COMPARISON OF SOLUTION METHODS FOR COMPUTER NETWORK MODELS*

Leonard Kleinrock
School of Engineering and Applied Science
University of California
Los Angeles, California 90024

Summary

A discussion of the need and structure of computer networks precedes a comparison of a number of solution methods for models of these networks. Various analytic, simulation, and measurement approaches are discussed and referenced and the interplay between these methods is investigated.

1. Introduction

First came the vacuum-tube computers in the early 1950's.[1] They were of moderate speed (tens of thousands of instructions per second) and of small memory (a few thousand words). Those machines were used mainly by scientists for mathematical and scientific calculations in an on-line fashion whereby one man had exclusive use of the full machine for hours at a time. This procedure was highly efficient from the user's point of view, but sadly wasteful of the machine's capability for work.

Next came the organizers in the mid-to-late fifties who introduced batch-processing in order to use the faster larger machines (hundreds of thousands of instructions per second with 32 to 65 thousand word memories) in a more efficient fashion.[2] Thus the user was thrown out of the computer room and punched cards or tape were his sole input, yielding reams of printed paper output. This succeeded in achieving the efficient use of machine time, but was accomplished at the expense of the highly inefficient use of the user, resulting in many hours for turn-around from when a job was submitted until it was returned complete. This procedure was a disaster when it came to debugging programs.

In the early 1960's a major breakthrough was made with the advent of time-sharing. One of the first demonstrated time-sharing systems was developed at M.I.T. with more than one flexowriter simultaneously using the computer.[3] The principle behind time-sharing is that no single user typically requires all the resources of a computer facility at one time and therefore many simultaneous users can share these resources. As shown in Ref. 4, a useful measure of the number of simultaneous and non-interfering users is approximately the ratio of the average user "thinking" time plus average required computer time divided by the average computing time. For example, if each user spends 19 seconds generating a request which requires one second of computing time, then approximately 20 such users can be scheduled on a continuous, simultaneous basis without significant interference. These time-shared systems clearly are efficient in the user's time and are also rather efficient in the use of the computer.

Time-shared facilities have grown up across the country during the past

---

*This work was supported by the Advanced Research Projects Agency of the Department of Defense (DAHC15-69-C-0285).

decade and a variety of different kinds of services are available at relatively low cost to the user.[5] The user was thus effectively allowed back into the computer room, except now he was coupled to the machine through a remote terminal. These remote terminals may be as simple as a teletype console; a much more powerful terminal includes graphical input and output devices for the display of characters, curves, diagrams, etc. Novel input devices have been invented, such as light pens, wands (three-dimensional input devices), mice (two-wheeled devices which detect X-Y motion), eyes (in the form of television cameras or head-mounted displays). This strongly interactive relationship between man and machine has naturally led people to consider what should be the proper interface between a human and a computer and what limitations each places upon the other.[6]

Under time-sharing, facilities and systems have grown and developed into sophisticated and rather unique sites. Across the country, we have seen rather specialized capabilities developing at universities and research laboratories.[7] These special features take the form of exceptional computer programs, data files, hardware devices, resources, and human talent. These capabilities are available only at the computer center where they have been developed and are, in general, not easily transferable. It is clear however that these special resources should be available for use by other than those at the site. This desire naturally led to the concept of computer networks and represents the next major breakthrough in the use of computers.

One of the earliest computer networks was the SAGE defense network in the fifties.[8] This highly specialized military net demonstrated for the first time that software requirements of large computer systems could be as costly and difficult as the hardware equipment. The American Airlines SABRE Reservation System came shortly thereafter;[9] this too was a collection of rather uniform equipment for a highly specialized application. The electronically switched telephone system has, for a time, been the world's largest computer network, which again is highly specialized and single-purposed.[10] Recently, Control Data Corporation announced their nationwide network.[11]

It is our intention in this paper to demonstrate techniques for analyzing and synthesizing computer networks. We begin by first describing a particular computer network which is currently being implemented. We then proceed to describe the use of modelling for such networks and elaborate upon the many ways of solving these models.

2. General Description of Computer Networks--An Example

A computer network is a collection of nodes (computers) connected together by a set of links (communication channels). Messages, in the form of commands, inquiries, replies, and file transmissions, etc., travel through this network over data transmission lines. At the nodes, the tasks of relaying messages (with all appropriate routing, acknowledging, error control, queueing, etc.) and inserting and removing messages which originate and terminate at that node, must be carried out. Often, these tasks are separated from the main computing functions required of the node, as will be seen in the example below.

It is convenient to describe the structure and operation of computer networks through the use of an example.* The system from which this example is drawn is the Defense Department's Advanced Research Projects Agency (ARPA) Experimental Computer Network.[13] The concepts basic to this network were clearly stated in Ref. 14 by L. Roberts of ARPA who originally conceived this system. A more recent paper on the development of this particular network is given by B. Wessler[15] elsewhere in these proceedings. Moreover, there is to be a session on computer networks, chaired by L. Roberts, at the forthcoming ACM Symposium on Problems in the Optimization of Data Communications, Pine Mountain, Georgia, Oct. 13-16, 1969. This network will provide store-and-forward communication paths between a set of approximately 15 nodes in the continental United States. The computers located at each of the nodes are highly incompatible (e.g. S.D.S. 940, DEC PDP-10, IBM 360/67, UNIVAC 1108, GE 635, ILLIAC 4, TX-2, etc.), and one of the major challenges is to design a network in which this assortment of varied hardware and software systems can communicate and cooperate with each other.

The topology and identity of the various nodes in this net have undergone considerable change since the early design stages. Nevertheless, for purposes of our example, we will choose a given set of 19 nodes (an original network design); again we emphasize that the details below regarding topology and identification of nodes have been and still are being changed. This last also applies to some of the operating procedures.

The example set of 19 nodes (mostly ARPA research contractors at universities) is shown in Figure 1; two configurations are considered for purposes of this paper. Note that net 2 has three cross-country links, whereas net 1 has only two.

In order to interfere least with the existing operation of these various facilities, the message handling tasks (relay, acknowledgment, routing, buffering, etc.) will be carried out in a special purpose Interface Message Processor (IMP) collocated with the principal computer (denoted HOST computer) at each of the computer research centers. The communication channels will (in most cases) be 50 kilobit/sec fully duplex telephone lines, and only the IMPS (not the HOSTS) will be connected (through type 303 data sets) to these lines. This communication net, consisting of these lines, IMPS and data sets, serves as the store-and-forward system for the HOST computer net.

When the HOST has a message ready for transmission, it will be broken into a set of smaller packets (each of size approximately 1024 bits, or less) with appropriate header information. The IMP will accept up to eight of these (an assembly set) at a time. The packets will then individually make their way through the IMP network where the appropriate routing procedure will direct the traffic flow. For each IMP to IMP packet transmission, a positive acknowledgment is expected within a given time; absence of an acknowledgment (caused perhaps by channel noise detected by a cyclic error detecting code, or by lack of buffer space, etc.) will force the transmitting IMP to try the same or some different channel for retransmission. We assume that the messages fall into two broad categories: one-packet short messages, typically averaging approximately 350 bits due to short commands and acknowledgments; and long, or multi-packet, messages due to larger data transfers.

---

*Material for this description has been drawn in part from Ref. 12.

This short description will serve as a point of departure for our modelling techniques. (Further details of the ARPA net may be found in Ref. 12.)

3. Solution Methods for Computer Network Models

In order to gain understanding and insight into the behavior of computer networks, it is essential that one construct a model of the network. This model must contain enough salient features of the true network so that a successful solution to the model behavior will correspond sufficiently well with a meaningful understanding of the real-world situation. Whether we admit it or not, we do always create models in one form or another whenever we ponder the real world, and it is with respect to such a model that we relate our analyses, measurements, etc.

Below, we consider five methods of "solving" computer network models and compare them with regard to usefulness and cost.

3.1. Pure Analytic Solution. The approach here is obviously to describe the details of the model in a mathematically precise fashion and then to solve the resultant mathematical structure for certain quantities of interest. The principle advantage of this method (aside from its elegance) is that it allows one to perceive the entire set of solutions to the problem through the statement of one or a few equations. Thus, graphs of performance over many parameter variations are easily generated. Furthermore, mathematical optimization is possible.

The disadvantage of such an approach is that one must typically discard considerable detail in constructing a simple enough model so that it is mathematically tractable. Only the relatively simple models yield to precise mathematical analysis.

Examples of successful analyses of computer and communication networks are relatively few. H. Frank and I. Frisch describe a number of problems which yield to the methods of network flow theory.[16] R. T. Prosser has studied routing procedures in communication nets,[17,18] as has P. Baran[19] and B. Boehm and R. Mobley.[20]

This author has carried out an analysis of communication nets in some depth.[21] In this work, considerable treatment was given to the special case (of a more general result) of optimally assigning capacity in a network where the cost per unit of capacity was the same for all channels ($d_i = 1$, see below). More recently, we have examined the more appropriate case (for computer networks) where we take cost as proportional to channel length as well as channel capacity.[12] The pertinent equation from these latter studies is

$$T = \frac{\bar{n}\left(\sum_{i=1}^{N}\sqrt{d_i\lambda_i/\lambda}\right)^2}{\mu\left(D - \sum_{j=1}^{N}\lambda_j d_j/\mu\right)} \qquad (1)$$

where T = average time for a message to pass through the store-and-forward communication net

$\bar{n}$ = average path length for messages

d_i = cost per unit of capacity on i^{th} channel

N = number of channels in net

λ_i = average message rate on i^{th} channel

$$\lambda = \sum_{i=1}^{N} \lambda_i$$

$1/\mu$ = average message length

D = total cost of net

For the case $d_i = 1$, we found that the two competing effects in optimum* network design were: (i) concentration of traffic into a few large channels; and (ii) minimization of $\bar{n}$, the average path length. We now observe for $d_i = L_i$, where L_i is the length of the i^{th} channel** that minimizing the numerator of Eq. (1) requires that we (i) concentrate the traffic on a few large <u>short</u> (cheap) channels, whereas maximizing the denominator requires that we (ii) minimize $\overline{nd}$ where

$$\bar{d} \equiv \sum_{i=1}^{N} d_i \lambda_i / \lambda = \text{average capacity cost.}$$

This last can be seen as follows

$$D - \sum_{i=1}^{N} \lambda_i d_i/\mu = D - \frac{\lambda}{\mu}\sum_{i=1}^{N} d_i \lambda_i/\lambda$$

$$= D - \frac{\lambda}{\mu}\bar{d}$$

$$= D - \overline{nd}(\gamma/\mu)$$

and where γ = average message rate into the net from external sources and $\bar{n} \equiv \lambda/\gamma$. Since γ, μ and D are fixed, we must minimize $\overline{nd}$ in order to maximize the denominator. Note that $\overline{nd} = \bar{n}\bar{L}$ ($\bar{L}$ = traffic-weighted average channel length) may be interpreted as the average length of lines traversed by a message.

As an example of the use of Eq. (1) for the case where $d_i = L_i$, we consider the case of two possible nets connecting two nodes A and B as shown in Figure 2. For case (a) with an intermediate relay node C, we have

$$T = \frac{4}{\mu D - 2\lambda} \qquad (2)$$

For case (b) with no intermediate relay, we have

$$T = \frac{2}{\mu D - 2\lambda} \qquad (3)$$

Thus from Eqs. (2) and (3) net (b) is always twice as good as net (a). (The optimum capacity for each of the channels in Figure 2a,b is equal to D/2 bits/sec.) It is clear that more attention must be given to the case $d_i = L_i$ due to its importance in modelling computer nets.

<u>3.2. Approximate Analytic Solution</u>. Here we refer to the case where an approximation is made in the solution to a precise mathematical problem. A related method is that which simplifies (or approximates) the given mathematical model. In either case, one obtains an approximation to the solution of the original model.

The advantage here is that one can often obtain rapid solutions to problems and still consider a broad range of parameter variation. Clearly, the difficulty is that one is left with the task of evaluating how serious the approximations are with regard to the original problem. Some of the work available here may be sound in Ref. 12, 17, 18, 19, 21 and 22.

<u>3.3. Iterative Numerical Solution</u>. Often it is possible to write down the equations which govern the

---

* Optimum in the sense of minimizing T, the average message delay.

**For the ARPA net, $d_i = 48$ where d_i is expressed in dollars per unit of capacity, and L_i is in miles (for 50 kilobit lines).

dynamics of a system, but it is not feasible to solve them. In such a case, it may be possible to obtain a solution by various techniques (e.g. iterative) if one is willing to choose numerical values for the various system parameters.

Such is the case in queueing networks. This observation has led to the development of a digital computer program which accepts graphical input (via a light pen) for describing the network configuration and supplies histograms and measures thereof for numerous system variables as output. This effort has been carried out at the University of Michigan and a portion of it is described in Ref. 23.

The advantage of such an approach is that more difficult problems may be solved for particular parameter choices. The program may tax the storage and computational capacity of the computer when the problem gets at all large.

3.4. Simulation. A fairly common approach to solving the analytically intractable models is that of digital simulation.* The principle advantage here is that considerable detail can be retained in the model so as to correspond very well with the real world. Moreover, the results of such simulation studies often point the way to analytic studies by identifying those portions of the system which may be approximated or studied independently.

On the other hand, digital simuations are typically costly, slow, inconvenient, and often require extensive programming effort.

The ARPA computer network has been simulated at UCLA and many of the results are available in Ref. 12. A collection of 10-node sub-nets of the 19-node network has since been simulated and studied through the use of Eq. (1) as well. These comparative results make clear that the two approaches agree reasonably well, considering that the analytic model was significantly simpler and less detailed than the simulation model.

3.5. Measurement. One of the ultimate approaches for obtaining solutions to a model of the real world is to go out and measure the true system. One must then relate these measurements to variables in his model. Clearly one thus obtains a view of the world which includes all of the pertinent details. Of course one is not often allowed this alternative for "solving" systems and predicting performance. However, even after the network is constructed, we often have the opportunity to optimize its performance through a study of measured behavior. Moreover, such measurement may be built-in so that control of the network, or perhaps accounting, etc., may be carried out.

Indeed all of the previous solution methods remain suspect (as regards their ability to predict and describe the real-world) prior to their validation with respect to measured system performance. Fortunately, this need is currently being recognized. For example, Jackson and Stubbs[24] and Fuchs and Jackson[25] have measured the user interaction process at the terminal of time-shared systems. These measurements are of great significance.

Moreover, careful preparation has been made for taking considerable measurements on the performance of the ARPA network. For example, it is anticipated that periodic "snapshots"

---

*Unfortunately, it is sometimes the case that extensive simulation studies are conducted on systems which yield to analytic techniques.

of the IMP activity for dynamically selected IMPs will be taken and sent back to a "network measurement center" (UCLA) for analysis. These snapshots will include such things as: queue lengths; buffer conditions; routing tables; synchronization information, etc. Histograms will be collected at each selected channel to include: message rate distribution; message length distribution; number of errors, retransmissions, etc. Also HOST activity will be monitored with regard to: input message length distribution; output message length distribution; control message traffic, etc. We also intend to collect inter-arrival time statistics on network message traffic. In addition we will have the capability of sending "trace" messages through the net in order to track the path and path delay, etc., for portions of the network.

4. Conclusions

In discussing the various solution techniques for models of computer networks, we found that we were trading cost of obtaining results with usefulness and applicability of these results. The analytic approaches provide answers which allow overall understanding at basically a broad qualitative level and require that we further assess the effect of idealization built into the model. Measurement and simulation, on the other hand, provide detailed quantitative results for a narrow range of operation at relatively high cost.

It is always wise to combine these approaches so that the measurement and simulation serve to validate (or suggest improvements in) the mathematical solutions; conversely, the mathematical analyses serve to direct the simulation and measurement and to suggest the important areas and range of values which need to be studied more carefully in the measurement and simulation.

In the final analysis, however, one must always verify the input data to any of these models by making measurements which identify the nature of the data and structure of the system being modelled.

References

1. Rosen, S., "Electronic Computers: A Historical Survey," Computing Reviews, Vol. 1, No. 1, March 1969, pp. 7-36.
2. Rosin, R. F., "Supervisory and Monitor Systems," Computing Reviews, Vol. 1, No. 1, March 1969, pp. 37-54.
3. Corbató, F. J., M. Merwin-Daggett, and R. C. Daley, "An Experimental Time-Sharing System," Proceedings of Spring Joint Computer Conference, May 1962, pp. 335-344.
4. Kleinrock, L., "Certain Analytic Results for Time-Shared Processors," Proc. of the IFIP Congress '68, Edinburgh, Scotland, pp. D119-D125, August 5-10, 1968.
5. Hammerton, J. C., "Business Time-Sharing: User Economics," Datamation, Vol. 15, No. 6, June 1969, pp. 70-81.
6. David, E. E., Jr., "Physiological and Psychological Considerations," pp. 107-128, in W. J. Karplus (Editor), "On-Line Computing," McGraw-Hill Book Company, 1967.
7. Corbató, F. J., and V. A. Vyssotsky, "Introduction and Overview of the Multics System," Proc. AFIPS 1965 Fall Joint Computer Conference, Vol. 27, Part 1, Spartan Books, Washington, D.C., pp. 185-196.

8. Everett, R. R., C. A. Zraket, and H. D. Benington, "SAGE: A Data Processing System for Air Defense," *EJCC*, pp. 148-155, 1957.

9. Evans, J., "Experience Gained from the American Airlines SABRE System Control Program," *Proc. ACM National Meeting*, August 1967, pp. 77-83.

10. Keister, W., R. W. Ketchledge, and H. E. Vaughan, "No. 1 ESS: System Organization and Objectives," *Bell System Tech. Journal*, September 1964.

11. *Business Week*, February 8, 1969, p. 38.

12. Kleinrock, L., "Models for Computer Networks," *Proc. of the International Communications Conference*, University of Colorado, Boulder, Colorado, June 1969, pp. 21-9-21-16.

13. Dickson, P. A., "ARPA Network will Represent Integration on a Large Scale," *Electronics*, September 30, 1968, pp. 131-134.

14. Roberts, L. G., "Multiple Computer Networks and Intercomputer Communications," ACM Symposium on Operating Systems Principles, Gatlinburg, Tenn., October 1967.

15. Wessler, B., "The ARPA Computer Network," *Proc. of the Computers and Communications Conference*, Rome, New York, Sept. 30-Oct. 2, 1969.

16. Frank, H., and I. T. Frisch, "Communication, Transmission and Transportation Networks," to be published.

17. Prosser, R. T., "Routing Procedures in Communication Networks, Part I: Random Procedures," *IRE Trans. on Communication Systems*, CS-10(4), pp. 322-329, 1962.

18. Prosser, R. T., "Routing Procedures in Communication Networks, Part II: Directory Procedures," *IRE Trans. on Communication Systems*, CS-10(4), pp. 329-335, 1962.

19. Baran, P., "On Distributed Communication Networks," *IEEE Trans. on Communications Systems*, pp. 1-9, March 1964.

20. Boehm, B. W., and R. L. Mobley, "Adaptive Routing Techniques for Distributed Communications Systems," Rand Corporation, RM-4781-PR, February, 1966.

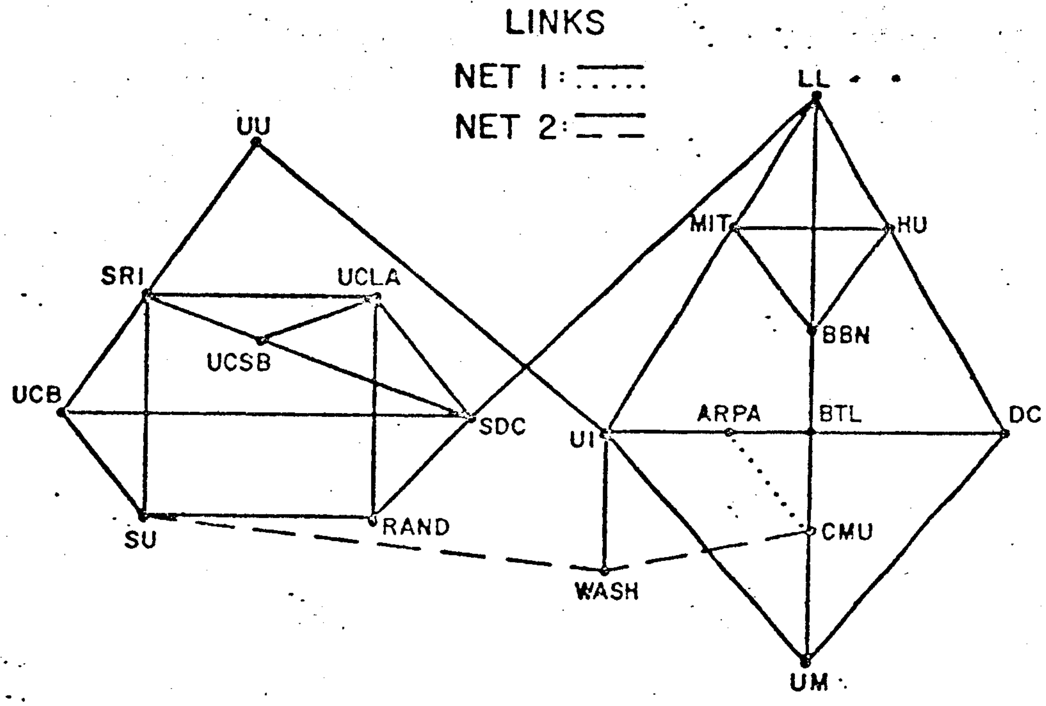
21. Kleinrock, L., *Communication Nets: Stochastic Message Flow and Delay*, McGraw-Hill, New York, 1964.

22. Kleinrock, L., "Sequential Processing Machines (SPM) Analyzed with a Queueing Theory Model," *JACM*, Vol. 113, No. 2, pp. 179-193, April 1966.

23. Wallace, V. L., and R. S. Rosenberg, "RQA-1, The Recursive Queue Analyzer," Tech. Rpt. No. 2, Systems Engineering Laboratory, University of Michigan, Ann Arbor, 1966.

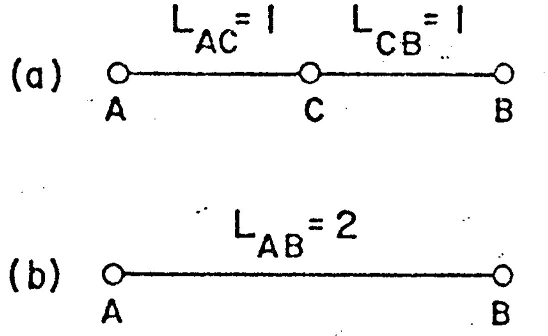
24. Jackson, P. E., and C. D. Stubbs, "A Study of Multiaccess Computer Communications," *Proc. of the Spring Joint Computer Conference*, 1969, pp. 491-504.

25. Fuchs, E., and P. E. Jackson, "Estimates of Distributions of Parameters for Certain Computer Communications Traffic Models," to be presented at the ACM Symposium on the Problems in the Optimization of Data Communications Systems, Oct. 13-16, 1969, Pine Mountain, Georgia.



TOPOLOGICAL NETWORK DIAGRAM FOR THE TWO CONFIGURATIONS

FIGURE 1



TWO POSSIBLE NETS CONNECTING
NODES A AND B

FIGURE 2