6

AN INTERCAMPUS DATA NETWORK FOR THE UNIVERSITY OF CALIFORNIA

CRAIG A. WILL

UNIVERSITY OF CALIFORNIA

DEPARTMENT OF
INFORMATION AND COMPUTER SCIENCE

TECHNICAL REPORT Number 6

December 1970

1. INTRODUCTION

This study considers the ways in which data communication between different campuses of the University of California can be provided. The purpose of this communication is to permit the sharing of computing resources among the campuses.

There are various reasons for desiring the sharing of computing resources.

An obvious need is the use of resources that are not available on the user's campus. These resources may be interactive languages, batch processors, or data banks.

If, however, intercampus data communication could be provided inexpensively enough on a large-scale basis, then other things would become feasible. A user could use remote facilities that were better suited to his needs (and thus more efficient) than those available on his campus; if the communication costs were low enough, overall economies would result.

Another prospect is that campus computing centers could specialize, each offering only a few well-supported services, depending upon the network to offer variety. This specialization could lead to greater efficiency and better service, and perhaps economies in software purchases. The network could also offer more hours of service and better overall reliability, since service from other campuses could be used whenever the local service was unavailable.

A degree of load-sharing would also result, since if the response time or turn-around time on one machine became unacceptably large users would take their business elsewhere. The net could also allow the remote storage of important data for security reasons.

The study first examines the characteristics of interactive and batch computing, and in particular those related to communications.

Then the ways in which intercampus data communication can be provided are examined, including circuit switching systems such as the ITS and the IDSS. The problems with circuit switching are examined, and the characteristics of message switching are presented, motivated by the desire to increase the efficient utilization of communication lines and to solve compatibility problems.

Two different types of message switching networks are presented, the "star" net and the "distributed" net, and their characteristics compared on the grounds of cost, software complexity, capacity, and reliability. The "star" net is tentatively chosen as the most cost effective for current needs.

A cost estimate of the equipment, development, and operating costs of the star net is presented, and a model formulated to show cost and performance. Curves are shown that present the estimated cost-per-terminal-hour for interactive terminal use of the net and cost-per-million-bits for remote batch processing. The costs of a circuit switching system are also presented for comparison. Presented also is a development plan for the implementation of a prototype network for interactive traffic only among the five southern U. C. campuses.

The conclusion reached is that a message switching system can provide interactive terminal communications at a cost low enough to allow large-scale sharing of interactive computing resources among the campuses, much less expensively than can be provided with a circuit switching system.

Message switching is also useful for remote batch processing, although the cost savings are not as dramatic as in the case of interactive computing.

2. WHAT ARE THE NEEDS FOR INTERCAMPUS DATA COMMUNICATION?

Computing facilities on other campuses may be used either interactively, meaning via on-line interaction with the computer, or through remote batch processing, where jobs are entered into a batch stream from a remote location. The characteristics of these two uses and their requirements for data communication are now considered.

Interactive Use

On-line interaction requires access via a terminal, such as a teletypewriter, typewriter-like device, or CRT display. Such terminals send and receive data a rates of from 110 to 2400 bits per second. These are peak rates; the average rate is much lower than this due to factors such as the user's typing speed, delays introduced by the computer, and "think time" experienced by the user. The average data rate of a Teletype or typewriter-speed terminal is about 18 bits per second, while a fast CRT display may average 100 bits per second. The first figure is from a study by Stubbs and Jackson[1], while the second is extrapolated from the model they present. These rates are for alphanumeric terminals; graphic CRTs such as those used with the Culler-Fried system at Santa Barbara will have a higher average rate, perhaps 600 bits per second.

A problem with interconnecting terminals and computers is compatibility. Terminals operate at a number of data rates; common ones are 110, 134.5, 150, 300, 1200, 1800, and 2400 bits per second. Transmission of data is usually asynchronous but may also be synchronous. Terminals may require echoing or not, transmission may be either half or full duplex, and various codes, character formats, and communication protocols are used. For terminals to communicate with a CPU the characteristics of the terminal must match the characteristics of the connection to the computer.

Terminals usually have fixed characteristics, although some CRTs have switch-selectable speed, and at least one typewriter terminal has a selectable character code.[2] Some terminals can in addition select whether or not they require characters to be echoed or not, but this is about the extent of the versatility that terminals can have without internal modifications. Recently, more "intelligent" terminals have been introduced that have greater versatility, such as the Imlac terminal[3] and the Datapoint 2200[4]. These terminals contain a built-in stored program CPU, allowing the characteristics of the terminal to be easily changed.

At the computer end of the terminal-computer connection, some of the characteristics are determined by the operating system or the interactive language in use, especially the character set and whether characters are to be echoed or not. Other characteristics are determined by hardware options in the CPU's terminal communications equipment.

Of some consideration with interactive computing is the delay introduced by the data communication network, since appreciable delays can increase the response time experienced by the user, slowing down the computing that can be done, and irritating the user. Minimizing this delay is particularly important when characters are echoed by the remote CPU.

The "holding time", the length of time that a terminal is connected and in use, is quite long for interactive terminals, ranging from 10 or 15 minutes to an hour or more.

Remote Batch Processing

The usual way in which remote batch processing is done is with a card reader and line printer. This may be a remote batch terminal (such as an IBM 2780 or a Univac DCT 2000) or a small teleprocessing computing system such as an IBM 360/20 or 1130. Remote batch work can also be done by the use of peripherals on a local large-scale computer, provided the communications equipment and software will permit it.

Remote batch processing terminals generally run at near the capacity of a voice grade communication line. Card readers read cards at a rate of

100 to 600 cards per minute, corresponding to a data rate of about 1000 to 6000 bits per second. Line printers typically operate at 300 to 1000 lines per minute, or 3600 to 13,000 bits per second. These figures are peak rates. Average data rates are lower than this, although the difference between peak and average data rate is not nearly as pronounced with batch work as it is with interactive. In the cost-performance estimates presented later the ratio of average to peak data rate is approximated at 50%.

A typical example of remote batch processing is as follows: A job is submitted on punched cards and read by a card reader, with the data being transferred to the remote CPU. When the deck has been completely read in, the remote CPU begins processing. After the processing has been completed, the output is transmitted back to the local station, and printed on a line printer. The average data rate, then, depends not only upon the speed of the card reader and line printer, but also on the type of job being run and on characteristics of both the remote and local (if any) operating system, such as buffering and the ability to multiprogram.

Compatibility is also a problem with remote batch terminals, although the extent of the problem is not clear. Transmission is almost always synchronous, at speeds of 2000, 2400, or 4800 bits per second. There are, however, variations in character sets, error checking methods, and communication protocols.

Holding times for remote batch work are usually much shorter than for interactive terminals, a few minutes per job. There is no need for rapid real-time response, and delays in processing messages are therefore of little consequence.

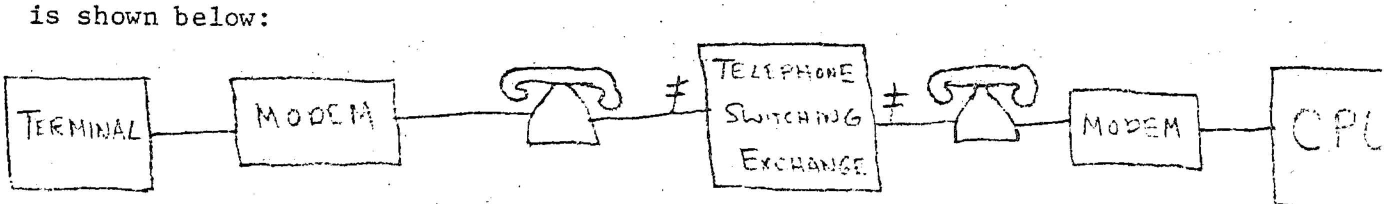
3. HOW CAN THESE NEEDS BE SATISFIED?

For intercampus resource-sharing to become practical, a sufficient number of terminals must be able to access computing facilities on other campuses. The more terminals that have this capability, the easier it will be for services from other campuses to be utilized.

To allow this access, some means of switching must be provided so that the terminal can communicate with the particular campus desired. Also, since distances to other campuses are great enough to make the cost of the communication lines rather expensive, it is important that these channels be used in an efficient manner. Two means of switching will be investigated, "circuit switching" and "message switching".

Circuit Switching

Circuit switching consists of switching the physical wire that provides the data transmission path. Once the connection is established, the two ends can communicate in any convenient manner, as long as they are compatible with each other. This technique is used by the ordinary telephone switching exchange (dialing establishes a connection, after which communication can begin), and in fact the campus telephone exchange is often used to connect terminals to computers. This is done by the use of a terminal and "modem" at the user's end, and an automatic-answering telephone and modem at the computer end. The purpose of the modem is to convert digital data into audio tones suitable for transmission over telephone lines; such devices are also known as "Datasets", "Data-phones", or "couplers", and may connect to the telephone linereither directly or accoustically through the telephone handset. A typical arrangement is shown below:



The use of the telephone system as a switch to connect terminals with computers gives two advantages over a dedicated, "hardwired" connection. First, it gives each terminal flexibility, allowing it to connect to many different CPUs. Secondly, it allows efficient use of "ports" into the computer, since only those terminals actually using the system are monopolizing a port. This is significant because many timesharing systems will only allow as many ports as simultaneous users, and dedicating a port to a little-used terminal reduces the use of the machine. Even where this is not true, current computer pricing policy places a premium price on terminal connections, making circuit switching economically advantageous. The problem of port efficiency can also be solved by use of a "portfinder", a circuit switching device that allows a number of terminals to compete for a lesser number of ports. Such a device is being installed on the Sigma 7 at Irvine. This will allow higher speed connections than practical with a telephone system, at a lower cost. Signals are sent at up to 2400 bits per second, without modems, over a full duplex (four-wire) connection.

Intercampus data communication via circuit switching can be accomplished by two systems, both of which are extensions of the telephone exchange idea. One is the University tie-line or ITS system, while the second is a circuit switching scheme planned exclusively for data, known as IDSS.

The ITS System

This system allows a user with a telephone on any campus to connect to a computer on another campus, by simply dialing it up. Since this system already exists, with communication lines connecting all campuses, it can provide a flexible data communication service without a large fixed cost. Also, the ability to share use of the system with voice traffic allows terminals that are only occasionally used inexpensive access, requiring only an ordinary telephone. A user may call either a local or a remote computer with the same equipment.

A disadvantage to the ITS is its poor quality, causing a high error rate. Also, the lines cannot be conditioned for data, limiting the transmission speed to about 2000 bits per second. The ITS provides only a single two-wire line, and so full duplex communication (necessary for echoplex operation) is not feasible above 300 bits per second.

The use of ITS also presents an administrative problem. The long holdling times that occur with data cause a disproportionate use of the intercampus lines by data users, who are still billed the same flat rate as voice users. Because of this problem, the ITS is presently restricted to voice traffic only except during evenings and weekends.

The ITS is planned to be replaced in 1973 by a system of higher quality. The new system will have a means of billing intercampus calls, allowing the costs of the system to be distributed more accurately. Use of the system for data would then be possible.

<u>The IDSS</u>

The other system is the Intercampus Data Switching System, a circuit switching scheme proposed with a central switch in Berkeley, and leased lines to each of the other campuses. A user on a campus buys a "port" on the switch in Berkeley, including a line to his campus, for a flat rate of about $ 150 to $ 300 per month. This is a four-wire, full duplex channel, conditioned to allow data transmission at up to 4800 to 9600 bits per second in each direction. Included with each connection is an ordinary telephone to allow other ports to be dialed, with a two-digit number. With the switch presently planned, a maximum of 60 ports can be used, and a maximum of 10 simultaneous connections at any one time are allowed.

<u>Problems with Circuit Switching</u>

There are two main problems with circuit switching, both of which are most troublesome with interactive terminals. One problem is the inefficient use that interactive terminals make of the capacity of a line, while the other problem involves the inefficiencies and difficulties resulting from the requirement that both ends of the line be compatible with the other.

Ineefficient Use of Line Capacity

Interactive terminals have a rather slow average data rate, especially when the terminals are slow to begin with, such as a Teletype. A voice-grade line has a capacity of 2000 bits per second unconditioned, and 4800 to 9600 bits per second if suitable conditioned. A Teletype-speed terminal, then, with an average data rate of 18 bits per second, does not make very efficient use of the line.

A common solution to this problem is "multiplexing", a means of allowing a single voice-grade channel to carry several channels of terminal-computer conversations. Multiplexing can be done by either time-division or frequency-division techniques, although it is useful, in general, only for slow-speed terminals, up to 300 bits per second.

Multiplexing, however, is not really a solution to the problem, since it is only useful when a line has been set up between two campuses, and more than one computer-terminal conversations are desired between these campuses. This might be desirable in special cases, such as providing communication over a single line for a cluster of terminals that all use the same remote CPU much of the time. However, in such cases it would seem cheaper to obtain a dedicated line between the two campuses rather than go through the switch in Berkeley, since the connection must exist only if one terminal is in use.

The only case where multiplexing might be advantageous with circuit-switching would be in a situation where it was desirable to use a cluster of terminals only within certain hours, and where the line could be used for other purposes the rest of the time.

Compatibility and Port Fragmentation

The other problem with circuit switching is that each end of the circuit must be compatible with the other. They must operate at the same data transmission rate, have the same method of transmission (asynchronous or synchronous), the same character format, and use modems with compatible signalling frequencies.

Usually, each computer "port" has a specified speed, character format, and possibly other characteristics. For each different type of terminal that is to be connected to the CPU, a seperate port must be available with the proper characteristics.

This requirement leads to a problem known as "port fragmentation", where empty ports exist but are of the wrong type and are therefore unusable. This problem is most serious with interactive terminals because of the wide variety of characteristics common, but can also be a problem with batch terminals.

Message Switching

A message switching system accepts data in the form of messages, and routes these messages to their proper destination. Unlike circuit switching, data is not sent directly in real-time from the source to the destination, but instead is stored and forwarded, eventually reaching the destination.

Message switching solves the two main problems that plague the use of circuit switching with interactive terminals--inefficient use of line capacity, and compatibility. Message switching also provides increased reliability in that it can have the ability to detect and correct transmission line errors.

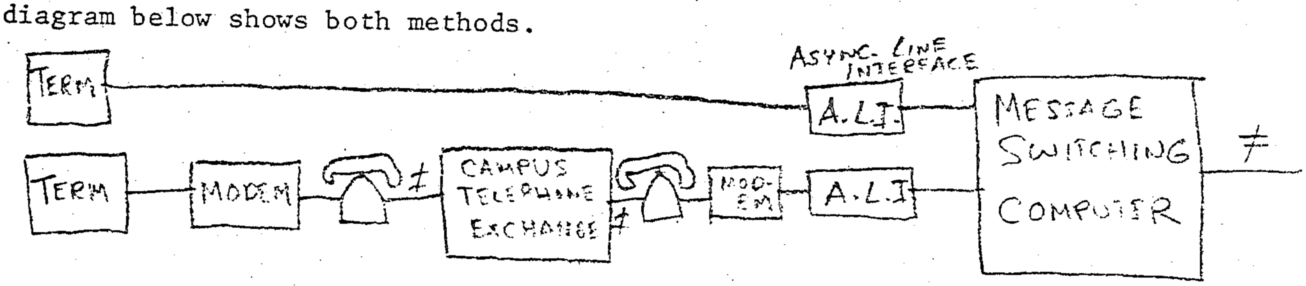
Message switching allows more efficient use of communication line capacity, since data is packed together and sent across the line at high speed. Also, individual characters are not sent immediately across the line but are buffered to remove peak bursts, and thus the required line capacity is that of the average data rate of the terminal rather than the peak rate.

The problem of compatibility and port fragmentation is reduced since the message switching system can accept and deliver data at whatever speed is required by the terminal or CPU. A terminal need only be connected to a message switching computer set up for the proper characteristics of that terminal, and it can then communicate with remote computers without regard to these characteristics. Message switching solves the compatibility problem inherently for transmission speed, character format, and type of transmission. Other characteristics, such as character sets, special communication protocols (such as locking and unlocking the keyboard of a terminal), or echoing, can be provided by a message switching system it it has been programmed to translate from one set of characteristics to another. Thus, a strange terminal can "look like a Teletype" to a CPU.

A message switching system includes a computer on each campus. This computer does data packing to allow the use of highspeed modems, does buffering to smooth out peak loads, and provides reliability by detecting transmission errors when they occur and retransmitting the data. This CPU also provides a way of connecting terminals directly to the network without going through a large-scale computer.

Connecting Terminals to a Message-Switching Network

Terminals may be connected to the message switching computer either directly or by use of dialup Datasets and the campus telephone exhange. The diagram below shows both methods.

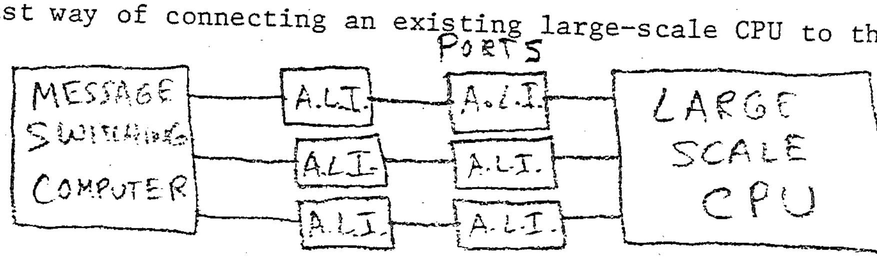


The direct method can work quite well. The connection itself can be made rather inexpensively with MOS terminal interfaces--such devices have recently been introduced for about $ 40 for both transmitter and receiver.[5] For most distances within the same campus, four-wire twisted pair can be used without modems for the transmission path, at speeds up to 2400 bits per second. The cable itself can be either University-owned or leased from the telephone company. (current rates for local lines are $ 4/month per half mile for a four-wire connection). The direct connection method is inexpensive, allows full duplex transmission at high speed, and trouble is easy to track down.

The other terminal connection method uses dialup Datasets. This presents some of the same problems of compatibility and port fragmentation as does circuit switching. A possible solution is the adaptive terminal interface--a port that samples the characters being received and adjusts itself to the characteristics of the terminal being connected. However, there are difficulties in implementing this and it is probably not useful at speeds above 300 bits per second, partly because the different type of Dataset required present another compatibility problem.

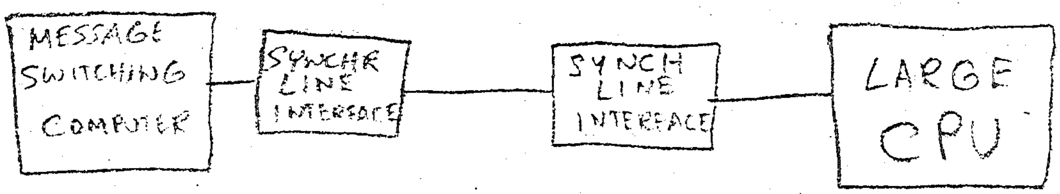
<u>Connecting CPUs to the Message Switching System</u>

The simplest way of connecting an existing large-scale CPU to the net is shown below:



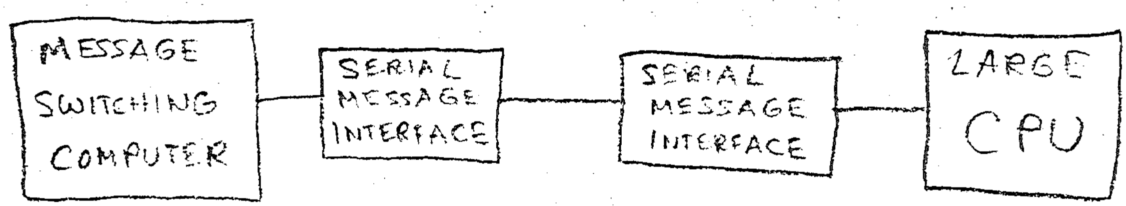
The message-switching CPU connects to the large-scale CPU as if it were a dedicated, hardwired terminal. It is desirable that the ports connecting the m-s computer run at a high speed; this would allow both high and low-speed terminals to efficiently use the CPU. A low speed port could also be used, however highspeed terminals logged in from the net would be limited to the speed of the port. This scheme has the advantage of requiring no hardware or software modifications to the existing CPU. However, it does require that some ports be permanently assigned to the net. Terminals connected to the m-s computer can use either a remote CPU or the local CPU, although terminals connected directly to the local CPU cannot gain access to the net without modifying the software of the local CPU.

A second method of connecting to a CPU is shown below:



This method allows all data to pass through a single connection, using synchronous communication equipment. This type of connection can be provided as standard equipmnt by almost all CPUs; the only limitation usually is speed. This connection requires modifications to the software of the CPU, although if necessary, the M-s computer could be programmed to emulate rather closely the function of the large CPU's communications equipment so that a minimum of changes would be required. The m-s computer might, for example, have a list of port numbers that it can assign, and when terminals converse with the large CPU the m-s computer simply sends a port number and character across the interface, and the large CPU handles this very similarly as it would a character received with its own terminal communications equipment. Further modification of the software would allow terminals connected to the large CPU access to the network.

A third method for connection to a CPU is shown below:



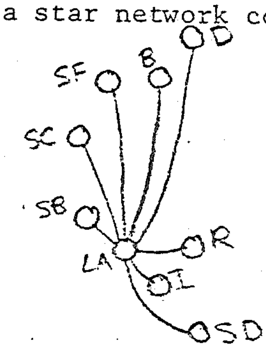
This method is very similar to the second method, the only difference being that it allows a connection to be made at greater speed and efficiency. The interface shown is serial, eliminating problems of variations in word length. A good interface specification to use for this connection is the IMP-Host interface used in the ARPA net,[6,7,8] since interface equipment for several large CPUs has already been built.

4. TYPES OF MESSAGE SWITCHING NETWORKS

Two different types of message switching networks will be examined as to their suitability. One type is the "star" network, while the other is the "distributed" network.

A Star Network

The diagram below shows a star network connecting the nine campuses of the University.



Each campus is connected to a central switch, shown here at Los Angeles. This is similar to the IDSS circuit switching system (which has a central switch in Berkeley). However, with message switching, a computer is used as the central "switch", accepting messages from each line and routing them to their proper destination.

The central switch in a star network introduces two important problems--reliability and the limitation of network capacity.

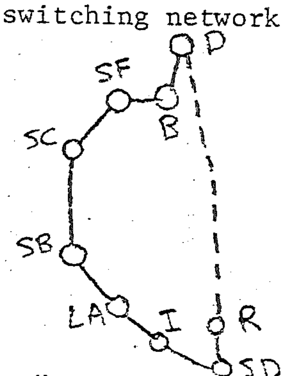
Reliability is a problem because the integrity of the network depends upon the proper functioning of the central switch. If the software or hardware in the central switching computer fails, the network does also. Also, communication line failures can isolate a campus from the central switch and thus from the network.

A limitation is placed on the capacity of the network because all data in a star net must pass through the central switch, and thus the total capacity of the network depends upon the ability of the central switching computer to process data.

To see more clearly the flow of data through the network, consider the case of a user with a Teletype at Berkeley who is communicating with a CPU at Irvine. Data from the Teletype is placed into a message by the message switching computer at Berkeley, and then sent to the central switching computer at Los Angeles. The central switching computer deciphers the destination address and routes the message to the Irvine m-s computer, which unpacks the data from the message and transfers it to the CPU. If the central switching computer (at UCLA) should fail, or either the communication line from Berkeley to UCLA or UCLA to Irvine fails, the Teletype/CPU conversation dies.

A Distributed Network

The other type of message switching network is the "distributed" network, shown below.



This network is "distributed" in that there is no central switch or control whose failure destroys the functioning of the entire net. This type of network has more line connections than absolutely necessary, such as the Davis-Riverside path shown as a dashed line. This redundancy provides an alternate path so that the failure of one line or one computer does not prevent a pair of nodes from communicating. The alternate paths also allow messages to be "adaptively routed" in order to equalize the flow of data through the network.

An example of the distributed type of net is the ARPA network[6], a nationwide net connecting together large-scale computers at various universities and research centers, including UCLA and UCSB. The ARPA system is a high-performance system designed to transport large amounts of data over long distances.

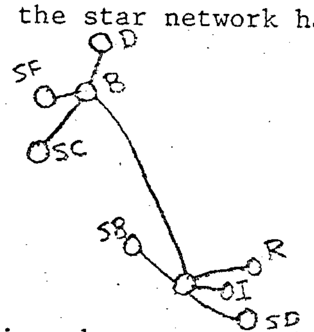
A distributed net is not subject to the reliability and capacity limitations that the star net, since there is no central "switch". However, a distributed net requires much more software complexity than does a star net, with each message switching computer keeping track of the status of other nodes, determining the best path for the messages it has to route. The message switching computer used in the ARPA net is known as an Interface Message Processor, or "IMP".[7]

We will again consider the case of a Teletype at Berkeley conversing with a CPU at Irvine, to show the flow of data in a distributed net. As before, the local m-s computer at Berkeley accepts data from the Teletype and makes up a message. (Although in the case of the ARPA net terminals are not directly connected to the IMP but instead through another computer. This will be ignored in the present discussion). There are now two paths that the message could follow. The shortest one is through Davis, Riverside, and San Diego. The message does not pass directly through the nodes in these locations; instead the message is stored in core memory at each of the nodes, and then forwarded to the next. If this shortest path is not available, however, due to the failure of a communication line or node computer along the way, the message could still get through, via the alternate path through San Francisco, Santa Cruz, Santa Barbara, and Los Angeles. The alternate path might also be used if, for example, heavy traffic was flowing between Riverside and San Diego, and quicker routing was available through the alternate path, even though it is longer and passes through more IMPs.

The topology shown for the distributed net is optimized for minimal line distance. As traffic through the net increases, more communication lines may be added, arranged so that the flow of data through the net is equalized.

A Two-Star Network

A variation of the star network has two central switches, as shown below:



The configuration shown uses Berkeley and Los Angeles as the locations for the central switches, with a trunk line connecting them.

The double-star net not only requires shorter communication lines, but more importantly, allows a central switch to be located at the two centers of data traffic, Berkeley and UCLA. Depending upon the actual pattern of traffic experienced on the net, the double-star configuration may significantly increase the efficiency of line utilization over a star net with a single central switch. The double star net also has nearly twice the capacity of a single-star net, assuming that most of the traffic in each star does not travel through to the other star.

Picking a Network Topology

The important differences between the star net and the distributed net are cost, software complexity, data capacity, efficiency of use of communication lines, and reliability.

Cost and Complexity

A distributed net has significantly greater software complexity than the star net. Also, a distributed net, is more difficult to develop and to fully debug, since the alternate paths make malfunctions difficult to trace. For these reasons, the development of a distributed network in-house would probably not be feasible. An alternative way to build a distributed net is by purchase of IMPs from Bolt, Beranek, and Newman, the designers of the ARPA net.

IMPs, however, were designed for high-volume traffic and long communication lines, and are rather expensive. Also, terminals cannot be directly connected to an IMP but must be connected through a "Host" CPU. This requirement would probably necessitate the addition of another mini-computer connected to each IMP so that terminals can be connected without going through an existing CPU. This is desirable for two reasons. First, it allows terminals to be used when service from the regular large-scale CPU is not available. Secondly, it allows access to the net without software modification to the large CPU, and from terminal types that may not be supported by the large CPU.

Even if a new distributed net was designed, it would probably still be wise to use two CPUs on each campus. Besides the question of whether one CPU has enough processing power for both the store-and-forward and message formatting and terminal control functions, there are reasons for separating these functions into two CPUs. Since each campus may want to modify the software for terminal handling to their own special requirements, the CPU that does this should not be the same CPU that handles messages passing through from other campuses. In other words, the distributed net should be a sealed-off system, to prevent the proliferation of user-introduced bugs from degrading the net in random ways that would be difficult to detect.

A star net not only has simpler software, but would require only one CPU on each campus plus one or two more CPUs for the central switch(es). Each campus node computer handles only traffic for that campus and thus cannot affect other parts of the net.

The capital costs for a distributed net using IMPs are $750,000 to $1,000,000, including message switching and terminal control CPUS, interfaces to terminals and Host CPUs, and software. A star net would cost about $350,000, including a message switching and terminal control CPU for each campus, infaces, and software.

Capacity and the Efficient Use of Communication Lines

The capacity of a star net is limited by the ability of the central switch to process data. This limitation is heavily dependent upon message length. Messages resulting from batch processing traffic, due to their greater length, allow more data to be passed through the net for the same amount of processing than do shorter messages that result from interactive traffic. The capacity also depends upon the processing power of the CPU and the complexity (and efficiency) of the line interface hardware. The capacity of a single star net, then, could range from 200,000 to 1,500,000 bits per second.

Whether the distributed net or the star net makes the most efficient use of communication lines depends largely on what the traffic will be. If the major source of traffic is one (or two) campuses, then a star net might be best with the central switch(es) at the traffic source. The number of nodes in the net is also a consideration. A distributed net with only nine nodes is probably not as effective as a net with a greater number, since not enough paths exist to efficiently equalize the flow of traffic. A distributed net might be more practical if the net served not only the university but also the state colleges and other state agencies.

Reliability

The distributed network clearly offers a higher degree of reliability than does the star net. However, the price of this reliability is high, and we must carefully consider how much reliability is really needed.

For uses of the net that are presently proposed, the need for reliability is basically to minimize the downtime, when services are not available. There is no need for absolute reliability as might be required in, for example, on-line patient monitoring in a hospital.

To compare the reliability of the star net with the distributed net, we will consider a user at one location, and see how often he is denied access to remote computing services.

A message-switching computer can be very reliable, since it does not need to depend upon mechanical devices such as discs. A ruggedized IMP has a mean-time-to-failure of 10,000 hours, or about one year.[6]

With a distributed net, a user is denied access to remote computing when either the message switching computer fails or the terminal controlling computer fails, or when there is a simultaneous line or node computer failure in both paths so that the message cannot get through. The computers can be expected to fail and cause downtime about two days out of the year, for a few hours each. The probability of failure in both paths is very low, and has been estimated for the ARPA net at about 30 seconds per year.[6]

With a star net, a user is denied access to computing when either the local computer fails, the central switching computer fails, or there is a line failure cutting the campus off from the central switch. As in the case of the distributed net, the failure of CPUs cause downtime two days out of the year. The most serious problem with the star net is line failures. Line outages of a few seconds or less can be handled by the retransmission of lost messages. However, outages longer than this will cause interruption of service for varying amounts of time. Accurate figures are not available for the frequency and length of line outages or for the probability of complete failure, however it might be estimated that a typical campus-to-central switch link would have a total downtime of a few hours per year. When traffic becomes sufficient to require more than one line between a campus and the central switch the duplication may help reliability, although it is not clear how much since much of the equipment likely to fail may be common to both channels.

So far we have only considered message switching computer failures due to hardware malfunctions. However, a main cause of the failure of a computing system is design bugs in the software. In order to have a central switch that is highly reliable, this problem must be coped with in some reasonable fashion. This is probably best done by keeping the software in the central switch as simple as possible. The central node software should be frozen very early in the development of the system, fully debugged, with as much functional capability in the campus local nodes rather in the central switch.

This goal of simple central switching software is a reasonable one, since the switch can perform with very few functions. Messages must be accepted from each line, transferred into core, and the destination address deciphered. The message is then set up for routing to the line leading to the proper destination. While these tasks are done error checking must be performed on each received message, and acknowledgement messages generated for each correctly received message. Each message passing through the system must be held until an acknowledgement has been received indicating that it has been correctly received by the destination node.

A further means of increasing the reliability of the central switch would be to duplex the processors, so that a standby CPU could take over in the event of failure. However, line failures are probably more of a problem than failure of the central switch, and this should not be necessary.

A Conclusion

The star net appears to be the best choice, since the cost is much less than that for a distributed net. The capacity and reliability characteristics of the star net, although inferior to those of the distributed net, are adequate to meet the needs of the University for perhaps the next 5 years. The next section presents a cost-performance study of the star net, and later a development plan is presented for a prototype star net.

LINE DISTANCES AND MONTHLY LINE CHARGES FOR VARIOUS NETWORK TOPOLOGIES

Rates are based upon TELPAK charges of $.52 per mile, channel terminals at each end at $35 per end, and line conditioning charges of $38 per end for C2 conditioning. Each line is a four-wire (full duplex) connection.

SINGLE STAR NEWWARK -- 9 campuses

Central Switch Location	Mileage	Monthly Charges
Berkeley	1957	$2186
Santa Cruz	1784	2096
Santa Barbara	1608	2004
Los Angeles	1622	2011
Irvine	1769	2088
Riverside	1835	2122

DOUBLE STAR NETWORK -- 9 campuses

Berkeley	Los Angeles	770	$ 1568
"	Irvine	786	1577
"	Riverside	842	1606

SOUTH-ONLY STAR NET -- 5 campuses

Central Switch Location	Mileage	Monthly Charges
Los Angeles	301	$ 741
Irvine	272	725
Riverside	325	753
Santa Barbara	540	865

NORTH-ONLY STAR NET -- 4 campuses

Berkeley	131	$ 506
San Francisco	140	511
Santa Cruz	235	560

DISTRIBUTED NETWORK

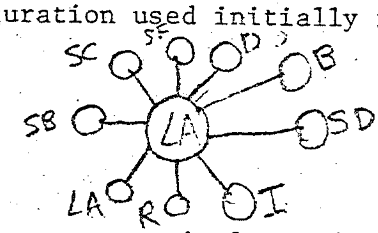
The topology with minimum line distance is: Davis-Berkeley-San Francisco-Santa Cruz-Santa Barbara-Los Angeles-Irvine-San Diego-Riverside-Davis.

This configuration has 9 lines with a total distance of 1011 miles, and line charges of $1840/month.

5. A PROPOSED NINE-CAMPUS NET--COST AND PERFORMANCE

This section presents a proposed store-and-forward message switching network, using the "star" topology. Costs of the net are estimated, and the capacity of the net is calculated for both interactive and batch processing use. The costs of the net are then shown in terms of estimated cost-per-terminal-hour (for interactive use) and cost-per-million-bits-of-data (for batch processing). The costs of a circuit switching system similar to IDSS are shown for comparison.

The network configuration used initially is shown below:



A central switching computer is located at UCLA, while a local message switching computer exists on each campus, and is connected to UCLA via a communication line. All communication lines are full duplex voice grade with C2 line conditioning, and operate at 4800 bits per second.

Data Capacity of the Net

For the configuration shown above, the aggregate data capacity of the net is 4.5 times the capacity of each line, or 43,200 bits per second.

This figure represents the total amount of data that can pass through the net under optimum conditions. However, this is not all usable--some of it is used by the message switching system itself as overhead, while the total net capacity cannot be fully utilized under less than optimum conditions of load distribution. Finally, the usable capacity of the net is not fully used all of the time, and so the costs must be based upon average use although the net must have sufficient capacity to handle periods of peak loading.

Overhead with Message Switching

With message switching, some part of each message must contian addressing information, reduncancy to allow detection of errors, and miscellaneous service bits.

This overhead is greatest for interactive traffic. One reason for this is that such traffic generally occurs in short bursts of about 20 characters in length. Another reason is that many time-sharing systems operate in "echoplex" mode, where hcaracters are echoed by the timesharing CPU in real-time. This necessitates two messages for every character typed by the user. Fortunately, input accounts for only about 10% of the total interactive data traffic, so one-character messages are not too frequent. Also,many systems operate in non-echoplex mode, allowing a whole line to be saved up before transmission to the CPU. This can also be done with echoplex operation, but is difficult because of variations in echoing characteristics and input termination characters with different operating systems.

For the purposes of the cost model, it will be assumed that 60% of the net capacity is useful for interactive traffic, while 40% is consumed by overhead.

Overhead for remote batch processing is less since the bursts are much longer, and in fact message length is usually limited by the lack of buffer storage. For the cost model, the 80% of the net capacity will be assumed to be useful for batch traffic.

Line Capacity Fragmentation

The calculation of basic capacity for the "star" net--where the capacity is calculated at 4.5 times the capacity of each line--assumes that the use of the lines is evenly distributed among all the campuses. This is not necessarily the case, however. Suppose, for example, that all the terminals on the net wished to log onto a computer at Irvine. In this case the usable capacity of the entire net would be limited by the capacity of the line from the central switch to Irvine. This waste of capacity due to a less than optimum distribution of load is called here "line capacity fragmentation". In the cost model, 50% of the network capacity will be assumed to be unusable due to this effect.

To reduce the effect of line capacity fragmentation, the central switch should be located on the campus with the major source of traffic, since lines from a central switch on the same campus can be run cheaply at very high speed. For this reason, UCLA was picked for locating the central switch for a single-star net. A double-star net with switching sites at UCLA and Berkeley would further reduce this effect and reduce line charges as well.

Line capacity fragmentation can also be reduced when the net has enough traffic to require more than one communication line between campuses, since the number of lines can then be adjusted to compensate for long-term averages of uneven load distribution.

Average Use of the Net

Cost figures for the net are based upon use 60 hours per week, or about 260 hours per month. However, the full capacity of the net cannot be expected to be used all of the time. The costs of the net should be based upon use of the net something less than full-time, although the net capacity must still be great enough to handle periods of peak loading where the entire capacity of the net is used. It will be assumed that the average use of the net is about 50% of capacity.

Unlike circuit switching, a message switching system has the ability to degrade, allowing more terminals to use the net that the actual rated capacity. When this happens, the terminals will simply run at slower than their maximum rated speed. This effect is not considered in the cost model, but is advantageous because it can smooth over peak loading problems.

The "rated capacity" of a message switching net will be defined here as the maximum number of terminals that can be used simultaneously at their full rated speed. It does not mean the number of terminals connected to the net, since there is no limit to this.

Costs of the Net

The costs of the model net are now shown. Capital costs for the net are estimated at $350,000, including an equipment cost of $280,000 and a development cost of $70,000. Operating expenses for a minimal net include line charges of $2011 monthly and $1000/month for hardware and software maintenance.

Equipment Costs

10 minicomputers with ASR 33 Teletypes and 8 K core	133,000
Synchronous line interface hardware	9,000
Asynchronous line interface hardware	14,400
Modems, 4800 bit per second synchronous	64,000
Power supplies and misc., hardware existing	9,000
Interface equipment to CPUs	50,000
	280,000

Development costs

The costs below include of hardware and software for the central switch, terminal interface hardware for the local nodes, and a basic software package for the local nodes that supports both interactive terminals and batch processing.

Personnel Requirements

Telecommunications Programmer	2 man-year	$22,000
Development Engineer	1 man-year	$10,000
Engineering aids	2 man-year	$12,000
Drafting	1 man-year	$ 6,000
		$50,000
Administrative overhead, 40%		20,000
		$70,000

Operating Expenses

Operating expenses include line charges and hardware and software maintenance. Line charges for a minimal single-star net are $2011/month, and maintenance is estimated at $1,000/month.

Expanding the Net

The cost estimates presented earlier are based upon a minimal net, with a single voice grade line connecting each campus with the central switch. As the traffic on the net increases, more lines can be added. Estimates of the useful capacity of the net for various line configurations are shown in the table below.

It is assumed that the same number of lines will be run between each campus and the central switch, although this will probably not be the case in a practical system due to the varying traffic requirements of each campus. The cost of each additional line is the increase in line charges, plus the cost of the line interfaces and modems at each end.

ESTIMATED USABLE NETWORK CAPACITY

No. of Lines	For Interactive Use			For Batch Use
	bits/sec	no. of TTYs	no. of hs CRTs	bits/sec
1	6,480	360	64	8,640
2	12,940	720	128	17,280
4	25,920	1440	256	34,560
8	51,840	2880	512	69,120

(for a single star net with 4800 bit per second modems).

Cost-Performance Curves

Using the model that has been described, the cost of the network to the user was calculated. The curves presented include all costs of the network, including equipment costs, development costs, line charges, and operating expenses. Capital costs are amortized over 60 months.

The model initially has a single line connecting each campus. As the load on the net becomes heavier, more lines are added to increase the capacity of the net.

Figure 1 (on the next page) shows a curve of per-terminal-hour costs for an interactive-only network. Costs are shown for terminals that operate at four different speeds. The cost per terminal-hour for a circuit switching system is also shown. This cost is similar to but no the same as the IDSS system. The details of how this was calculated is covered in the next section.

Figure 2 shows a similar curve for remost batch processing expressed in cost-per-million-bits-of-data. The cost of communicating with circuit switching is also shown, for transmission rates of 2400 and 4800 bits per second.

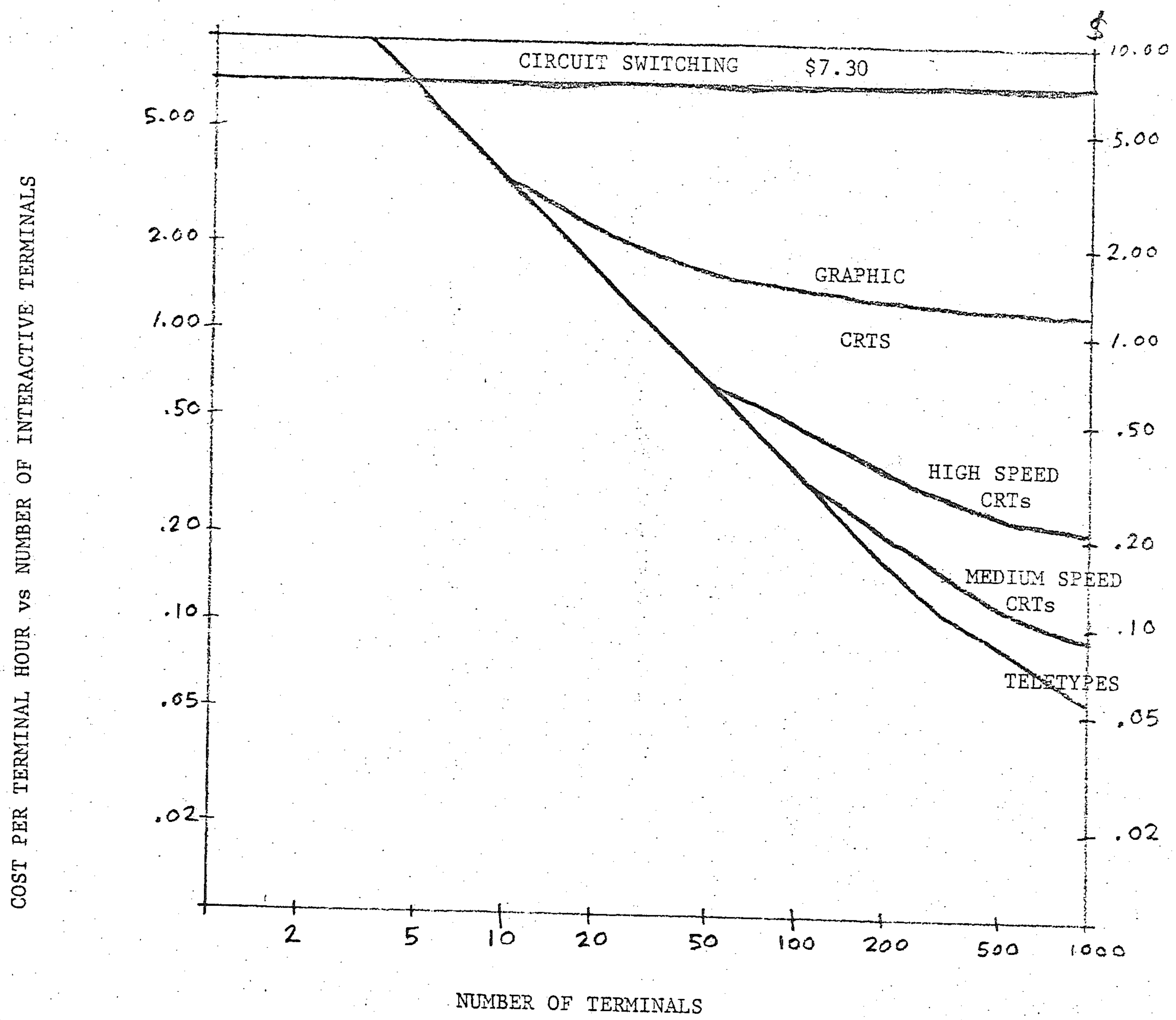
Cost of Circuit Switching

In calculating the cost of circuit switching, a configuration similar to the IDSS system was used. The location of the central switch is Los Angeles, however, for comparison with the case of the message switching net. Cost data for the circuit switch itself is not available, and so the costs shown are line charges and do not include the switch.

As in the case of message switching, not all the lines in the system are usable under less that optimum conditions. This "line fragmentation" results from an even distribution of load, and is especially serious with circuit switching because of the compatibility problem. We will assume that 50% of the lines cannot be used at any given time due to this problem. Cost figures for circuit switching are, like the ones for message-switching, based upon use 60 hours per week. Also, it is assumed that average use is only 50% of capacity. The cost-per-hour of a terminal connection between two campuses, then, is about $7.30 per hour.

FIGURE 1

COST PER TERMINAL HOUR vs NUMBER OF INTERACTIVE TERMINALS



"Graphic CRTs" are estimated to have an average data rate of 600 bits per second. (peak rate is usually 2400 b/s)
"High-Speed CRTs" (alphanumeric) have an average data rate of 100 bits per second. (peak is 1200 to 2400 b/s)
"Medium-Speed CRTs" have an average data rate of 35 bits per second. (peak is 300 bits per second)
"Teletypes" and similar terminals have an average rate of 18 bits per second. (peak is 110 to 150 bits per second)

COST PER MILLION BITS

vs

AMOUNT OF DATA TRANSFERRED

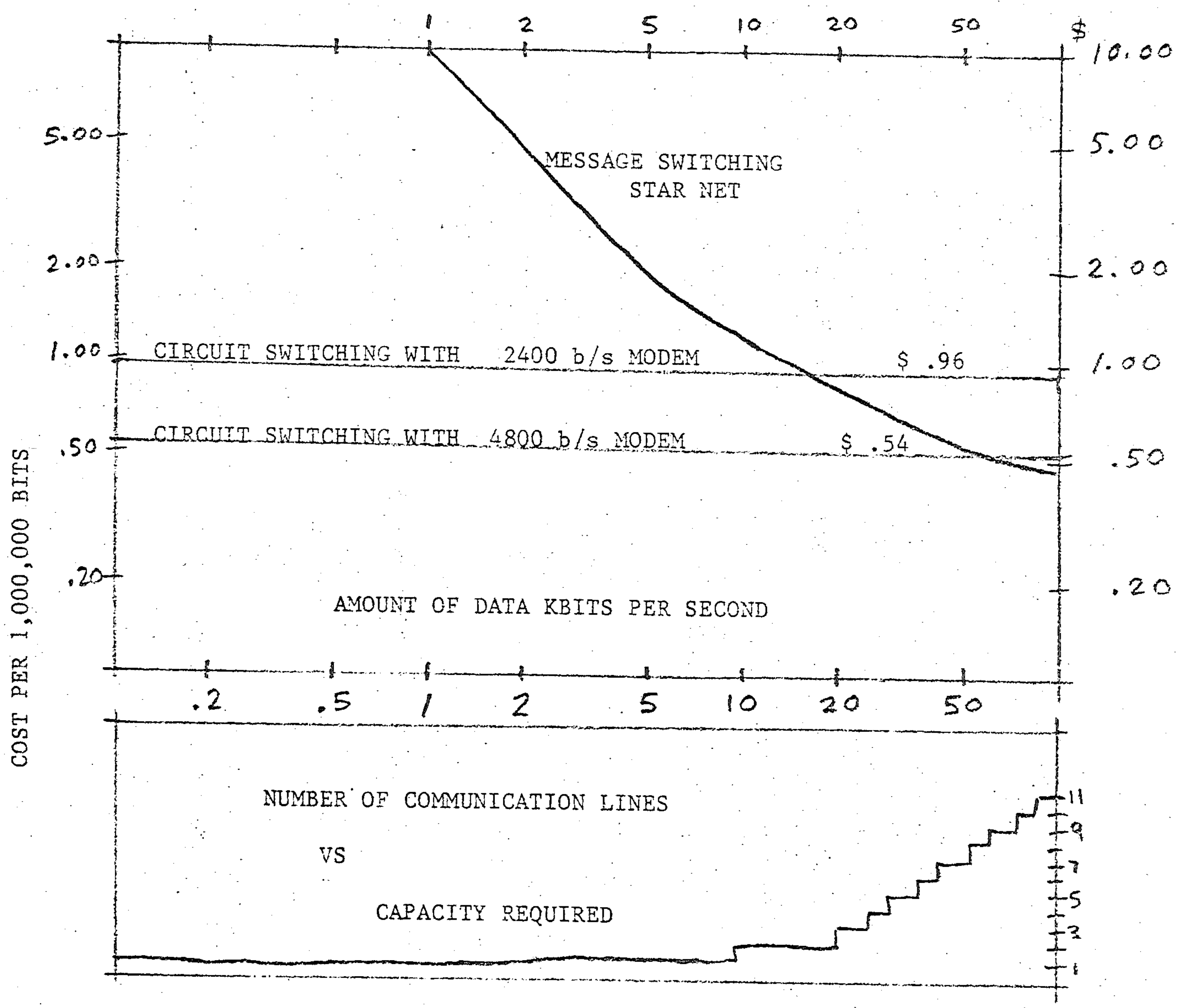


FIGURE 2

In the case of remote batch processing, we wish to figure the cost per million bits. Since the cost depends upon the speed at which data is sent over the line, we will figure two cases, at 2400 and 4800 bits per second. The cost of modems is estimated at $2000 for 2400 b/s and 4000 for 4800 b/s, and is amortized over 60 months in the calculation. The typical remote batch terminal is assumed to have an average data rate of about 50% of its peak rate. Then, the costs are $.96 per megabit at 2400 b/s and $.54 per megabit at 4800 bits per second.

These costs are biased in the direction to make circuit switching less expensive than it really is, since the cost of the switch is not included, and most remote batch use probably has an average data rate of less than 50% of peak. Also, circuit switching man be more expensive if the remote batch terminal is used less than 30 hours per week.

6. DEVELOPING A PROTOTYPE NETWORK

The details of the implementation of a prototype network will now be presented. This network is designed to serve the five southern campuses of UC--Santa Barbara, Los Angeles, Riverside, Irvine, and San Diego, with a central switching node at Irvine.

The initial network--referred to as "Version 1"-- is intended for interactive terminal use only. Remote batch processing can be done with the IDSS circuit switching system.

Version 1 is designed to be as simple as possible, to speed development, while still retaining the major advantages of message switching. No hardware or software modifications are required to an existing CPU in order to allow terminals to log on from the network. Terminals can be directly connected to the local node computer, or Datasets can be connected, allowing terminals to dial up via the campus telephone exchange.

Terminals supported with Version 1 will be model 33 and 37 Teletypes, IBM 2741s, and Teletype-compatible CRTs that operate at speeds up to 2400 bits per second, such as the CTC Datapoint 3300. This is a tentative list, and a survey of the different campuses can better tell which terminals are most necessary.

Interfacing to the CPUs with the local node computers will be done by the simple expedient of connecting to a port as if the node computer were an ordinary hardwired terminal.

Figure 3 shows a diagram of the prototype net, while Figures 4 and 5 show the hardware configurations for a local node computer and a central switching computer, respectively.

Estimated Hardware Costs for a Prototype Network

The hardware costs here are for a central switching node plus a node computer for each of the five southern campuses. Each campus node has one synchronous communications channel (for communication with the central node), and 8 asynchronous communications channels, which can connect to terminals, Datasets, and CPU ports. More lines can be added if desired in groups of 8. Modems are included that run at 4800 bits per second, synchronously.

Cost of a local node for each campus:

1 Minicomputer, with 4K of core and ASR 33 Teletype	$ 10,800
1 Synchronous communications channel	500
1 Synchronous modem, 4800 bits per second	4,000
8 Asynchronous lines and scanner	1,600
Power supplies, miscellaneous hardware, connectors, rack panel	1,100
	18,000

Cost of a central switching node:

1 Minicomputer, with 4K of core and ASR 33 Teletype	$ 10,800
1 4K core memory module	2,500
5 Synchronous communications channels	2,500
3 Synchronous modems, 4800 bits per second	12,000
Power supplies, hardware, connectors, rack panel	1,200
	29,000

5 Local Nodes @ $18,000	$ 90,000
1 Central Node @ $29,000	29,000
Total hardware costs	$119,000

Development Costs for Hardware and Software

Personnel Requirements

Telecommunications Programmer	1 man-year	$11,000
Development Engineer	1/2 man-year	5,000
Engineering aids	1 man-year	6,000
Drafting	1/2 man-year	3,000
		$25,000
Administrative overhead (40%)		$10,000
		$35,000

A Development Plan

A plan will now be proposed for the development of the 5-campus prototype network.

Phase 1: General planning for the network is done, and specifications decided upon. Message formats are specified and some of the software is specified for the local node computers. A minicomputer is selected, the design of the asynchronous and synchronous communications equipment is done, and prototypes are constructed and tested.

Phase 2: During this period initial software is written for Version 1. Software for the central switching node is completed, and a first version of software for the local node computers is written, allowing the net to operate with one type of terminal, a model 33 teletype. Meanwhile, copies of the asynchronous and synchronous communications equipment are being constructed, and a central switching CPU and several local node CPUs are configured. Initially, these nodes are connected together in the same room, with some terminals, and the basic message switching software is tested out. Ports from nearby computers (such as Irvine's Sigma 7 or PDP-10) are connected to the net, and terminal users on the net can log onto these machines. At first, only the simple message formatting and switching software is written, later error detections and retransmission software is put into the system.

Phase 3: After the network has been tested out locally on one campus, the local nodes are moved to their respective sites, and the communications lines are connected, together with the modems. When this is done, the network should work well, with the only restriction being the use of model 33 teletypes only. The last part of this phase involves the completion of software to allow other terminals access to the net. Version 1 is now fully implemented, and terminals on any campus can now log onto any computer on any of the five other campuses.

Figure 3. A PROTOTYPE NETWORK

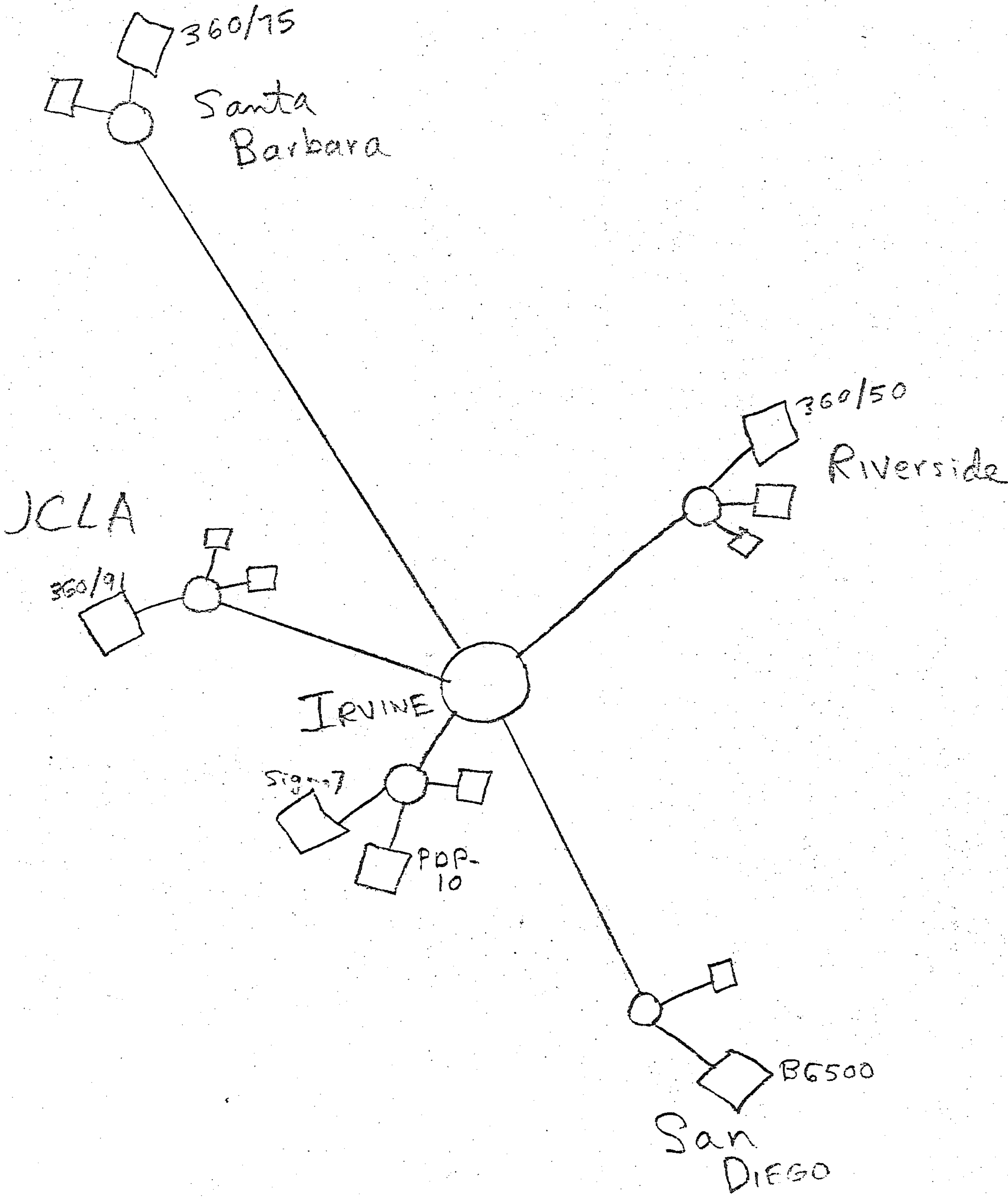


FIGURE 4

HARDWARE CONFIGURATION

LOCAL NODE COMPUTER

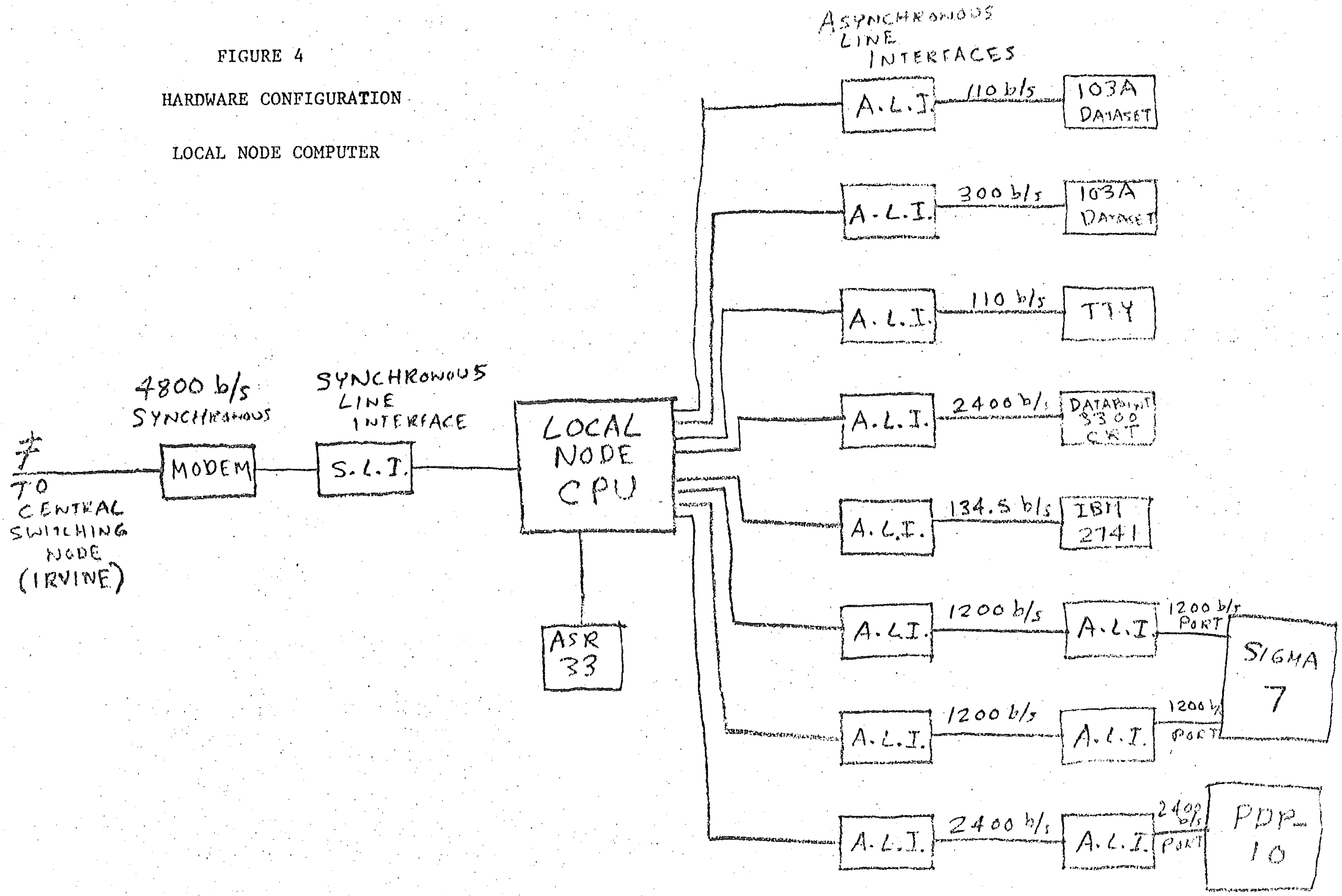
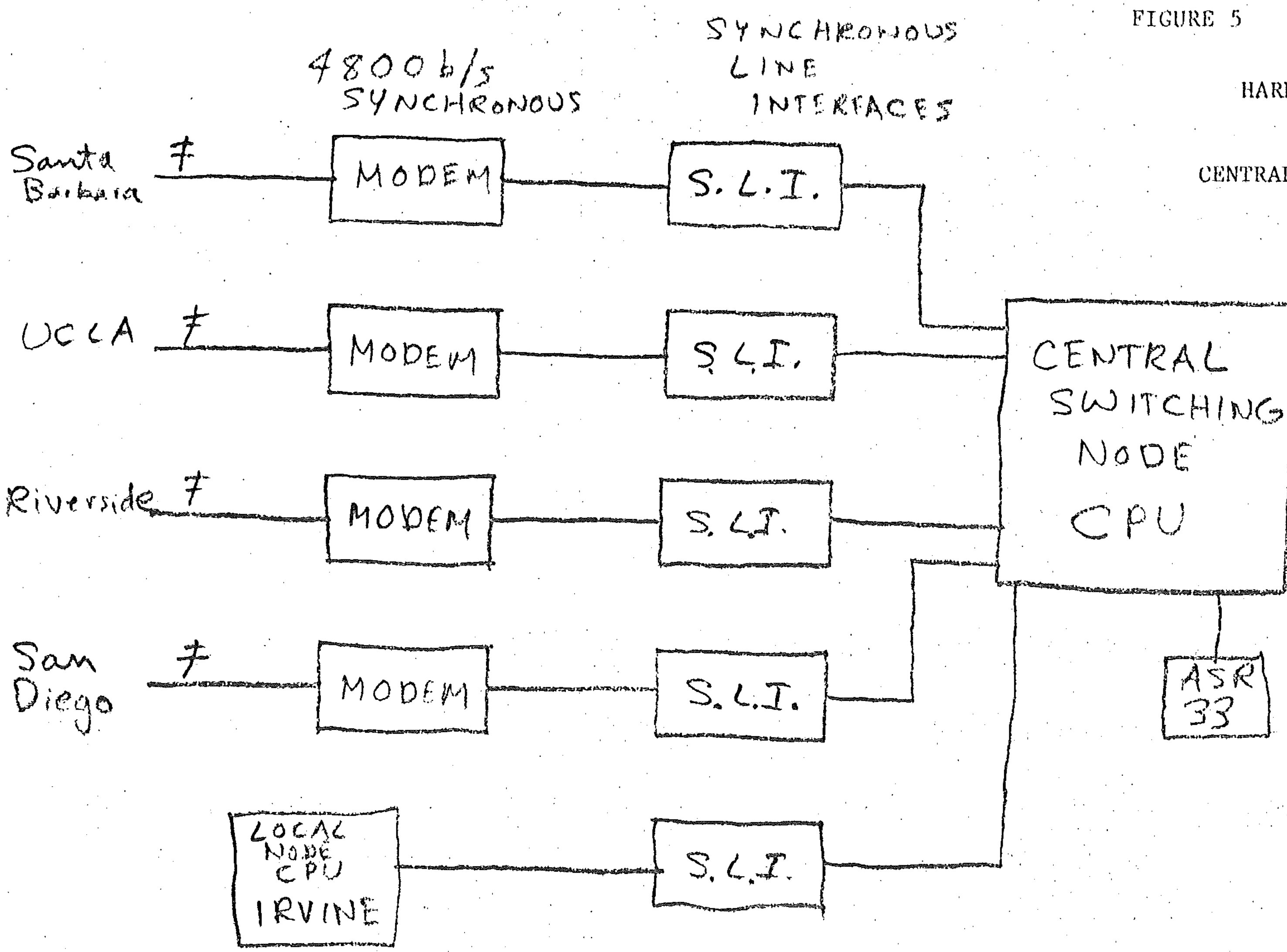


FIGURE 5

HARDWARE CONFIGURATION

CENTRAL SWITCHING COMPUTER



7. A CONCLUSION

This study has attempted to present the basic ways in which inter-campus data communication can be done, and examine the ways in which some ways are better than others. Some of the more promising methods have been explored in greater detail than the rest, to the point of presenting a development plan for implementation.

This section, then, is an attempt to answer the question, "what should be done?".

The main questions confronting us are:

1. Should the data network be a circuit switching or a message switching system?

2. If it is message switching, should the net be a star or a distributed net?

These questions must be answered for both interactive and remote batch processing. They also involve time--what is best now (or what can we afford) and what should be done in the long run.

It is clear that message switching offers the only way to provide interactive terminal access inexpensively enough to encourage massive interactive resource-sharing among the campuses.

The question is whether circuit switching or message switching should be used for remote batch processing. This decision is influenced by the advantages of having a single system for both interactive and batch use. (This would allow capital costs and the fixed line costs to be shared among more users, lowering the cost to each user).

Rather than attempt to analyze the subject further, a brief recommendation will follow:

A message switching network should be developed for both interactive and batch processing. The type of network initially used should be a star net, with a message switching CPU on each of the nine campuses.

A star net is recommended for initial development because it is inexpensive and because the capacity and reliability will be adequate for the next few years.

It is possible that the eventual network to be used will be a distributed net. This will depend upon the traffic patterns that develop, the capacity required in the future, whether or not the net is expanded to include the state colleges and other users, and the results of the star net's performance, especially with regard to reliability.

If a star net were converted to a distributed net, the local node computer hardware and much of the software could be utilized as terminal control CPUs.

To prepare for this eventuality, interfaces between large CPUs and local node computers should use an interface specification compatible with the IMP interface.[8]

REFERENCES

1. P E Jackson C D Stubbs
"A study of multi-access computer communications"
Proceedings of the Spring Joint Comp. Conf. 1969 p. 491

2. Mfg. by American Data Systems, Canoga Park, California

3. Mfg. by Imlac Corp. Waltham, Massachusetts

4. Mfg. by Computer Terminal Corp. San Antonio, Texas

5. Electronics, August 3, 1970, 116-117

6. L G Roberts B D Wessler
"Computer network development to achieve resource sharing"
Proc. of the SJCC 1970

7. F E Heart et al
"The interface message processor for the ARPA computer network"
Porc of the 1970 SJCC

8. "Specifications for the interconnection of a Host and an IMP"
Bolt Beranek and Newman report No. 1822 1969