AD-A086 947 MITRE CORP MCLEAN VA F/G 17/2
CABLE BUS APPLICATIONS IN COMMAND CENTERS.(U)
NOV 79 S F HOLMGREN, A P SKELTON, D A GOMBERG F19628-80-C-0001
UNCLASSIFIED MTR-79W00383 NL

1 of 2
AD
A086947

FY79 Final Report: Cable Bus Applications in Command Centers

The MITRE Corporation

9

FY79 Final Report. for FY 1979

6

Cable Bus Applications in Command Centers.

10

Steven F. Holmgren
Anita P. Skelton
David A. Gomberg

October 1979 11

14

MTR-79W00383

12 142

Sponsor: Defense Communications Agency/
Command and Control Technical Center
Contract No. F19628-80-C-0001

15

~~This document was prepared for authorized distribution. It has not been approved for public release.~~

The MITRE Corporation
MITRE C³ Division
Washington C³ Operations
1820 Dolley Madison Boulevard
McLean, Virginia 22102

402364

EXECUTIVE SUMMARY

Microprocessor technology has advanced to the extent that the three or four lowest layers of a typical long-haul network protocol architecture may be implemented on one microprocessor-based circuit board. The impact of this evolutionary step on the capabilities of microprocessor-based networks will be profound.

This report documents the FY 79 activities of a MITRE/DCA program to determine the applicability of such a futuristic microprocessor-based cable bus local area network to World Wide Military Command and Control System (WWMCCS) command centers. The major emphasis of this program has been on a comparative analysis of cable bus and message switch local area network architectures. Potential cable bus and message switch command center applicability is measured here in terms of security, functionality, reliability, cost, flexibility and performance.

Specific tasks during FY 79 included the installation of a cable bus test bed with a gateway to the ARPA network, measurement of cable bus performance, and partial implementation of a cable bus protocol architecture..

Message Switch - Cable Bus Comparison

Prototypes of the message switch and cable bus which support WWMCCS requirements were not available for direct measurement. In fact, further research and experimentation with both systems is necessary before absolute comparisons can take place. Thus it is felt that some comparisons, particularly cost, performance, and reliability are premature enough so that no definitive opinion on applicability of the cable bus to WWMCCS command centers should be rendered at this time.

The following is a brief summary of the cable bus and the message switch comparison in the areas of security, functionality, reliability, cost, flexibility and performance.

Security

Simply stated, the goal of data security is to prevent the unauthorized or unintended disclosure, modification or destruction of information and to prevent denial of service. This goal is achieved by implementing countermeasures to a set of perceived security threats.

To achieve multi-level security, cable bus and message switch security measures must convincingly counter threats previously countered primarily by administrative and physical means. The message switch and cable bus security strengths and weaknesses lie in their differing abilities to implement the additional countermeasures imposed by this considerably more demanding environment.

Present day command center security measures rely largely upon physical and administrative steps to contain command center components. These steps include clearing all personnel to a system-high security level and placing components within a limited access vault.

The message switch mainframe or the cable bus backbone and interface units are components that would be installed within the vault. The vault prevents unauthorized physical access thereby preventing the introduction of specialized equipment or use of existing equipment by unauthorized personnel.

It is felt that sometime in mid-1982 or, possibly later, a multi-level secure NFE providing access to the AUTODIN II network will be available. The NFE and AUTODIN II multi-level data handling capabilities will necessitate a requirement for handling single level terminals and computers which are not necessarily all cleared to a system high security level. In this architecture, ADP measures must be implemented in both the cable bus and message switch to provide separation of sensitive data streams.

Because the message switch terminates all terminal and computer communication lines, a security kernel or kernel-like executive is needed to mediate invocations of the different communications line software modules. To insure that this mediation is correct and fair, the kernel software must be formally verified. The executive in turn provides support for other software modules that actually implement message switcning functions. To insure that multi-level data is handled correctly, the message switch specific software modules must also be formally verified.

Separation of multi-level data streams may be enforced somewhat differently in the cable bus architecture. Since each interface unit terminates one or two communication lines, each unit may still be operated at a single level. This means that only address detection and security level checking software must be verified. Address detection software monitors messages on the cable bus backbone to determine when one is addressed to the local interface unit. The amount of interface unit software to be verified is significantly less than that required in the message switch. While each interface unit only handles messages at a single security level, the cable backbone transports messages of varying levels. To separate data streams and provide protection against analysis of messages and traffic on the backbone, data encryption may be used to encode backbone message header and data information.

When mainframes support multi-level data streams, cable bus interface units may at any one time have messages of varying security levels within them. This means that software other than address detection and security level validation modules must be verified. It is believed that the same sofware constructs used to provide verified separation of message switch data streams will be directly applicable to the interface unit.

In the late 1980's the first components of the next generation of the WWMCCS Information System will begin to appear in operational commana centers. These components will include newer WWMCCS mainframes as well as teleconferencing and data base handling mainframes.

The loading induced by the secure handling from these systems will non-linearly increase the complexity of the message switch, resulting in steadily decreasing performance and the real possibility of overload failure, increased difficulty in maintenance of the software and the prospect of reverification of sizable amounts of software with the addition of each new device.

The multi-level secure interface unit, on the other hand, will not degrade as various components are added to the command center. Each additional component adds an interface unit to provide interface buffer capacity and protocol processing cycles. The increased load will in turn increase traffic on the cable bus backbone. However, the application of backbone technologies such as fiber optic bundles which support gigabit transfer rates should reasonably support command center backbone requirements.

It is believed that the discussion contained in the body of this report demonstrates that the cable bus will be at least as secure as the message switch architecture.

Functionality

The message switch and cable bus architectures are both functionally equivalent. They provide terminal-to-computer and computer-to-computer communication functions. The cable bus, however, has an inherent broadcast capability since all interface units see each message. With the introduction of a name table in the form of an associative memory, interface units can be logically grouped together. The message switch, on the other hand, must perform all of the broadcast functions in software by replicating the message for each communications line that is to receive it.

The MITRE cable bus system is unique among cable bus systems in that it uses radio frequency modems to transport digital information. This offers the capability to transmit video and audio signals along with the digital information on the same wire. Finally, this radio frequency transmission technique enables the creation of several logically distinct network systems using the same wire. The interface units may be "tuned" to the appropriate digital network. In the command center this lends itself to the separation of communities of interest without major software overhead.

Reliability

Since system reliability cannot be directly measured without historical records, a reliability comparison is limited to a discussion of architectural weaknesses which would make the message switch or cable bus inherently more or less reliable. The major architectural weakness of the message switch is clear. A failure in any one of the message switch software modules or a failure of the message switcn hardware halts all command center communications. Protection against these failures is usually implemented in the form of redundant systems. This is expensive; the message switch and its interfaces with other components must be replicated.

The cable bus architecture distributes its processing load to each interface unit. Each interface unit is a passive device whose only system wide failure mode, continuous transmission, can be protected against by a simple, inexpensive watchdog timer. The timer disables the interface unit if it has been continuously transmitting for an extended period. The cable bus, because it uses a single wire, is also subject to overload. Here, redundancy is also a solution. However, second or even third communication backbones are inexpensive.

Finally, the physical separation of interface units increases reliability. A failure in one unit will not damage the tables or data of another unit. It is thus concluded that the cable bus has the potential for significantly more reliable operation than the message switch.

Cost

Costs are divided into fixed initial purchase costs and recurring maintenance costs. A ten year life cycle is used to estimate recurring expenses. To make the comparison as valid as possible, message switch costs are estimated conservatively low and cable bus costs are estimated conservatively high.

Fixed message switch costs are estimated for terminal interfacing hardware, computer interfacing hardware and minimal hardware mainframe support for these interfaces at $51,554.

The recurring cost over a ten year life cycle is $28,230. The total estimated message switch cost is $79,784.

Fixed cable bus costs are estimated for the cable backbone and for interface units at $45,700. A recurring cost of $12,600 is estimated for each cable bus system. The total estimated cost for each cable bus system is $58,300.

Given that WWMCCS installs these systems at the 23 sites where NFE installations are planned, the comparative costs for the message switch and cable bus are:

Message Switch	($79,784)	$1,835,032
Cable Bus	($58,300)	$1,340,900

The cable bus is approximately 27 per cent less expensive than the message switch.

When reliability issues are factored into the cost in terms of duplicating message switches at each site versus duplicating critical interface units and cable backbones, the savings almost double.

Message Switch	($159,568)	$3,670,064
Cable Bus	($ 78,200)	$1,798,600

The cable bus is now approximately 51 per cent less expensive than the message switch.

Flexibility

The cable bus is a physically modular system. This physical modularity makes the cable bus amenable to growth and shrinkage to meet evolving communication requirements. As new command center devices are added to the cable bus, interface units can be correspondingly added to provide incremental increases in buffering capacity and cpu cycles. Message switch incremental increases in buffering capacity and cpu cycles tend to be both large and expensive.

This physical modularity also has the advantage that new device types may be interfaced to the cable bus without affecting the operation of existing software in other interface units. In message switcn architectures, software complexity increases non-linearly with the addition of each new device type; often existing software and buffer space has to be compressed to make room for new devices. On the other hand, interface unit software can be tailored to specialized requirements of one device without disturbing software for other devices of the same class. Therefore, it is felt that the cable bus architecture is significantly more flexible than the message switch.

Performance

Performance comparisons are difficult to make without actual measurement of message switch and cable bus systems. As a basis for making performance predictions, measurements of the NFE and MITRE cable bus are used.

NFE measurements are obtained from measurements of an early version of the NFE called the Experimental Network Front End (ENFE). Performance measurements of a newer NFE called the Interim Network Front End (INFE) are not yet available. Both NFEs are supported by a general purpose operating system, UNIX. The general purpose nature of UNIX severely limits throughput. A new operating system now under construction may increase NFE performance by as much as a factor of three to five.

Cable bus measurements were obtained by direct measurement of a cable bus test bed installed as a part of this program. Throughput and bandwidth measurements were made of the basic cable bus interface units and of the cable bus Transmission Control Protocol (TCP) implementations. The TCP implements command center virtual circuit and datagram functions. Modifications were made to the cable bus protocols to eliminate duplicate functions and decrease overhead. These modifications and the experimental results that provide the impetus for them are documented in the report.

The ENFE was measured as having an average maximum bandwidth of 33.5K bits per second (bps). Cable bus throughput was measured at the virtuai circuit level at 28.9K bps. The slowness of the cable bus can be attributed to its outdated microprocessor-base and unsophisticated (byte-at-a-time) test bed hardware interfaces.

If the new NFE operating system performs as expected, a three to five fold increase in performance may be forthcoming. Thus NFE performance estimates of 100.5K bps to 167.5K bps might be expected. During FY 80 a new 16-bit microprocessor interface unit will be tested. It is expected that this system will also perform from five to ten times faster than present test bed interface units. Thus cable bus performance estimates of 144.5K bps to 289K bps might be expected.

Effective cable bus and message switch performance is determined largely by buffering capacity and the arrival rate of "to-be-transmitted" messages. The number of "to-be-transmitted" messages is dependent upon the number of processor cycles available to set up message transmission. Since the message switch architecture uses a single cpu, the number of available message-set up processor cycles is limited by the single cpu speed. The buffering capacity is limited by the capacity of the message switch main memory. The cable bus, on the other hand, uses a cpu for each interface unit. Thus, cable bus cpu cycles and buffering capacity are determined by the aggregate number of interface units effectively handled by the cable backbone.

For performance comparison purposes, it is assumed that the cable bus can effectively handle four interface units (a larger number is somewhat more realistic). If as estimated each interface unit is abie to operate at five to ten times its present rate, from 144.5K bps to 289K bps, the aggregate cable bus performance of four interface units will be somewhere between 578K bps and 1156K bps. Based on these ranges, the cable bus may be significantly faster than the message switch in the near term.

Conclusion

Due to the early stages of development of both the cable bus and the message switcn, conclusive opinions as to their applicability to the command center are necessarily analytic and not empirical. However, based on the foregoing comparison it appears that the cable bus will be as secure as the message switch, from 20 per cent to 50 per cent less expensive, significantly more reliable and have better overall performance with a flexibility for incremental growth surpassing the message switch architecture. FY 80 development and testing should enable less subjective comparisons. FY 80 work will continue cable bus protocol development, a cable bus virtual terminal protocol with graphics capabilities will be investigated and a cable bus system will be installed at DCA Reston in order to begin evaluating the system in a command center setting. In a related MITRE internal research and development effort, a new, faster interface unit will be developed and tested.

ACKNOWLEDGMENT

Our efforts, as reported in this paper, would not have been possible without certain invaluable contributions by our colleagues both within MITRE and from the networking community at large.

Within MITRE, we wish to acknowledge our debt to the many people at MITRE Bedford for developing and refining the cable bus technology and for sharing their expertise with us. In our own group, John K. Summers and David C. Wood provided much in the way of direction and an abundance of support and Milton C. Harper has been an integral part of the entire project.

The DARPA Internet community generously shared their ideas and software with us. Jim Mathis at Stanford Research International implemented the MOS operating system and the TCP for the LSI-11 and Mike Wingfield at Bolt Beranek and Newman developed the UNIX TCP.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF ILLUSTRATIONS

LIST OF TABLES

1.0 INTRODUCTION

This report documents the FY 79 activities of a MITRE Mission Oriented Investigation and Experimentation (MOIE) program initiated in April 1978 and sponsored by the Defense Communications Agency (DCA). The purpose of MOIE programs is to develop MITRE expertise and to investigate new technologies of potential interest to the sponsor.

1.1 Background

The World Wide Military Command and Control Systems (WWMCCS) Command Centers are interconnected via an ARPANET-like long-haul network. In the near future, these command centers will be connected to the AUTODIN II long-haul network. A Network Front End (NFE) minicomputer will be used to interface command center mainframes and terminals to AUTODIN II. An expanded NFE could conceivably serve as a centralized local network message switch to interconnect additional computers, terminals and other components of a WWMCCS Command Center.

This program is investigating the applicability of a distributed cable bus local area network to interconnect command center components to each other and to offload NFE terminal handling functions. To measure cable bus applicability to the command center, the FY79 program focused on a comparison of cable bus and NFE-based message switch interconnection architectures in terms of security, functionality, reliability, flexibility, cost and performance.

1.2 Report Organization

This report is divided into eight sections. Section 2 outlines the WWMCCS command center, its purpose and components, and defines future WWMCCS command center communication requirements. Section 3 describes the components of local area networks and distinguishes between centralized and distributed architectures. Specfic instances of these architectures, viz., the message switch and the cable bus, are described. Section 4, Analysis Activity compares the two architectures in terms of security, functionality, reliability, cost and flexibility. Section 5, Development Activity, reports on the establishment of a cable bus test bed set up to provide an initial starting point for the experimental activity. Section 6, Experimental Activity, contains a comparison of the performance of a postulated WWMCCS message switch and the cable bus test bed. Section 7 describes a protocol architecture that evolved during the year to support command center requirements. Section 8 contains a synopsis and indicates the areas of future emphasis.

2.0 THE WWMCCS COMMAND CENTER

This section contains a brief outline of the WWMCCS command center and its present and projected communication requirements.

2.1 The Command Center Environment

The World Wide Military Command and Control System (WWMCCS) is responsible for the coordination of all U.S. military activities. World events are monitored by interconnected warning systems and electronic information management tools concentrated in geographically distinct military command centers. The management tools employed might include devices for receiving information (such as sensors or telemetry equipment), large data base machines for information retrieval, devices for processing or updating the data collected and computer controlled display devices. The output from these management tools facilitates high-level decision actions and/or tactical control for military activities.

Thus, a smoothly functioning, well-designed command, control and communications system enables the speed, processing power and retention/retrieval capabilities of the computer to be successfully combined with human intuition. The following scenario, depicted graphically in Figure 1, is an example of the desired integration:

> Geographically separated computers and sensors transmit information over communication channels. Received data is displayed as a diagram or sketch on a CRT. A specialist manipulates the display under computer control. The display is modified with data base information resulting in a visual chart with a color-coded decision variable. The enhanced chart is then transmitted to a command center for viewing on wall-sized screens by groups of senior decision makers.

2.2 The Command Center Components

The WWMCCS Command Centers of today consist of an enormous proliferation of communication devices for the receipt and dissemination

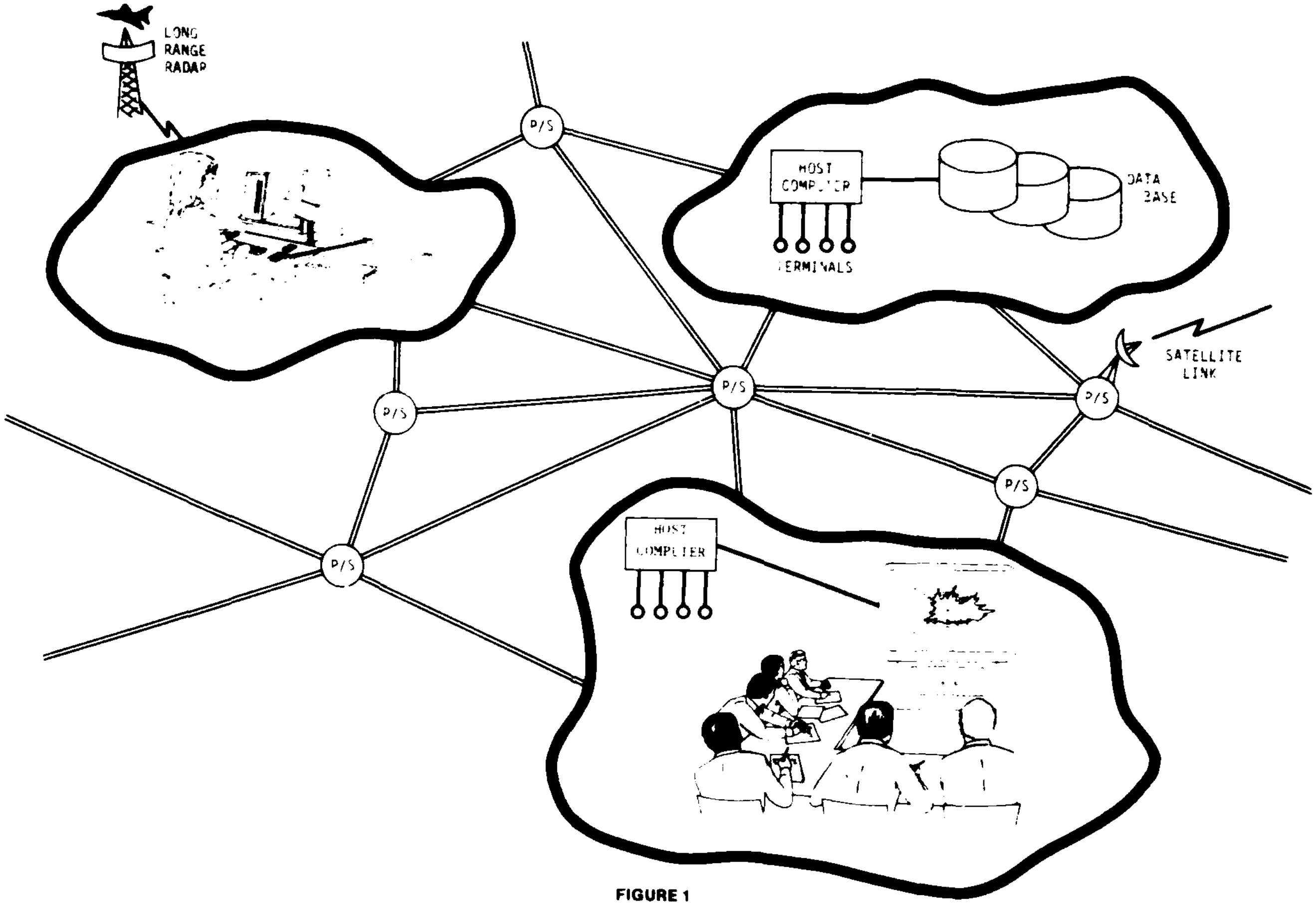


FIGURE 1
INTEGRATED COMMAND CENTER

of messages with varying levels of security. These devices provide service for voice, typed text, digital facsimile, closed-circuit T.V., commercial T.V., commercial radio and a variety of recording facilities.

The command center computer facility now houses Honeywell 6000 mainframe computers and a variety of terminal devices including Visual Information Processors (VIPs), IBM 2741s, Tektronix 4014 grapnics terminals, Tektronix 4632s, remote line printers (RLP 300s) ana IBM 2250s with light pens. A detailed discussion of command center components is contained in Acker et al.[1]

In addition to present day WWMCCS Honeywell 6000 series computers, individual command centers of the 1980's are likely to include the next generation WWMCCS Information System mainframe, Network Front Ends to offload networking software, the next generation WWMCCS terminals, such as the Tektronix Colorgraphics 4027, plasma panel terminals, intelligent terminals, teleconferencing machines, data base machines, an automated text and message handling system (including a document preparation capability via a text editor and formatter), hard-copy graphics capabilities, an electronic mail capability for voice, typed text, digital facsimile and, possibly, an intelligence system.

This expanded configuration enables the integration of all the monitoring, analysis and communications activities throughout the command center. A local area network is the ideal vehicle for this needed integration.

2.3 Command Center Local Network Models

Two models of local area networks receive primary attention in this report, the message switch model and the cable bus model.

Figure 2 depicts the message switch architecture. The message switcn is characterized by a medium scale mini-computer terminating

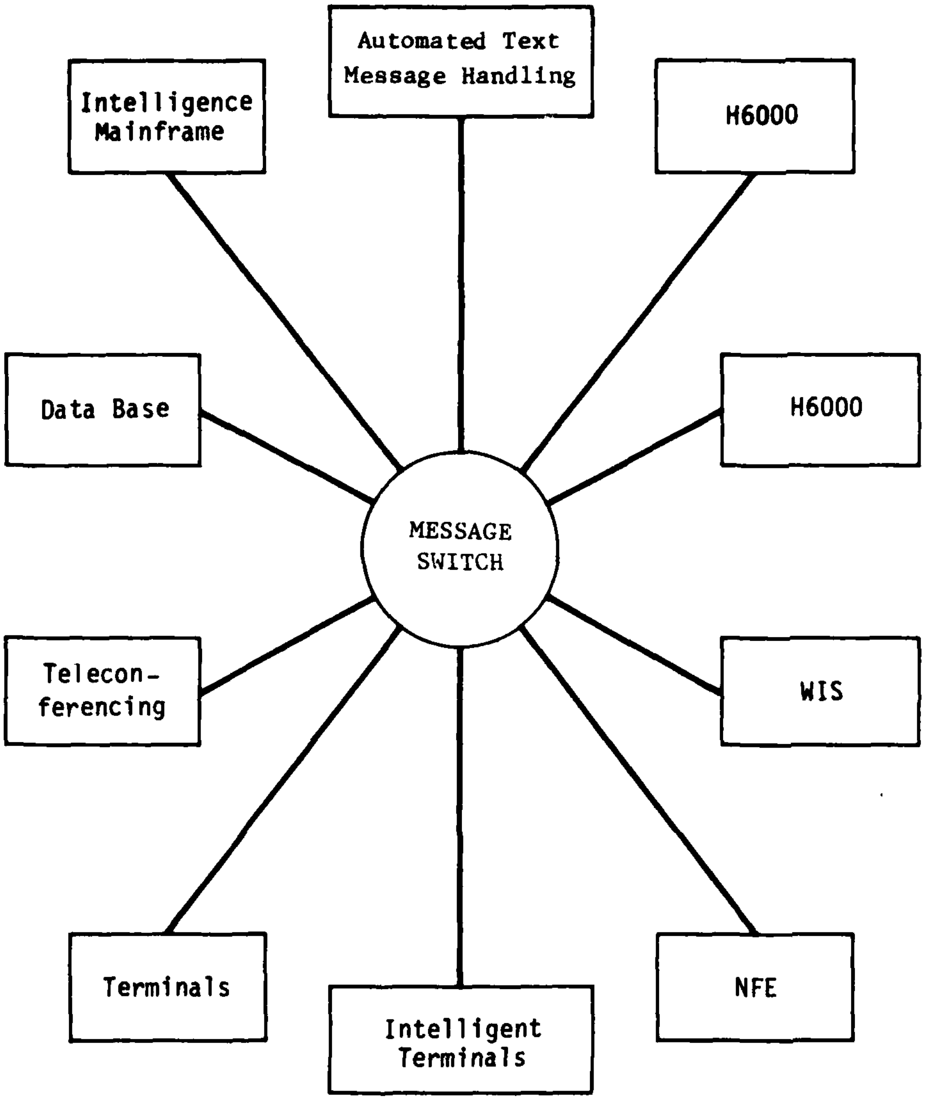


FIGURE 2
MESSAGE SWITCH ARCHITECTURE

communications lines from command center components. Software within the message switch provides virtual interconnection of the communication lines.

In the WWMCCS environment, the message switch must provide interconnection of the present day WWMCCS Honeywell 6000 series (H6000) mainframe computers and a variety of terminals. In the future these requirements will be expanded, as discussed above, to include interconnection of the next generation WWMCCS Information System (WIS), a Network Front End (NFE) which interfaces the command center to a long-haul packet-switched network, a teleconferencing facility, a data base machine, an automated text message handling facility, possibly an intelligence mainframe and new intelligent terminals with graphics capabilities.

Figure 3 depicts the cable bus architecture. The cable bus is characterized by a single wire communications medium interconnecting microprocessor-based interface units. Each command center communication line is terminated by an interface unit. Software in the interface units provide virtual interconnection of the communication lines.

Although the two models must address the same problems to provide the required integration among the command center components, the solutions are often strikingly different. It is the comparison of these models in the command center environment which is the focus of much of this report.

2.4 The Communications Network Requirements

A local area network consists of a set of communication lines to carry data and a set of protocols to manage the data flow. A protocol is a set of mutually agreed upon conventions which enables the orderly exchange of information between two functionally related elements. Whenever there is communication between functionally dissimilar entities, an interface must be defined. The communications

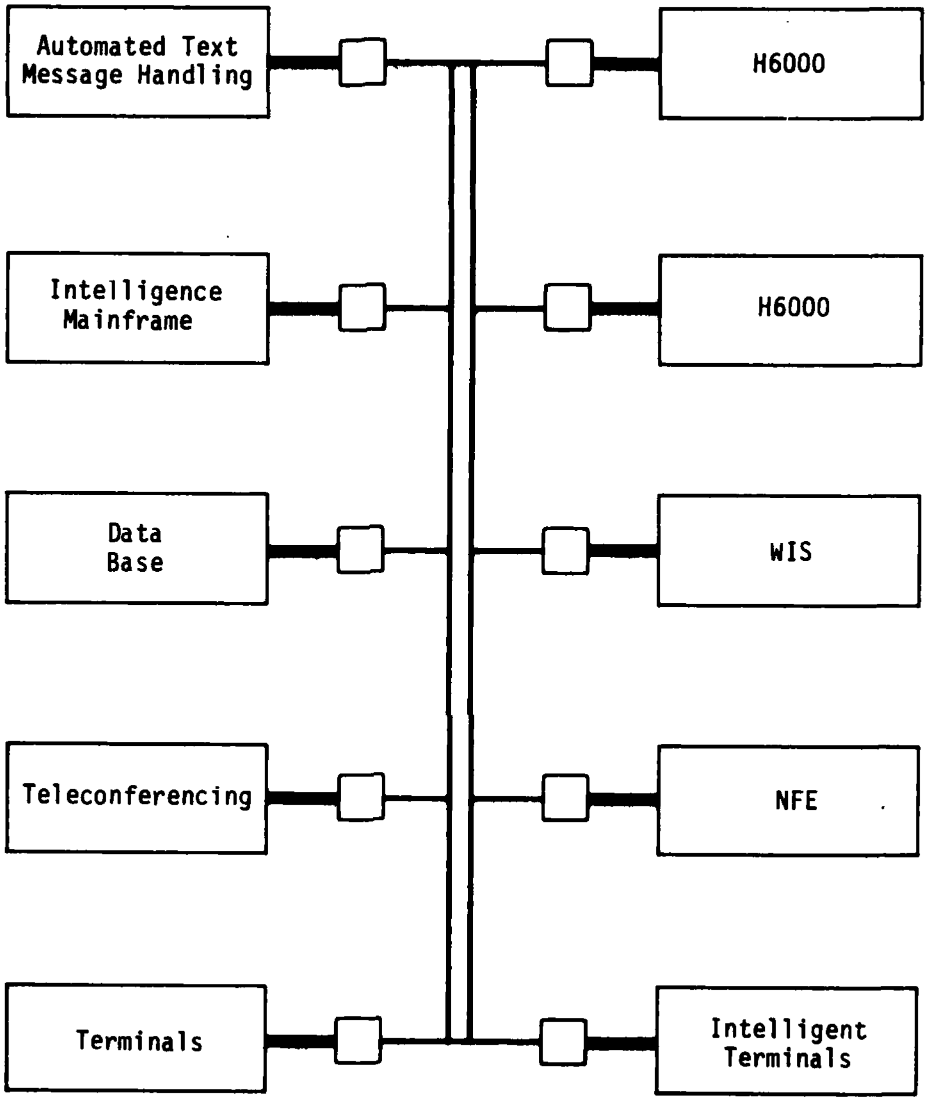


FIGURE 3
CABLE BUS ARCHITECTURE

network protocols and interfaces delineate the capabilities of the network. Based on the command center components described above, the capabilities which must be supported by a military command center network are as follows:

- interconnection between hundreds of terminals of various types and tens of computers ranging from microcomputers to large systems scattered through out a command center
- local terminal to host communications at up to 19.2K bps and terminal virtualization by mapping characteristics for various terminals and computers into a standard representation
- interfacing of terminals to the network inexpensively, i.e., not more than several hundred dollars per terminal
- local computer-to-computer communication with data rates of millions of bits per second
- interfacing of computers via high speed multiplexed interfaces capable of supporting many virtual connections
- offloading network specific interface software from mainframe computers by implementing such interface protocols as X.25 and HFP (Host-to-Frontend Protocol) in the network
- load leveling and other resource sharing concepts to enable efficient utilization of resources and to minimize delay during day-to-day and crisis operations
- accessing of remote resources via gateways to other networks

Additionally the network should support the following applications:

- exchange of critical messages (voice, typed text, facsimile) with multi-precedence levels
- preparation and transmission of documents (with graphics and text) of print quality
- access of remote data bases
- execution of standard command center application programs whose "location" (host/net) is not known and access of data bases whose physical location is not known

Of utmost importance is the requirement for:

- the capability to exchange intelligence and other sensitive information among qualified parties without fear of its accidental or intentional compromise, modification or destruction

2.4.1 The Protocol Requirements

The ability of the command center network to support the requirements delineated above is very much a function of local area network protocol architecture and its implementation. In order to enable remote devices, processes or users to exchange information, certain standard protocol mechanisms must exist. A hierarchical multilayered protocol structure separates required communication functions. In general, each level in the hierarchy is responsible only for the management of a particular class of resource (such as channel capacity, processors or address space). This separation results in both simplicity and flexibility. The replacement of any protocol layer should be transparent to the surrounding layers if the interfaces to the adjacent layers are preserved.

Certain protocol mechanisms are essential if the multiplicity of disparate computing elements in the command center model are to be joined into a unified whole. File access mechanisms are necessary to enable data base accesses between computers within a command center and globally between centers. Virtual terminal and graphics protocols are necessary to support the many terminals within command centers by mapping terminal characteristics into standard virtual terminal representations. Virtual circuits are necessary for extended interactions requiring the exchange of reliable data streams. Datagrams are useful for one-way and isolated exchanges.

Virtual circuits and datagrams require the support of additional basic protocol mechanisms. These mechanisms are: addressing (using a hierarchical address space to enable both intranetwork and internetwork addressing), congestion and flow control, reliability and out-of-band signaling (to enable urgent information to be designated). Taken together, all of these basic protocol mechanisms must be flexible and robust enough to support communications over diverse transmission media with characteristics that vary in bandwidth, delay, error rates and broadcast capability.

At the very lowest levels of the architecture are the network specific mechanisms used for the access and control of the particular transmission medium. This architecture is shown in Figure 4.

2.4.2 The Interface Requirements

In addition to basic transport facilities, the network should offer a variety of standard hardware and software interfaces so that a wide spectrum of devices can be attached with relative ease.

2.4.2.1 Hardware. The network should offer standard electrical serial interfaces for terminals such as RS-232C and, perhaps later, RS-422/3. Both synchronous and asynchronous serial interfaces should be supported. A parallel interface must be provided to enable high-speed direct memory access (DMA) transfers from computer devices to and from the network.

2.4.2.2 Software. The network should support data link control interface protocols to provide error checking and failure control of transmitted data. An international standard which does this at the bit level is HDLC (high-level data link control); an IBM standard performing the same functions at the byte level is BSC (binary synchronous data link control). Classically, these have been implemented in software. Recently, Western Digital has announced an HDLC integrated circuit chip.

If a mainframe has the capabilities to support virtual circuit software, that software should be implemented in the mainframe. However, for any one of a number of reasons, the mainframe may not be able to cleanly support a virtual circuit implementation. For these

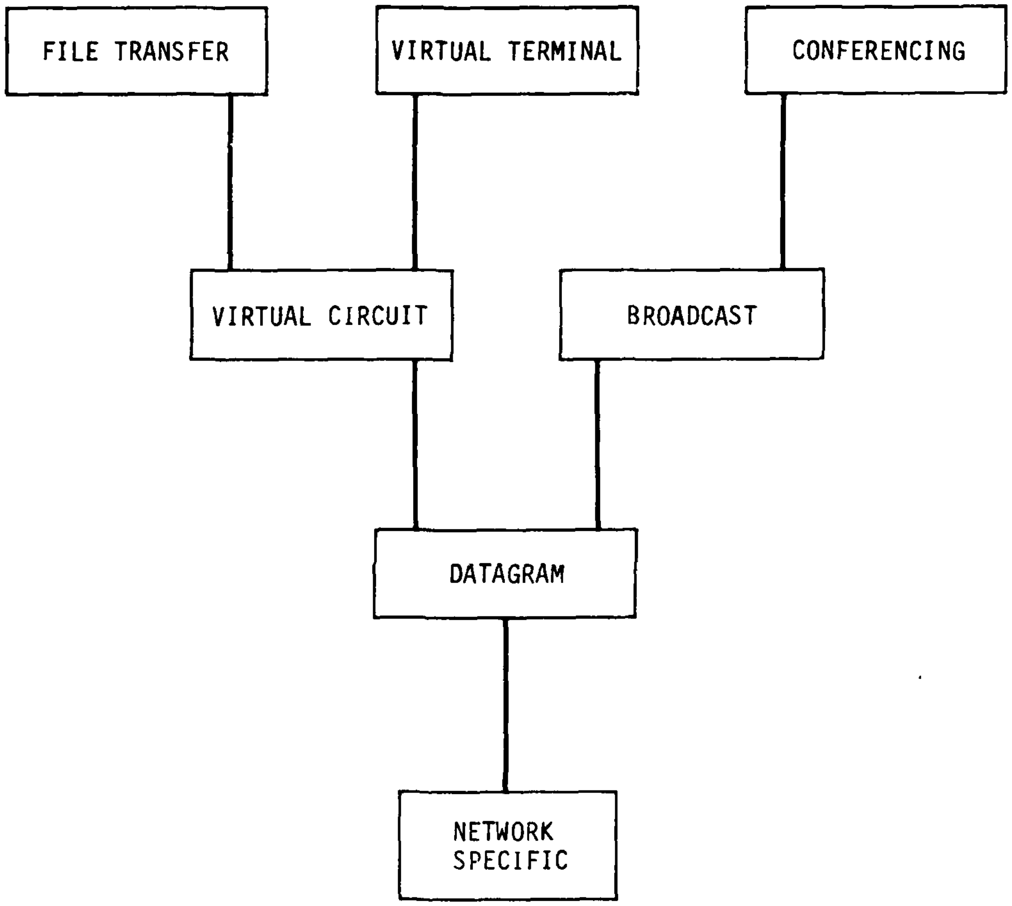


FIGURE 4
COMMAND CENTER PROTOCOL LAYERING

cases, the local area network should provide the capability to offload the mainframe virtual circuit software. To do this, an interface protocol is required which allows the mainframe to direct the activities of the local interface unit. A prime candidate for this interface protocol is the Host-to-Frontend Protocol (HFP)[2] currently used between the NFE and the H6000 host. Another possibility is to use the CCITT X.25 interface protocol. Implementation of an interface protocol in the local network would enable the network specific protocols to be offloaded from the host computers. Ideally, this offloading provides a simplified and easily implemented network interface. The resultant network interface unit would then be capable of being directed by devices and processes with diverse requirements.

2.4.3 Security Requirements

The design of a command center communications system must respond to all recognizable threats to data security. It is reasonable to assume that appropriate physical and procedural security measures are followed in and around the command center so that certain kinds of threats lie outside the scope of this design. The possible threats generated by cleared and uncleared personnel.which could lead to exposure, modification or destruction of information or to denial of service must be catalogued and appropriate countermeasures employed. Threats arising from accidental causes or from system component failures must also be considered. The goal of the design is an efficient communications network which can simultaneously serve users of sensitive information at all security levels without the possibility of information compromise.

3.0 LOCAL AREA NETWORKS

A local area network is a high speed communications network interconnecting a set of computing elements over an area potentially spanning several kilometers. The success of a network is determined by how well these computing elements are able to exchange information (messages) with one another. A network's ability to transmit and receive messages with speed and reliablity is determined, to an extent, by the degree of independence of its elements. The degree of independence is a function of the kind of control that a network node applies to each message as it traverses from source to destination.

Different local area network control architectures are named by their interconnection topologies: star, ring and bus. The star-type architectures fall into a group of networks which provide centralized control of message traffic. Bus and ring networks generally render decentralized control of message traffic, thereby providing nodal independence. Centralized control increases the complexity of the resultant network, while decreasing the reliability. A message switch based on the WWMCCS Network Front End (NFE) is an example of a star configuration, and the MITRE cable bus is an example of a distributed architecture. To provide background for a comparative analysis, the distinguishing characteristics of a star architecture like the NFE, a bus architecture like the MITRE cable bus and, for completeness, the ring architecture are discussed.

3.1 Central Control - Stars and Modified Stars

Centralized topologies evolved out of the economics of producing computers in the mid-1960's. To be cost effective, large computers were built and maintained centrally. Centrally controlled architectures are typified by complex primary nodes and relatively simple secondary nodes. The secondary nodes implement only very basic communications functions. Primary nodes support the remaining functions; routing, flow control, reliability, protocol translation and, sometimes, security.

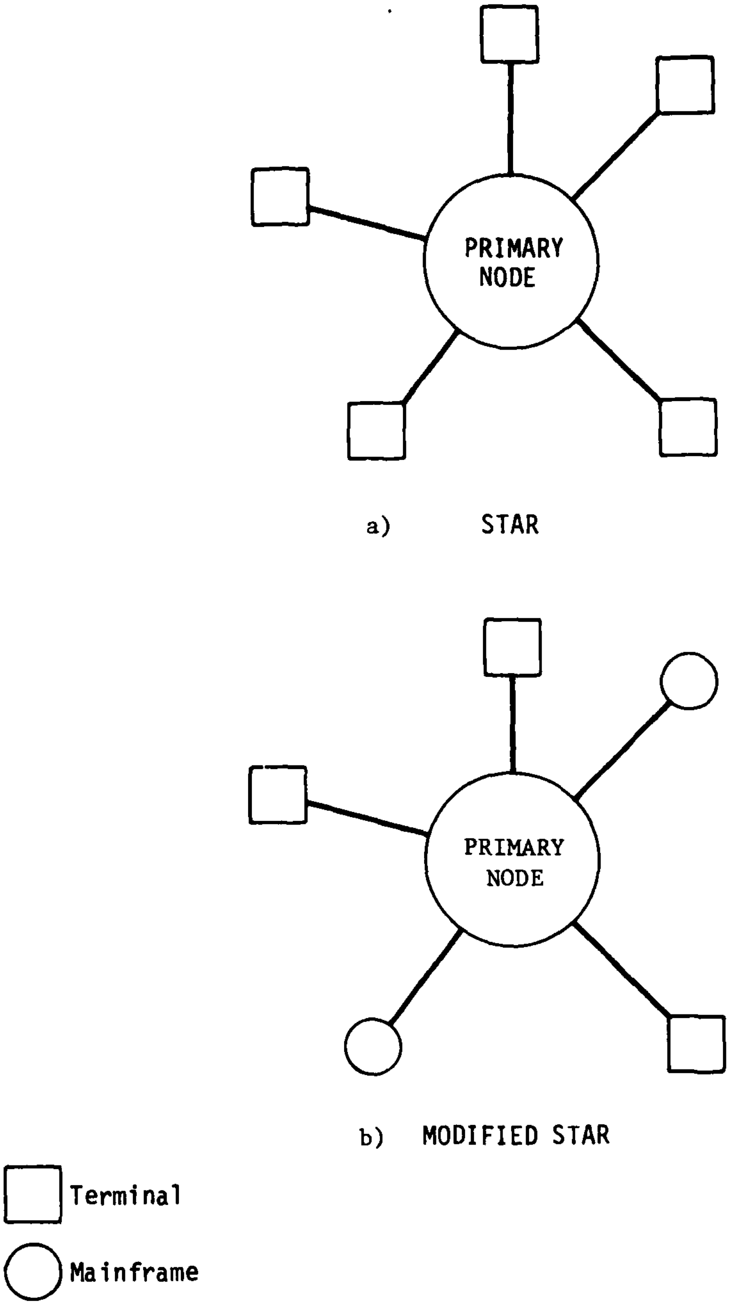
The star network, shown in Figure 5a, is represented by a single primary node and terminal nodes. The primary node supports message routing and character set translation. The terminal nodes rely completely on the primary node for all communications functions. This simple architecture is commonly used by time sharing systems in which messages flow between a number of terminal terminals and a primary central processor. The advantage of this architecture is that terminal nodes are inexpensive to construct.

When computer mainframes with terminal node communication capabilities are added to the star network architecture, a modified star architecture, shown in Figure 5b, results. Since failure by one computer node should not affect the operation of the primary node or other secondary nodes, communications are implemented with elements of defensiveness not found in simple star architectures. Since the mainframe nodes may be different, elements of generality are increased in the primary node implementation. Taken together, these elements support a mutually suspicious environment which is significantly more complex than the simple star architecture described above.

The clear drawback to the topologies shown in Figure 5 is reliability. If the primary node fails, network communications are halted. The primary node must operate correctly in the face of simultaneous high speed communications with computer secondary nodes and, in addition, have the capacity for generalized protocol translations, virtualization and dynamic routing functions.

3.2 Decentralized Control - Rings and Cable Buses

Initially, the centralized topologies were implemented with comparatively large mainframes. As technology advanced, some of the communications functions were offloaded to large mini-computers to



a) STAR

b) MODIFIED STAR

FIGURE 5
CENTRALIZED MESSAGE CONTROL

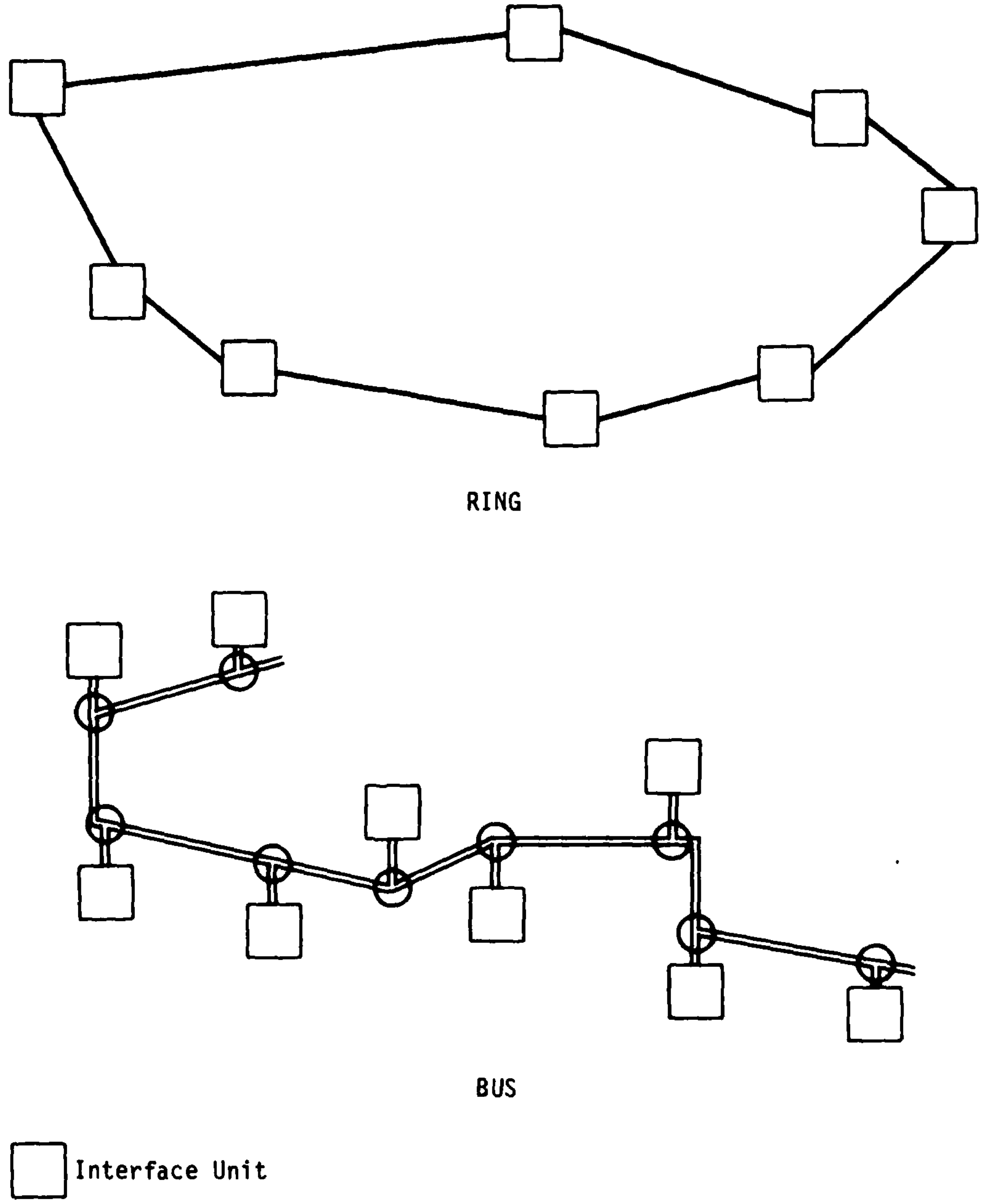
relieve the mainframe of communications overhead. Decreasing hardware costs are making it feasible to distribute primary node capabilities to each secondary node. The cable bus and ring architectures are examples of this evolutionary step.

Ring and cable bus architectures, shown in Figure 6, are characterized by a high bandwidth single-wire communications medium interconnecting intelligent interface units. The interface units have two major functions: they manage message communications over the medium, and they interface user devices (computers, terminals and other peripheral devices) to the communications medium. The cable bus and ring bus architectures are an intermediate hybrid of the computer peripheral bus and the centrally connected network. The speeds attainable through these architectures are tantamount to those of a peripheral bus but, at the same time, in interconnecting standalone computer devices both the ring and cable bus must take on aspects of defensiveness and generality common to centrally connected networks. To some extent, the homogeneity of the interface units simplifies intra-interface unit protocols by allowing assumptions to be made about interface unit processor speed, word size, buffering capacity and timeout values which are untenable in the fully connected environment. These architectures eliminate the singular dependence on a central node by distributing message control to each network node without compromising the simplicity of the nodes. This is made possible by several architectural factors:

- The homogeneity of the cable and ring relaxes the degree of mutual suspicion; each of the nodes are running a copy of the same software.

- No routing is required; each node sees all messages and captures those messages whose destination address matches its own.

- Each node provides its own translation from device specific protocols to generalized virtual protocols. Since the translation is "closer" to the device, the degree of generality found in centralized translations can be reduced. The node's



RING

BUS

FIGURE 6
DECENTRALIZED MESSAGE CONTROL

recognition of a specific terminal type or of a computer's word size is an immediate example of this relaxation.

The character of these architectures is affected by interface unit algorithms used to access the transmission medium. Various algorithms exist to establish temporary communications medium ownership by one interface unit in the face of a set of interface units which simultaneously wish to transmit a message. If these algorithms are centrally controlled, decentralized architectures will take on many of the deficiencies of centralized architectures. The distinction between the ring and cable bus architectures is found in these algorithms.

Ring architectures have employed a number of different algorithms for this purpose. These algorithms are described in Liu.[3] The major disadvantage of ring architectures from a command center point of view is that each of the algorithms requires an interface unit to become an active element in the network. Each interface unit must pass on a "time-to-transmit" token, or "take-a-message-off" the network, or be in timed synchrony with other units to store data in "slots". If an interface fails while it is in this active mode, manual or semi-automatic recovery measures are required.

The cable bus architecture does not have this problem. Various cable bus architectures are described in Trooper.[4] Cable bus interface units implement a contention algorithm to establish medium ownership. There are a variety of contention algorithms. The most successful ones check to see if the medium is in use before transmitting a message. This lessens the probability of one interface unit damaging the transmission of another unit. Due to signal propagation delay, there is a small time lag between the moment of transmission by one unit and the moment that another interface unit is able to detect a busy medium. Simultaneous transmissions during this time interval cause collisions. Thus the effectiveness of these contention algorithms is determined by interface unit adeptness at detecting collisions.

Recent advances in contention algorithm technology have produced theoretical communication medium bandwidth utilization of 85 to 95 per cent.[5] These advances and the decreasing cost of interface unit hardware are the major reasons for recent interest in the future of cable bus systems.

3.3 Centralized and Decentralized Implementations

The message switching functions of the Network Front End exemplifies a centralized architecture and the MITRE cable bus exemplifies a decentralized architecture. These systems are used for quantitative comparisons in this report.

3.3.1 Network Front End

The Network Front End (NFE) uses a PDP-11/70 minicomputer to support message switching software. The system was developed for DCA to interface command center Honeywell computers and terminals to packet switched networks. Originally, the NFE was interfaced to the ARPA network. It will soon be interfaced to the forthcoming AUTODIN II network. The message switching software is programmed in a high level language called 'C'. Both the message switching software and the C language are supported by UNIX, a general purpose operating system developed by Bell Laboratories.

The PDP-11/70 has 512K bytes of RAM, a disk, a synchronous terminal multiplexor interface, an asynchronous terminal multiplexor interface, as well as high-speed direct memory access interfaces to the packet switched network and the Honeywell computer.

The message switching software implements the Host-to-Frontend Protocol (HFP) to interface with the Honeywell computers. It also implements packet switch network specific software to interface to the ARPA and AUTODIN II networks. The HFP software is composed of several layers. One of the layers is reponsible for management of virtual channels between a Honeywell computer and a front end. The other layers interface commands and data from Honeywell-based programs to the packet network specific software.

The NFE is still in an experimental form. Extensive measurements of the NFE in this form have been taken.[6] These measurements are used as a basis for comparison with the MITRE cable bus.

3.3.2 MITRE Cable Bus

The MITRE cable bus is based on a system which was developed at MITRE Bedford. It is currently in operation there and at a number of government sites. Details of the system can be found in Hopkins,[7] Malis[8] and Hopkins et al.[9]

The system uses standard Community Antenna Television (CATV) coaxial cable components and microprocessor based interface units to interface subscriber computers and terminals. The cable bus consists of two parallel coaxial cables, one inbound and the other outbound, which are connected at the headend as shown in Figure 7. This architecture takes advantage of well-developed unidirectional CATV components. The CATV components include coaxial cable, wideband linear amplifiers to boost the signal where needed, splitters to divide the cable into a branched structure which easily conforms to building topology, and taps to provide access to the cable.

The interface units are designed to transmit on the inbound cable and receive on the outbound cable. The interface units contain Radio Frequency (RF) modems and microprocessor systems.

The Radio Frequency modems modulate a carrier signal to transmit the digital information. A data rate of 307.2K bps is used on the cable which occupies about 1 MHz of bandwidth around the 24 MHz frequency. The remaining bandwidth can be used to carry other digital

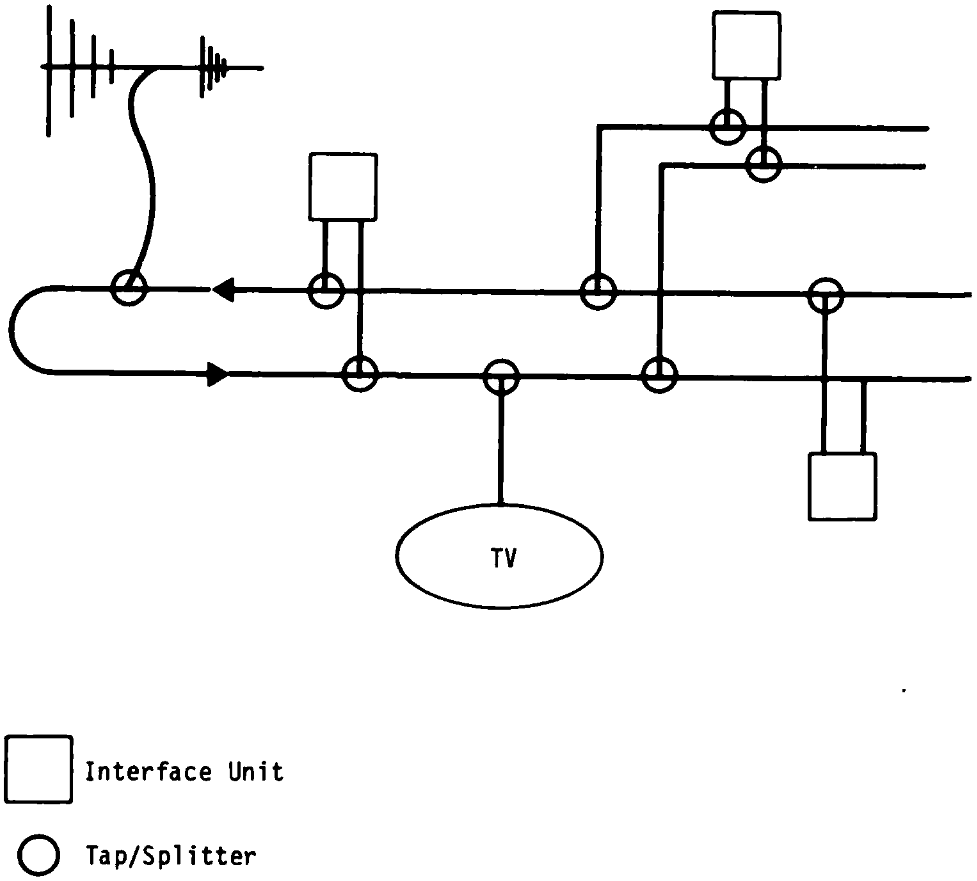


FIGURE 7
MITRE CABLE BUS

channels, video in the form of closed circuit and off the air TV and audio.

The interface unit microprocessor system uses about 20 MSI and LSI integrated circuits. The integrated circuits include a MOS 6502 Microprocessor, 2K bytes of programmable read only memory (PROM), 2K bytes of random access memory (RAM), a clock, and interface circuits to the modems and user devices.

4.0 ANALYSIS ACTIVITY

The analysis activities discussed in the following sections compare the capabilities of the cable bus and the message switch architectures to meet five command center requirements:

- Security
- Functionality
- Reliability
- Cost
- Flexibility

Versions of the NFE and cable bus which support all perceived WWMCCS command center requirements do not exist at present. Therefore, NFE and cable bus comparisons are necessarily analytic and not empirical in nature. This is particularly true in the Security, Reliability, and Cost analysis sections where experience with operational prototypes is required to make definitive statements.

4.1 Security

Simply stated, the goal of data security is to prevent the unauthorized or unintended disclosure, modification or destruction of information and to prevent denial of service. This goal is achieved by implementing countermeasures to a set of perceived security threats.

The analysis approach has been to look at the evolving WWMCCS security environment in the context of comparative message switch and cable bus security measures. The WWMCCS command center security policies may evolve from a single-level, system-high security policy to a multi-level security policy in the mid-to-late 1980's.

To achieve multi-level security, cable bus and message switch security measures must convincingly counter threats previously countered primarily by administrative and physical means. The message switch and cable bus security strengths and weaknesses lie in their differing abilities to implement the additional countermeasures imposed by this considerably more demanding environment.

4.1.1 The Data Security Problem

The data security problem can be defined in terms of the security threats that must be countered by a command center local area network. In discussing these threats, comparisons are made between cable bus countermeasures and message switch countermeasures in order to point out the potential advantages and disadvantages of each. At a minimum, one would desire that the cable bus system would afford at least as much data security as the message switch.

There are a large number of potential threats to data security, many of which apply only in the broadest sense to communications processors. In many instances the appropriate countermeasures are entirely unrelated to the particular implementation. For example, countering physical theft of data by a trusted person from within a facility is entirely a matter of physical and administrative safeguards such as inspection of materials leaving the site and individual clearances. Thus, threats such as penetration of a facility by an overwhelming force and threats related to administrative countermeasures such as periods processing, guarded restart procedures and system high personnel clearance are seen as equally applicable to both the cable bus and message switch. These threats are mentioned here and in the sections that follow as a baseline security policy with little or no further comment. More complete discussions of these threats and countermeasures are contained in Courtney[(10)] and NSA.[(11)] While it can never be demonstrated that a particular list of threats in a given environment is complete, an attempt has been made to deal with the perceived major threats to the cable bus and message switch systems. Table I is a capsule description of the remaining threats that must be effectively countered by both the cable bus and

TABLE I

COMMAND CENTER DATA SECURITY THREATS

		RESULT			
		Information Disclosure	Information Modification	Information Destruction	Denial of Service
	ACCIDENTAL THREATS				
A1	MISROUTING Header of properly prepared message in the clear is accidentally changed during transmission so that the address or authorization information is incorrect	●			
	SPILLAGE Misapplication of authorization or accidental appendage of parts of one message to another	●			
	LABELING FAILURES System errors which cause incorrect or missing classification or compartment indicators	●			
A2	SYSTEM COMPONENT FAILURES				●
	INTENTIONAL THREATS				
I1	PHYSICAL DESTRUCTION Cutting communications lines or other deliberate action				●
I2	UNAUTHORIZED SYSTEM ACCESS An outsider with a legitimate terminal or other means, or by an insider's violation of authorization level, need-to-know or community of interest	●	●	●	●
I3	EMANATIONS Interception or analysis or stray electronic signals used to acquire information directly or through traffic analysis; if message lengths, sources and destinations, traffic volumes and flow patterns are discernable, useful information may be gained without message contents	●			
I4	PROBING The introduction of spurious command sequences in an attempt to drive software into unknown or untested states which may allow unauthorized access	●	●	●	●
I5	SPOOFING Various masquerading techniques allowing attacker to take on rights of legitimate user	●	●	●	●
	PLAYBACK Repetition of recorded command sequences to acquire access	●	●	●	●
I6	UNREASONABLE SERVICE DEMAND Total consumption of available resources; overloading				●
I7	ADVANCE PLANTING Previously planted devices (Trojan horse, trap door, time-bombs) triggered by outside signal or event sequence or calendar event allowing access or service denial	●	●	●	●

message switch. Data security threats may be divided into two categories, accidental and intentional. The accidental threats are labeled as A1 and A2. The intentional threats are labeled I1 through I7. (These labels are used in Table II to refer to specific threats.)

4.1.2 The Evolving WWMCCS Security Environment

The command center security model of today is a single-level mainframe attached to a single-level long haul network. Over the next decade the WWMCCS security policy will evolve from this relatively simple single-level, system-high policy to more complex security policies involving mixed single-level entities and eventually to a full multi-level security policy.

In the present simple model, physical and administrative procedures can insure the integrity of the environment. As the transition to a multi-level security policy occurs, sophisticated software validation techniques and encryption methods are of paramount importance.

The ensuing sections explore the WWMCCS security policy evolution in incremental steps. At each stage, for both the cable bus and message switch systems, a determination is made of the requisite security countermeasures and an assessment is made of the relative ease of implementation of these countermeasures. This assessment demonstrates that the distributed nature of the cable bus does not preclude the effective use of classical physical containment security methods, and moreover, has a favorable effect on the implementation of the more sophisticated ADP countermeasures.

4.1.2.1 The Present Security Environment. Present day command center security measures rely largely upon physical and administrative steps to contain command center components. These steps include clearing all personnel to a system-high security level and placing components within a limited access vault.

TABLE II

EVOLVING WWMCCS COMMAND CENTER SECURITY CONSIDERATIONS

PRESENT ENVIRONMENT

WWMCCS Mainframes Connected To Single-Level Long-Haul Network

COUNTERMEASURES

MESSAGE SWITCH	CABLE BUS
Entire system within secure vault [I1,I3]	
System high operation [A1,I4]	
Remote access by name/password [I2]	
Separation of commands and data by making message switch or interface unit non-programmable [I4,I6]	
System limitation on resource usage	
by operating system scheduling algorithm [I6]	by contention, guarded with watchdog timer [A2,I6]
each device on individual line [A2]	

TABLE II (Continued)

MID-1982 ENVIRONMENT

Interconnection Via Multi-Level Secure Long-Haul Network
(AUTODIN II) With Hosts And Terminals At Different Single Levels

COUNTERMEASURES

MESSAGE SWITCH	CABLE BUS
Individual Authentication [I2]	
in hosts and message switch	in hosts and interface units
Device Authentication [I2,I5]	
kernel mediates transfers between different devices (complex software)	interface units tailored to device (modular, simple software)
Verified software in message switch [A1,I4,I7] (difficult for large amounts of code)	Verified address detection software in interface units [A1,I4,I7] (reasonable for small amounts of code)
Security level and compartment check in message switch [A1,I4]	Security level and compartment check in interface units [A1,I4]
Monitoring and auditing by kernel reported to security officer [I2.I4]	Monitoring and auditing by control point on cable bus reported to security officer [I2,I4]
	Replication of fields, fixed message sizes, cyclic redundancy checks [A1,I3]
	Encryption hardware and fixed key distribution [A1,I3,I4,I5]
	Device connection monitor to detect failures, connections, disconnects [A2,I3,I4,I6]

TABLE II (Concluded)

1985-1987 ENVIRONMENT

Addition of Multi-Level Hosts

COUNTERMEASURES	
MESSAGE SWITCH	CABLE BUS
End-to-end encryption and expanded access control techniques [A1,I3,I4,I5]	Encryption with PKD or other dynamic key distribution [A1,I3,I4,I5]
	Verified software in interface unit [A1,I4,I7]

The message switch mainframe is one of the components that would be installed within the vault. Figure 8 depicts this architecture. The vault prevents unauthorized physical access thereby preventing the introduction of specialized equipment or use of existing equipment by unauthorized personnel. To an extent, it also prevents emanations due to the shielding of the vault. Other emanating equipment also installed in the vault usually prevents recognition of message switch emanations. ADP security measures are rather limited currently. Attempts are made to test hardware and software to demonstrate correct operation under both expected and unexpected circumstances. Certain denial of service threats are moderated or defeated by operating system scheduling algorithms which attempt to distribute system resources equitably. Simple password mechanisms are usually employed to provide individual authorization. Additionally, the message switch operating system can be configured to allow no user programming which aids in preventing some probing attacks.

The cable bus backbone must also be protected from physical attack by installation in some form of protected environment. If, as shown in Figure 9, the cable bus backbone and all interface units are similarly protected by a vault, all message switch physical security measures apply. An additional component in each interface unit, known as a watchdog timer, inhibits a failed interface unit from denying service to other interface units by continuous transmission. If an interface unit has been continuously transmitting for an extended time, the watchdog timer disables the interface unit. The interface unit is not a user programmable device, so it is not subject to being used in probing attacks. The current interface unit has been subjected to TEMPEST emanation testing.(12)

The variable length of a cable bus backbone allows it to expand or contract to meet a variety of physical considerations. On one hand, one might envision a very short cable backbone, on the order of five or ten feet, which connects interface units which have been

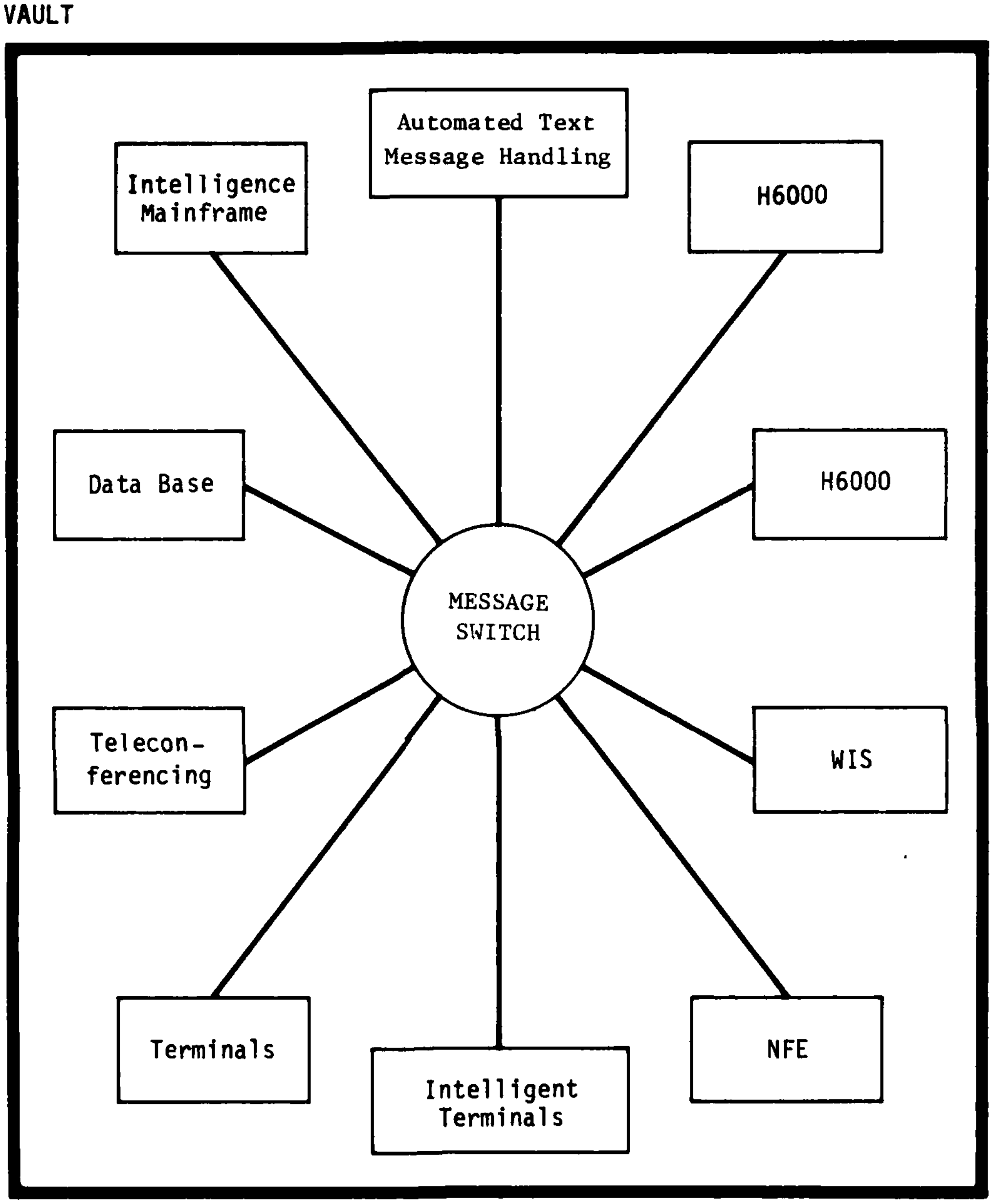


FIGURE 8
SECURE MESSAGE SWITCH

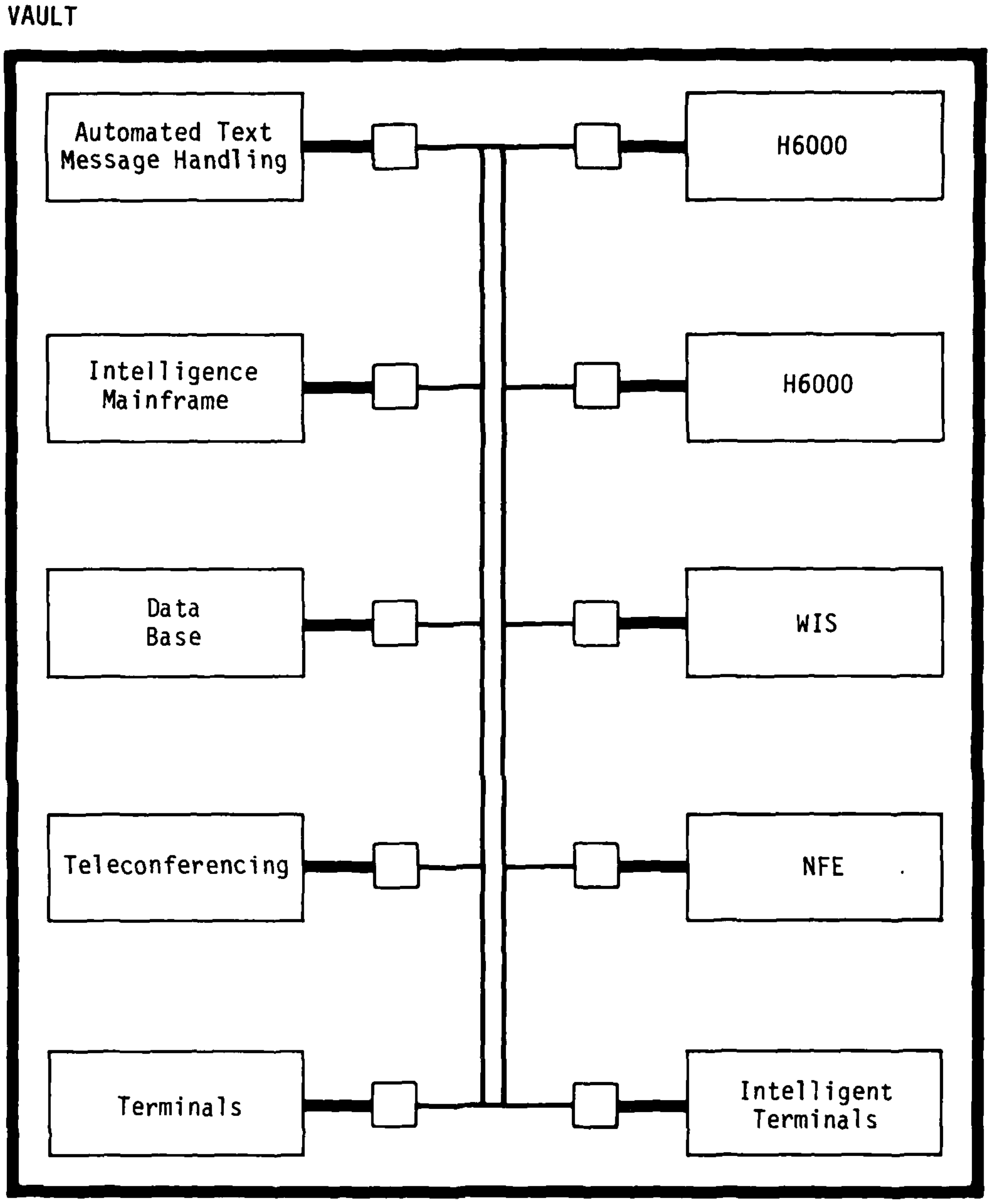


FIGURE 9
SECURE CABLE BUS

concentrated in a very local area. This architecture is shown in Figure 10. The communications lines terminated by these interface units are the same lines that would be terminated by a cabinet containing a message switch mainframe. Thus by physically installing the cable bus in a vault the cable bus can be as physically secure as the message switch. On the other hand, a longer backbone, as shown in Figure 11 may be desirable. If this is the case, the backbone may be extended to be physically closer to command center components possibly increasing the reliability of the component to interface unit transmissions and possibly easing the addition of new equipment.

It is expected that single-level secure mainframes and many of their physical and administrative security measures will continue to be a part of the command center environment for the foreseeable future.

4.1.2.2 The Environment of the Early-80's. It is felt that sometime in mid-1982 or, possibly later, a multi-level secure NFE providing access to the AUTODIN II network will be available. The NFE and AUTODIN II multi-level data handling capabilities will necessitate a requirement for handling single level terminals and computers which are not necessarily all cleared to a system high security level; some terminals may access the AUTODIN II network at a confidential level while some mainframes are operating locally at top secret. In this architecture, not all personnel, terminals and computers are necessarily cleared to the highest security level. Thus, ADP measures must be implemented in both the cable bus and message switch to provide separation of sensitive data streams.

Because the message switch terminates all terminal and computer communication lines, a security kernel or kernel-like executive is needed to mediate invocations of the different communications line software modules. To insure that this mediation is correct and fair, the kernel software must be formally verified. The executive in turn

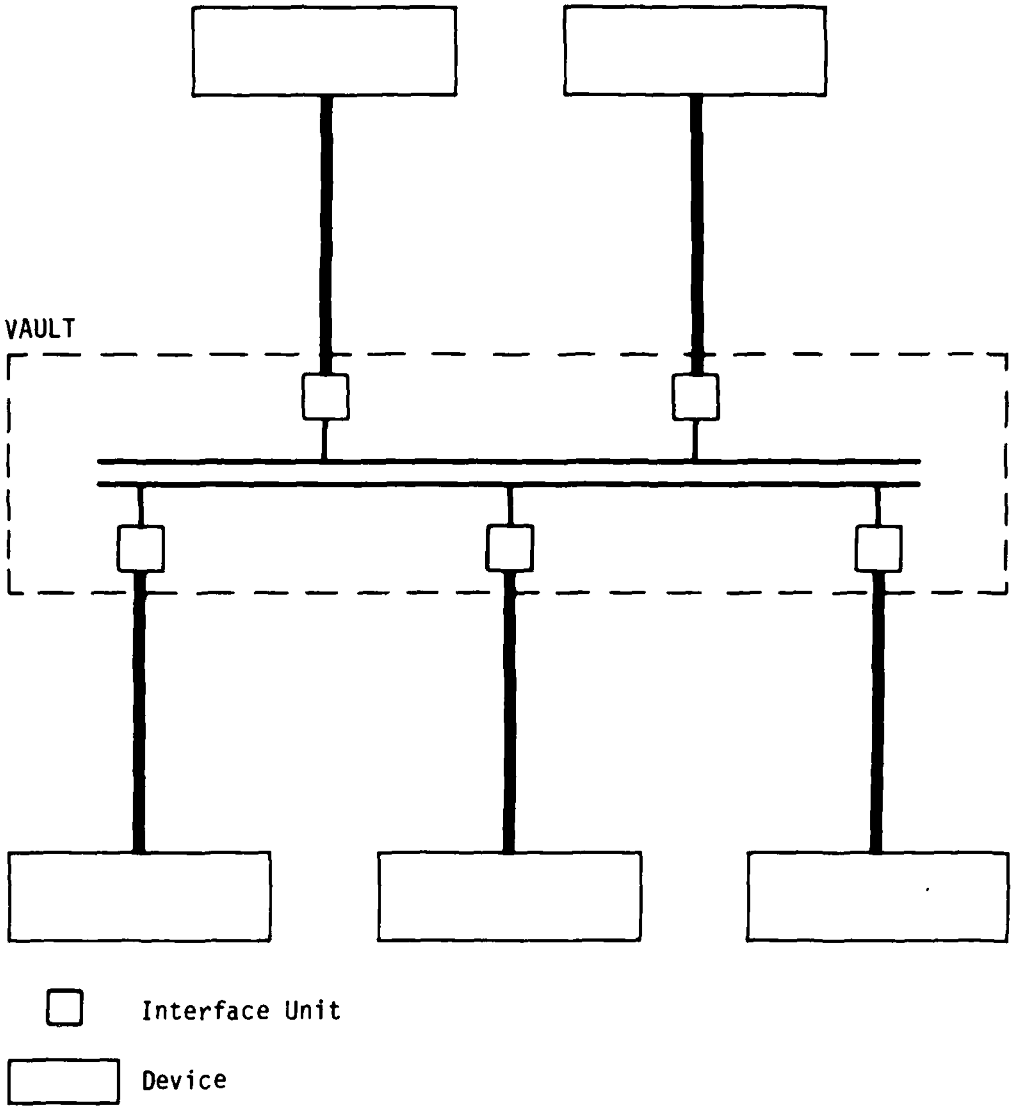


FIGURE 10
GEOGRAPHICALLY CONCENTRATED INTERFACE UNITS

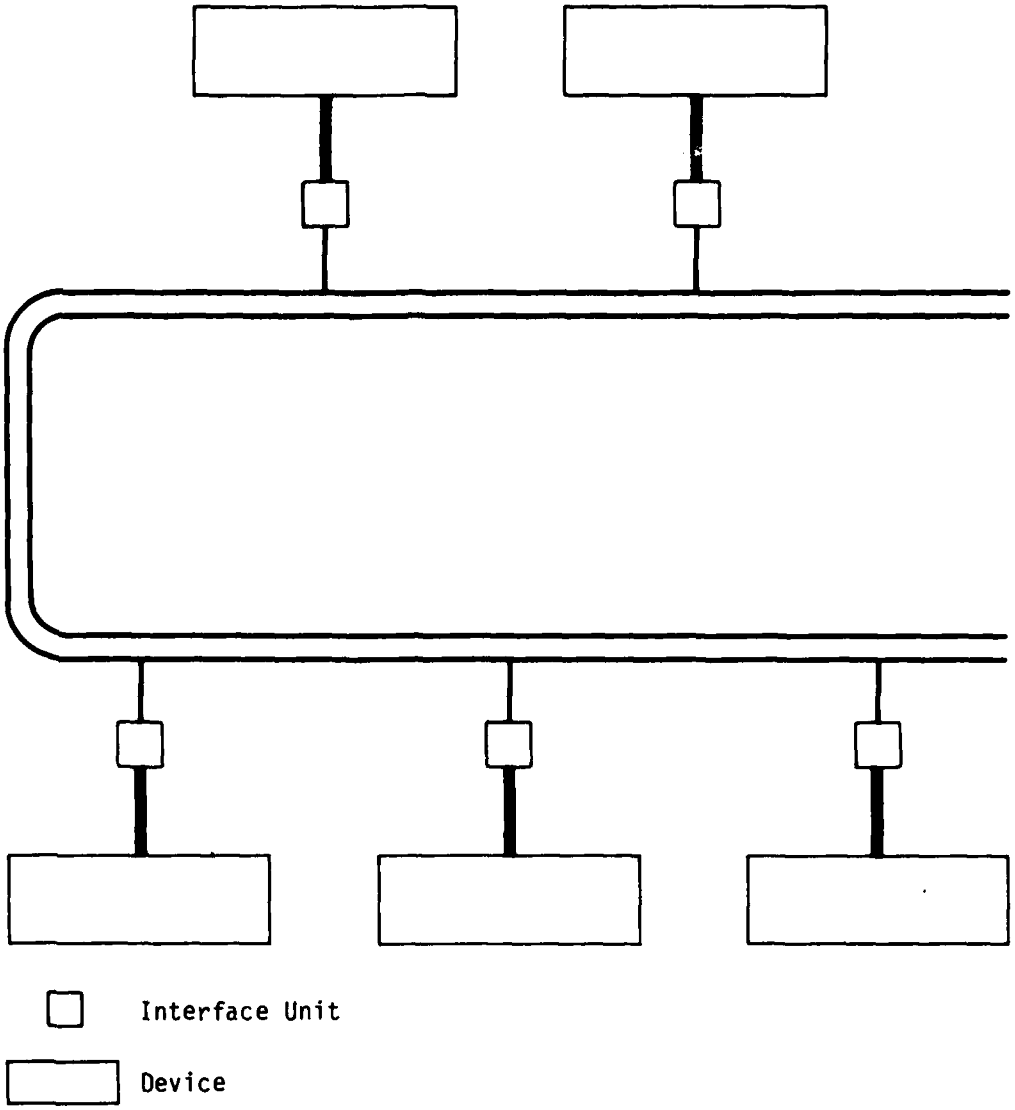


FIGURE 11
GEOGRAPHICALLY DISTRIBUTED INTERFACE UNITS

provides support for other software modules that actually implement message switching functions: routing of data from one communication line to another; checking the security level of a message to determine whether it may be routed to a second communication line; and implementing individual authentication in the form of a usercode and passwcrd. To insure that multi-level data is handled correctly, the message switch specific software modules must also be formally verified. As a part ot the verified software, monitoring and auditing functions may be employed to report attempted violations to a security officer for further action.

Separation of multi-level data streams may be enforced somewhat differently in the cable bus architecture. Since each interface unit terminates one or two communication lines, each unit may still be operated at a single level. This means that only address detection and security levei checking software must be verified. Address detection software monitors messages on the cable bus backbone to determine when one is addressed to the local interface unit. The amount of interface unit software to be verified is significantly less than that required in the message switch. While each interface unit only handles messages at a single security level, the cable backbone transports messages of varying levels. To separate data streams and provide protection against analysis of messages and traffic on the backbone, data encryption may be used to encode backbone message header and data information. While formal software verification and end-to-end data encryption technologies are both in early stages of development, it appears that during the next 3-5 year term, encryption hardware will be available to provide a greater level of confidence with higher overall performance than formal verification of kernel software. Integrated circuits, which operate at or above cable bus backbone speeds, are now available to implement the Data Encryption Standard.(13) Stronger encryption methods, such as the Public Key Distribution method, will enable dynamic key modification over much shorter timeframes. Thus, the cable bus trades formal verification of all software for partial verification and encryption to accomplish the same effective data security.

When encryption is used, it is unlikely that a message damaged in transit will decrypt to anything sensible, but if a legitimate address is produced, the verified address detection software is the first-level defense against misdelivery of messages. As secondary protections against misdelivery, a cyclic redundancy check (CRC) may be used and address and security level fields may be replicated in message headers. After decryption, the CRC can be recalculated and the contents of the replicated fields can be compared. In the unlikely event that a message is damaged and not detected through encryption, these additional mechanisms will insure against misrouting.

The cable bus must also implement monitoring and auditing software and hardware to report attempted security violations. In the cable bus, this takes the form of a continuity detection circuit and individual authentication software. The continuity circuit reports any changes in the continuity of the cable bus backbone and attached interface units. The continuity circuit may be implemented by measuring resistance, capacitance, or voltage changes on the cable bus backbone. The individual authentication software is driven by security level checking and login procedures which verify an individuals security level clearance.

Since all messages on the backbone will be encrypted at some point in the future it is not unreasonable to expect that interface units, placed in hardened enclosures, can be located outside of the vault. This arrangement is depicted in Figure 12.

The coaxial cable connecting the interface unit to the backbone is subject to physical attack. If this line is cut, it should be

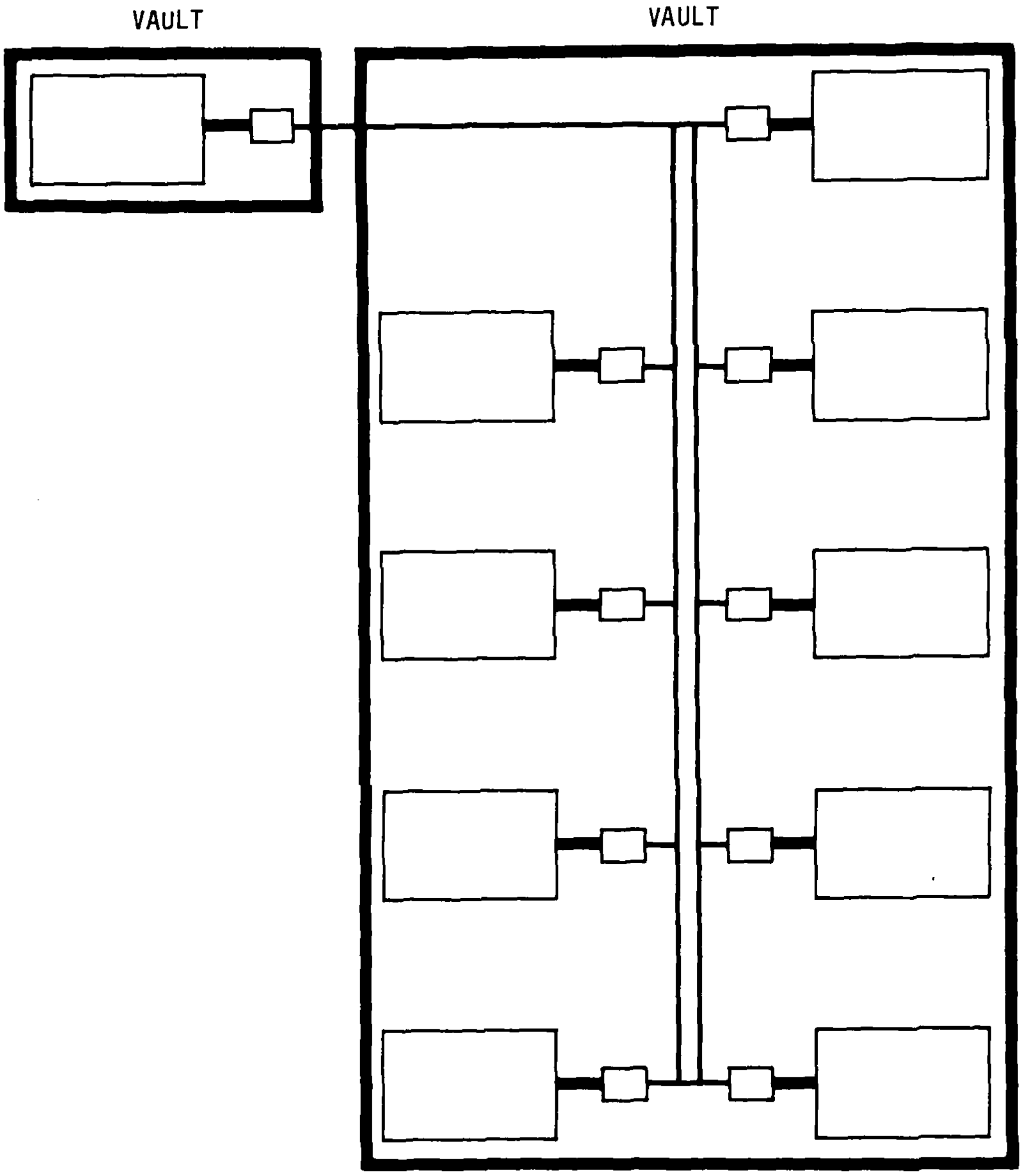


FIGURE 12
ALTERNATE SECURE CABLE BUS

made clear that only communications with the remote interface unit are disrupted. The main backbone continues to function.

4.1.2.3 The Environment of the Mid-80's. It is expected that sometime in the mid-1980's the introduction of multi-level secure mainframes based on approaches such as KVM will necessitate the requirement for multi-level security throughout the command center.

Since the message switch was already required to handle data streams at differing levels of security no further software functions are required.

If mainframes support multi-level data streams, cable bus interface units may at any one time have messages of varying security levels within them. This means that software other than address detection and security level validation modules must be verified. It is believea that the same sofware constructs used to provide verified separation of message switch data streams will be directly applicable to the interface unit.

It is felt that stronger encryption algorithms, using a Public Key algorithm will be available by this time in hardware form. The addition of stronger encryption methods will make it possible to perform user authentication via interface unit and, if it makes sense, a central username/password facility.

4.1.2.4 The Environment of the Late-80's. In the late 1980's the first components of the next generation of the WWMCCS Information System will begin to appear in operational command centers. These components will include newer WWMCCS mainframes as well as specialized functional elements such as teleconferencing and data base handling mainframes.

The loading induced by the secure handling from these systems will non-linearly increase the complexity of the message switch, resulting in steadily decreasing performance and the real possibility of overload failure, increased difficulty in maintenance of the software and the prospect of reverification of sizable amounts of software with the addition of each new device.

The multi-level secure interface unit, on the other hand, will not degrade as various components are added to the command center. Each additional component adds an interface unit to provide interface buffer capacity and protocol processing cycles. The increased load will in turn increase traffic on the cable bus backbone. However, the application of backbone technologies such as fiber optic bundles which support gigabit transfer rates should reasonably support command center requirements.

4.1.3 Security Conclusions

It is proper to point out here that until a detailed design, and in some cases actual implementation, is completed, some of the proposals and extrapolations presented here must necessarily be regarded as speculative. It is certainly the case that before recommending establishment of a cable bus system in a command center, careful experimentation and testing needs to be carried out. In fact, no local area network architecture can solve the data security problem in the general sense since it must interact with other elements (e.g., host computers with outdated operating systems, communications links to other systems) whose deficiencies will not disappear with the installation of a cable bus or other local communications system. The intention was to determine the affect of a cable bus on the overall system. It is believed that the foregoing discussion demonstrates that the cable bus will be at least as secure as the message switch architecture.

Thus, the cable bus may employ security measures at least as strong as the message switch. Furthermore, despite the physical limitations on the layout of the interface units, the cable bus provides a flexible interconnection architecture superior to the centrally located message switch mainframe.

4.2 Functionality

The centralized message switch and distributed cable bus architectures both provide local terminal-to-computer and computer-to-computer communications functions. A distinction between the architectures lies in a potentially larger cable bus bandwidth for supporting computer-to-computer communications. Coaxial cable bandwidths are bounded by the effective electrical bandwidth of the wire, 295M Hz in the MITRE system. Message switch bandwidths are bounded by the speed at which software can route messages from input interface to output interface. This bound is naturally message switcn hardware and software dependent. Effective cable bus and message switcn bandwidth is determined largely by buffering capacity and the arrival rate of "to-be-transmitted" messages. The cable bus buffering capacity is the the sum of the buffering capacity of all interface units. The message switch buffering capacity is the amount of main message switch memory dedicated to buffering. The number of cable bus processor cycles is the aggregate sum of all of the interface unit processor cycles. The message switch is limited to the number of message switch processor cycles.

The cable bus has an inherent broadcast capability not available in the message switch since all interface units "see" each message. The introduction of name tables implemented in an associative memory allows interface units to be treated as groups for broadcast transmissions. The message switch must replicate and transmit each broadcast message for each communication line which is to receive the information. This broadcast capability is potentially useful in implementations including teleconferencing.

The MITRE cable bus system is unique among cable bus systems in that it uses radio frequency modems to transport digital information via frequency division multiplexing. This offers the capability to transmit video and audio signals along with digital information on the same wire. Finally, this radio frequency transmission technique enables the creation of several logically distinct network systems using the same wire. The interface units can be "tuned" to the appropriate digital network. This lends itself to the transmission of information at differing security levels on different frequencies of the same cable.

4.3 Reliability

A system is reliable when it operates correctly for a length of time which is satisfactory to a large percentage of its users. Reliability is measured in terms of the average number hours between failure. Failures result from both software and hardware induced errors. System reliability is difficult, if not impossible, to measure, without direct experience with the system. Software failure rates are dependent on the quality of the initial implementation and the quality of software maintenance personnel; fixing one software failure may uncover or induce other failures. The reliability of hardware logic devices is also difficult to measure effectively. Computer systems use such a large number of components (from 100 to 1,000,000 devices) that device reliability must be at least as good as 10^{-6} failures per hour to be useful. It is not hard to test to this level. However, testing to 10^{-7} becomes very expensive in production numbers, primarily because of the length of the test. Newer more capable devices are being produced at such a rapid rate that it is not cost effective for manufacturers to demonstrate reliability of 10^{-6} to 10^{-7} failures per hour even though true reliability may be better. Thus, measurement of operational prototype system is required to determine reliablity. Since neither the NFE nor MITRE cable bus systems are operational, a message switch/cable bus reliability comparison is limited to a discussion of architectural weaknesses which would make one or the other architectures inherently less reliable.

The major architectural weakness of the message switch is clear: a single point message switch failure will take down the whole command center communications backbone. The message switch may also fail in overload situations by denying users service during what may be a crisis. Protection against these failures is usually implemented in the form of redundant systems. This is expensive since the message switch and its interconnections with other components must be replicated. Other approaches to nearly redundant systems are typified by Pluribus(14) and TELENET TP4000(15) which may offer significant improvement.

The cable bus architecture does not suffer from this particular malaay. Each of the interface units are passive devices whose only cable bus wide failure mode, continuous transmission, can be protected against by a simple, inexpensive watchdog timer which disables an interface unit if it has been continuously transmitting for an extendea period of time. The cable bus, because it uses a single wire, is also subject to overload. Here redundancy is also a solution, in the form of second, or third cable bus communication·backbones which are inexpensive. The MITRE system is particularly amenable to this because of its use of radio frequency to carry digital data. To move an interface unit from one "logical" cable backbone to another is a matter of tuning interface unit modems to a different channel.

Finally, the physical separation of interface units prevents a failure in one unit from affecting the operation of another unit. A bug in one interface unit software will not damage the tables or data of another interface unit. Thus, development of newer software may even reasonably take place alongside operational units. This is very difficult to do in the message switch environment.

4.4 Cost

An attempt has been made to determine, as accurately as possible, the predicted fixed and recurring costs of a cable bus system and a comparable message switch configuration. For the purposes of comparison, recurring costs are estimated over a 10 year life cycle. There are a number of overall system components which are common to both systems which are not included in this comparison such as communications lines and packet switch specific software.

4.4.1 Cable Bus

The cost of a cable bus is primarily incurred in the acquisition of its two major component groups: (1) the cable backbone, and (2) the interface units.

4.4.1.1 Cable Backbone. Since standard CATV components are used, the cost of a cable backbone is well known and minimal. The components are:

Headend	$1000
Cable at $500/mile	750
Splitters at $25	250
Taps at $9	270
Miscellaneous	230
Total	$2500

The $2500 cable backbone shown above is for a system covering approximately 1/2 mile. Components are included to branch the cable 10 times to conform to building architecture and tap into the cable at 30 locations; each tap can support one to four interface units Other one time purchase installation equipment (spectrum analyzer, field strength meter) may run as high as $5000.

4.4.1.2 Interface Units. The interface units utilized in the MITRE Washington test bed are hand-built, wire-wrapped circuit boards costing $1000 per unit. The units are constructed with older, inexpensive MOS technology and Motorola components. Commercially available units using newer Zilog components are being manufactured at a cost of $1500 per unit. Both units provide the same capabilities, and consequently, are not capable of supporting WWMCCS Command Center security and offloaded protocol requirements.

From the work described in this report, a set of components for an interface unit capable of supporting WWMCCS command center communication requirements has evolved. Estimated prices for those components are shown below:

	1979	1982
64K bytes RAM/ROM	$550	$250
RF Modems	300	300
Power Supply	100	100
Name Table	150	25
CPU	120	100
Encryption	100	80
DMA Interface Circuits	80	80
Cable Transmit Watchdog	25	10
Clock Timer	20	20
Misc. (Cabinet, Connectors)	355	235
Totals	1800	1200

The estimates shown in the first column above give 1979 costs for the commercial construction of a WWMCCS capable bus interface unit. The estimates were obtained by taking the list prices for existing integrated circuits and adjusting the value upward to account for assembly of the component into the interface unit. The prices include no figures for profit, overhead, or recovery of development costs.

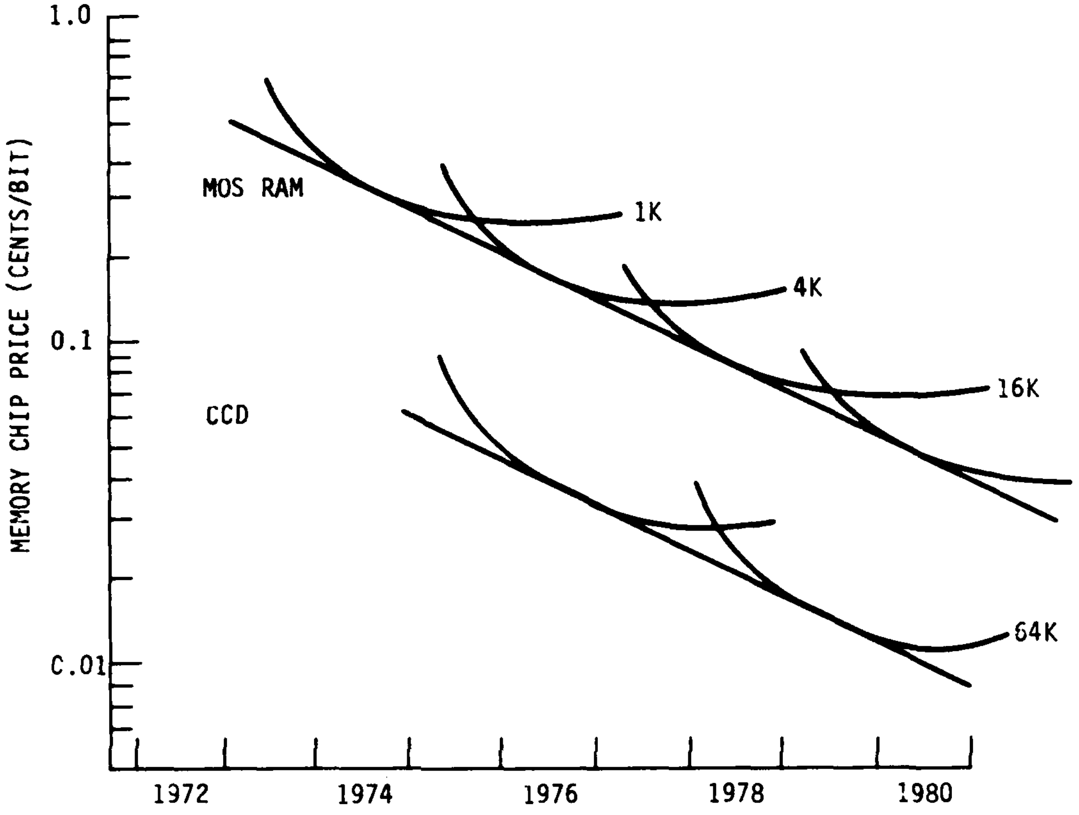
The price of memory represents almost 1/3 of the total cost of the unit. The memory was sized at 48K bytes of RAM and 16K bytes of EPROM. It is believed that this is significantly more memory than will actually be needed.

The second column above gives an estimate of the 1982 commercial cost of the same interface unit, taking into account the kinds of decreases in price that have been seen over the past three years. Memory in particular has been rapidly decreasing in cost in recent years. The price decrease of RAM and CCD memory circuits over the past seven years in cents/bit is shown in Figure 13. The price per bit in 1976 was approximately 0.2 cents/bit. The price per bit at the beginning of 1979 is shown as 0.07 cents per bit. Now, towards the end of 1979, the actual cost per bit is closer to 0.035. Digital Equipment Corporation, for example, has decreased its price for an LSI-11 64K byte RAM two times this year alone, from $1149 to $774 to $582. If the price trends shown in Figure 13 hold as expected, a cost of 0.02 cents/bit can be projected for mid-1981, resulting in the interface unit memory estimate of approximately $250.

The power supply also represents a significant element of the cost. This cost could be reduced significantly by integrating the interface unit circuit boards directly into the chassis of newer terminals, thereby allowing for a sharing of the terminal power supply.

Although the advantages of employing RF modems discussed elsewhere would be sacrificed, base-band signalling techniques could be used for which nothing comparable to a modem is required. The elimination of the RF modem could reduce the 1982 estimate by 25 per cent.

Based on the estimates outlined above, the cost of a cable bus system is largely dependent upon the number of its interface units. Each interface unit will support one high speed computer interface or from one to four terminal interfaces, depending on the type and complexity of the terminal. For the purposes of this report, it is estimated than an average of two terminals will be supported by each interface unit. The cable bus system fixed cost, C, as a function of the number of attached computers and terminals is:



Source: Dave House, "CCD vs. RAM for Bulk Storage Applications," Digest of Papers, COMPCON Spring 76 San Francisco, Calif., p. 60.

FIGURE 13
MEMORY COST—CENTS/BIT

C(#Comp, #Term) = #Comp * $1800 + #Term * $900 + $2500

Recurring cable bus maintenance costs are difficult to estimate. The maintenance policy for interface units is similar to the typical terminal maintenance procedure which is to handle malfunctions by swapping spare circuit boards into the unit until operation is restored. Therefore, maintenance costs are estimated by supplying two spare interface units for each cable bus installation and providing a service for the repair of damaged interface units. It is estimated that each interface unit will require repair five times over a ten year period and that each repair will cost approximately four percent of the unit, or $75. Thus, a recurring cable bus cost of $12,600 for each cable bus system is estimated.

4.4.2 Message Switch Cost

The cost of a message switch is incurred in the acquisition of computer interfacing hardware, terminal interfacing hardware, and a cpu to support movement of data between interface communication lines. Message switch cost is estimated by extracting message switcning hardware costs from NFE cost estimates; ignored here are the costs associated solely with its front end functions. This estimate includes NFE computer interfaces, NFE terminal interfaces and a percentage of NFE cpu costs.

The cost of a network front end implemented on six alternative hardware bases is estimated in a Digital Technology Incorporated (DTI) report.(17) The average price of these systems is $95,130. The average price of the terminal interfacing hardware is $18,362. This hardware interfaces 32 terminal lines to the NFE. From the DTI report an estimate of the memory required to support 32 terminals may be derived:

Code	15.3K bytes
Buffers	167.2K bytes
Total	182.5K bytes

The total amount of memory recommended is 512K bytes. The terminal handling memory may then be roughly estimated at 35 per cent of the overall NFE memory requirements. It is also estimated that as many as half of the NFE processing cycles will be exposed at any one time to handling character-at-a-time or block-at-a-time interrupts from directly connected front end terminals. Based on these admittedly rough estimates and direct experience with NFE measurements, a conservative measure of 25 per cent of the NFE as a whole may be attributed to terminal handling. Thus, 25 per cent of the NFE cpu cost, $19,192, may be directly attributed to message switching.

The cost of providing a high-speed direct memory access interface between the message switch and computer components is estimated from the DTI report at $2000 per interface.

The total fixed message switch cost is then estimated as:

$$\$19{,}192(\mathrm{cpu}) + \$18{,}362(\mathrm{term}) + \$2000 * 7(\mathrm{comp}) = \$51{,}554$$

The average recurring monthly maintenance for the six systems described in the DTI report is $941. Taking the estimated 25 per cent yields a monthly message switch maintenance figure of $235.25.

4.4.3 Cable Bus - Message Switch Cost Comparison

The comparison then between the projected cable bus costs and the derived message switching costs is relatively straightforward.

The cost of a command center local area network with 32 terminals and 8 larger computer components may be computed for the cable bus and the message switch.

Cable bus costs are computed via the cable bus cost formula:

C(8,32) = 8*$1800 + 32*$900 + $2500 = $45,700

Message switch costs were derived above:

$19,192 + $18,362 + $14,000 = $51,554

Added to these fixed costs are recurring maintenance expenses over a 10 year life cycle.

Cable bus:

Spare interface units		$3600
Repair service		$9000
	Total	$12,600

Message Switch:

$235.25 * 120mo = $28,230

Given that WWMCCS installs these systems at the 23 sites where NFE installations are planned, the comparative costs for the message switcn and cable bus are:

Cable bus:

Fixed Cost at $45,700		$1,051,100
Recurring at $12,600		$ 289,800
	Total	$1,340,900

Message Switcn:

Fixed Cost at $51,554		$1,185,742
Recurring at $28,230		$ 649,290
	Total	$1,835,032

Thus the cable bus-message switch price differential is significant in that the cable bus could save as much as 27 per cent over a centralized connection architecture.

When reliability issues are factored into the cost in terms of duplicating message switches at each site versus duplicating interface units to critical nodes and duplicating cable backbones, the savings almost double:

Cable bus:
(32 interface units and two backbones)

Fixed Cost at 62,600 * 23 sites = $1,439,800
Recurring at $15,600 * 23 sites =$ 358,800
Total $1,798,600

Message Switch:
(2 message switches per site)

Fixed Cost at $103,108*23 sites = $2,271,484
Recurring at 56,460 * 23 sites = $1,298,580
Total $3,670,064

In this case, the cable bus would save almost 51 per cent.

4.5 Flexibility

A cable bus is an extremely flexible system. A message switch is a relatively fixed architecture. The physical modularity of the interface units is the key to cable bus flexibility. This physical modularity makes the cable bus very amenable to growth or shrinkage in communication requirements. As new devices are added to the cable bus, interface units are correspondingly added to provide incremental increases in buffering capacity and protocol processing capabilities. NFE incremental increases in buffering capacity and processing capability tend to be both large and expensive. This physical modularity also has the advantage that new device types may be interfaced to the cable bus without affecting the operation of existing software in other devices. With message switch architectures, NFE software complexity increases non-linearly with the addition of each new device type.

The degree of software complexity is somewhat smaller in interface units because they are "closer" to the interfaced device. Assumptions, untenable in the message switch, can be made based on a specific "knowledge" of the user component being interfaced. These simplifying assumptions lessen the amount of generality and defensiveness that must be programmed into the interface unit in comparison to the kind of validity checking software that must be programmed into the message switch.

5.0 DEVELOPMENT ACTIVITY

The purpose of the development activity was to provide starting place for experimental activity performance and protocol investigations. This included four tasks:

- Installation of a MITRE Bedford developed cable bus system with six interface units in the MITRE Washington facility.
- Installation of a cable bus gateway minicomputer to the ARPA network.
- Installation of an initial set of command center protocols on an LSI-11 cable bus host and the PDP-11/70 gateway.
- Construction of medium speed interfaces between the LSI-11 and the cable bus and the PDP-11/70 and the cable bus.

The resulting test bed is shown in Figure 14. Each of the tasks are discussed below.

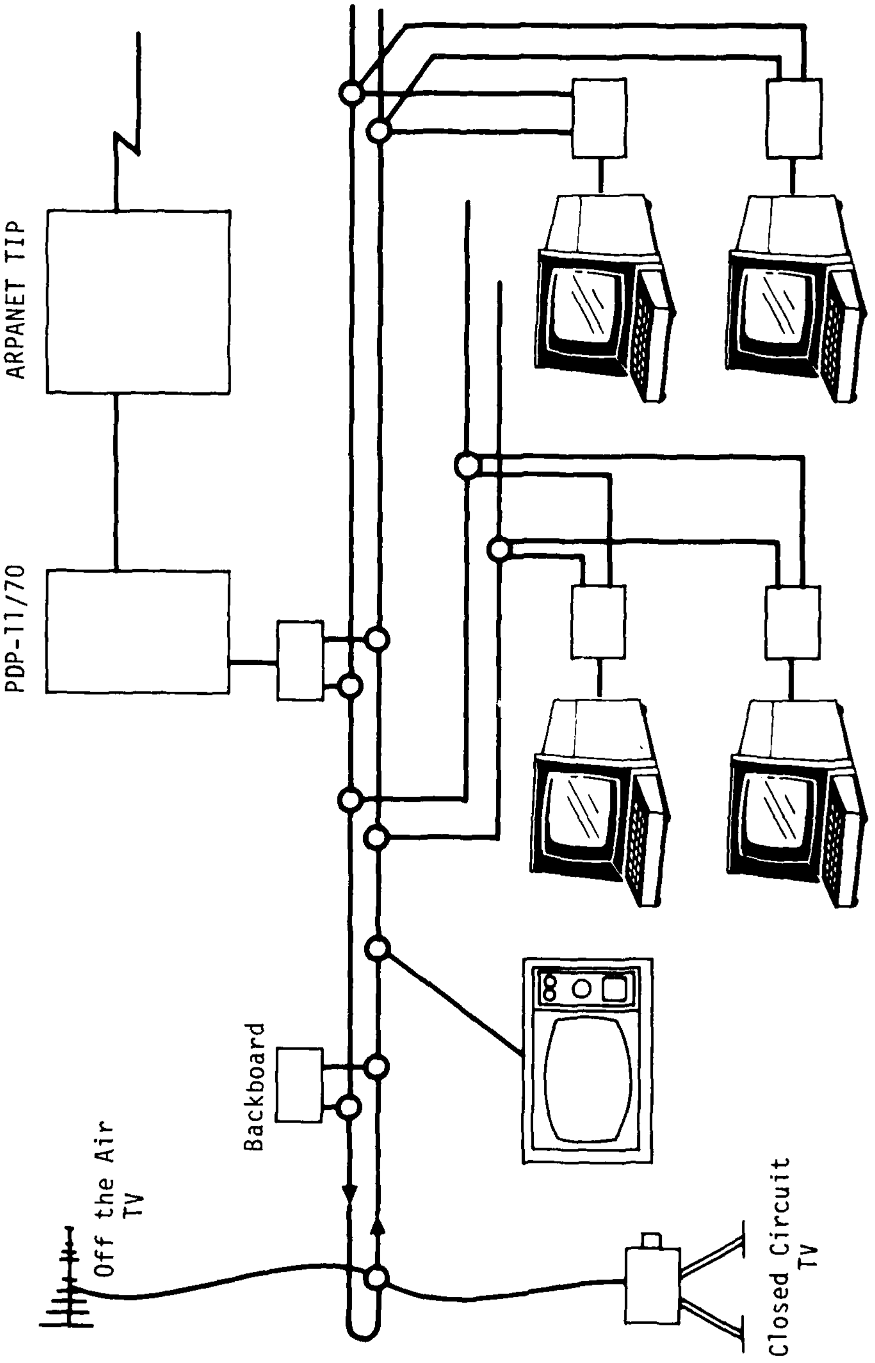
5.1 Cable Installation

The cable bus consists of coaxial cable, amplifiers, taps, splitters, and microprocessor-based interface units.

The coaxial cable was installed in the first floor and basement levels of the MITRE Westgate building. The total length of the cable is approximately 3000 feet. The layout of the cable and the location of splitters, taps, and amplifiers is shown in Figure 15.

System integrity was verified by testing cable continuity to detect broken or missing connections. This was followed by a signal level alignment using a spectrum analyzer and RF signal level meter.

Operational integrity was verified by attaching a backboard interface unit and a tester interface unit to the cable. The tester interface unit sent a continuous stream of messages to the backboard which reflected them back to the tester. This series of tests verified that system delay and noise were within specified tolerances. Since tester software may be temporarily installed in any interface


ARPANET TIP
PDP-11/70
Backboard
Off the Air TV
Closed Circuit TV

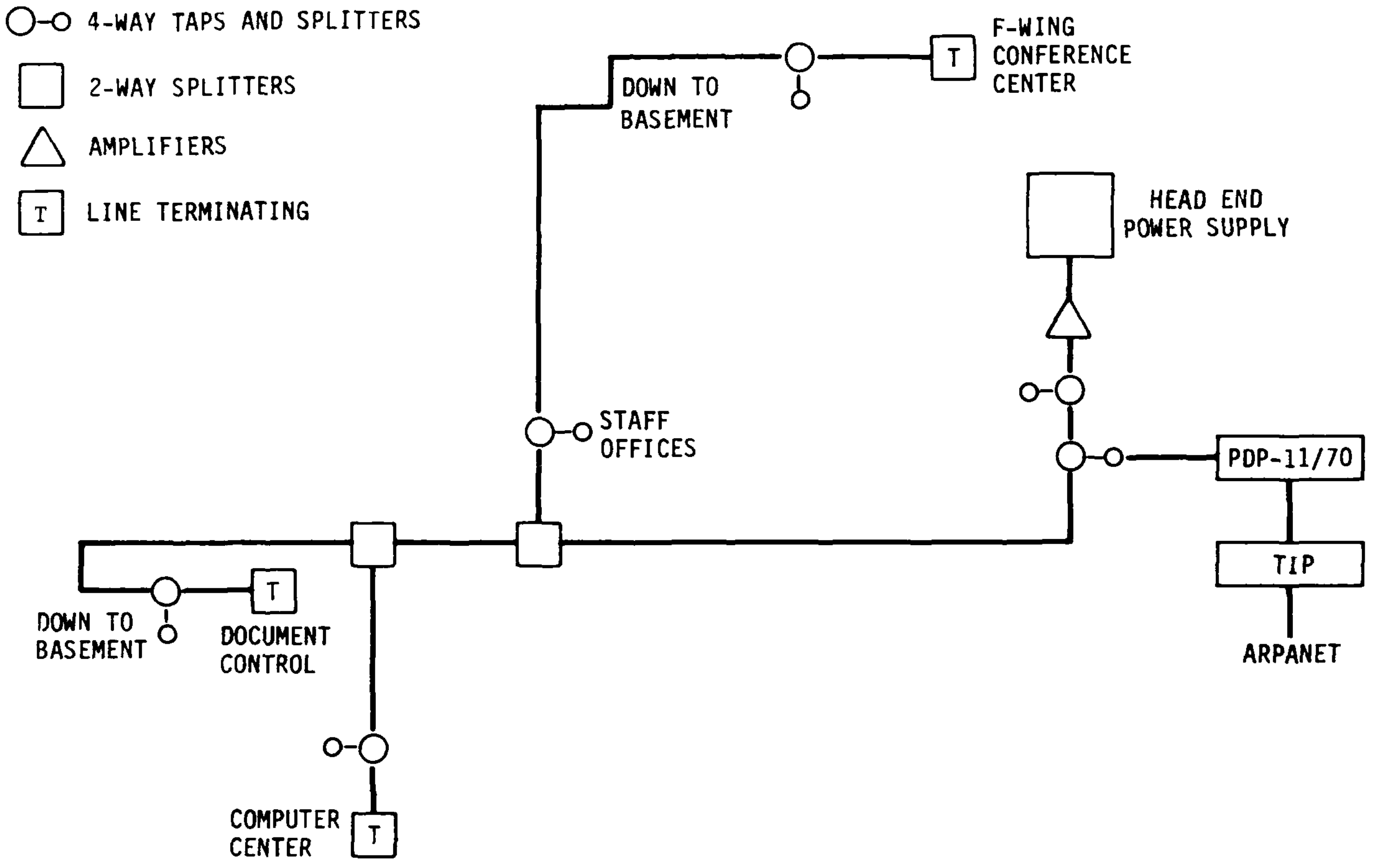


FIGURE 15
MITRE WASHINGTON LAYOUT

unit, these tests were also used to verify the correct operation of each of the interface units.

5.2 Gateway Installation

A PDP-11/70 minicomputer system was used as a gateway between the ARPA network and the cable bus. The PDP-11/70 system consists of:

- PDP-11/70 Central Processor
- 128K words of main memory
- 67 Million words of disk storage
- 9-track tape drive
- 16-line terminal multiplexor interface
- ARPA network interface - IMP11-A
- byte-at-a-time DR11-C interface

The UNIX operating system was installed to support network protocol software, terminal access and program development.

Finally, to provide for fine-grained timings, a high-speed programmable clock was installed. Software based on the clock was developed to enable the timestamping of messages and to measure the utilization of the system under load conditions.

5.3 Command Center Protocol Installation

The Transmission Control Protocol (TCP) was selected as an initial baseline to support internetting between the cable bus and the ARPA network. This selection was made for several reasons:

- The TCP meets the Command Center requirement for internetwork operations.
- TCP meets the Command Center requirement for a robust, highly reliable operation.

- A variant of TCP is being used for the DCA Autodin II Network, and is consequently being implemented in the WWMCCS Network Front End.

- TCP has been adopted as a Defense Department Standard.

There have been several versions of TCP. The original ideas were published in 1974, and the first specification, now known as Version 1, was produced later that year. In 1976, a TCP specification was produced for AUTODIN II. In connection with ARPA internetwork research, the original Version 1 has been through several revisions, resulting in Version 2 in 1977, Version 3 in 1978 and most recently Version 4. In Version 4,(16) TCP has been split into two distinct protocols; TCP itself and a lower-level Internet Protocol (IP).(18)

TCP manages the opening and closing of virtual connections via a three-way handshaking procedure and provides for a mechanism whereby processes exchange letters with each other. Letters may be broken up into segments. The basic function of regulating the flow of segments between processes resides in the TCP module. In addition to flow control, TCP performs sequencing, duplicate detection, end-to-end acknowledgement and error control services.

TCP depends upon the Internet Protocol to handle certain functions including addressing, fragmentation, and reassembly. The IP embeds TCP segments into Internet segments. The Internet Protocol handles the delivery of the Internet segments from source to destination through a system of interconnected networks.

To construct the test-bed as expeditiously as possible, existing implementations of TCP Version 4 and the Internet protocol were obtained for the PDP-11/70 and the LSI-11. The PDP-11/70 TCP and IP were obtained from Bolt Beranek and Newman, Incorporated which had implemented TCP and IP as part of the Exploratory Data Network (EDN) for DCA. The LSI-11 TCP and IP implementations for the MOS operating system were obtained from the Stanford Research Institute which is developing a microprocessor-based network access host as part of an ARPA Packet Radio networking project. The initial layering of protocols for the test-bed cable bus system was shown in Figure 4.

5.4 PDP-11/LSI-11 Cable Interfaces

The cable bus interface units were originally constructed to support RS-232C terminal access to the cable bus. RS-232C restricts data transfer speed between the interface unit and the user device to 19.2K bps.

To support realistic cable bus computer-to-computer data transfer speeds a higher speed interface was required. The RS-232C interfaces were replaced in two interface units with faster 8-bit parallel interfaces. The parallel interfaces have a theoretical maximum operating speed of 200K bps. Since the transfer speed of the cable is 307.2K bps, this speed was thought to be sufficient to support experimental measurements.

The parallel interfaces also had the secondary benefit of providing control handshaking signals to govern the flow of data between the interface units and the host computers. These handshaking signals provided a rudimentary form of flow control needed to keep one side of the interface from sending data too fast for the other side to receive.

6.0 EXPERIMENTAL ACTIVITY

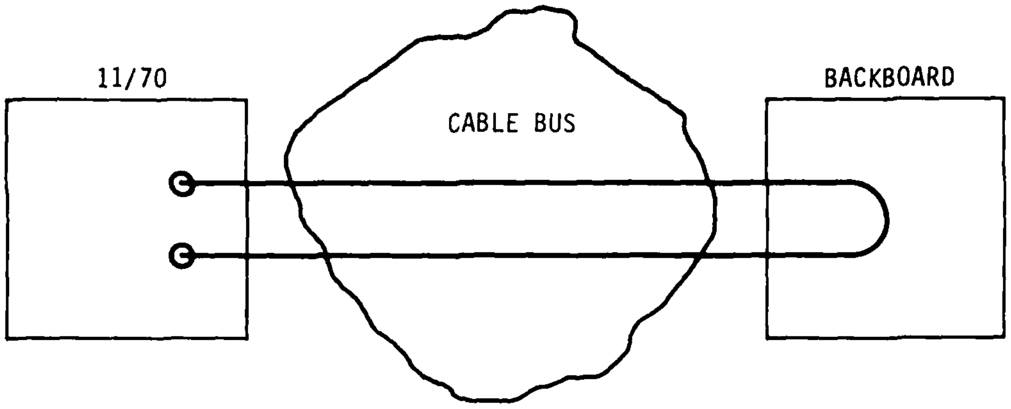
The purpose of the experimental activity was to use the test bed facility to gain hands-on insight into the existing capabilities and performance of the MITRE cable bus system and to evolve a set of protocols to support command center communication requirements.

Four experiments were performed. The first experiment met a need to determine the effective throughput and bandwidth of the cable bus backbone. The second experiment measured the throughput and bandwidth of a virtual circuit loop implemented by user processes and TCPs on the PDP-11/70 and LSI-11. The results of this experiment served as a catalyst which lead to the restructuring of the mechanisms within the initial protocol layering. The third experiment measured the throughput and bandwidth of the new restructured protocol implementations. The fourth experiment verified the capability of the restructured protocol to interface with the ARPA network protocols.

6.1 Experiment 1: Cable Bus Backbone Performance

This experiment met a need to determine the effective throughput of the backbone cable bus system by measuring the maximum number of messages per second processed by an interface unit and the effective bandwidth of the backbone cable bus system by measuring the maximum number of bits per second transferred through the cable bus.

The experiment, shown pictorially in Figure 16 below, relied on placing the value of a 100K Hz clock (timestamping) in an outgoing stream of messages from the PDP-11/70 to a backboard interface unit which "reflected" each message back to the PDP-11/70 where it was again timestamped. Timestamping took place at the very lowest levels of the PDP-11/70 operating system softwares so as to eliminate the effect of operating system performance on the measurements. The outgoing and incoming timestamps were compared to synthesize a round-trip measurement.



O - TIMESTAMP

- Throughput - 337 MSGS/SEC
- Bandwidth - 36K BPS
- Poor Performance
  - Older Slow BIUS - MOS 6502
  - Chatacter-at-a-time Interface
  - Small Maximum Packet Size

FIGURE 16
CABLE BUS BACKBONE EXPERIMENT

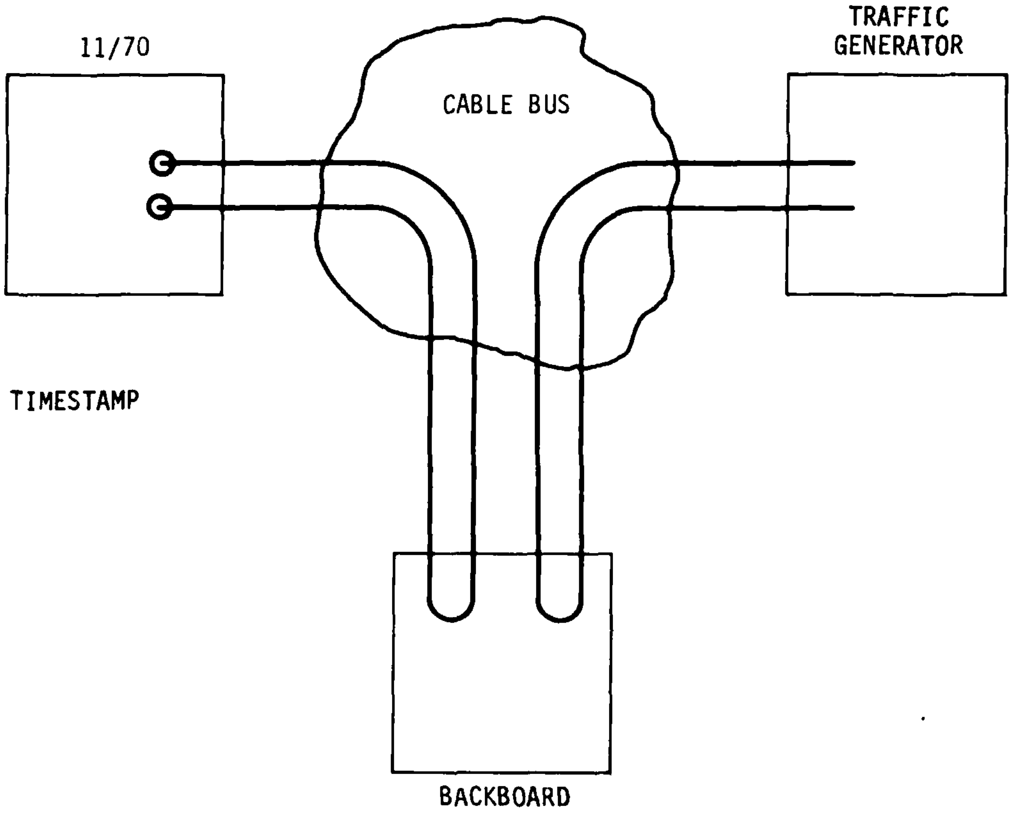
The message size was varied in order to measure throughput and bandwidth. The sizes of the messages were set at the smallest length possible (12 bytes) to measure throughput and set to the largest possible (128 bytes) to measure bandwidth. The throughput was measured at 337 message per second. This throughput is rated as acceptable. The bandwidth was measured at 36K bps. This is rated as poor performance.

The performance was primarily limited by the speed of the microprocessor, the interrupt-per-character PDP-11/70 interface unit interface, and the small maximum packet size (128 bytes/packet). All of these factors will be improved in newer versions of the interface unit. These results graphically demonstrate that cable bus systems in their present state of development will not support command center requirements.

Experiment 1 was performed with no background traffic on the cable bus. To determine the degradation in message flow induced by background traffic, a traffic generator interface unit was added to the experiment. The traffic generator interface unit simply sent a continuous, uncontrolled stream of messages to the backboard interface unit and ignored messages received from the backboard. The traffic generator was allowed to consume as much of the cable bus resources as it could, but it was not capable of saturating the cable bus. The experiment, shown in Figure 17, resulted in no significant changes in the previous throughput or bandwidth measurements.

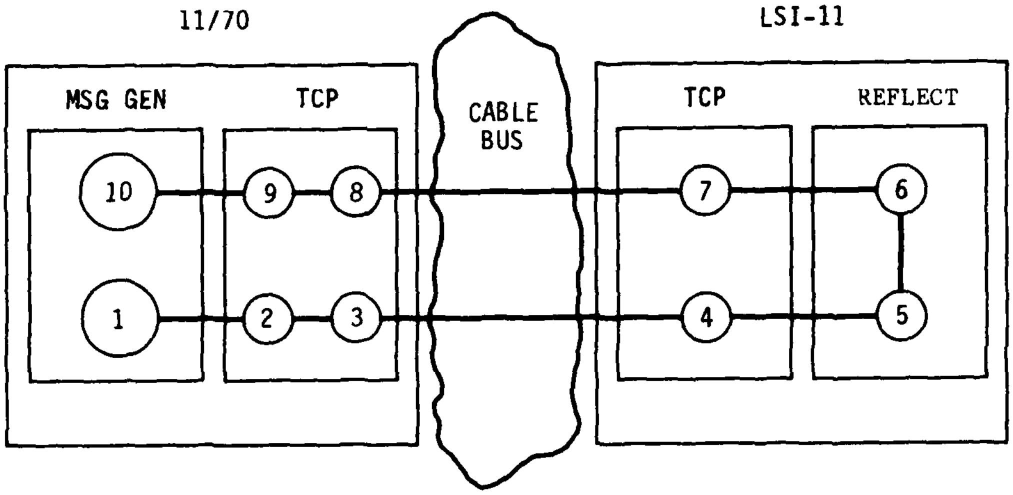
6.2 Experiment 2: Standard TCP Performance

Given that the first experiment provided a measurement of the basic speed of the cable bus backbone, it was of interest to measure the throughput and bandwidth of TCP-based communications over the cable. To do this, the second experiment expanded on the concept of timestamping minimum and maximum sized messages in transit around a loop. The loop, shown in Figure 18, was comprised of four processes:



NO SIGNIFICANT PERFORMANCE CHANGE
WITH ADDITIONAL TRAFFIC

FIGURE 17
MODIFIED CABLE BUS BACKBONE EXPERIMENT



- Unix TCP-4 slow - 6.4K bps

  Unix Ipc
  TCP - Device Driver Interface
  Character-at-a-time

- MOS TCP-4 Respectable - 25K bps

FIGURE 18
TCP CABLE BUS EXPERIMENT

two located in the PDP-11/70 and two located in the LSI-11. Each message was timestamped at ten locations spaced unevenly around the loop. The reason for the asymmetry of the timestamp locations was based on the difference in the available timestamp clock rates between the PDP-11/70 (100K Hz) and the LSI-11 (60 Hz). The LSI-11 clock is so slow that meaningful timestamp values are not obtained unless the timestamp locations are spaced far enough apart to let the clock "tick".

The table below describes the location of each timestamp.

Timestamp No.	Timestamped when:
1	Message generated
2	Message received from the MSG Generator
3	Message given to cable device driver for transmission
4	Message received from the cable
5	Message received from LSI-11 TCP
6	Message sent to LSI-11 TCP
7	Message transmitted to cable
8	Message received from cable device drivers
9	Message transmitted to user
10	Message received from PDP-1/70 TCP

By comparing the timestamp values of different timestamp locations, one can get an indication of the overall bandwidth and throughput as well as a detailed breakdown on where that time is being spent.

The results of this experiment are summarized in Table III. Considering the 36K bps speed of the communications backbone, the 25K bps throughput of the LSI-11 system is respectable. However, the 6.4K bps speed of the PDP-11/70 TCP is poor. This poor performance is due to 11/70 TCP implementation and TCP protocol overhead factors.

6.2.1 TCP Implementation

The TCP implementation on the PDP-11/70 makes heavy use of UNIX inter-process communication primitives, UNIX pipes. Use of these primitives represents almost one-third of the total packet

TABLE III

TCP PERFORMANCE MEASUREMENTS

EXPERIMENT 2 RESULTS

0 - 10K SAMPLE PACKETS

AVERAGE TIME BETWEEN POINTS	MSECS/BYTE
1-2	5.602
2-3	3.697
-	-
4-5	0.628
5-6	0.000
6-7	0.613
-	-
8-9	0.105
9-10	0.487

AVG. FLOW RATE	MSEC/BYTE	MSGS/SEC	BPS(IN/OUT)
11/70	2.472	3.16	6472 (3236/3236)
LSI-11	0.621	12.58	25K(12.5K/12.5K)

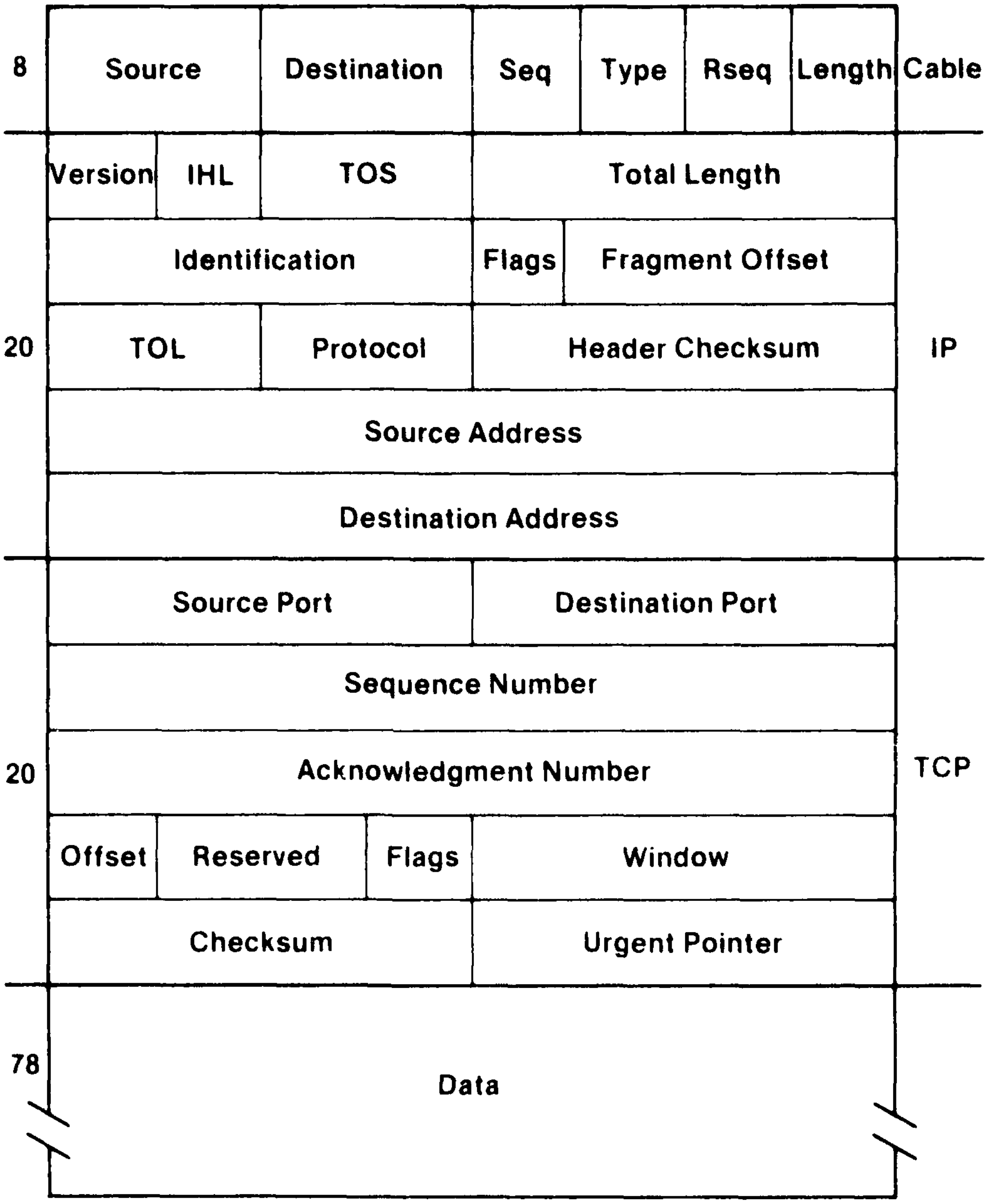
transmission time. This time is almost equal to the amount of time spent sending a packet to the cable, looping it through the LSI-11 TCP and user process and receiving it again by the 11/70 TCP. Considering that the 11/70 processor is ten times faster than the LSI-11, there is clearly room for improvement. An effort of several months would be required to change the basic transport architectures of the UNIX pipe mechanism. An unscheduled task of this magnitude was not possible given the available resources in FY 79.

6.2.2 TCP Protocol Overhead

The amount of per-message overhead required to transfer a message was the second major cause of the poor throughput. A message, as shown in Figure 19, requires 8 bytes of local cable bus header, 20 bytes of Internet protocol overhead, and 20 bytes of TCP protocol overhead. This means that each full message (maximum size 128 bytes) is composed of 50 bytes of header and 78 bytes of data. Thus 39 per cent of each messager is header. This is significant when compared with ARPA network overheads of 0.8 per cent per maximum size message. This overhead can be reduced in two ways: 1) by increasing the maximum size of the message, and 2) by decreasing the number of overhead bytes.

Increasing the maximum message size requires an increase in the buffering capacity of the interface unit and/or an increase in the basic transfer speed of the coaxial cable. Increased buffer capacity is a matter of enlarging the memory in the interface unit. Increasing the cable transfer speed requires a modification to the interface unit modem and an increase in the basic operating speed of the interface unit microprocessor. Thus, increasing the maximum message size requires a moderate amount of work. Unfortunately, time and personnel constraints prevented work in this area.

Reducing per message overhead by decreasing the number of overhead bytes is somewhat simpler. The reductions lead to the protocol



8	Source	Destination	Seq	Type	Rseq	Length	Cable
20	Version	IHL	TOS	Total Length			IP
	Identification		Flags	Fragment Offset			
	TOL	Protocol	Header Checksum				
	Source Address						
	Destination Address						
20	Source Port		Destination Port				TCP
	Sequence Number						
	Acknowledgment Number						
	Offset	Reserved	Flags	Window			
	Checksum		Urgent Pointer				
78	Data						

FIGURE 19
CABLE BUS TCP HEADER

architecture described in Section 7. The resulting message fc served as a basis for Experiment 3.

6.3 Experiment 3: Modified TCP Performance

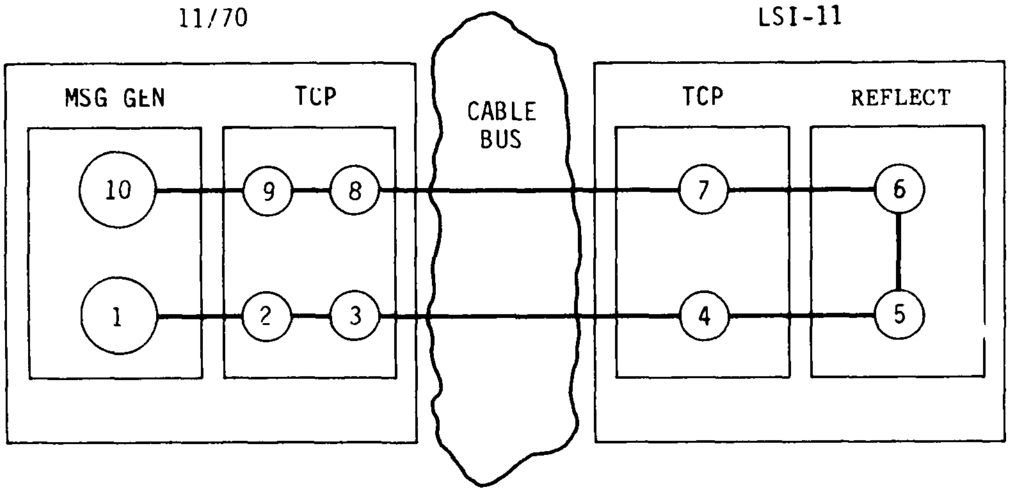
Experiment 3 was a repetition of Experiment 2 as shown in Fi 20, with the message format shown in Figure 21. The purpose of experiment was to insure that no significant increases in per-mes processing time had been introduced by the reprogramming effort. results of the experiment are summarized in Table IV below. decrease in header size resulted in an expected corιespon increase in the bandwidth of the PDP-11/70 (from 6.4K bps 8.3K bps) and LSI-11 (from 25K bps to 28.9K bps).

6.4 Message Switch - Cable Bus Performance Comparison

At present, the test bed cable bus performance compares unfa ably to the message switch. This is due to the fact that both message switch and the cable bus do not accurately represent the formance capabilities of the cable bus and the Network Front architectures. Nevertheless, the performance measurements are indication of what exists today.

NFE performance measurements were made for an early version the NFE called the Experimental Network Front End (ENFE). T measurements were used in comparisons with the cable bus. Per mance measurements of a newer version of the NFE, called the Int Network Front End were not available. Both the Interim and Exp mental Network Front End versions are supported by a general pur operating system, UNIX. The general purpose nature of UNIX seve limits data communications throughput. A new operating system under construction may increase NFE throughput by as much as a fa of three to five (for a non-secure version).[19]

The NFE performance measurements are described in detail in <u>et al</u>.[6] In that report, the NFE was measured as havin



- Unix TCP-4 slow - 8.3K bps

  Unix Ipc
  TCP - Device Driver Interface
  Character-at-a-time

- MOS TCP-4 Respectable - 28.9K bps

FIGURE 20
MODIFIED TCP CABLE BUS EXPERIMENT

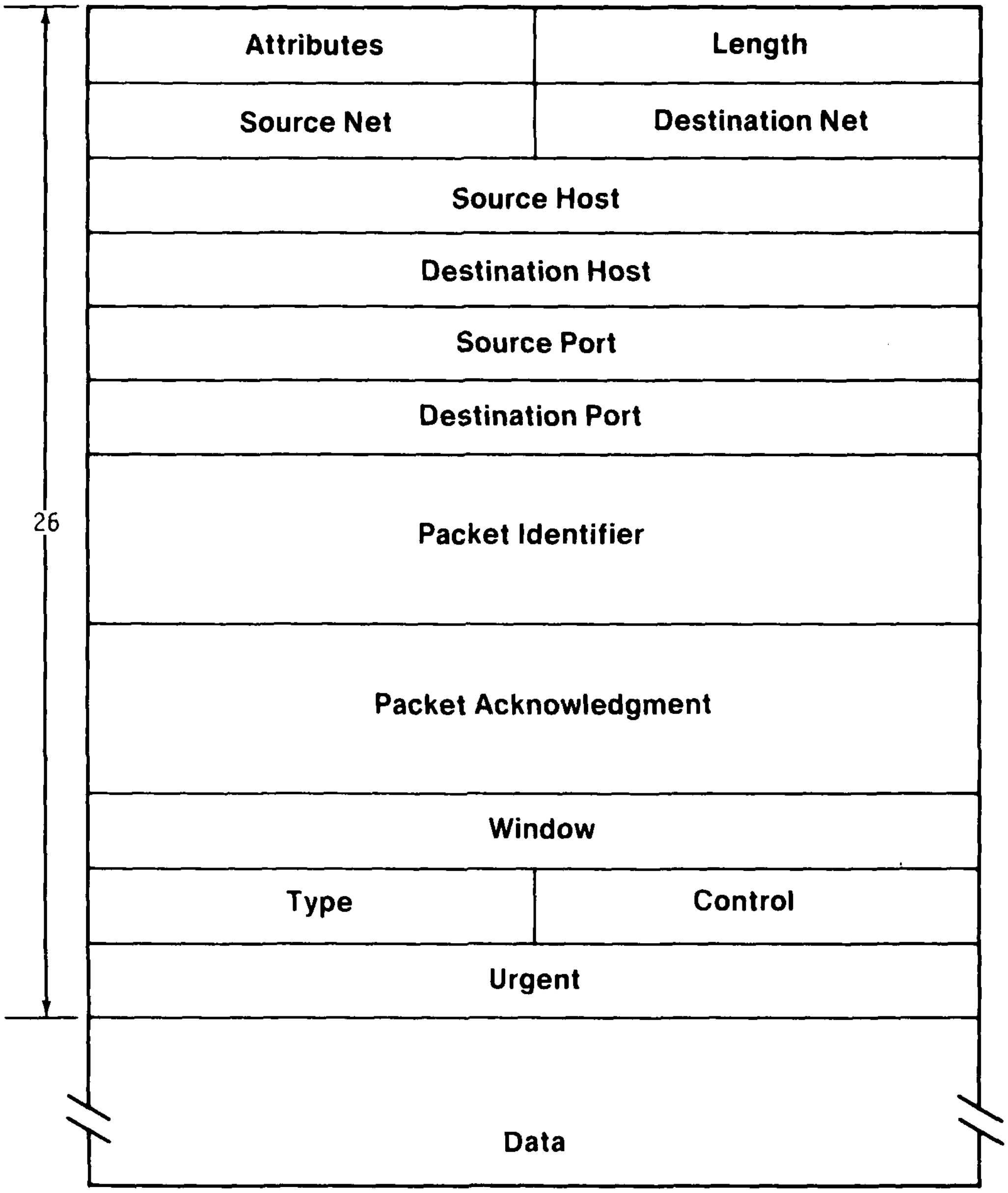


FIGURE 21
MODIFIED CABLE BUS TCP HEADER

TABLE IV
MODIFIED TCP PERFORMANCE MEASUREMENTS
EXPERIMENT 3 RESULTS

0 - 10K SAMPLE PACKETS

AVERAGE TIME BETWEEN POINTS	MSECS/BYTE
1-2	3.774
2-3	2.970
-	-
4-5	0.568
5-6	0.000
6-7	0.537
-	-
8-9	0.176
9-10	0.558

AVG. FLOW RATE	MSEC/BYTE	MSGS/SEC	BPS(IN/OUT)
11/70	1.869	4.18	8360 (4280/4280)
LSI-11	0.552	14.13	28.9K(14.47K/14.47K)

average maximum bandwidth of 33.5K bps. If the highest throughput from Experiment 3 (28.9K bps) is compared to the average NFE measurement (33.5K bps) a differential of approximately 14 per cent results.

Currently, the test bed cable bus system is supported by older 8-bit microprocessors. The processor clock rate is slow, 2 MHz. The instruction set of the microprocessor is rudimentary, requiring a number of instructions to perform meaningful operations. Newer microprocessors have clock rates that are twice as fast as existing interface unit clock rates. These new microprocessors also have more efficient instruction sets. Microprocessors, which will be available within the year, are purported to operate at ten times the speed of their older counterparts.

The test bed hardware interfaces are also rudimentary, requiring an interrupt to handle each character received or transmitted. Faster direct memory access interfaces will significantly decrease the interrupt overhead, thereby increasing the transfer speed between the interface units and the test bed computers.

Present measurements show that the test-bed system is 14 per cent slower than the ENFE. It is felt that this relationship will change with the advent of new cable bus interface units and a new operating system for the NFE.

If the new NFE operating system performs as expected, a three to five fold increase in NFE performance may be forthcoming. Thus an NFE performance estimate of 100.5K bps to 167.5K bps might be expected.

During FY 80 a new cable bus interface unit based on a fast 16-bit microprocessor (Zilog 8000) will be tested. This microprocessor is purported to be ten times faster than the existing interface unit microprocessor. Furthermore, the instruction set of the new microprocessor is more efficient, allowing more meaningful functions to be performed with fewer instruction fetch cycles. It is expected that the combination of this new microprocessor and direct memory access I/O will massively increase cable bus performance. Speculative estimates are that an increase of from five to ten fold in performance may be realized.

Since each interface unit is capable of this increased performance, the overall performance of the cable bus becomes a function of the number of active interface units. Each interface unit then may be capable of from 144.5K bps to 289K bps. If four interface units are active, 578K bps to 1156K bps may be measured. Thus, four interface units are a minimal number to support an acceptable traffic rate if all four operate at maximum capacity. In an operational installation, several more interface units operating at only fractional capacities might be better suited to the application, but acceptable traffic flows will easily be maintained.

The relative performance of the NFE and the cable bus then are seen as changing to a point where the cable bus will be significantly faster than the message switch. FY 80 measurements will support a less speculative comparison.

6.5 Experiment 4: Interneting Experiment

The fourth experiment verified the capability of the newly evolved protocol tested in Experiment 3 to function in an internet environment with long-haul networks, specifically the ARPA network.

The verification was in the form of opening a series of connections between the LSI-11 and various ARPA network sites via 11/70 based gateway translation software. The expeiment is shown pictorially in Figure 22 below. The translation software, in the form of a modified cable bus TCP implementation, took "open-virtual-circuit" requests and translated them into ARPA network TCP "open-virtual-circuit" requests and standard ARPA network Host-to-Host "open-virtual-circuit" requests. Once the connection was open, a process

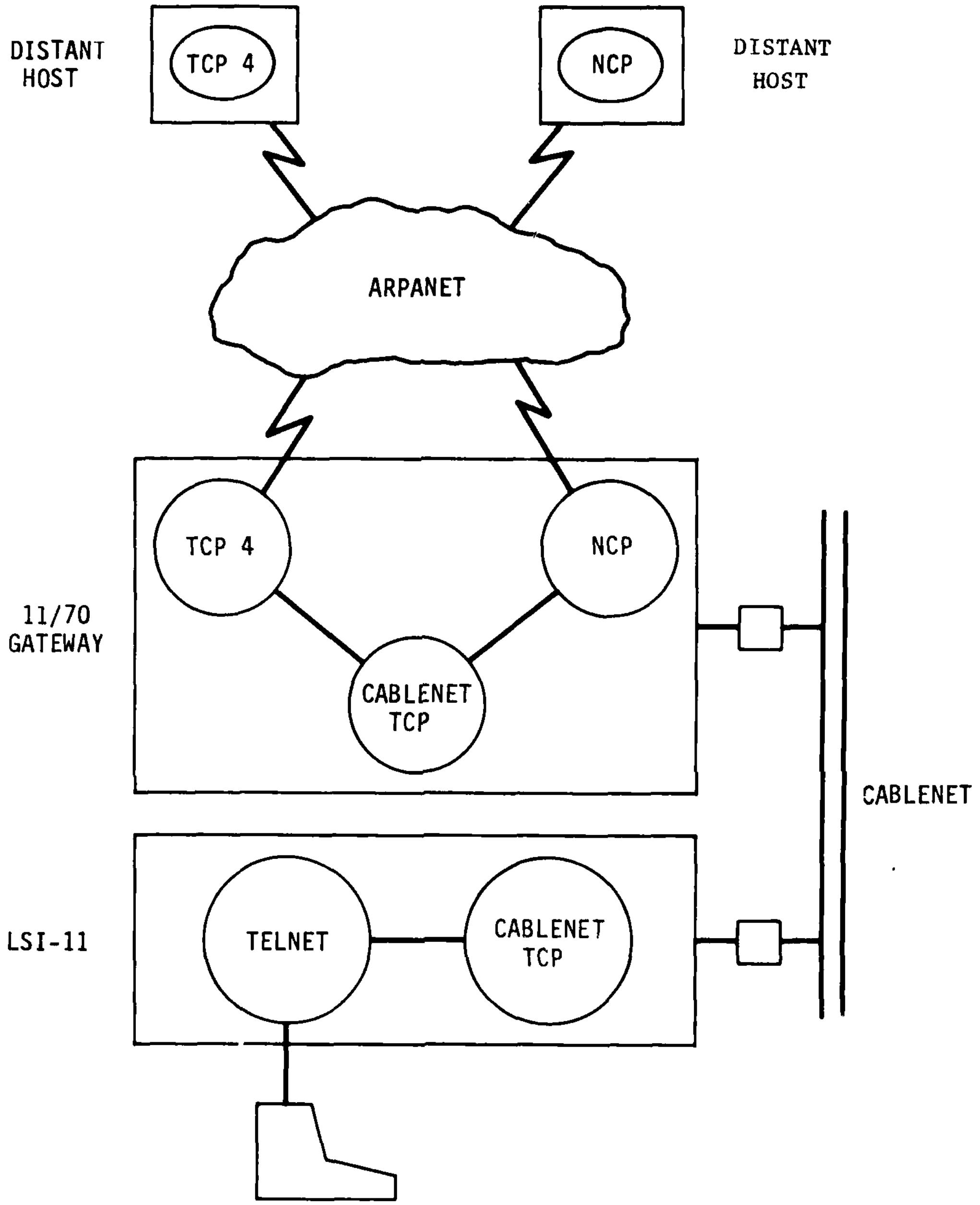


FIGURE 22
INTERNETTING EXPERIMENT

was started to transfer data between the ARPA network and the cable bus network.

Telnet connections were completed to various ARPA network hosts using the Host-to-Host translation software. A TCP loop was made through the ARPA TCP and back into the LSI-11. On the whole, the translation software was reasonably easy to construct and was put together quickly. However, there are significant problems such as exception condition and out-of-band signal handling which need to be factored into the translation handling software. These problems will be attacked during the coming fiscal year.

7.0 EVOLVED PROTOCOL ARCHITECTURE

The next generation WWMCCS functional requirements and the cable bus characteristics, coupled with our early experimental findings, and our view of the future microprocessor interface unit, dictated the necessity for a WWMCCS protocol architecture. Our intention was to construct a protocol architecture to fit these requirements and still remain open enough to evolve nicely to the large number of perceived command center applications. Our approach has been to provide a good set of established protocol mechanisms within an extensible structure such that protocols may evolve on a stepwise basis as new protocol mechanisms and command center applications arise. In addition, the design of this protocol architecture does not preclude the transparent transport of other existing protocols such as TCP.

The protocol architecture is also based on the idea that there are really very few new protocols. Most new protocols are recombinations or reformattings of such functions as opening a virtual connection or flow controlling data as it passes over a virtual connection.

The protocol architecture that results from this point of view is shown in Figure 23.

- Level 0 provides the cable bus contention mechanism. This level decides when a packet should be transmitted to the cable and handles packet collisions.

- Level 1 provides a flexibly structured set of protocol mechanisms which in combination provide protocol capabilities ranging from raw, unsequenced, unreliable datagrams to a fully sequenced, reliable data stream. This layer (the flexible transport) is described in detail below.

- Level 2 represents a series of protocol implementations which are constructed by combining level 1 mechanisms. Note that a Transmission Control Protocol (TCP) implementation rests comfortably at this layer by using only a very few of the level 1 mechanisms. The datagram implementation provides a user interface to the level 1 address mechanisms and adds the capability to perform fragmentation and reassembly of larger sized datagrams. The connection management implementation

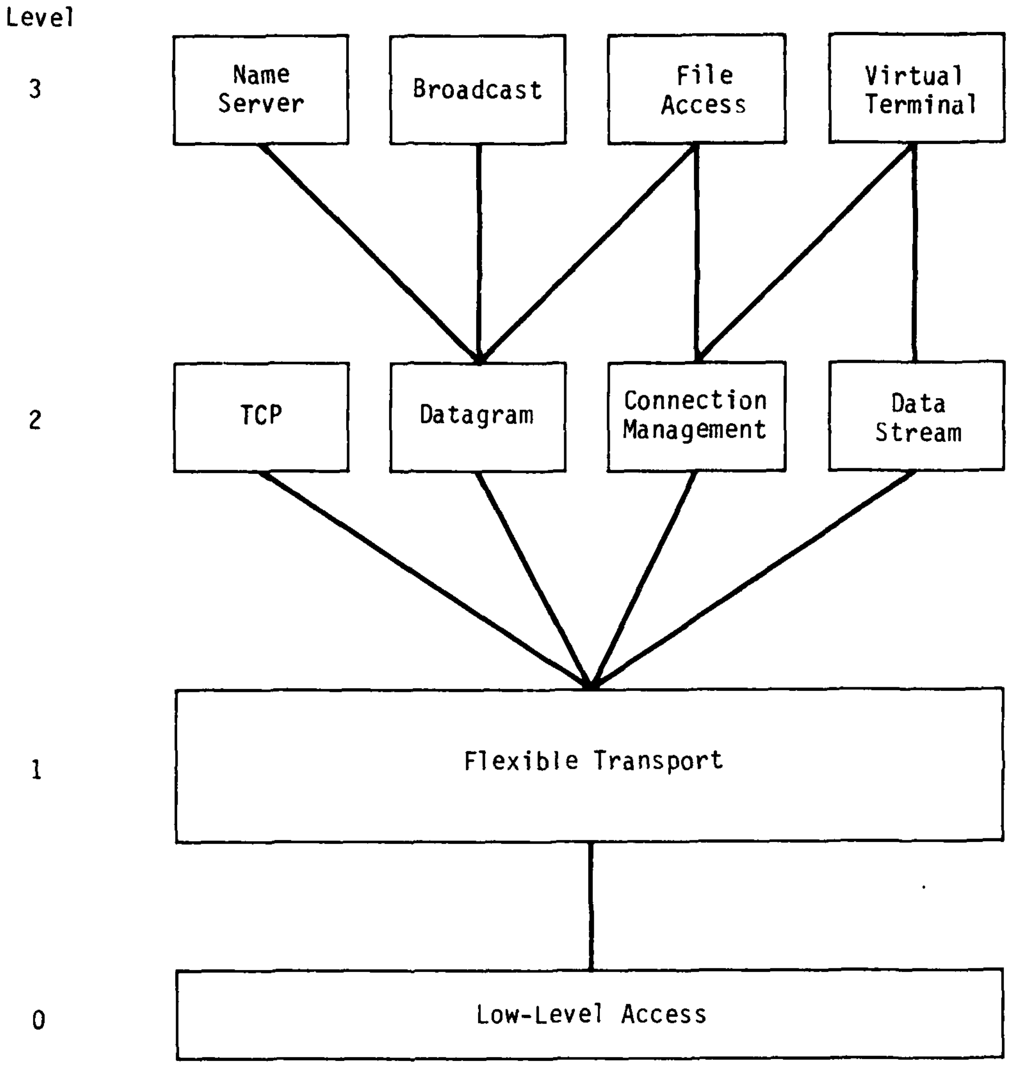


FIGURE 23
EVOLVED CABLE BUS PROTOCOL LAYERING

provides management of virtual circuit connections using a timer based protocol described in Fletcher and Watson.[20] The data stream implementation provides a user interface to the reliable, sequencing and out-of-band signaling mechanisms of level 1 to form a reliable, sequenced data stream capability.

- Level 3 represents examples of higher level protocols which make use of the functions provided by layers 1 and 2.

Thus level 2 software may exist as a simple combination of level 1 mechanisms providing an interface to the next higher level or it may implement a full set of its own mechanisms using few of the level 1 mecnanisms.

The name server implementation provides a mapping between string names of objects and internetwork addresses described in Postel.[21]

The broadcast implementation provides a capability to group a set of users into a conversation such that a message generated by one user is received by all users.

File Access provides access to remote file data. The File Access implementation, based on a protocol suggested by Day,[22] allow data transfers to be described as a series of file byte pointer and length requests followed by data responses.

The virtual terminal implementation maps terminal characteristics into standard virtual terminal representations. It includes graphics functions such as line, curve, and shaded polygon drawing as well as the more mundane record mode, stream mode, and terminal option negotiation.

7.1 Flexible Transport Protocol

The Flexible Transport Protocol[23] defines a set of rules to govern the transport of blocks of data over interconnected cable bus networks with binary degrees of reliablity, flow control, addressing, and other common end-to-end and transport level mechanisms. These mechanisms are grouped at this level to allow the optional specification of each of the mechanisms in terms of its effect on the format of the resulting protocol header. Each header contains a "bit-map" specifying the "shape" or attributes of the remaining portion of the packet. These attributes are groups of data fields which, if specified, cause the protocol mechanisms referred to above to be invoked. If an attribute is not specified, default processing mechanisms, such as always accepting a packet as in sequence if the sequence attribute is not specified, are invoked. The obvious advantage of this scheme is that if a mechanism is not required to support a particular type of data transfer no price is paid in terms of header overhead and processing cycles.

If in the future a general environment exists where the overall packet length is not restricted, the flexible header scheme may easily evolve into a fixed scheme. We should also note that in practice, we don't envision the header length changing from packet to packet, but rather for the attribute specification to be a function of the application process.

The Flexible Transport Protocol supports the following mechanisms:

> <u>Packet Assembly/Disassembly</u>. This mechanism provides packet formatting for out-going packets from a global set of variables and a specification of attributes to be applied to any packet data. It also provides for the disassembly of received packets and the passing of any packet attributes and data to a pre-defined sequence of mechanisms within the flexible transport layer.
>
> <u>Packet Addressing</u>. This mechanism provides for the detection of packets addressed to the local host. It detects packets addressed to a remote network if the local host is a gateway to that network. It detects packets addressed to the local host. It locates local port data structures. It also handles broadcast messages should any of the destination host or port attributes of the packet be absent.

Packet Typing. The packet typing mechanism provides for the manipulation of a registered set of packet types which include host reset requests, protocol error messages, and host status messages.

Packet Sequencing. The packet sequencing mechanism provides a means for identifying whether a packet is in sequence, out of sequence, or a duplicate.

Flow Control. The flow control mechanism provides a sliding window describing a range of acceptable sequence numbered packets. It also provides for the acknowledgement of previously received packets.

Out of Band Signal. The out of band signal mechanism provides a means of specifying a location in the data stream where "interesting" information resides.

Figure 24 illustrates the hierarchical structure of the flexible transport attributes. Flow control, for example, makes use of the packet sequencing, and out of band signaling needs to be free of flow control constraints. Packet assembly/disassembly is of course always needed, and one can select any combination of the other mechanisms.

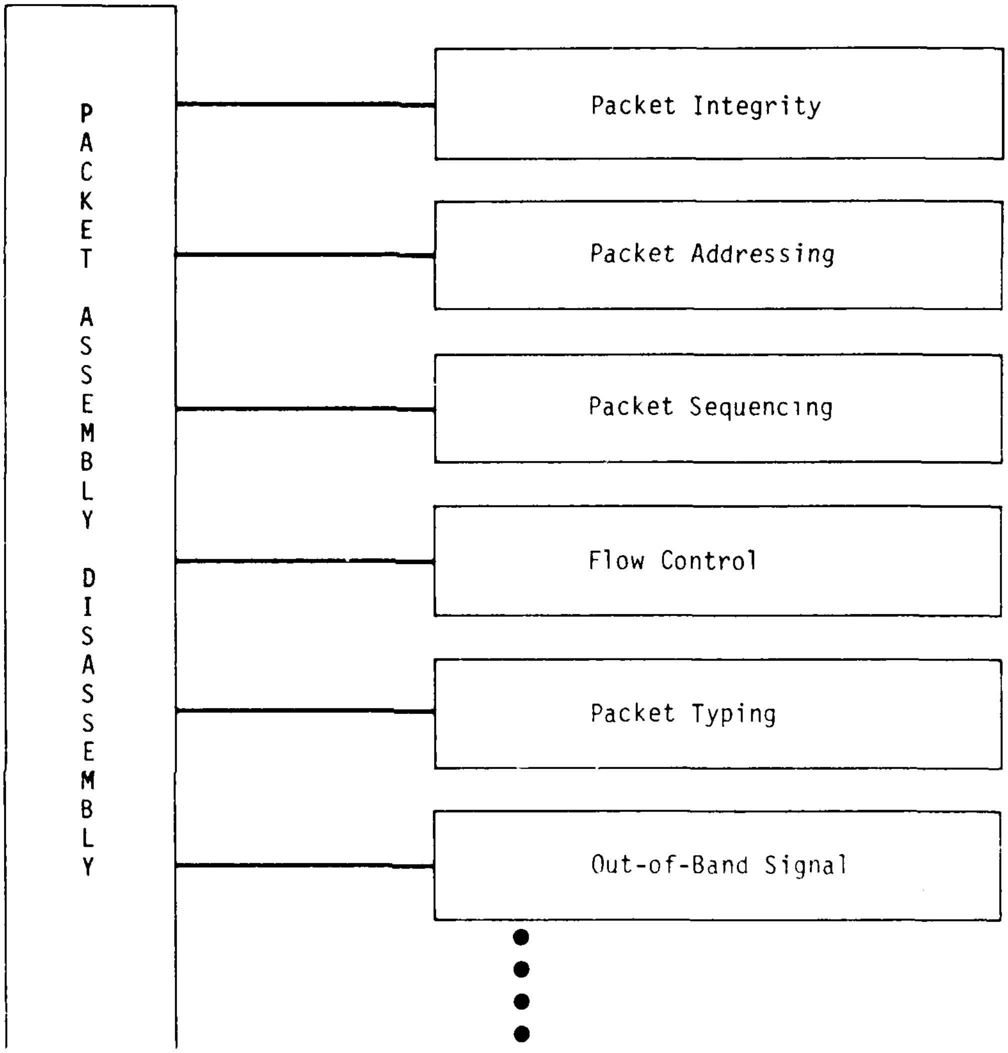


FIGURE 24
FLEXIBLE TRANSPORT PROTOCOL STRUCTURE

AD-A086 947 MITRE CORP MCLEAN VA F/G 17/2
CABLE BUS APPLICATIONS IN COMMAND CENTERS.(U)
NOV 79 S F HOLMGREN, A P SKELTON, D A GOMBERG F19628-80-C-0001
UNCLASSIFIED MTR-79W00383 NL

2 of 2
AD
A086947

END
DATE
FILMED
9 80
DTIC

8.0 CONCLUSION

Based on the analysis activity, it appears that the cable bus will be as secure as the message switch, from 27 per cent to 51 per cent less expensive, significantly more reliable, and have better overall performance with a flexibility for incremental growth surpassing the message switch architecture. However, due to the early stages of development of both the cable bus and the message switch, conclusive opinions as to their applicability to the command center are necessarily subjective. FY 80 cable bus protocol development and testing should enable less subjective comparisons.

The FY 80 experimental cycle will insure the soundness of the implementation of the protocol architecture described in Section 7. The protocol implementation will be expanded to include a virtual terminal protocol with color graphics capabilities. It is planned that a cable bus system will be installed at DCA Reston in order to begin evaluating the system in a command center setting. In a related MITRE internal research and development effort, a new, faster interface unit will be developed and tested.

REFERENCES

1. Acker, L. R., MacKay, D. J., Nixon, R. T., National Military Command System Description (U) SECRET, MTR-5316, MITRE Washington, February, 1979

2. Day, J. D., Grossman, G. R., Howe, R. H., WWMCCS Host to Front End Protocols: Specifications Version 1.0, DTI Document 78012.C-INFE.14, November, 1979

3. Liu, M. T., Distributed Loop Computer Networks, Advances in Computers, Volume 17, Academic Press, 1978, pp. 163-221

4. Trooper, C., Models of Local Computer Networks, MTR-3783, MITRE Bedford, May, 1979

5. LaBarre, C. E., Analytic and Simulation Results for CSMA and Contention Protocols, MTR-3672, MITRE Bedford, November, 1978

6. Poh, S. S., Stoneburner, P. D., Wood, D. C., Network Front End Evaluation Report, MTR-5312, MITRE Washington, August, 1978

7. Hopkins, G. T., A Bus Communications System, MTR-3515, MITRE Bedford, November, 1977

8. Malis, A. G., The MITRE Carrier-Sense Multiple-Access Listen-While-Talk Interface Unit Description, Unpublished Draft

9. Hopkins, G. T., Meisner, N. B., Willard, D. G., Cable Bus Contention Comparison, MTR-3296, MITRE Bedford, August, 1976

10. Courtney, R. H., Security Risk Assessment in Electronic Data Processing Systems, IBM Corporation, November, 1975

11. NSA, ADP Security Design Goals and Standards (U) CONFIDENTIAL, NSA-S86, March, 1979

12. Overman, W. W., Final Report, MITRE RF Data Bus: Unintentional - Undesired Signal Radiation and Conduction (U) CONFIDENTIAL, NSA-R61, July, 1979

13. Diffie, W., Hellman, M. E., Privacy and Authentication: An Introduction to Cryptography, Proc. IEEE, Volume 67, Number 3, March,1979, pp. 441ff

14. Katsuki, D., Elsam, E. S., Mann, W. F., Roberts, E. S., Robinson, J. G., Skowronski, F. S., Wolf, E. W., Pluribus-An Operational Fault-Tolerant Multiprocessor, Proc. IEEE, Volume 66, Number 10, October, 1978

15. Telenet Communications Corporation, TP2200 & TP4000 Hardware Description Manual, TCC-TPGHD-0179-HM-16, Vienna, Virginia, 1977

16. Digital Technology Inc., An Assessment of NFE Engineering Alternatives, DTI No. 79001.C-INFE.20, March, 1979

17. Postel, J., ed., Transmission Control Protocol, Internet Experiment Note 112, ISI, August, 1979

18. Postel, J., ed., Internet Protocol, Internet Experiment Note 111, ISI, August, 1979

19. Digital Technology Inc., Statement of Work: Communications Operating System for AUTODIN II Network Front End, August, 1979

20. Fletcher, J. G., Watson, R. W., A Mechanism for a Reliable Timer-Based Protocol, Computer Networks, Volume 2, 1978

21. Postel, J., Internet Name Server, IEN 61, October, 1978

22. Day, J. D., A Proposal for a File Access Protocol, ARPA Network RFC 520, 1973

23. Holmgren, S. F., Flexible Transport Protocol, in preparation

DISTRIBUTION LIST

MITRE Washington

D-12 H. D. Benington
C. C. Grandy
S. W. Gouse

D-14 A. J. Tachmindji
J. S. McManus

W-30 D. S. Alberts
H. T. Gruisin
J. S. Quilty
W. B. Woodward
W-30 Administrative Office

W-31 J. W. Benoit
M. J. Gordon
S. F. Holmgren (100)
A. J. Kleiboemer
R. J. Nieporent
H. K. Resnick
J. K. Summers (2)
D. C. Wood
Technical Staff

W-32 W. P. McQuiggan
H. I. Ottoson
R. Rubin

W-34 W. A. Eliot
R. A. Joy
S. J. Turner

W-36 C. E. Bowen
R. P. Granato
W. B. Stevens

W-37 F. A. Frangione
W. B. Hall

MITRE Honolulu - T. C. Hilinski

MITRE Washington Library

MITRE Bedford

D-64 V. A. DeMarines
G. T. Hopkins (5)

D-75 E. L. Burke (5)

MITRE Bedford Library

DCA/CCTC

Chief Scientist, 101A
Dr. Stillman, 101D
Director CCTC, C100
Deputy Director CCTC, C101
Lt. Col. Paul Davis, C110
Mr. W. Leary, C110
Chief Plans Division, C210
Technical Director, NMCS ADP Directorate, C301
Deputy Director, WWMCCS ADP, C401
Assistant Deputy Director, WWMCCS ADP, C401
Mr. M. Welch C402
Mr. M. Corrigan, C409 (20)
Chief, Software Development Division, C420
Chief, Network Branch, C421
Mr. E. Britton, C421
Maj. K. Foiles, C421
Mr. Ron Gray, C421
Mr. J. Thomas, C421
Chief, Systems Software Branch, C422
Chief, Plans and Analysis Division, C610
Chief, Reston Computer Operations Division, C720
Chief, Reston Computer Operations Branch, C721

DISTRIBUTION LIST CONCLUDED

Chief, Reston Computer Support
Branch, C722
Dr. C. Guffee, DCEC, R810

RADC

Communications and Control Division (DC), Dr. Diamond
Telecommunications Branch (DCL),
Mr. Kelly
Telecommunications Branch (DCLT),
Mr. Davis

WSEO

R. Bookman
Col. Pixton
J. Volpe

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
MTR-~~80~~79W00383 ✓

2. GOVT ACCESSION NO.
AD-A086 947

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
FY79 Final Report: Cable Bus Applications in Command Centers

5. TYPE OF REPORT & PERIOD COVERED

6. PERFORMING ORG. REPORT NUMBER
MTR-~~80~~79W00383

7. AUTHOR(s)
Steven F. Holmgren
Anita P. Skelton
David A. Gomberg

8. CONTRACT OR GRANT NUMBER(s)
F19628-80-C-0001 ✓

9. PERFORMING ORGANIZATION NAME AND ADDRESS
The MITRE Corporation ✓ /MITRE C^3 Division
Washington C^3 Operations
1820 Dolley Madison Blvd., McLean, VA 22102

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS
DCA/Command and Control Technical Center
Attn: C421 11440 Isaac Newton Square, North
Reston, VA 22090

12. REPORT DATE
November, 1979

13. NUMBER OF PAGES
88

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)

15. SECURITY CLASS. (of this report)
Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)
~~Controlled Distribution~~
Distribution Unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)
No restriction.

18. SUPPLEMENTARY NOTES

19 KEY WORDS (Continue on reverse side if necessary and identify by block number)
Cable Bus, Command Center, WWMCCS

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

An evaluation of Cable Bus Technology as applied to WWMCCS Command Centers is presented. The report compares message switch and cable bus architectures in terms of security, performance, flexibility, functionality, cost and reliability.

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

F

9