circuit/packet switching by a SENET (Slotted Envelope Network) concept", *1975 Nat. Telecomm. Conf. (NTC-75)*, New Orleans, LA. (Dec. 1-3, 1975), 42-12 - 42-17.

14. Cravis, H., *Communications Network Analysis*, D.C. Heath and Co., Lexington, MA, 1981.

15. Dirilten, H., and Donaldson, R.W., "Topological design of distributed data communication networks using linear regression clustering", *IEEE Trans. on Comm. Com-25*, 10 (Oct. 1977), 1083-1090.

16. Dreyfus, S.E., "An appraisal of some shortest-path algorithms", *Operations Research 17*, 3 (May-June 1969), 395-412.

17. Dysart, H., Krone, M., and Fielding, J., "Integrated voice/data private network planning", *IEEE 1981 Int. Conf. on Comm. (ICC-81)*, Denver, CO (June 14-18, 1981), 4.2.1-4.2.5.

18. Elovitz, H.S., and Heitmeyer, C.L., "What is a computer network?", *1974 Nat. Telecomm. Conf.* (NTC-74), San Diego, CA (Dec. 2-4, 1974), 1007-1014.

19. Forgie, J.W., "Voice conferencing in packet networks", *IEEE 1980 Int. Conf. on Comm. (ICC-80)*, Seattle, WA (June 12-18, 1980), 21.3.1-21.3.4.

20. Frank, H., "Plan today for tomorrow's data/voice nets", *Data Communications* 7,9 (Sep. 1978), 51-62.

21. Frank, H., and Chou, W., "Network properties of the ARPA Computer network", *Networks 4* (1974), 213-239.

22. Frank, H., and Chou, W., "Topological optimization of computer networks", *Proc. of IEEE 60*, 11 (Nov. 1972), 1385-1397.

23. Frank, H., and Gitman, I., "Economic analysis of integrated voice and data networks: a case study", *Proc. of IEEE 66*, 11 (Nov. 1978), 1549-1570.

24. Fultz, G.L., and Kleinrock, L., "Adaptive routing techniques for store-and-forward computer communication networks", *IEEE 1971, Int. Conf. on Comm. (ICC-71)*, Montreal, Canada (June 14-16, 1971), 39-1 - 39-8.

25. Gallager, R., "Distributed network optimization algorithms", *IEEE 1979 Int. Conf. on Comm. (ICC-79)*, Boston, MA (June 10-14, 1979), 43.2.1 - 43.2.2.

26. Garey, M.R., and Johnson, D.S., Computers and Intractability, *A Guide to the Theory of NP-Completeness*, W.H. Freeman and Co., New York, NY, 1979.

27. Gerla, M., "The design of store-and-forward networks, for computer communications", Ph.D. dissertation, School of Eng. and Appl. Sci., Univ. of California, Los Angeles (Jan. 1973).

28. Gerla, M., and Kleinrock, L., "On the topological design of distributed computer networks", *IEEE Trans. on Comm. Com-25*, 1 (Jan. 1977), 48-60.

29. Gerla, M., and Mason, D., "Distributed routing in hybrid packet and circuit data networks", *IEEE Proc. of Comp. Comm. Networks: COMPCON 78*, Washington, DC (Sep. 5-8, 1978), 125-131.

30. Gerla, M., Frank, H., Chou, W., and Eckl, J., "A cut saturation algorithm for topological design of packet switched communication networks", *1974 Nat. Telecomm. Conf. (NTC-74)*, San Diego, CA (Dec. 2-4, 1974), 1074-1079.

31. Gitman, I., Hsieh, W., and Occhiogrosso, B.J., "Analysis and design of hybrid switching networks", *IEEE Trans. on Comm. Com-29*, 9 (Sep. 1981), 1290-1300.

32. Gitman, I., Occhiogrosso, B.J., Hsieh, W., and Frank, H., "Sensitivity of integrated voice and data networks to traffic and design variables", *Proc. 6th Data Comm. Symposium*, Pacific Grove, CA (Nov. 1979), 181-192.

33. Greene, W.H., "Optimal routing within large scale distributed computer-communications networks", Ph.D. dissertation, Texas A&M University (May 1978).

34. Gruber, J.G., "Delay related issues in integrated voice and data networks", *IEEE Trans. on Comm. Com-29*, 6 (June 1981), 786-800.

35. Hoard, B., "Integrating voice and data - sharing the lines", *Computerworld 15*, 52 (Dec. 28, 1981), 33-35.

36. Hsieh, W., Gitman, I., and Abernathy, J., "Topological design issues in network interconnection", *IEEE 1977 Int. Conf. on Comm.* (ICC-77), Chicago, IL (June 12-15, 1977), 22.5.126-22.5.131.

37. Hsieh, W., Gitman, I., and Occhiogrosso, B.J. "Design of hybrid-switched networks for voice and data", *IEEE 1978 Int. Conf. on Comm. (ICC-78)*, Toronto, Canada (June 4-7, 1978), 20.1.1-20.1.9.

38. Jenny, C.J., Kummerle, K., and Burge, H., "Network nodes with integrated circuit/packet switching capabilities", IBM Research Report RZ-720 (Aug. 1975).

39. Kleinrock, L., and Kamoun, F., "Optimal clustering structures for hierarchical topological design of large computer networks", *Networks 10*, 3 (Fall 1980), 221-148.

40. Kozicki, Z., and McGregor, P.V., "An approach to computer-aided network design", *IEEE 1981 Int. Conf. on Comm. (ICC-81)*, Denver, CO (June 14-18, 1981), 4.4.1-4.4.7.

41. McAuliffe, D.J., "An integrated approach to communications switching", *IEEE 1978 Int. Conf. on Comm. (ICC-78)*, Toronto, Canada (June 4-7, 1978), 20.3.1-20.4.5.

42. McQuillan, J.M., "Adaptive routing algorithms for

distributed computer networks", Report AD-781467, NTIS (May 1974).

43. Metcalfe, R.M., "Packet communications", Report AD-771-430, NTIS (Dec. 1973).

44. Occhiogrosso, B.J., Gitman, I., Hsieh, W., and Frank, H., "Performance analysis of integrated switching communications systems", *1977 Nat. Telecomm. Conf. (NTC-77)*, Los Angeles, CA (Dec. 5-7, 1977), 12:4-1 - 12:4-13.

45. Ozarow, L., and DeRosa, J., "A combined packet and circuit-switched processing satellite system", *IEEE 1979 Int. Conf. on Comm. (ICC-79)*, Boston, MA (June 10-14, 1979), 24.5.1-24.5.5.

46. Phillips, D.T., and Garcia, A., *Fundamentals of Network Analysis*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1981.

47. Rich, M.A., and Schwartz, M., "Buffer sharing in computer-communication network nodes", *IEEE Trans. on Comm. Com-25, 9 (Sep. 1977)*, 958-970.

48. Roberts, L.G., and Wessler, B.D., "Computer network development to achieve resource sharing", *1970 Spring Joint Computer Conf., AFIPS Conf. Proc. 36*, Atlantic City, NJ (May 5-7, 1970), 543-549.

49. Rosner, R.D., "A digital data network concept for the Defense Communications Agency", *1973 Nat. Telecomm. Conf. (NTC-73)*, Atlanta, GA (Nov. 26-28, 1973), 22C-1 - 22C-6.

50. Rosner, R.D., "Large scale network design considerations", *1974 Int. Conf. on Computer Comm. (ICCC-74)*, Stockholm, Sweden (Aug. 12-14, 1974), 189-197.

51. Ross, M. J., "System engineering of integrated voice and data switches", *IEEE 1978 Int. Conf. on Comm. (ICC-78)*, Tronoto, Canada (June 4-7, 1978), 20.5.1 - 20.5.4.

52. Ross, M.J., Tabbott, A.C., and Waite, J.A., "Design approaches and performance criteria for integrated voice/data switching", *Proc. of IEEE* 65, 9 (Sep. 1977), 1283-1295.

53. Rudin, H., "Studies on the integration of circuit and packet switching", *IEEE 1978 Int. Conf. on Comm. (ICC-78)*, Toronto, Canada (June 4-7, 1978), 20.2.1 - 20.2.7.

54. Schmitz, H.G., Saxton, T.L., Huang, C.S., and White, J.A., "Application of associative processing techniques to an integrated voice/data switched network", Report AD-A040636, NTIS (June 1976).

55. Schneider, K.S., "Integrating voice and data on circuit-switched networks", *IEEE Trans. on Aerosp. Electron. Syst. AES-15*, 4 (July 1979), 481-493.

56. Steiglitz, K., Weiner, P., and Kleitman, D.J., "The design of minimum-cost survivable network", *IEEE Trans. on Circuit Theory CT-16*, 4 (Nov. 1969), 455-460.

57. Takehiko, Y., and Schimasaki, N., "A study of future integrated service digital networks", *1975 Nat. Telecomm. Conf. (NTC-75)*, New Orleans, LA (Dec. 1-3, 1975), 7-1 - 7-6.

58. Tanenbaum, A.S., *Computer Networks*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1981.

59. Thurber, K.J., "Circuit switching technology: a state-of-the-art survey", *IEEE Proc. of Comp. Comm. Networks: COMPCON 78*, Washington, DC (Sep. 5-8, 1978), 116-124.

60. Weinstein, C., McLaughlin, A., and Bially, T., "Efficient multiplexing of voice and data in integrated digital networks", *IEEE 1980 Int. Conf. on Comm. (ICC-80)*, Seattle, WA (June 8-12, 1980), 21.1.1 - 21.1.7.

61. Whitney, H., "Congruent graphs and the connectivity of graphs", *Amer. J. Math. 54* (1932), 150-168.