INTRODUCTION

The planning, design, and management of a large telecommunications network is a complex process involving many aspects of the network's cost and performance. Throughput capability, reliability, response time performance, and cost must all be considered in evaluating alternatives. For a large network, obtaining an accurate analysis of any one of these aspects is a formidable task involving cumbersome calculation and many decisions. The problem of obtaining an accurate analysis of the entire system is magnified by the fact that these individual factors interact strongly with one another. Thus, in order to design a “good” network, or verify that an existing network is operating properly, one must not only be able to solve each of the above problems, but also be able to evaluate the solution of each one in the context of solutions to the others.

Computer programs, if they are carefully designed and properly used, can be powerful tools in the planning and management process. While one cannot reasonably expect that such programs will solve all problems or even that they will solve any one problem perfectly, such programs can take a large burden off the analyst by providing accurate and reliable answers to subproblems which can then be used to answer global questions about the overall network.

Thus, these programs are used by the qualified analyst as tools. If the man-machine interface is good, the analyst and the computer together can solve problems that neither could solve as well separately and each works on the aspects of the overall problem which he can handle best. The analyst decomposes the overall problem into subproblems and couches his questions to the computer in such a way as to reflect the peculiarities of each particular network and the relative importance of each design criterion, which also varies from system to system. The machine answers specific questions in the context posed and enables the analyst to consider many more alternatives than he would otherwise have the time or inclination to approach using hand analysis alone.

This paper functionally describes programs which can be used as design and analysis tools, and the uses they can be put to. Examples of the benefits obtainable through the use of such tools in the design and analysis of actual networks are presented. Although much of the material presented can be applied to the design and analysis of all types of networks, the discussion is limited to centralized telecommunications networks.
each individual subproblem solution. More so, the subproblem solutions often depend upon one another. This task falls mainly upon the analyst. He may choose to coordinate the modules by specifying input parameters to each module, by examining outputs, by interacting with the modules during their execution, or even by using a program manager module to control some of the flow of information among individual modules. In each case however, it is the analyst who decomposes the original problem, specifies the context of the solution of each subproblem, and assembles the solution of the global problem from its component parts. This places the responsibility upon the analyst of not only understanding the problem, but also of understanding what his tools can do for him and what they can not. The situation which thus arises is more realistic than one where this responsibility is placed on the tools and is a significant improvement over working with no tools at all.

INPUT MODULE AND DATA BASE

Before one can make use of design and analysis tools, he must first specify the network and his objectives to the programs he is using. While this is not the most conceptually difficult task in the design/analysis process, it can be the most time consuming unless it is done efficiently. This can best be done using an input module and a data base which interact with one another.

The input module reads in user supplied data including parameters and pointers into the data base. It then fills in the defaults and fetches user requested information from the data base, collecting all specified data in one place. Next it checks the collected data for completeness, correctness, and consistency, outputting error messages, and possibly halting if any sufficiently serious errors are encountered. Finally, it organizes the data into the structural form required by the other modules, and if the input module is run as a separate program, outputs this data structure to be read in by other modules.

The ideal mode of input is the interactive mode where the user is permitted to enter data while the program is running. Error checking and corrections can be done immediately, and the user can specify some parameters after having seen output from some of the modules. An important property of an input module running in an interactive mode is that parts of it will run after other modules are run or, in some cases, even while they are running. This differentiates it structurally from an input module run in a static mode, which usually does its whole job at the start of execution. Thus, an interactive input module should consist of several sections which are executable separately and closely linked to the modules controlling the overall program flow. For each particular network design, the task of inputting all the required data can be greatly simplified by incorporating an easily usable set of defaults, allowing the user to omit parameters and have the system fill them in. The NAMELIST facility, present in many higher level languages, which allows the user to input data in free format specifying both the variables’ name and value and omitting variables which are to take default values, is one good way of implementing defaults.

The data base is a body of information which is not, in general, dependent upon a particular network (e.g., device characteristics or tariffs) and is necessary input to the program. The user can specify a block of information he wishes to make use of, and the input module fetches it from the data base thereby relieving the user of the burden of specifying each piece of information individually. One section of the input module, which in general may run entirely independently of the rest of the sections (and the rest of the program), is a procedure for creating and modifying the data base.

It may be necessary to input parameters which overwrite and supplement values gotten from the data base. A common example of this is pricing information which is rightfully part of the data base but is often varied from system to system because it is volume dependent. By providing this flexibility, the data base can be kept to a manageable size, excluding devices of marginal utility without fear of limiting the use of the program. It is also not necessary to constantly change the data base when a new line of equipment is introduced. Manual override parameters and system default parameters together provide the user with the ability to freely intermix input data with information gotten from the data base, using the program under a wide variety of circumstances with a minimum of effort.

NETWORK CONFIGURATION AND HARDWARE ALTERNATIVES

One of the first questions which must be answered in order to design a network is what hardware devices should be used in the design and how should the hardware be configured. The choice of central processor, peripherals, concentrators, multiplexers, terminal controllers, and terminals greatly influences the network’s cost and performance, and forms the basis of all further analysis and evaluation of the system.

Central site configuration

Given a central processor (or several processors) and a set of peripheral devices (tapes, disks, drums, etc.), one must evaluate the throughput capability and cost effectiveness of the configuration. One can thus compare different lines of equipment and different configurations of a specific line of equipment.

A program of this nature can range from a simple closed-form queueing formula to a very elaborate brute-force simulation. The former approach, though simple, must adhere to rigid assumptions. Thus, the scope of its application is limited. The latter one requires enormous efforts both in developing and using the program, and is in general only useful for one particular system. Thus, it is quite impractical to use, as two essential features of a simulation program are that it be inexpensive for repetitive usage and versatile enough to be used in a varying operating environment. There-
fore, this module should be a simulation program with embedded queuing models.

A properly constructed queuing model can represent many different systems by changing parameters and can accomplish many important tasks, such as considering systems containing peripheral devices with different characteristics, pinpointing traffic bottlenecks, and allowing arbitrary distributions for CPU processing time and message arrival rates.

**Terminal configuration**

The basic function of such a module is to evaluate the cost effectiveness and throughput capabilities of different types of terminals and terminal configurations. In particular, one may wish to evaluate the effectiveness of using terminal controllers with simple slave terminals as opposed to using more sophisticated terminals which can operate without controllers. For each configuration, the module calculates the number of terminals required as a function of throughput and the total cost versus throughput capacity.

**Choice of multiplexers and concentrators**

The decision whether or not to use concentrators and multiplexers in a network design must be partially based upon how much money can be saved in other parts of the network and how their inclusion will affect the network's reliability and response time performance. Configuration modules can help in making this decision by evaluating alternate devices and calculating the number of devices required and their cost for given traffic loads.

Each of these modules takes, as input, device characteristics and traffic characteristics and yields as output an evaluation of a hardware configuration. A final decision as to what hardware configuration should be used need not be made on the basis of such information alone, but instead, can be made using the output of other modules as well. By treating problems of hardware configuration in these modules however, rather than as part of a large program which considers other issues as well, one can obtain clear answers to these configuration questions and greatly simplify the tasks of the design and analysis modules.

**RESPONSE TIME/THROUGHPUT ANALYSIS**

One of the most important, and most difficult problems in analyzing a telecommunications network is getting an accurate analysis of response time and throughput capability. Without a clear picture of the system's response time performance, the analyst must make conservative assumptions during the design process, and runs the risk of introducing so much slack into the design that it cannot be sold in today's highly competitive market. Worse yet, he runs the risk of designing a system which will not work. When the system to be analyzed includes concentrators, multiplexers, terminal controllers, and complicated central processor hardware configurations, the problem becomes especially complex.

The output of the hardware configuration modules can be used to great advantage in solving this problem. One can explicitly simulate message flow along a single path between a terminal and the central computer by using the parameterized analytic queuing models in the configuration modules to represent system hardware and software. In this way, it is possible to accurately account for the delays introduced by hardware, software, and the presence of traffic from other lines incident to the path being simulated.

This approach to the problem is more effective than pure simulation or analytic modeling alone. Analytic models of the complex network of queues present in real systems can only be formulated by making simplifying assumptions which result in inaccuracies in the response times obtained from the model. Pure simulation, while more accurate, is cumbersome and results in a model which is specific to the particular system under consideration. Such models must be modified extensively for each new system considered. The hybrid technique, using analytic models for each device as part of a simulation model of the entire system, results in an accurate model which is at the same time flexible enough to be used in the analysis of a wide variety of different types of systems.

By means of input parameters, the user should be able to specify message length distributions for input and output messages, keying and display (or printing) times, distributions for message interarrival times, number of terminals (or terminal controllers) per line, number of lines per concentrator, line speeds, line discipline, polling and addressing sequence lengths, number of bits per character, propagation times, device turnaround times, message processing times, and parameters describing the delays in hardware devices. Additional input parameters controlling the operation of the simulation package might include multiple simulation option, trace feature, output control parameters, maximum simulated time, maximum number of transactions simulated, and transient response suppression.

The output of the response time/throughput module may be formatted in many ways, dependent upon the user's needs. Response times may be presented on an overall basis or components of response time such as waiting times in various devices may be presented separately if they are of interest. Response times may be given as functions of facility utilization (e.g., line utilization, or CPU occupancy). Empirical distributions may be presented if a detailed analysis of systemwide variation in response time is required, in particular if a constraint at a given percentile of traffic must be checked. In some applications, where leased lines are used, queuing delays may be of primary interest while in others, where contention or dialup is used, blocking probabilities may be required.

Thus, with the aid of output from the configuration modules, the response time/throughput module can produce an accurate and reliable analysis of a network even when it is complex and contains several levels of hardware. This analysis is not only interesting in its own right as it provides one of the most useful measures of a system's performance, but
also, it can greatly simplify the task of the topological design module by providing it with useful relationships between response times and facility utilizations.

RELIABILITY ANALYSIS

One of the most neglected areas in the network design process is reliability analysis. Many networks are designed without giving any thought to this important measure of network performance, and only after painful experience with frequent failures and degraded response times following periods of downtime is it given belated consideration.

The most fundamental decisions in the configuration of a network—hardware selection, use of alternate paths to provide backup for critical communications lines, use of spare components for backup, and connection of on-line devices, in series or in parallel—relate strongly to the network's reliability. Here, as in the case of response time, inadequate analysis may lead to conservatism in design, or to excessive optimism, which can result in unacceptably poor system performance.

Given a topological design of a network and the failure rates of components (mean time to failure, mean time to recovery, or probability of failure) one must calculate reliability measures for the entire network. Here, as in the case of response time and throughput analysis, the output may be organized in different ways, depending upon the user's needs.

Usually, one wishes to obtain the average value of the percentage of time a terminal or location (with several terminals) can communicate with the central computer. Sometimes this information is useful on a terminal by terminal or location by location basis. One may also want the average reduction in system throughput capacity due to equipment failure.

If the network under consideration is topologically simple (i.e., a tree, a series parallel network, or a loop network), such reliability measures can be obtained analytically. A combination of analytic and simulation techniques is often highly effective in obtaining solutions in many practical situations which do not fall into any of these simple topologies. In particular, many networks can be analyzed as a collection of trees or loops which are interconnected by a more topologically complex backbone network. Each tree or loop can then be collapsed analytically into an equivalent node in the backbone network, and the problem can thus be reduced to that of a small number of nodes which can be analyzed using simulation. Alternatively, when the network under consideration is composed of many small topologically complex subnetworks linked by a backbone, each subnetwork can be separately reduced to an equivalent node in the backbone, and the backbone can then be analyzed efficiently. In each case, the network is collapsed from the extremities toward the center, replacing increasingly large subnetworks with equivalent nodes. If the overall network is not complex topologically, both the running time and memory requirements of the procedure need be only linearly proportional to the number of nodes in the network.

TOPOLOGICAL OPTIMIZATION

The task of topological optimization, i.e., deciding how to interconnect network locations as economically as possible while still meeting all performance constraints, is the most complex task facing the analyst, as it encompasses all the previously mentioned problems. The other modules at his disposal provide great assistance in this task; but he is still left with many decisions, especially if the network is large and geographically diverse enough to benefit from the economies offered by concentrators and multiplexers.

In order to design networks using concentrators and multiplexers, one must first choose the number of concentrators and multiplexers and their locations. Next, each terminal in the network must be associated with a particular concentrator or multiplexer. Then the routing and speed of the multidropped lines connecting the terminals to the concentrators and multiplexers must be decided upon. Finally, the routing and speed of the lines connecting the multiplexers and concentrators to the central computer must be decided.

A module which is to be a useful tool in solving these problems must itself be composed of submodules, each of which is designed for a specific purpose.

First, because of the variety of tariff structures currently in use, one needs a submodule for evaluating the cost of connecting any pair of points with a given speed line. Where certain tariff structures are applicable, such as the high/low density tariff, or where existing TelpakS can be used, intermediate points may be used in routing circuits, and this cost calculation may itself be complex. By separating the cost calculation from other functions, the topological optimization module can be used with a variety of tariff structures without becoming unduly complex.

The problems of associating terminals with concentrators and routing the multidrop lines connecting terminals to concentrators are separable from the other problems and should be treated by a separate submodule dedicated to this purpose. One such efficient algorithm has been the subject of another paper by this author.

The remaining problems, which concern the location and interconnection of concentrators, can often be dealt with manually by the analyst, since the number of concentrator or multiplexer locations are often limited by the existence of manned facilities. If, despite these factors, the remaining problems are not tractable manually, a straightforward automated procedure can be implemented by embedding solutions to the multidrop line layout problem into an algorithm which evaluates different numbers of concentrators and different concentrator locations.

The topological optimization module requires, as input, information describing the applicable tariffs, traffic at each location, terminal locations, and utilization factors for lines and devices. In addition to this, concentrator and multi-
plexer locations must be input unless an automated procedure for their selection has been included. Notice, however, that performance criteria need not be input directly. It is the function of other modules to evaluate such criteria and produce as output the utilization factors and specific device types which are input here. This operation greatly simplifies the task of the module, allowing it to solve its problem accurately and efficiently.

**COORDINATION OF THE MODULES**

As has already been mentioned, the task of coordinating these modules usually falls on the analyst. Nonetheless, it is possible to lighten this burden to some extent by designing another module, the Program Manager, to assist the analyst in the performance of part of this task. The point must be stressed, however, that an attempt to use such a module as a substitute for insight on the part of the analyst can easily complicate the Program Manager and all the other modules, and seriously impair their functions.

The basic function of the Program Manager is to link the other modules together while they are executing, allowing them to pass information to one another which would help them in performing their individual tasks.

Some of the parameters passed to each module as input can be gotten from the output of other modules. Thus, the utilization constraints on lines and devices input to the topological optimization module (TOM) can be gotten directly from the output of the response time module (RTM) rather than having the user specify them as input. This not only makes the TOM easier to use; it also provides for greater flexibility in specifying these constraints, since the RTM can generate combinations of constraints that are too complex for a user to specify by hand. The RTM may also pass several different combinations of constraints and allow the TOM to choose the best set.

Similarly, the TOM can pass detailed information about a particular design to the RTM, allowing the RTM to produce a more exact analysis of that particular design than could be provided using only general parameters describing the network. The RTM could then reevaluate constraints and pass them back to the TOM, and the entire process can iteratively improve the network design.

A detailed description of alternate CPU and terminal configurations can be passed to the RTM, enabling it to produce a more exact response time and throughput analysis of the network using each configuration. The results of this analysis can then be passed back to the configuration modules, allowing them to produce a more detailed comparison of the alternatives. Again, this process can be iterated. Similarly, the topology of a design can be passed directly from the TOM to the reliability evaluation module. Note, however, that in each case, the functions of the module are kept distinct, even though each module makes use of information provided by other modules. This is as close as one can reasonably expect to come to having one's cake and eating it too.

**EXAMPLES**

The techniques described in the previous sections have in fact been used to design and analyze real networks. Many problems which would have been intractable to manual solution or which would have necessitated the development of costly software to solve them entirely automatically were solved by analysts using the basic design tools described above. The following illustrative examples were chosen from among these problems.

*Response time analysis and central site configuration*

Figure 1 is a simplified schematic of a network whose response time capability was recently analyzed using the techniques described earlier in this paper. A detailed schematic of the central processor configuration is given in Figure 2. Pairs of remote concentrators are linked to the CPU by high speed trunk lines ranging in speed from 7200 bps to 50,000
bps. Terminal controllers are linked to the remote concentrators by multidropped full duplex voice grade lines. When economical, groups of up to five low speed lines are multiplexed onto a single high speed channel. Each terminal controller handles up to 12 terminals. The network contains over 1,000 terminal controllers and almost 2,000 terminals.

Because of the stringent response time requirements used in the design of this system, the line discipline is unusually complex. The system handles two kinds of messages with different length distributions and different arrival patterns. Polling sequences are nested into acknowledgments in order to reduce queueing delays on the regional lines. In addition, flow control procedures are imbedded in the central processor and concentrator software to prevent any possibility of buffer overflows at the concentrators and CPU. The simulation package is organized in such a way as to allow for several
distinct line disciplines and, by a change of input parameters, for a variety of polling and selection sequences for each type of line discipline. Thus, even a system as complex as the above one could be analyzed accurately.

Part of a sample output from the simulation package is shown in Figure 3. Facility utilizations, parameters describing the system, line discipline, and the length of the simulation are given first. Next, the average values of several common measures of response time performance are given. Empirical distributions of this type are useful, not only because they provide information about systemwide variation in response times, but also because they often relate directly to design constraints, such as "90 percent of the messages must have terminal response times less than 20 seconds."

Figure 3—Sample output for response time analysis

From the collection of the Computer History Museum (www.computerhistory.org)
Terminal response time is defined as the time elapsed between the keying in of the last character of the input and the start of printing of the first character of the acknowledgment. Overall response time includes terminal response time, keying, printing, and waiting time at the terminal before the input is keyed in.

Empirical data collected from the system after it was put into operation was used to check the validity of these simulation results. The response times were found to be accurate to within five percent under normal operating conditions. Most of this variation was due to noise in the statistics. This simulation package is currently running on a CDC 6600 and requires roughly five seconds of CPU time to simulate 1,000 transactions, an adequate number for most purposes. It requires roughly 30,000 core locations to run. Because of the program's small running time and its ability to do multiple simulations on the same run by setting input parameters, it is possible to generate response times under different operating conditions very easily, thus facilitating a detailed analysis of the system's performance under a variety of assumptions about the system's loading.

On the basis of output from this simulation package, a detailed analysis of the throughput capability of the system was performed. The levels of traffic at which the central computer, the trunk lines connecting concentrators to the CPU, the concentrators, and the regional lines connecting terminal controllers to the concentrators became saturated were each identified for future planning and management of the network's operation.

**Economical network design using multiplexers and concentrators**

As an example of how much can be saved by incorporating concentrators and multiplexers into a telecommunications network design, a cost comparison between designs with and without concentrators and multiplexers for a network linking 102 locations across the United States is presented. The design was produced subject to the constraints that 90 percent of the transactions have response times no greater than 25 seconds during ordinary operating conditions and no greater than 35 seconds during the peak hour. Furthermore, the expected number of terminals communicating with the central
computer in a network using multiplexers and concentrators must be no more than .1 percent lower than the expected number communicating in a comparable network without multiplexers and concentrators. A simulation package was run to determine what combinations of facility utilizations (line, concentrator, CPU, and buffer) would safely meet the response time constraints under a variety of assumptions about traffic load and mix. Designs were then produced with and without multiplexers and concentrators, using a sophisticated network design package. The design using concentrators and multiplexers is summarized in Table I.

The design using multiplexers and concentrators saved over $900,000 in annual line charges when compared with a similar design without multiplexers and concentrators. Not only was the reliability of the concentrator/multiplexer design nearly indistinguishably different from the design without them, but also, because of the concentration of key network components in a small number of areas, it was possible to provide more effective and economical backup to the system when concentrators and multiplexers were used.

As an indication of the quality of designs which can be produced by a sophisticated network design package, some of these designs were compared with proposals sent by vendors for the same network. Designs supplied by the vendors did not consider reliability, and utilized a single line utilization constraint. The comprehensive design procedure using multiplexers and concentrators had reduced line costs by over 25 percent and runs made solely for the purpose of this comparison showed savings of up to three percent over vendor supplied designs, even when identical design constraints were used.

The running time and core requirements for the design package are essentially linearly proportional to the number of locations in the system. Designs of this system required an average of 7.5 seconds of running time and 35,000 core locations on a CDC 6600. The module can be used to design networks connecting as many as 1,000 terminals.

The user can even specify the tariff to be used, subject to the restriction that it is not dependent upon the topology. Thus, even the newly proposed high/low density tariff can be used in network designs. Figure 4 shows a design obtained from this module using AT&T's proposed high/low density tariff. Not only is the design of high quality, but also, the linear dependence of core and running time on the number of locations is unaltered.

This last fact is noteworthy. The high/low density tariff charges a lower rate for lines connecting high density points, reflecting AT&T's lower costs for providing service between these points. Thus, in general, when evaluating the cost of a connection between two low density points, one must also consider the possibility of routing the line through one or more high density points, in order to take advantage of the lower tariff between these points. If done naively, this calculation would greatly increase the complexity of any algorithm for designing networks using this tariff. If proper care is taken in developing the design module, however, it is possible to incorporate the flexibility of allowing the high/low density tariff in its design procedure without sacrificing efficiency in the design process or in the design itself.

CONCLUSION

We have shown that problems of network design and analysis can be successfully solved by an analyst equipped with tools in the form of modular programs, each of which is designed for a specific purpose-input, reliability analysis, configuration, response time and throughput analysis, and network design. These tools, when used together with the analyst's experience and judgment, can produce better solutions with less effort than could be obtained by a man or machine alone.

REFERENCES
