SLAC-PUB-3767
September 1985

A TALE OF TWO NETWORKS

R. L. A. Cottrell

Submitted to *Data Communications*

Stanford Linear Accelerator Center, Stanford University, Stanford, CA 94309

Work supported by Department of Energy contract DE–AC03–76SF00515.

At the Stanford Linear Accelerator Center (SLAC), we've experienced dramatic growth in interactive computing over the past decade. From supporting a few small clusters of terminals shared amongst our users, we've advanced to providing hundreds of users with terminals at their workplace. And we've also expanded the size and capabilities of the terminal pools we provide for intermittent users.

With more than a ten-fold growth in the number of terminals located at SLAC, our local data communications requirements have grown exponentially in the last decade. Our early clusters communicated over dedicated links to their hosts: basically, it took only line drivers and cabling to provide the link. As we added terminals, things got more complicated. Instead of a few pool locations, we began supporting many terminals scattered about the SLAC facility; at the same time, our computing capabilities grew, and many users found need to access several computers. Simple cables and line drivers no longer did the job: we saw a very real need for a network to support our local data communications.

In 1981, we took a major step forward by installing a data PABX as the hub of a terminal access network. This system controls and coordinates interactive access to our various computers; due to its modular design, we've been able to expand our network as new users and computers come on line. Two years later, we took another major step by installing an Ethernet and cross-connecting it into the data PABX network. In addition, using a PDP-11/60, we have developed sophisticated statistics gathering, monitoring and multiplexed control (i.e., shared access to a virtual operator's console for multiple authorized network management personnel) capabilities for the terminal network.

Today, about 900 terminals and personal computers, two dozen Digital Equipment Corporation (DEC) VAX research computers, and an IBM 3081 central facility are interconnected via the networks at SLAC. Figure 1 shows a schematic of the current SLAC network. Each month the network handles roughly 60,000 terminal sessions totaling 40,000 connect hours. Researchers from the U.S. and other nations, including Switzerland, Germany, England, and Japan, can connect their terminals to the SLAC network via the dial telephone network, over the Tymnet X.25 packet switched network, or through multiplexors and leased lines. Conversely, terminal users at SLAC use the same channels -- dial modems, packet switching, multiplexed private circuits -- to access various remote computing resources, both in the U. S. and abroad.

THE PROBLEM

Just as successful businesses tend to expand, successful research almost inevitably results in pressure for further research - the deeper we delve into the universe, the more we find there is to learn. When the Voyagers flew by Saturn, rather than provide an explanation of the rings, they showed us that these fascinating structures are more complex than anyone had ever imagined.

Similarly, in our own subatomic world of high energy physics at SLAC, each breakthrough tends to bring with it a multitude of new questions. And just as new questions about extra-terrestrial conditions require more sophisticated space probes to find answers, new questions about subatomic particles necessitate bigger, more complex experiments to gather the data needed to bring us closer to a better understanding of the basic building blocks of the universe. Because of this, the pressure for new research tools at a national lab like SLAC is enormous, and that includes pressure to add or expand computer systems and networks.

THE FACILITY

SLAC, shown in Figure 2 and located on 480 acres of the Stanford University campus, south of San Francisco, is operated by Stanford University for the U.S. Department of Energy. It is devoted to experimental and theoretical research in elementary particle physics, and to the development of new techniques in high-energy accelerators and elementary particle detectors.

The two-mile long linear electron accelerator is the major experimental facility at the center. It generates the highest energy electron beams available in the world. It was experiments with this machine, which may be thought of as a very powerful microscope, which showed us that the heart of the atomic nucleus, the proton, is itself composed of still smaller particles.

Following this discovery, an electron beam storage ring was constructed in which particles from the linear accelerator were made to collide. These collisions resulted in the release of energy from which new particles emerged. The discovery of the psi particle led to the award of a Nobel Prize to SLAC's Burton Richter and after that to the construction of an even larger colliding-beam storage ring, which provides collision energies four times those attained with the original ring. Today, it has become obvious that the future of this field of basic scientific research depends on machines that will produce colliding beams of even higher energy and SLAC is currently constructing a $114 million linear collider to enable it to stay in the forefront of high-energy physics research.

For all of this concentration on physical research, as recently as 1974, our data communications capability was confined to 60 IBM Selectric typewriter terminals and a few CRTs in pool areas for the physicists. Until 1979, terminals still were located only in pools. But as SLAC's terminal use expanded, this arrangement increasingly hindered people's ability to work efficiently.

Since data acquisition and analysis are performed by different computers, many terminal users needed access to more than one computer. Initially, this was solved by the simple expedient of placing a terminal for each computer in terminal pool areas. This required extra cabling, terminals, and ports. Eventually, we found ourselves running out of computer ports at the same time that the demand for additional terminals was outstripping our ability to supply them.

THE DATA PABX SOLUTION

We began to bring the situation under control in 1981 when we installed a Micom Micro600 Data PABX in the computer facility. The data PABX, shown in Figure 3, acts as the hub of the terminal network. All cabling extends from this unit to terminals, computers, and other services so we have a central point from which we can provide control and monitoring of network operation.

With the data PABX in place, we have, in essence, a private telephone network for data. Instead of a telephone set at each extension, we have either a terminal or a computer port. The data PABX makes connections between terminals and ports in response to requests entered at the keyboard. Once a connection is established, the data PABX passes data transparently until it recognizes a terminal's request to disconnect.

The data PABX provides features similar to those found in today's newer voice exchanges, but adapted to the data-only environment. A basic example of this adaptation is how the user specifies the destination when asking for a connection: where the voice user can most easily enter a series of digits (because telephones have numeric keypads or dials), the terminal user can easily request a resource by "class" name, such as SYSTEMA. We get the effect of a telephone rotary - where one number selects the first available extension among several - because a class can designate any number of ports. Our data PABX also provides features akin to "camp-on-busy" and "call forwarding." The terminal speed need not be known in advance -- the data PABX's autobaud feature provides automatic terminal speed recognition up to 9600bps.

Terminals are connected to the data PABX in a variety of ways. We find that we can directly cable terminals at distances of up to 700 feet at SLAC, although we realize the EIA RS232C standard only specifies signal integrity for 50 feet. For intermediate distances of up to one or two miles, terminals are connected by line drivers or local multiplexors. For longer distances, terminals are connected via modems communicating over dial-up, or multiplexed leased, telephone circuits. Figure 4 shows some of these lines and their modems and multiplexors in our computer center. Each incoming interface is monitored by the data PABX just as a local interface would be, and its access to ports can be restricted for security purposes by entering instructions through the command port.

For managing the network, the Micro600 has a command port. Using a terminal connected to the command port we can redefine the class names that refer to ports, enter new security restrictions, take entire classes of ports out of service for preventive maintenance, etc. In short, the command port lets us react to changes in our environment.

For an on-site user, our data PABX makes accessing the network simple - one simply depresses the RETURN key on the terminal. The data PABX responds with a message requesting the class of service the user desires and the user replies by typing in the class name. (The class designations can specify a certain computer, specific ports on a particular computer, or other services that the network offers.) Once the user enters the class designation, the data PABX makes the connection and displays a GO message on the terminal. If the connection cannot be completed, the data PABX displays a message on the user's terminal that indicates the reason. If all of the ports in the requested class are in use, the data PABX responds with a BUSY message and tells the user how many others are waiting for connection to the class. Then, the user can opt to get into the queue and be connected automatically when a port becomes available or to disconnect and try again later. This puts contention for available services on an orderly basis, and the interactive communication with the data PABX makes the system "friendly" to work with.

We try to keep the number of times users get a BUSY message to less than 5% of the total number of successful attempts. Two-thirds of the times when users get a BUSY message, they choose not to wait and either give up or request another class of service.

Figure 5 shows examples of the terminal dialog for a few common types of sessions. Since terminals are not hard-wired to computer ports, a single terminal has access to all service classes. Experience with about 660 different terminal lines accessing the switch per month, shows that each one selects, on average, 5 (median 3, maximum 32) different service classes. Service classes may be allocated resources based on level of usage provided economically and infrequently used services can be shared by many users (e.g. we have 4 dial-out modems which are accessed by about 60 different terminal lines per month).

Also, fewer ports than terminals are required. Our data PABX has more than 750 terminals and personal computers connected, and 425 computer ports. The contention ratio (terminals/ports) would be much higher if we did not provide access to over 45 different service classes. For example, one of the major service classes has 85 computer ports which are accessed by over 480 different terminal lines each month. The ability for a single terminal to select multiple service classes, and the contention feature allowed us to make major cost savings while providing a maximum level of service.

SPECIAL FEATURES

With the data PABX, we've found it easy to provide special services for our users. Users access these services by class name, just as they would request access to a host. Some examples include:

Loop-Back: One port on the data PABX is wired to loop back to the sending terminal. Thus, if users experience difficulty, they can quickly check their terminals by connecting to this port to see their input as (or if) it comes back. In order to prevent users inadvertently getting stuck on this service, this port is configured with a no-activity timeout of 2 minutes. Typically this port gets used by about 15 different terminal lines/month.

Tracing the Terminal Address: When users request the TRACE class their terminal is connected to a port on the PDP-11/60. This port has been set up to interrogate the data PABX command port and tell users the address of the line they are using. Thus, if they have problems, they can supply information to aid in troubleshooting.

Help: When users request the HELP class, they are provided with a listing of the classes of service available, important phone numbers, and instructions for using the network. They can also request the STATUS class which provides the current (updated automatically every 5 minutes) status of the main computer services. The HELP and STATUS functions are implemented on the PDP-11/60.

Special Connections

The Lawrence Berkeley Laboratory (LBL), another physics research facility, does a great deal of work using SLAC facilities. Since LBL and SLAC are in line of sight, the connection is made by a microwave link. Since LBL's network is built around a data switch too, we are able to make switch-to-switch connections that provide virtually universal access within the two locations. Similarly, Fermi National Accelerator Laboratory (FNAL) a physics research facility outside Chicago, has a Micro600 switch which is connected via a 9600 bps multiplexed satellite link to SLAC's data PABX. An additional eight sites are connected to SLAC by multiplexed leased land lines or satellite links.

SLAC also has a special purpose broadband network, to control the accelerator. So that terminal users on either network may access services on the other network, we have successfully cross-connected the two networks.

In addition to the asynchronous terminal network that this article explores, SLAC also has a terminal network of some 200 Lee Data Corporation IBM 3278 look-alike terminals, connected via co-ax cable links with controllers interfaced to the IBM mainframe. Some of these terminals also have asynchronous capabilities, so we installed a link between their controller and the data PABX.

A NEW DIRECTIUN

Just as our researcher-users continue to develop more powerful tools to study high energy physics, we continue to refine and improve our network to better support its users. We strive to keep ourselves abreast of the state-of-the-art, and, at the recommendation of a study group, in spring 1983, SLAC installed a small Ethernet terminal network. Its aims were:

- .to gain experience with the new technology;
- .to learn how to connect Ethernet terminal servers to the data PABX;
- .to compare the Ethernet approach with that of the data PABX when both are used for terminal access;
- .to find a smooth migration path from data PABX to Ethernet technology; and,
- .to make recommendations on how we can best serve terminal users in the future.

Initially, we installed about 1,300 meters of cable, a pair of Bridge Communications CS/1 terminal servers (both configured for eight RS232 connections), transceivers, etc. Our cost estimates at that time favored the data PABX solution. However, we felt the Ethernet approach was more open ended, and that distributing networking intelligence could provide a network more responsive to its users. We also expected rapid price erosion in the Ethernet market, reducing, or even reversing, the price advantage of data PABX networks.

Based upon these initial experiences and expectations, we decided to expand the Ethernet terminal network and to give it a "stress test." We upgraded our two terminal servers to their maximum complement of 32 ports each, and we added a third 32-port server. To make it easier for us to diagnose problems quickly, we initially put only experienced users on the net. At the same time we decided to expand the Ethernet installation, we decided to freeze any further acquisition of data PABX type equipment.

By the summer of 1984, our intial Ethernet experiment spawned two distinct networks. These networks comprise six Ethernet segments, roughly 2,200 meters of co-ax, and five repeaters. One Ethernet supports 14 VAX systems running DECnet, and the IBM mainframe running file transfer software written in-house. The other provides data PABX-like access, and includes:

.seven Bridge CS/1 terminal servers, each supporting an aggregate of 32 terminals and computer ports;

.two Bridge CS/100 terminal servers with 10 ports apiece;

.five additional VAX computers; and,

.the PDP-11/60 we use for terminal network management

Figure 6 shows a Bridge terminal server installation. We use DEC-supplied interfaces (DEUNAs) for the VAXs, an Interlan NI1010 for the PDP-11/60, and an IBM Device Attach Control Unit (DACU) with another Interlan NI1010 controller board to bring the IBM 3081 mainframe onto the Ethernet. Most of the transceivers are from DEC, with three being from Interlan.

There are 127 terminal devices and 72 computer ports on our terminal servers, leaving 45 terminal server ports open for expansion. The terminal servers support terminals, auto-answer modems, IBM PCs, as well as ports on VAXs, microVAXs, an IBM Series/1 acting as a channel-attached 3270 terminal emulator, and an IBM 3705 communications controller. Additionally, 32 terminal server ports attach to the data PABX: eight channels provide access from the data PABX to services on the Ethernet, and 24 channels provide access from the Ethernet to services available through the data PABX.

We've observed that our Ethernet terminal server network typically has a maximum of ~~34~~ 40 simultaneously active terminals. This total includes, on average, 16 terminals on the Ethernet accessing resources through the data PABX. Our data PABX, heart of a network comprising roughly 1,200 devices, typically has seven of its more than 220 (peak use) simultaneous connections going to devices through the Ethernet. These "crossovers" are due to resource availability: either the resource does not exist on the originating network, or it exists on both, but all available ports on the originating network are unavailable.

We found that the cross-over requirement was fairly easy to meet. Connecting the data PABX network to the Ethernet co-ax network was accomplished simply by running RS232 cables between data PABX interfaces and terminal server interfaces. However, this approach does require the user to deal with both networks when making a connection. For a data PABX attached terminal, the user requests access to the class "BRIDGE" and then the terminal behaves as if it was directly connected to the Ethernet terminal server; the user must then make a final connection request of the terminal server. Conversely, an Ethernet-attached terminal can ask to be connected to the data PABX by issuing the command "DO MICOM"; the user will then participate in the Micro600's dialog to complete the connection.

ETHERNET EXPERIENCE

Because this was our first experience with a new technology, we encountered abundant learning opportunities. For instance, we learned to mark off the co-ax cable at 2-1/2 meter intervals, so we could more easily comply with the Ethernet standard requirement for spacing transceivers. In addition to closely observing the standard transceiver spacing, we found it important to comply with the standard's maximum cable segment length of 500 meters. We even went so far as to check the lengths of already installed segments using a time domain reflectometer.

Transceivers, we learned, need secure mounting, and the cable between transceiver and the station it supports should be anchored firmly for strain relief. We replaced the Ethernet-specified slide connectors on the transceiver cables with more secure screw-down connectors. Attaching a transceiver to the cable requires drilling a hole in the co-ax to accomodate the "sting" that makes contact with the cable's center conductor; while this is not an especially tricky task, it does require due care, since a faulty installation can bring down the Ethernet segment or make the device attached to the transceiver inoperative. Also, we tried to standardize on equipment from as few vendors as possible. (Had Micom's early 1985 acquistion of Interlan occured two years prior, we might have had an essentially single-vendor solution.) Despite this, we ran into problems between Bridge terminal servers and DEC transceivers, resulting in increased packet alignment errors, and consequent multiple error retries.

Our Ethernet terminal servers required more attention than the more mature, less complicated multiplexors that we use with our data PABX. We've learned to make certain that servers are always accessible (e.g., if it's in a locked closet, the key must always be available), and that there is room to work around the units (e.g., don't install terminal servers on the top shelf of a utility closet). We learned these lessons as we found ourselves frequently changing floppy diskettes or updating circuit boards.

Diagnosing problems on an Ethernet can prove a challenge. In part, this is due to the newness of the technology and our lack of experience with it. The Bridge terminal servers do provide some statistics reports on traffic and errors, which were some help. Still, a major problem facing the network management team is the current lack of simple-to-use, inexpensive diagnostic devices akin to the line monitors and RS232 breakout boxes we use on the data PABX network. A corollary of this is that the intellectual effort to run an Ethernet is greater than that required to run a data PABX.

Nonetheless, most of our problems to date have turned out to be installation problems and faulty transceiver cables. Our transceivers have failed only when we have "played" with them; left undisturbed, they seem to work indefinitely. When we did have a transceiver failure, it's been fairly easy to find, since the station - the terminal server or computer using the transceiver - also died. Also, we find that the indicator lights on our repeaters can help us isolate a fault to an Ethernet segment; for instance, seeing the Collision Detect light shine indicates that there is trouble nearby.

Limitations of current terminal servers

We encountered a few problems with the terminal servers we installed at SLAC. Some were bugs: spontaneous disconnections in mid-session, servers losing some or all of their configuration data, interference between DECnet and Bridge communications over the same Ethernet cable, and occasional long delays - measured in seconds - between the time a key is hit and the time a character echo returned from the computer. We isolated this last problem to the Ethernet terminal server network, having ruled out the terminals and ports as the time-delaying culprit, and believe it is due mainly to retries associated with error recovery. We see other shortcomings as simply a lack of features (or perhaps, in some cases, design flaws). For instance, at the conclusion of a terminal session, the disconnection sometimes happened before all characters from the end of the dialog were displayed at the terminal. This bothered some users, as the information often contained accounting data regarding the user's session. Another problem occurs because our terminal servers don't check EIA control signals on a computer port before making a terminal connection. This means that the servers don't skip over ports with the Data Terminal Ready signal low when searching for an available port in a requested class. In turn, this leads to users being connected to nonworking ports, which both confuses and aggravates users.

Also, we find the automatic data rate detection provided by our terminal servers somewhat awkward. Instead of performing autobaud to 9600bps, as we'd like, the Bridge terminal servers provide autobaud _in two ranges_: below 2400bps and 2400bps to 9600bps. This means that we must fragment some services into two classes based upon speed. In fact, at the moment, it means it is not easy to connect one of the new 300/1200/2400bps auto-answer modems to the Bridge terminal servers.

Centralized network management also appeals to us. Because our terminal servers are scattered about a half-mile radius, it is tedious to visit each to install a new software release on floppy diskette, or to reboot the server, or to enter a new "welcome" message. We also want the time and date stamps in each server to remain synchronized. Ideally, we'd like to perform all of these functions from a central site.

Comparing network service

Using an IBM PC, we measured the elapsed time between entering a character at a keyboard and having it echoed by a host via the Ethernet; this takes roughly 1/12 second. On terminals that send repeating characters at a rate in excess of 12 per second users will see the cursor move in a jumpy fashion if the host (and not the terminal or its local server) echoes the character, since several characters will be buffered into a single packet. This can be slightly disturbing, and it makes positioning the cursor difficult. On the data PABX we see no problems of this type, even when the character repeat rate is 30 per second. To minimize jumpiness on the Ethernet, we configure the terminal server ports to tie off packets and send them with minimal delay. In turn, this short packet tie-off increases the Ethernet loading, as each packet usually contains only a single data character and several dozen bytes of packet overhead.

XON/XOFF flow control caused lengthy debates. In general, we set our terminal servers to pass XON/XOFF flow control characters transparently, the conventional way our data PABX network operates. Thus, it is up to the computer port or terminal to act on the XON or XOFF character. We find this the right way to handle most of our terminals and ports. Nonetheless, in the case of graphics terminals and other semi-intelligent devices, we find we need to have the terminal server act on XON/XOFF characters. Otherwise, the terminal may send an XOFF when its buffers fill to a threshold, but characters coming from buffers in the terminal servers may overflow the receiving device unless the terminal server knows to invoke flow control. Since the data PABX doesn't buffer data, we don't encounter this problem on that network.

We found need for nontransparent flow control on terminal server ports supporting IBM PCs doing file transfers to mainframe hosts. Avoiding speed-matching, we found that PC file transfers worked well through the Bridge network; we haven't needed to change any PC code or timeouts to handle this application on either the Ethernet or data PABX terminal networks. We did, however, observe that the actual throughput for a disk-to-disk file transfer through a 9600bps channel was about 7% to 10% less through the Ethernet terminal servers than through our data PABX. This likely is due to packet processing overhead in the terminal servers. (If Ethernet's speed-matching capability is used - say to support a dial-up PC at 1200bps transferring files between a computer port set for 9600bps -- the problem gets worse, and time-outs in the file transfer code probably need adjustment at the PC, the host, or both.)

Configured as they are, with short tie-offs to best support terminal access, the terminal servers send most packets with only a few data characters, and several dozen bytes of packet overhead. There is concern that there may be unwanted interference between the long packets (around 1,500 bytes) used by the file transfer applications and the short packets (averaging around 75 bytes) used by the terminal servers, resulting in unpredictable response times for interactive terminal users. In other words, Ethernets tailored for terminal access applictions may lose some of their effectiveness for file transfers.

The Ethernet terminal network performs "speed matching" (more accurately, "speed conversion"), which means that the user terminal speed needn't match that of the port being accessed. However, consider the case where the terminal runs much slower than the port, as for a 1200bps (or even 300bps) dial-in user accessing a 9600bps service. If the computer sends output at 9600bps, then, since the terminal can't accept it at that speed, the network buffers will quickly fill. Flow control still can manage this situation. However, if the user wants to interrupt the computer, then all the data in the network buffers pending delivery must be emitted to the terminal at its slower speed before the terminal can output the response to the user's interruption. In the case of a 1200bps to 9600bps mismatch, this is annoying and leads to poor responsiveness; for a 300bps to 9600bps mismatch, the service borders on unacceptable. The data PABX does not perform this type of speed matching, as it only makes connections between terminals and ports configured for the same speed. While this requires that the terminal speed be preset to that of the port, the improved response more than makes up for any inconvenience.

One of the flexible features of Bridge terminal servers is their ability to have ports convert XON/XOFF character flow control to RS232 Clear-to-Send (CTS) signal flow control, without passing the character through the port. Since our IBM 3705 communications controllers are half-duplex devices they cannot respond to character input while they are sending output characters to a terminal. We, therefore, modified the Commpro EP software in the IBM 3705 to respond to CTS flow control, so we now have flow controlled IBM 3705 ports. To do the same thing with our data PABX we had to build our own special XON/XOFF-to-CTS flow control converters, since the standard interfaces available passed the XON/XOFF characters through to the IBM 3705, confusing the front-end. Micom has since developed an interface that can convert XON/XOFF to CTS flow control, however it not only converts the XON/XOFF, it also passes them through which confuses the IBM 3705 control program.

While the data PABX network adheres to a one-user, one-session philosophy, we like the Ethernet terminal server capability of supporting multiple simultaneous sessions (one active, the rest suspended). This is particularly useful to network management personnel. However, the feature does have its attendent costs, since each session ties up a port on a terminal server and the computer port itself. This, in turn, reduces the effective contention ratio of terminals to ports, so extra ports may be needed. To reduce this negative impact, we have reduced from four to two the number of simultaneous sessions allowed for most terminal users, and have not heavily advertised this feature. Currently, the maximum number of terminals with more than one simultaneous session in progress typically peaks at three or four, and the maximum number of ports tied up by multiple-session users is around six, thus the impact on our contention ratio is relatively small.

Bridge provides a useful --and unique-- macro facility for its terminal servers. An authorized network manager can store a lengthy sequence of terminal server commands for subsequent invocation with a "DO" command. This facility lets us provide our users with a more friendly interface to the network. For instance, we use macros to initiate all connections. Such a macro may connect to a port within a named service class, set the speed and parity of the remote port, specify echoing of the local port, send an autobaud character to the host, etc. We also use this macro facility to provide a HELP command for our Ethernet users.

Controlling It All

Early on in our networking experience we recognized the importance of centralized control and monitoring facilities. Collecting and analyzing network statistics helps us assess the quality of our service. These analyses also help us spot trends and plan future network growth.

Two standard features of our data PABX address these requirements. The unit has a command port, to which a terminal can be attached to manage the network. At the command port, users can enter commands to redefine resource classes, to take classes out of service tor maintenance, to alter inactivity time-outs, to force connections between any two interfaces on the switch, and just about anything else a network manager might need to do.

Complementing the command port, a statistics log port on the data data PABX outputs an audit trail of all switch activity. The statistics log consists of time and date stamped records of every attempt to connect, whether or not successful, and every disconnection. For unsuccessful connect requests, the log shows the reason. The log also shows queuing activity.

To enhance our management capabilities, we've dedicated a PDP-11/60 computer to that task. We've connected both the command port and the statistics log output port to the PDP-11. We've programmed the PDP-11 to provide capabilities that the data PABX, by itself, does not offer.

Connecting the command port to the PDP-11 let us develop more convenient ways for our networking staff to deal with the data PABX. Where the command port supports but a single terminal, we can have the PDP-11 time-share the command port between several authorized users. Through a link to the IBM mainframe, these users also can be logged onto the VM operating system on the IBM host, and interactively issue commands to the data PABX from there. We also can store VM macro sequences of commands for reuse, and we can have the computer automatically issue commands to the data PABX for such predictable situations as taking a host off-line for scheduled preventive maintenance. The PDP-11 also can make adjustments (e.g., altering terminal inactivity time-outs) to our data PABX network based on performance changes noted from the statistics log output.

For management reporting, the PDP-11 monitors the statistics log port. ~~Every~~ each time a terminal accesses the data PABX it causes the PDP-11 to create a session record, complete with such information as start ~~and end~~ the time~~s~~, class, line id, port id, number of ports in the class currently in use, etc. When the terminal disconnects, the computer adds the disconnect time to the record a logs the~~se~~ record~~s~~ in a disk file which is passed on to an IBM mainframe on a daily basis.

For the Ethernet, we developed our own software to collect network activity data. We again used the PDP-11, bringing network management for both data PABX and Ethernets to a central point.

The PDP-11 connects to the Ethernet via an Interlan N~~M~~I1010 controller board; using Interlan RSX 11M XNS/ITP software, we have the PDP-11 communicate with the Bridge terminal servers. We interrogate all servers on our network every minute so we can log each server's status. We've programmed the PDP-11 to derive session records from these status snapshots. By watching for changes in status, our software deduces when sessions start and finish, etc. Our software then builds session records that contain the terminal server active port (the terminal end), the terminal server passive port (the computer port end), service class name, start and finish time of the session (resolved to our sampling interval), number of other sessions in progress for the terminal, number of ports in use and free for the resource class, and total number of Bridge sessions in progress. We save these records on disk, transmitting them on a daily basis to an IBM mainframe.

The IBM maintrame does the actual statistical analysis of our network session records - both those captured on the Ethernet and those coming from the data PABX. We use the SAS statistical package to help us identify usage patterns by service class, by terminal server, and by individual ports. Information available includes the maximum number of simultaneous sessions per day per service class, elapsed minutes and total number of sessions by class, how many different terminals use a given service class, and which ports and terminals are not used. With this information we can spot trends, decide where to add or remove capacity, and uncover strange behaviour (e.g., a particular port never has a long session) which may indicate problems.

For users in central areas of SLAC, we developed an additional, inexpensive aid for working with the network. The PDP-11 accumulates key information from the data PABX and Ethernet session records, on the tly, to arrive at totals for lines in use, ports in use, number of people waiting in queue, and so on. The totals are transmitted to an IBM PC where they are used to periodically

Recently we have acquired a Bridge network control and monitoring server which can aid in generating these session records update usage statistics and merged with other data, such as listings of units that are out of service. This information is passed to a Commodore VIC-20 computer which formats it for graphics and transmits it to other VIC-20s in more populated locations. The latter VIC-20s display the information on large, easy-to-read color monitors so users can check on the network at a glance. Figure 7 shows a typical monitor installation.

Looking Ahead

Contrary to our expectations, the cost differential between the data PABX and Ethernet network components increased significantly. This is due in part to new local multiplexing products introduced by Micom, which have actually considerably cut our cost to add a terminal. They have also allowed us to more efficiently use existing twisted pair wiring. For example the Micom 32 port time division multiplexers allow us to support up to 1,600 terminals on a 100 twisted pair trunk cable, where previously we could only support 33 to 50 depending on whether two to three pairs were used per terminal. Further, we did not see the price erosion we had expected in the Ethernet terminal server market.

Today it costs us about $565 to connect a terminal to the Micro600 network compared to about $965 to connect the Bridge network. These figures assume the terminal will locate beyond the RS232 distance from the computer port, and so in the data PABX case will need a line driver (often built into multiplexers). Also, for each terminal we add 50% of the cost of adding a computer port to the network so we can maintain a contention ratio of two terminals for each computer port. Our cost estimates also include miscellaneous cables, connectors, plus labor charges for installation; we don't include the cost of the computer port or the terminal.

Also, it is not economically attractive to install Bridge terminal servers with fewer than 10 ports. This means that for areas with only one or two terminals, one has to run long device cables from the terminal to the nearest Bridge terminal server. The Micro600 is more flexible in this respect, since one can get four port multiplexers, and one can easily and cheaply attach single terminals.

Based upon our experiences, and in light of some uncertainty about "standards" evolving in the Ethernet world, we decided to unfreeze data PABX expansion and at the same time to severely limit further expansion of the Ethernet terminal server network. We made this decision in September of 1984, and we immediately added 128 line/port interfaces to the data PABX to provide growth capacity for the next few months.

For the future, we will continue to track closely the developments in the industry. When and if the costs for Ethernet terminal servers more closely match those of data PABXs, and when and if Ethernet terminal services improve to more closely match data PABX functionality, then we will re-evaluate the direction in which our future network expansions should be made. We will continue to support both data PABX and Ethernet networks, since each provides useful functions not available on the other.

For the future, we expect to see the number of terminals at SLAC growing at 30% per year and the demands on the network will surely multiply. Even today there are about 200 visitors from about 40 nations stationed here, major experiments often are ongoing and can involve more than 100 physicists from all over the world plus additional technical people, so there is no alternative to data communications to coordinate their efforts. In the same vein, SLAC collaborates in experiments at other labs. In some cases SLAC scientists work at other facilities, such as the Deutsche Electron Synchrontron in Germany and CERN in Geneva, Switzerland, and then need a means of communicating with the resources here.

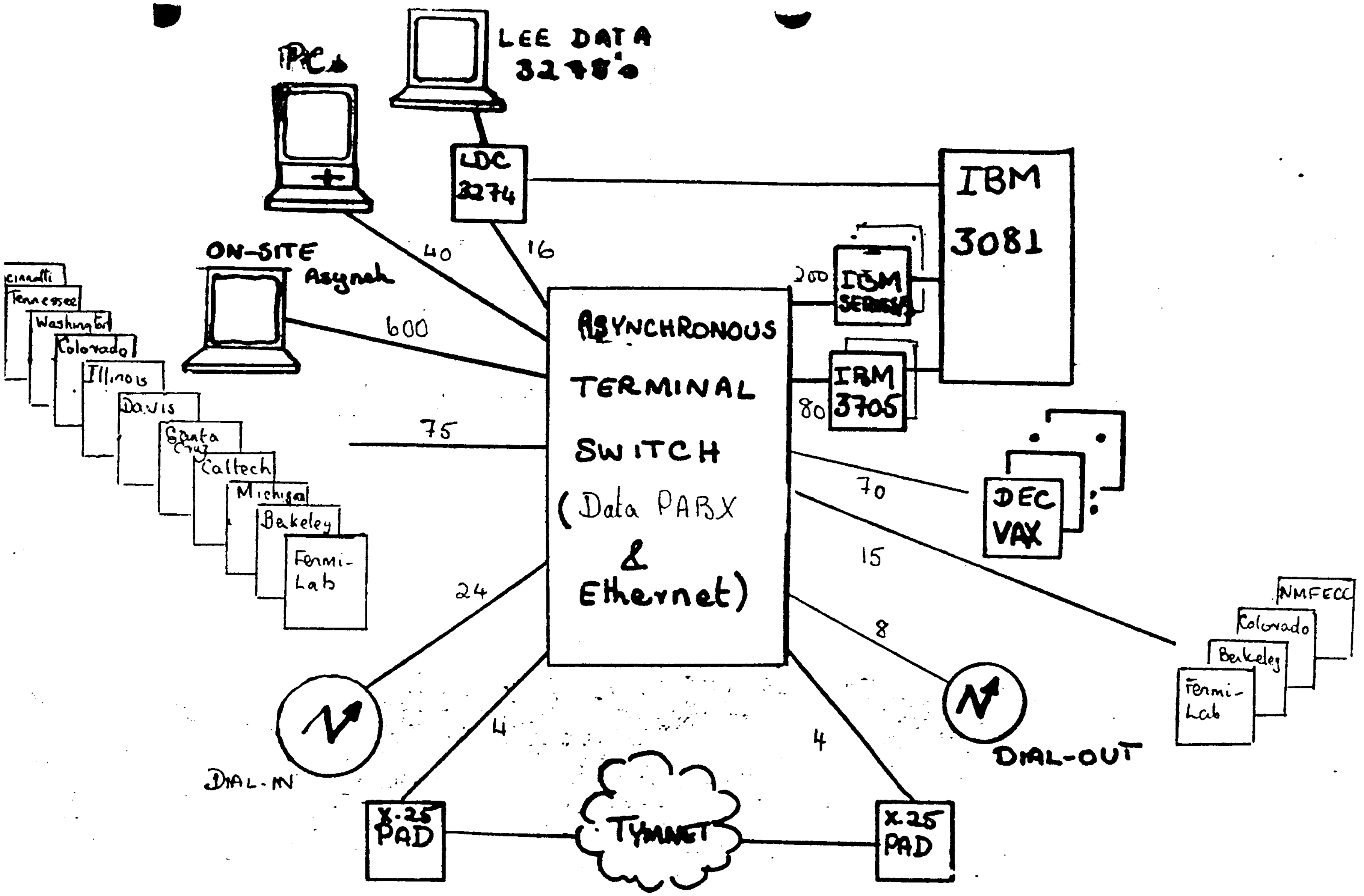

Figure 1: A schematic of the SLAC Terminal network.

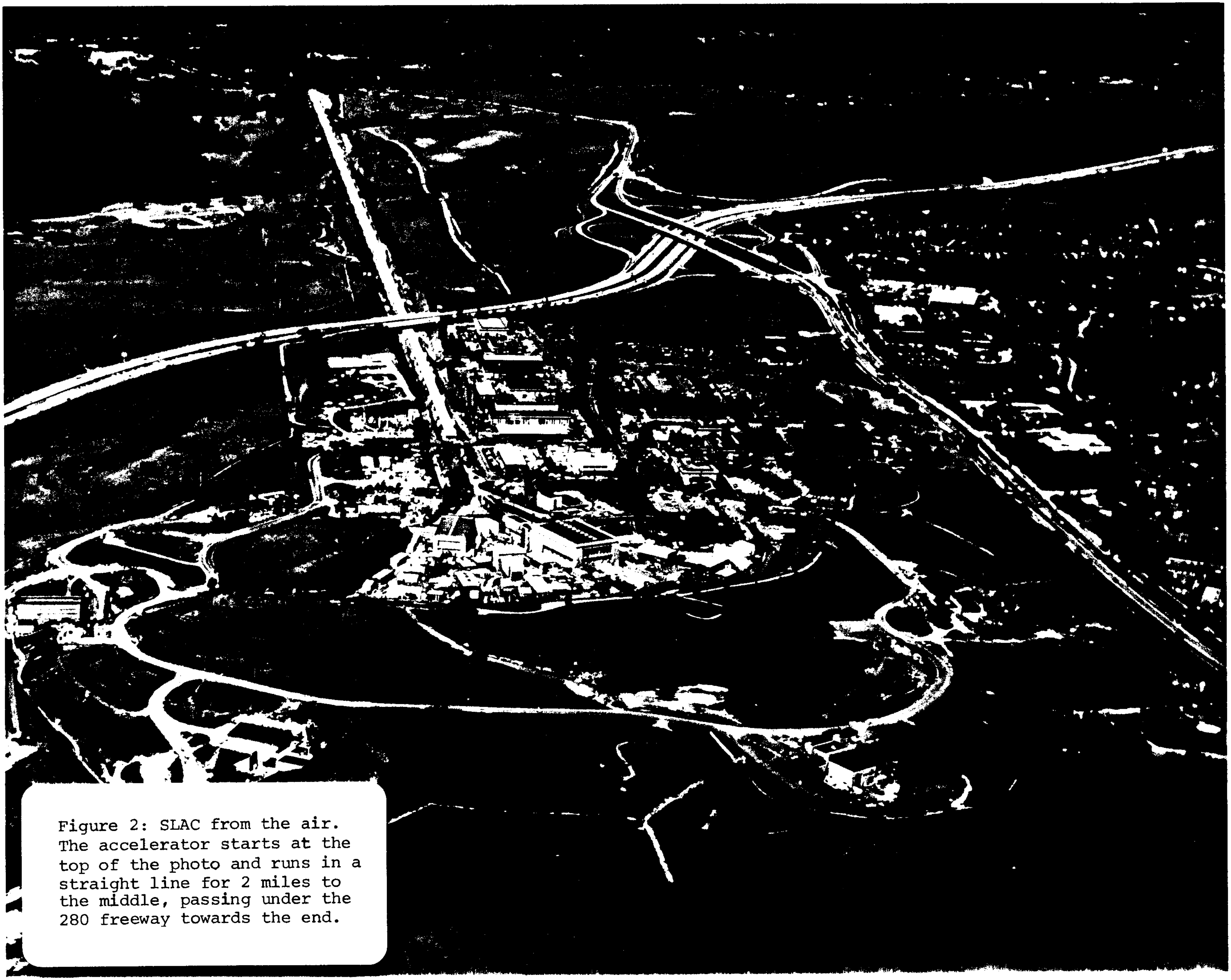

Figure 2: SLAC from the air. The accelerator starts at the top of the photo and runs in a straight line for 2 miles to the middle, passing under the 280 freeway towards the end.

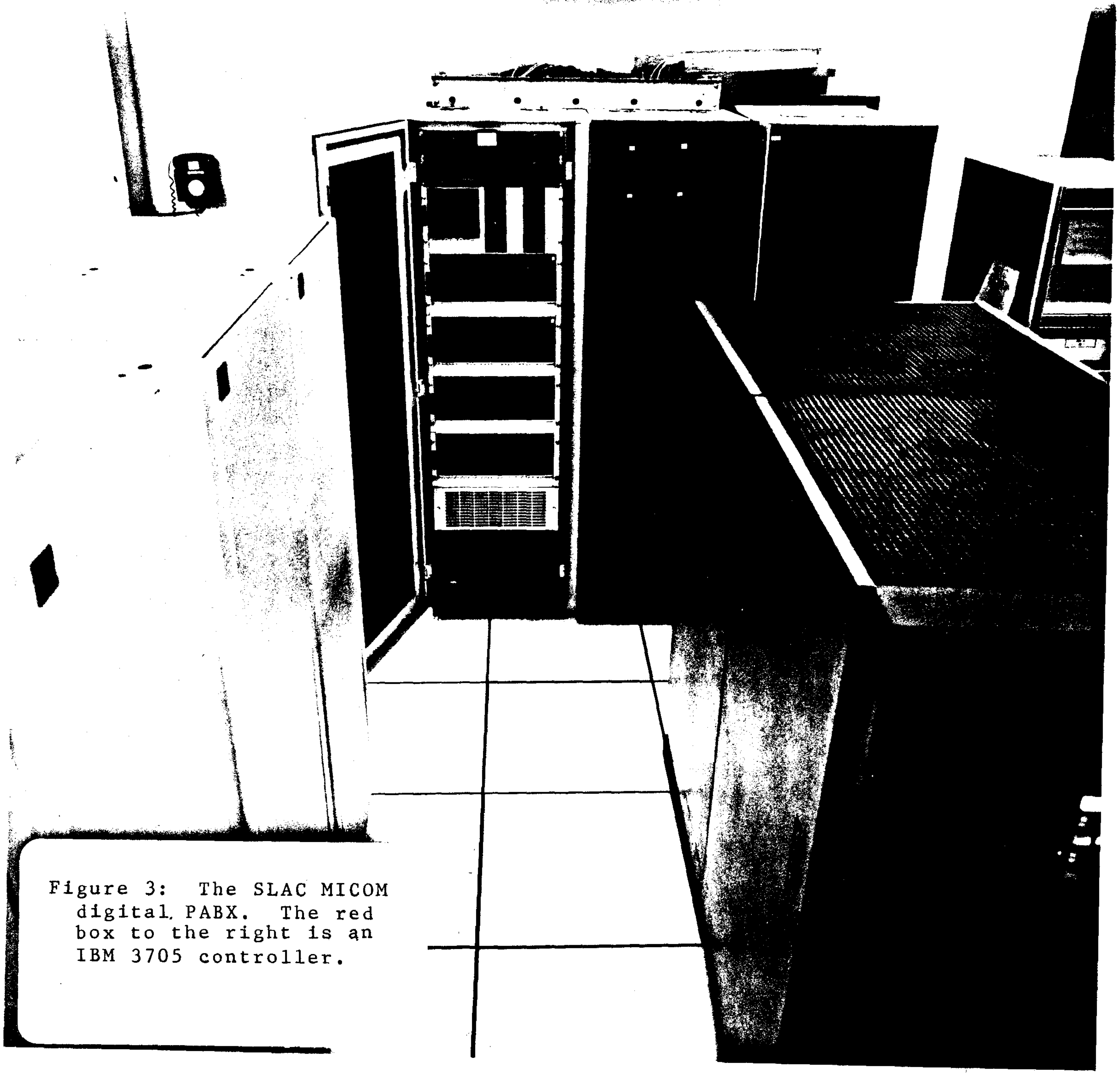

Figure 3: The SLAC MICOM digital PABX. The red box to the right is an IBM 3705 controller.

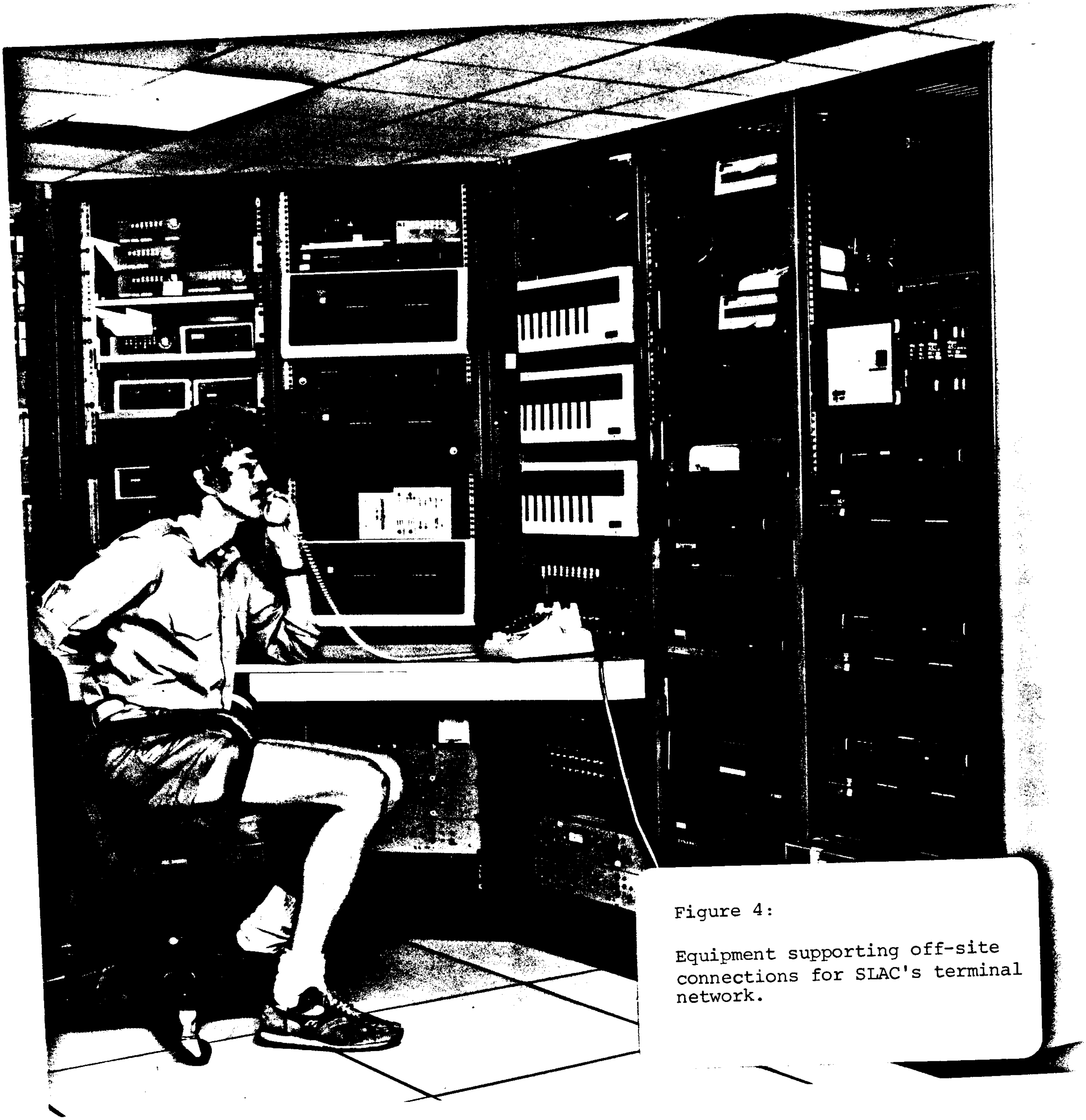

Figure 4:

Equipment supporting off-site connections for SLAC's terminal network.

```
SLAC MICOM SWITCH (Ver 2A-D)                                  Welcome message
Common Classes:  VMLINE, VM24, VM43, HELP, STATUS
Enter class:  VM24
*** Type <cr> after GO ***
GO                                                      successful connection
.
Enter class:  TRACE            The user wants to know the terminal's line number
Type:  HELLO TRACE/<cr>                             special class instructions
GO
> HELLO TRACE/
.
sta p256                              The PDP-11 asks the Micom about this line
P0256 C031 S9600 T00 I000 PT000 CONN 0091             The terminal is on line 91
>; Make a note of the of the number after CONN
>; on the lines above
SLAC MICOM SWITCH(Ver 2A-D)   The PDP-11 automatically returns to the Data PABX
Common Classes:  VMLINE, VM24, VM43, HELP, STATUS
Enter class:  DIAL                                 Attempt to use dial-out class
NOT AUTHORIZED                            Authorization is needed to use this class
Enter class:  VM43
BUSY-WAIT Y/C?003  C                  All vm43 class ports busy with 3 people waiting
Enter class:  JUNK                                            Try another class
NO SUCH CLASS
Enter class:  DISCONNECTED                   User did not respond within 12 secs.
```

Figure 5:
Examples of several kind of Data PABX connection attempts
`Text typed by the terminal is in this font`
RESPONSES TYPED BY THE USER ARE IN THIS FONT
Explanatory comments are in this font

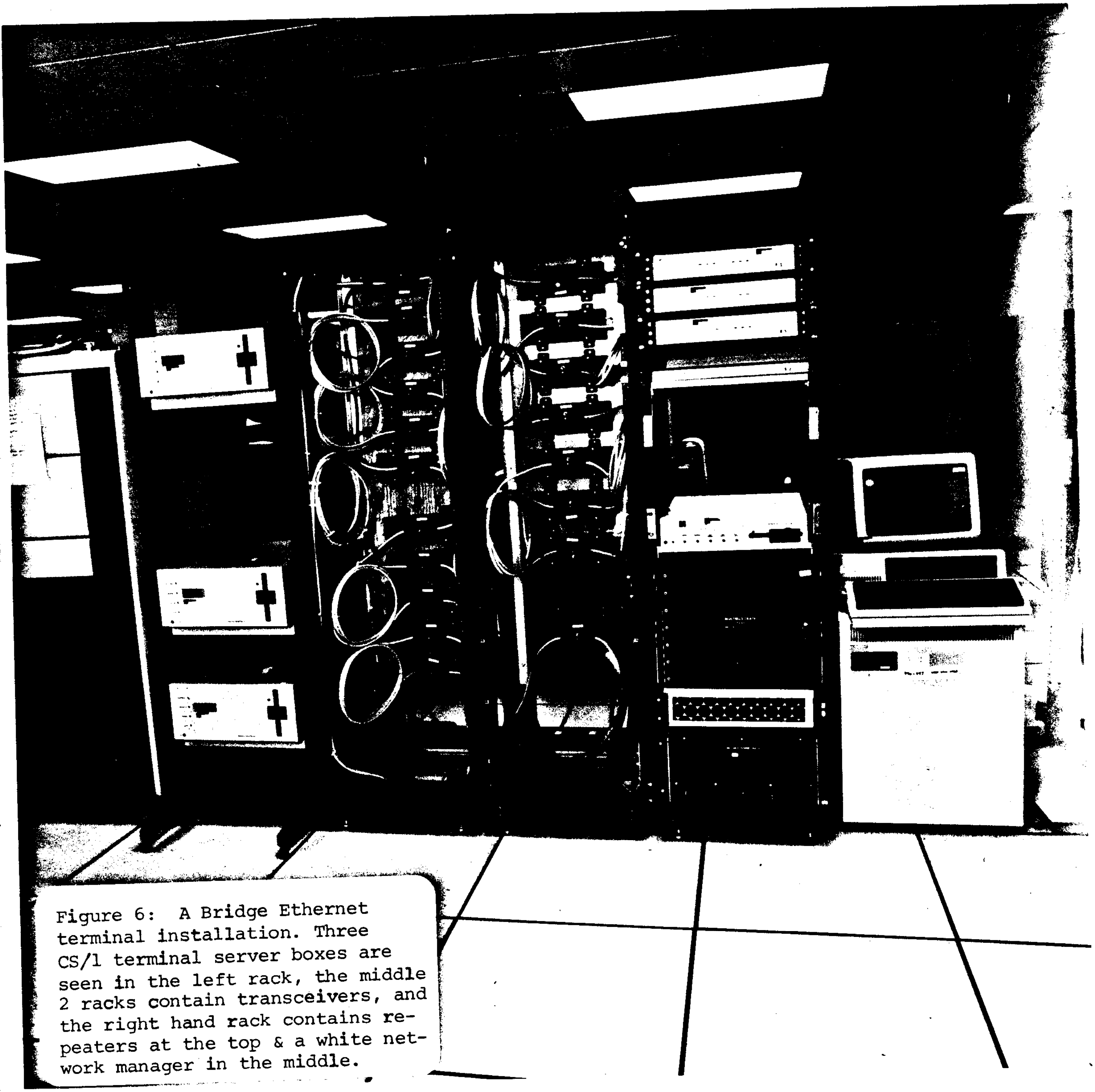

Figure 6: A Bridge Ethernet terminal installation. Three CS/1 terminal server boxes are seen in the left rack, the middle 2 racks contain transceivers, and the right hand rack contains repeaters at the top & a white network manager in the middle.