

Copyright © 1985, by the author(s).
All rights reserved.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission.

THE PROTOCOL WORKROOM: AN EXPERIMENTAL PROTOCOL
AND DISTRIBUTED SYSTEM RESEARCH FACILITY FOR
U.C. BERKELEY

by

Ayman Fawaz, Didier Giralt and Lester Ludwig

Memorandum No. UCB/ERL M85/82

(Protocol Workroom Document No. 84-1 (Version 3))

11 November 1985

COVER PAGE

THE PROTOCOL WORKROOM: AN EXPERIMENTAL
PROTOCOL AND DISTRIBUTED SYSTEM RESEARCH
FACILITY FOR U.C. BERKELEY

by

Ayman Fawaz, Didier Giralt and Lester Ludwig

Memorandum No. UCB/ERL M85/82

(Protocol Workroom Document No. 84-1 (Version 3))

11 November 1985

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

TITLE PAGE

THE PROTOCOL WORKROOM: AN EXPERIMENTAL
PROTOCOL AND DISTRIBUTED SYSTEM RESEARCH
FACILITY FOR U.C. BERKELEY

by

Ayman Fawaz, Didier Giralt and Lester Ludwig

Memorandum No. UCB/ERL M85/82
(Protocol Workroom Document No. 84-1 (Version 3))

11 November 1985

ELECTRONICS RESEARCH LABORATORY
College of Engineering
University of California, Berkeley
94720

**THE PROTOCOL WORKROOM: AN EXPERIMENTAL
PROTOCOL AND DISTRIBUTED SYSTEM RESEARCH
FACILITY FOR U.C. BERKELEY**

Ayman Fawaz, Didier Giralt and Lester Ludwig

*U.C. Berkeley "Protocol Workroom" Development Group,
Department of Electrical Engineering and Computer Sciences
University of California at Berkeley
Berkeley, California 94720*

Protocol Workroom Document No. 84-1, (version 3)

ABSTRACT

This paper describes a 10 Mb/s real time hardware facility, currently under development at U.C. Berkeley, to support research in communications protocols, network architectures, and distributed systems. The facility consists of 32 programmable node emulation systems, a programmable channel emulator system, and a backbone system. Each node emulator can independently emulate an intelligent digital device such as a host, server, switch, gateway, etc., with up to 8 distinct physical level ports. Each node emulator supports message generation, monitoring, and link layer protocol operation at 10 Mb/s. The multiport configuration includes an integrated switching fabric and an enhanced control processing. The programmable channel emulator emulates a wide variety of physical level functions, network architectures, and independent subnetworks. This system implements arbitrary topologies, tap functions, and inter-node delay values and provides carrier sensing, collision detection, and both synchronous and asynchronous operation. The backbone system is unobtrusively responsible for control, monitoring, analysis, display, and interactive features. It consists of an Ethernet and Sun workstations.

THE PROTOCOL WORKROOM: AN EXPERIMENTAL PROTOCOL AND DISTRIBUTED SYSTEM RESEARCH FACILITY FOR U.C. BERKELEY

Ayman Fawaz, Didier Giralt and Lester Ludwig

U.C. Berkeley "Protocol Workroom" Development Group,
Department of Electrical Engineering and Computer Sciences
University of California at Berkeley
Berkeley, California 94720

1. INTRODUCTION

This document describes a protocol and distributed system research facility currently under development at UC Berkeley. This facility, named the *Protocol Workroom*, is primarily intended to provide a hardware support system for research in the fields of communications protocols and networks. Interest in these areas has continuously increased since the introduction of Ethernet as a means of achieving economical, fast, and reliable local communication between heterogeneous digital devices. Interest has also accelerated from the promises offered by Integrated Services Digital Networks (ISDNs) for enhanced telecommunications services. Although much has been learned since the first release of Ethernet, surprisingly few of the many enhancements and alternatives to its basic CSMA/CD protocol have been implemented in hardware. In addition, there are many other architectures that merit investigation, both for theoretical purposes and in relation with developing commercial products and international communications standards. Finally, with the introduction of the first portable personal terminals with radio link communication, the topic of mobile packet radio takes on new importance.

The project goal is to provide a hardware facility supporting the study of a wide variety of these networks and protocols. In addition to offering a powerful

development tool through its emulation capabilities, the control and monitoring systems are designed to support 10 Mbps real-time stochastic simulation. Since protocols and architectures may be directly implemented, complex modeling and modeling approximation can be avoided. The real-time feature avoids the delay associated with software simulation. Thus the facility also creates a new complement to traditional mathematical analysis and software simulation techniques currently used in network research. Further, the facility provides through its architecture a distributed processing system incorporating a respectable number of powerful processors. It thus serves as a natural environment for research in the areas of distributed systems, distributed processing, distributed control, and distributed operating systems/algorithms, offering a centralized control and monitoring station for user control of investigations. Finally, in addition to its use in research, the facility can also serve as an instructional tool for use in academic curriculum at UC Berkeley.

This paper begins with a brief overview of the facility architecture. Key hardware details are then examined. A discussion of the facility control and monitoring systems is then presented. This is followed with a treatment of the various software and software development tools employed in the facility. The paper concludes with a discussion of the initial start-up configuration, development plan, and intended applications for the facility.

2. OVERVIEW OF THE FACILITY

Figure 1 shows a view of the architecture of the facility. Functionally, it may be viewed as a number of intelligent nodes (each emulating a digital device) connected to two separate communication networks, the *experimental* and *backbone* networks. The experimental network emulates the network whose behavior is under study. This network consists of a physical level channel

emulator, to which the node terminates, and an intelligent network controller, which is located within each node. The experimental network operates at a fixed data rate of 10 Mbps. The backbone network provides communications for configuration, control, monitoring, and analysis. It is realized with a standard Ethernet network and Sun workstations. A remote master computer (a dedicated Sun fileserver) and VAX-11/750 (providing database functions) are connected to the backbone network to support the software development, monitoring, and control of the nodes. The composite collection of processors and the backbone network are termed the "backbone system".

The channel emulator depicted in Figure 1 is a hardware system which, under the control of the master computer, inserts programmable time delays, tap features, interconnections, and fault conditions among the nodes. In this system there is no analog encoding of transmitted data, no transceiver, and no cable; rather the entire "physical layer" and collision detection systems are implemented in logic. The channel emulator provides flexible logical emulations of a wide class of actual physical layer networks.

As shown in Figure 1 each node emulator is realized as a two-board system consisting of:

1. A Pacific Microsystems 68010 board with 256K of dynamic RAM serving as a host computer connected to a network, or as any other digital device emulator; and
2. A custom designed system incorporating a Motorola 68020 processor, special hardware, and state machines. This system serves as an intelligent network controller.

This is illustrated in Figure 3.

In the single port configuration the Pacific board connects to its associated network controller over a private bus. In the multiple port case, a single Pacific

board hosts several network controller over the same private bus.

Each node reports key events (successful transmission or reception of a packet, error, collision, etc.) to the master computer to allow for monitoring and statistical analysis. Unlike the configuration and control features the monitoring and analysis functions are invisible to the operation of the experimental network. It is therefore convenient to view the facility as having a conceptual systems architecture that features distinct layers for experiment operation and monitoring. This is illustrated in Figure 2. Note the channel emulator is not explicitly involved in monitoring and thus resides only in the operations layer.

The combined MC 68020 and state machine controller is primarily used to realize software defined hardware implementations of link level protocols. Modification of node link layer protocol is permitted by software modifications of the 68020 code and the state machines control store.

Packet generation and most protocol features above level 2 (network, transport, etc.) required for the experiment are implemented in the Pacific 68010 system. In fact, each node emulator contains a local message generation scheme implemented in software by the Pacific 68010 CPU. The generation of data (e.g., packets, blocks, spurts, etc.) for transmission over the experimental network is programmable. Sample paths of a specified stochastic process modeling transmit times (e.g., Poisson, doubly-Poisson, etc.), destination addresses, and packet/block/spurt lengths can be easily realized with locally-implemented pseudo-random generators. Table-driven deterministic processes can also be used, as well as combinations of these and generated sample paths.

3. HARDWARE DESCRIPTION

There are three major hardware systems in the facility: the backbone system, the node emulators, and the channel emulator. These are discussed separately in of the following subsections.

3.1. Backbone System

The backbone system consists of the following: a VAX 11/750 used for database functions, a SUN fileserver used as a master computer, and SUN rack-mountable workstations used to control a cluster of node emulators. They all communicate over an Ethernet local area network. In this configuration, a group of 18 node emulators form a cluster and are controlled by a SUN host. In the current plan, two clusters would be used to implement the 32 node facility. The host computer collects the information gathered by the node emulators and forwards it to the master computer. This two-level architecture greatly reduces the cost of interfacing node emulators to the backbone system. Furthermore, the Sun-2/170 host provides a first stage of processing and filtering for the monitoring messages issued by the node emulators it supervises. This reduces the load on the master computer and adds processing flexibility. In the reverse direction, the Sun host directs control commands and facilitates software downloading. The Sun master computer also supports the facility monitoring and analysis systems and interactive features.

3.2. Channel Emulator

The channel emulator depicted in Figure 4 is a computer-controlled hardware system performing six basic functions:

1. Separate interfacing with each node emulator, obtaining transmit timing, data, carrier signals from each node while simultaneously presenting

- received timing, data, carrier signals and collision monitors to each node.
2. Support of a wide range of node connection topologies and configurations.
 3. Insertion of programmable propagation delays into the asynchronous serial data streams between any pair of connected node emulators.
 4. Real-time insertion of a number of possible fault conditions into the serial data streams between any pair of connected nodes.
 5. Simultaneous implementation of a variety of collision detection schemes at each experimental network node.
 6. Distribution of a global clock signal used for global time stamps and a common time base for synchronous network experiments.

The channel emulator thus provides for the flexible logical emulation of a wide class of actual physical layer network types and operating conditions. Through the incorporation of very long delay times, it also permits geometric stretching for time-distance scaled modeling of high-bandwidth networks. Further discussion of the channel emulator is provided in [1].

The channel emulator replaces analog encoding of transmitted data, transceivers, and cable traditionally employed in the interconnection of nodes with a centralized hard-wired digital logic system. This system, under programmable control of the master computer, implements functional equivalents to the entire physical layer and associated collision-detection mechanisms. The Sun-2/170 host is used for establishment and real-time modification of the channel emulator configuration and operational modes.

The channel emulator provides a separate interface for 32 node emulator ports. A single-port node emulator uses one of these ports while a multi-port configuration may use several. Figure 5 shows the interface. There are transmit and receive connections for data, and code violation, carrier, and timing signals.

The code violation signals are used in expanding the data symbol-space so as to carry out-of-band signals such as Manchester code violations. Such out-of-band signals are often used for control purposes. Carrier signals indicate that the signals on the data and violation lines are valid and can be interpreted as a logic signal indicating a state of active transmission. These signals are also used in clock arbitration during asynchronous operation, for one type of collision detection, and in directional sense (as used in Fastnet [2]).

Through use of the channel emulator the node emulators can be connected in a number of configurations. The current plan is to support the following network topologies (see [3],[4],[5] for discussions of these):

1. bidirectional bus, with and without directional sense,
2. unidirectional folded bus (both single-fold and double-fold),
3. unidirectional bus pair,
4. unidirectional ring,
5. double rings (counter-rotating or unidirectional),
6. collision-protocol star,
7. passive trees (as in CATV and ISDN passive bus),
8. point-to-point,
9. gateway configurations and hierarchical subnetworks, and
10. mobile/fixed-site radio topologies of up to 8 nodes.

Delay cells are used to introduce programmable delays between node emulators to simulate the propagation time introduced by a transmission medium. Each internode delay can be separately adjusted to give delays corresponding with separations between 10 and 20490 meters in 20 meter increments. For a 32 node network, this gives an effective network length in excess of 650 Km for use in time-distance scaling of high-bandwidth networks. When time-distance

scaling is used to simulate higher transmission rates, the resolution is enhanced by the scale factor (e.g., the increments are 2 meters for scaling to 100 Mbps). The long delays are also directly useful in the study of geographically large networks. In addition to time-distance scaled protocol studies, this capability also permits study of synchronization problems with applications in both communications and power systems that are physically distributed over large distances.

The architecture of the channel emulator is shown in Figure 4. It consists of tap/clock arbitration/feedback blocks, a node-port interface for each of these blocks, and flexible interconnection fabric. This interconnection fabric consists of 64 delay cells, a switching network that allows any delay to be driven by any one tap block, and a masked-OR array. This array can function as another switching network, permitting any tap block to be driven by any delay cell, but also can permit any group of selected multiple sources to drive any tap block. This mode creates radio topologies.

3.3. Node Emulator

The operation of the node emulator is quite complex and is described in a series of documents [6,7,8]. A simplified view of the node emulator architecture is depicted in Figure 3.

Each node emulator consists of a two-board system connected to the Multibus backplane of the Sun-2/170 host. In addition to the Pacific 68010 board, this processing system also contains a real-time 10 Mbps intelligent network controller consisting of a 68020 processor, direct-memory access controller, timers, transceiver, pattern recognizers, CRC hardware, and a bit-event driven state machine. The design philosophy is to complement the 68020 processor with an ensemble of specialized hardware and a fast state machine, thus relieving the 68020 of all bit-oriented actions. The processor and the state machine

are used together in implementing link layer protocols for the experimental network. The state machine makes decisions and takes actions on a bit-period basis according to an easily programmed stimuli-response table. The state machine does no computations. Computations and other higher level functions are performed by the 68020.

The MC 68020 was chosen for its computational speed, powerful high-level instruction set, and the existence of the Magic/L [9] software development system which provides the flexibility required for easy software development. Since the same Magic/L environment will support the 68020, the 68010 in the Pacific board system, and the various Suns making up the backbone system, a single software development environment can thus be used for the entire facility.

In addition to performing managerial and computational actions involved in implementing protocols, the 68020 also assists in the monitoring and file transfer processes between the host and the network controller. The 68020 implements the latter functions under interrupt control. The file exchange is done through the dual-port RAM shown in Figure 3, used here as a mailbox. This dual-port RAM is also used as buffer storage for incoming and outgoing packets. A separate memory is used for the control store containing the software executed by the 68020 and the stimuli-response table used by the state machine. The 68020 is assisted by a set of timers. These are programmed and started by the 68020 as needed to keep track of time-out conditions without burdening the 68020 with the actual time keeping. When a timer expires, the 68020 is interrupted and provides the service requested. In general, these features and the structure of the other special hardware making up the network controller permits simultaneous transmission and reception of data between the host/controller pair and controller/channel-emulator pair.

Special direct-memory-access hardware is used to transfer information from the dual port RAM to the transceiver hardware (which accesses the channel emulator). Information transfers are initiated by the 68020 or state machine and require only start pointer and end pointer values and an "initiate" command. The DMA hardware provides notification at the completion of a transfer, and can recover from an aborted transfer as needed. In this way the 68020 and state machine are freed from the detailed actions of the information transfer, and are free to non-destructively abort a transfer that is in progress. The information flowing between the transceiver and dual port RAM is passed through dedicated CRC hardware as the transfer occurs. Two separate CRC generators compute the checksum of outgoing packets (for CRC insertion) and incoming packets (for error detection), respectively.

The transceiver system consists of independent transmitter and receiver circuits which may be linked as needed by a programmable delay line and a programmable editor. This delay line is used to store a received packet when the node is transmitting another message. This permits pattern recognition hardware to identify control fields requiring specific action and retransmit an edited packet without having to perform massive storage and regeneration functions. This feature can react within one bit-period delay and can be used in a very flexible manner. Such a feature is very useful for ring topology protocols and general packet forwarding. In the multi-port configuration to be discussed below, this feature also permits the implementation of cut-through packet switching and can be very useful in gateway and head-end station arrangements. The pattern recognition hardware searches for several patterns of arbitrary length simultaneously. The patterns may be changed as a function of state machine commands, and these changes can be in the form of conditional branches. Patterns that may occur at any unforeseen time require the full

attention of the pattern recognizer, while a number of patterns whose expected start time is known at the beginning of the recognition search can be watched for simultaneously and separately identified by a single pattern recognizer. Such simultaneous searches are useful for handling aliases and other group addressing procedures. The number of patterns simultaneously searched is 4 in the current design.

As important as the emulation hardware itself is the unobtrusive monitoring of the events occurring at each node. The network controller includes a special event FIFO and time-stamp circuit for this purpose. A global timing signal, independent from the controller's own internal clock source, is transmitted to each node by the channel emulator over the interface shown in Figure 5. This global timing signal is also independent from the receive timing signal also provided over this interface. The global timing signal and an associated synchronizing signal are used to run a time-clock arrangement within the network controller. This permits each controller board to have access to a globally agreed time value for use in assembling event records for monitoring. Whenever an event occurs, an event record is generated and is marked with the current value of globally agreed time at that instant. Events are defined by the state machine and 68020. Some events are permanently defined, while others may be programmed in either the state machine or 68020. It is possible to instruct the 68020 to suppress records that are of no value in an experiment. It is important to note that simultaneous events can be recorded in the same record. The event records, including time-stamps, are stored in a FIFO. The contents of this FIFO can be transmitted to higher layers of the monitoring system at a time convenient for the 68020 and Pacific board. The monitoring system is described in further detail in the next section.

The multiport configuration is illustrated in Figure 6. Up to 8 network controller can share a single Pacific board via a common P2 bus. There are also two other buses as shown in Figure 6. In the multiport configuration the P2 bus is used for centralized control, employing the common Pacific board as the central controller. Another bus, the P3, is used as a high speed interconnection fabric among the controllers for exchanging data. It is implemented as a 64 bit bus operating at 1.25 Mbps although it is sampled at a 10Mbps rate. It is managed as a byte-oriented $10N$ Mbps TDMA bus, where N is the number of network controllers used in the configuration (N ranges from 2 to 8). Another key property is that the byte-orientation can be viewed itself as a TDM frame imposed on each 10Mbps data stream, and the space-division structure of the bus permits easy implementation of an effective time-slot interchange system.

The net result of this is that each 10Mbps stream can be viewed as eight 1.25Mbps channels which may be freely interconnected among the other $8(N-1)$ 1.25Mbps channels of the other $N-1$ nodes. Each effective 1.25Mbps time-slot of the P3 bus may be ignored or connected to either the message RAM or transceiver in either transmit or receive modes. This permits multiple bit-rate circuit switching and variable throughput packet and cut-through switching to be implemented over the same switch fabric (i.e., the P3 bus) and hence implements a true integrated switch fabric. Thus the multiport configuration can act as a circuit, packet, cut-through, or integrated switch as controlled by the executive (68020) and host (68010) processors. Since the rest of the network controllers are free to implement any compatible access protocol, the multi-port can also function as an integrated services gateway in addition to more traditional gateway functions. The channel emulator can be used to implement point-to-point connections or independent subnetworks for linking the multiport configuration to other node emulators. Store-and-forward, PBX, MAN, ISDN,

integrated LAN, and gateway networks can all be studied using these various features.

A P4 bus is also included to provide peer communication among the controllers 68020s. This permits decentralized control algorithms for use in control of the multiport configuration to be studied. As a result there are provisions for both centralized and decentralized control within the multiport entity. Thus various combinations of centralized and decentralized control systems elements can be compared in precisely the same environment under exact conditions. This could be quite useful given that the current trend in switching and PBX architectures is towards decentralized control via smaller processors. This trend is motivated by needs for modular architectures for smooth growth an evolution.

4. MONITORING AND ANALYSIS SYSTEM

To investigate and compare performance of different protocols and architectures a monitoring and analysis system is required. A highly flexible monitoring system, supplemented by off-line and on-line analysis systems interacting with presentation and display systems, is available in the facility for this purpose. A high-level view of the monitoring and analysis system was provided in Figure 2 where the monitoring and analysis system was shown as the background layer. The details of this system is illustrated in Figure 7.

The major features of the monitoring system are the programmable event classes and programmable record filtering systems. As discussed in the previous section, both predefined and custom defined events are noted and time-stamped by special purpose hardware within the network controller. These constructed records are queued in the event FIFO for convenient transmission to the Pacific board over the P2 bus. It may be possible for certain types of events

to be disregarded or processed before transmission to the Pacific board.

In the Pacific board the transfer records are stored in a FIFO data structure for filtering and processing. Protocol oriented processes within the Pacific board can also generate records which are processed as well. The processed records are then queued up for convenient transmission to the Sun cluster controller. This processor is very lightly loaded during an experiment and therefore has ample opportunity to process and consolidate records, possibly in coordination with the other Sun cluster controller via Ethernet communication. The final records are then passed up to the database and master computer for off-line and on-line analysis. On-line analysis systems can use moving-window statistical operations and also report run-time events. Both off-line and on-line analysis systems interact with a presentation process stage (providing graphics formats, etc.) and display system as shown in the Figure.

The filtering provisions provided within the facility permit the use of probe traffic monitoring techniques. In these techniques the facility is used to generate a specified ambient load by all but a small selected set of nodes. The selected set of nodes are used to insert monitoring traffic (i.e., "probe" traffic) into the network. Performance is measured by observing the network's treatment of this probe traffic. This allows accurate network behavior, since all protocol rules are being followed by a true multiple-node network, but does not require monitoring of every node in the system. It is important to note that in general the complexity of protocol operations in a multinode network prevents the generation of a statistical equivalent load by a fewer number of nodes, a tempting option to reduce facility size. The selected set of nodes may be chosen either arbitrarily or for the purpose of analyzing specific topological dependencies. Details of the monitoring system can be found in [10].

5. CONTROL SYSTEM

The control system is depicted in Figure 8 which is only a slight refinement of the foreground layer of Figure 3.

It supports message, file, and software transfers from the master to the node emulators. It also supports file transfer to the channel emulator.

An important part of the control system is the real-time modification of experiment conditions and parameters. These are controlled by the master computer via message transfer and may be either user initiated or generated automatically by preprogrammed routines. The real-time fault capability and geographic modulations of mobile radio locations supported by channel emulator are controlled by the master computer in this way. Message generation procedures and protocol parameters can also be controlled in a similar fashion.

The control system will be operated from the monitoring and analysis system's display terminal in order to form an interactive system. One provision will permit the control system to modify the record filtering and processing during an experiment to focus on interesting phenomena. Note that the display terminal will also be used for off-line software development. It may also be possible to couple the described interactive system with the software development environment. This will permit the modification of protocol software to be integrated with the control, monitoring, analysis, and display features.

6. SOFTWARE DESCRIPTION

This section outlines the software systems within the facility. There are two major classes:

1. Permanent software systems (operating systems, control, display, standard analysis packages, software development systems);

2. Parameterized software systems, and per-experiment custom software systems.

From the viewpoint of Figure 1, the software systems can also be organized by the system they reside in. The remainder of this section focuses on this view. Since the channel emulator has no processor but rather is initialized by sending it a configuration file, the only systems to consider are the backbone and node emulator systems.

6.1. Backbone System Software

A number of software systems are required to implement the backbone environment of this facility.

Device drivers have to be written for the SUN-2/170 host to supervise the Multibus control bus. They consist of programs responsible of downloading the operator specified code to the nodes for hardware configuration set-up and initialization of the overall facility. Another function of the device drivers programs is to collect and process the monitoring information stored in each node. The host SUN has to send the collected data to the master computer for further processing.

The programs running on the master computer will include programs that will be used to calculate the performance evaluation parameters from the received monitoring information. Another interactive program will help the user defining the protocol under investigation. Associated with this program is an executive program that will translate the input file to an executable code used for the hardware configuration. Some high level control command will be provided to enable, stop, and modify the running experiment. Database functions that will be used to manipulate the collected data will be available on the system.

6.2. Node Emulator Software

Programs that will generate packets randomly according to some statistical profile will have to be written and would run on the Pacific board.

Each protocol evaluation experiment involves development of a state machine for the link layer and machine level (68010, 68020) software for the other protocol layers.

Debugging node assembly code may be a fairly difficult process. Consequently, all 68010 and 68020 code will be developed using the Magic/L interactive software development environment [9]. In this environment, source programs are written in a high-level structured language and are easily debugged through execution. Programs are then assembled for faster execution.

The developed routines and programs for the node emulator will run under VRTX, a real time operating system [11] that would perform all the task switching functions. VRTX will be stored in an PROM and will reside in the Pacific board.

At a future date, a specification language and "compiler/assembler" for automatic code generation from high-level descriptions will be attempted. This effort is expected to borrow from the developing body of protocol specification and validation knowledge which is currently a topic of great interest.

7. INITIAL CONFIGURATION AND DEVELOPMENT PLAN

The initial 4 node prototyping configuration is illustrated in Figure 9. The 4 node size was chosen as a small manageable system capable of reasonable flexibility. A 4 port channel emulator version is used for the experimental network. The backbone system is simplified through the use of an existing Sun file server and the use of a single Sun-2/170 workstation for both cluster control and master computer functions. All node emulators will be single port versions with

provision for the multi-port configuration. An Ethernet-like protocol will be implemented on the experimental network to validate the facility. After thoroughly debugging the 4 node system, a 32 node version will be designed, employing commercially fabricated boards, gate array, and possibly custom VLSI chips designed within our department. A UC Berkeley VLSI group headed by Prof. R. Broderon is currently designing a chip set for the 32 port channel emulator which could potentially replace the discrete chip design currently envisioned. The special purpose hardware within the node emulator could also be excellent candidates for custom VLSI fabricated within the department.

8. APPLICATIONS OF THE FACILITY

Initial studies would focus on performance evaluation and comparison of existing Local Area Network protocols, such as CSMA/CD, Token Ring, etc., which are relatively well understood. The first goal would be to duplicate known results and behavior, both to verify the integrity and functionality of the facility and for later use as a laboratory instructional tool for research students. It is anticipated that student work in this area using the facility would allow identification of significant enhancements to these existing protocols. In addition, the great flexibility of the system will also permit later studies of new topologies and protocols. Additional research would address bandwidth allocation problems on an optical broadcast network (using scaling), counter-rotating double ring architectures (by using two link layer controller boards per node), and any interconnection of two, or more, networks of different topologies through a gateway.

A key topic for future study using the facility is Integrated Services Digital Networks (ISDN) and Metropolitan Area Networks (MAN). The most direct contribution would be a study of the proposed ISDN S-interface Passive Bus and

Extended Passive Bus. There has been some preliminary study of these arrangements, but much is not known for certain about the behavior of this access system. In addition, ISDN and MANs involve traffic integration, a topic also of great recent interest in Local Area Networks (LAN) and Private Branch Exchanges (PBX). The facility offers support for the flexible study of traffic integration for all of the topologies associated with each of these systems. In particular, the multiport configuration with its multiplexed variable-structure channels and integrated switch fabric can open a whole new dimension in integrated architecture and integrated gateway research.

At an appropriate point in the development of the system, the facility will be made available for laboratory coursework. At a much later time it will be opened to wider use for the study not only of protocols but also of distributed systems, distributed operating systems, and distributed computation. It is noted that the facility is relatively inexpensive with respect to the powerful emulation abilities and flexibility it offers in many areas of research. Once the final 32 node fabrication design is established, it is foreseeable that several copies of the facility can be built so that several independent experiments can be studied simultaneously by different user groups.

9. ACKNOWLEDGEMENTS

The authors are grateful to Professors P. Varaiya and J. Walrand for their suggestions, encouragement, support, and editorial review. Many thanks are also extended to Randy Cieslak for his constructive criticism of this paper. The research is supported in part by NSF grants ECS-8118213/85-06337, ONR contract N00014-80-C-0507, JSEP contract F49620-79-C-0178, and grants from Bell Communications Research and the MICRO program.

10. REFERENCES

- [1] A. Kao, L. Ludwig, "Channel Emulator for the U.C. Berkeley Protocol Workroom Facility," Protocol Workroom Document No. 85-1, February, 1986, UCB, EECS Dept.
- [2] J.O. Limb, C. Flores, "Description of Fastnet-A Unidirectional Local Area Network," *The Bell System Technical Journal*, Vol.61, No.7, Sept 1982, pp. 1413-1440.
- [3] A. Tanenbaum, *Computer Networks*, Prentice Hall, Englewood Cliffs, 1981.
- [4] C. Tropper, *Local Computer Network Technologies*, Academic Press, New York, 1981.
- [5] K. Kummerle, M. Reiser, "Local Area Communication Network-An Overview," *Journal of Telecommunication Networks*, Vol.1, No.4, 1982, pp. 349-370.
- [6] A. Fawaz, L. Ludwig, M. Peck, "Node Emulator Architecture for the U.C. Berkeley Protocol Workroom Facility," Protocol Workroom Document No. 85-3, February, 1986, UCB, EECS Dept.
- [7] R. Cieslak, A. Fawaz, "A General Architecture for Data Link Layer Controllers," Protocol Workroom Document No. 85-4, December, 1985, UCB, EECS Dept.
- [8] R. Yu, "Design of the Link Layer Controller for the U.C. Berkeley Protocol Workroom Facility," B.S. project report, U.C. Berkeley, EECS Dept., December, 1985.
- [9] Magic/L Reference Document (Sun version).
- [10] R. Cieslak, "Monitoring and Control Systems for the U.C. Berkeley Protocol Workroom Facility," Protocol Working Document No. 85-2, UCB, EECS Dept.
- [11] VRTX Reference Manual.

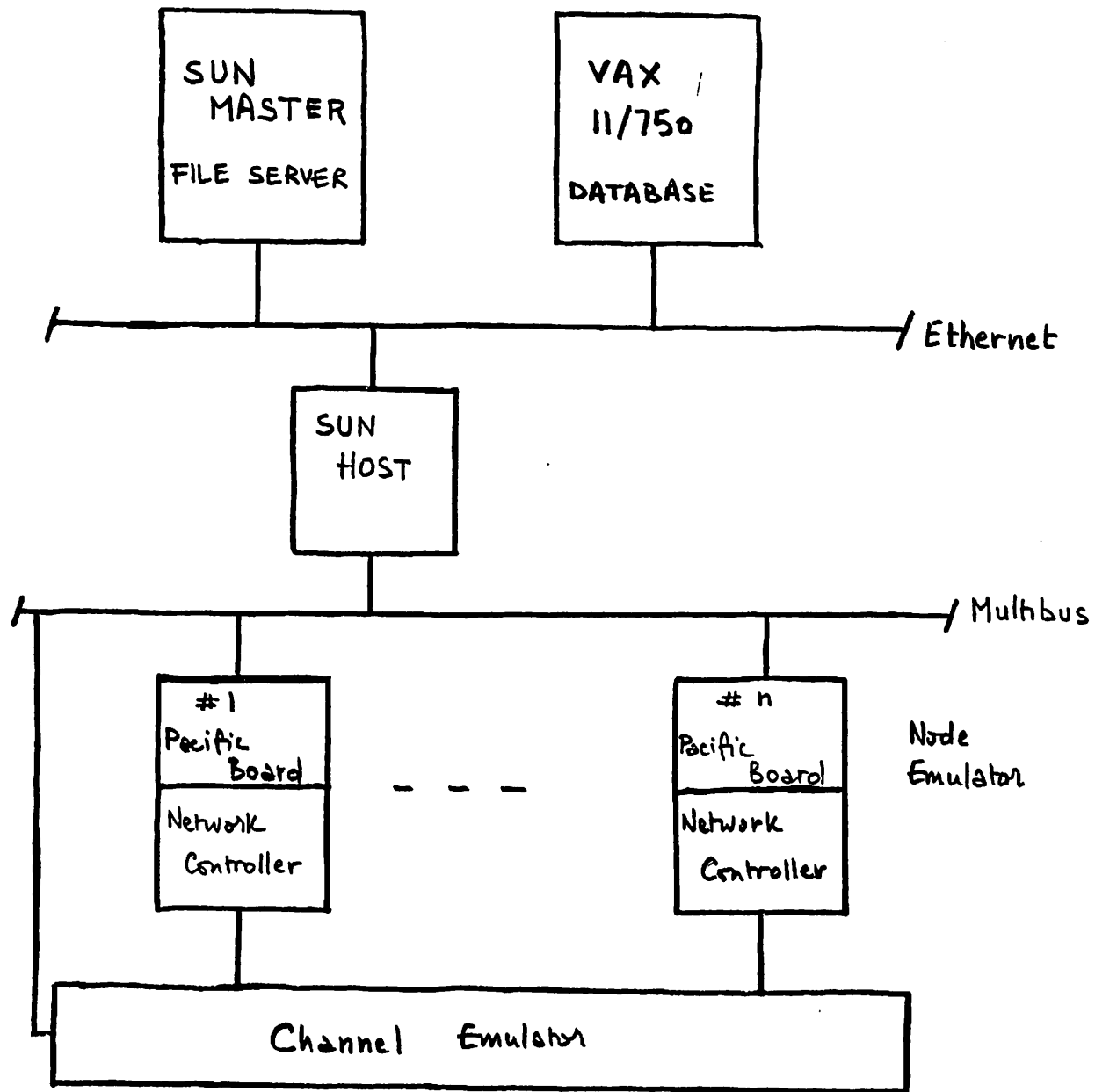


Figure 1. Overview of the Protocol Workroom Facility

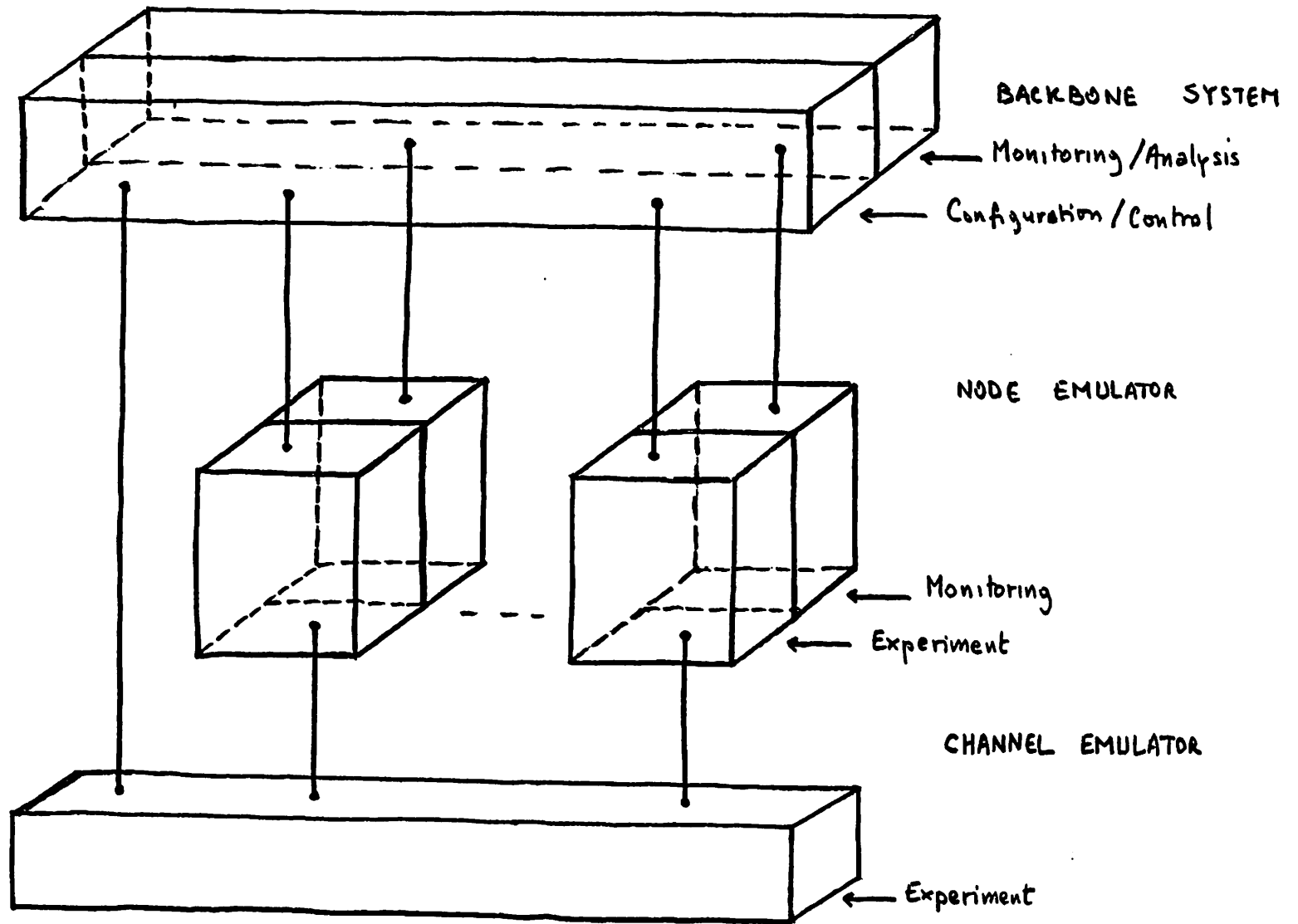


Figure 2. Conceptual Architecture Showing Different Layers of the Facility

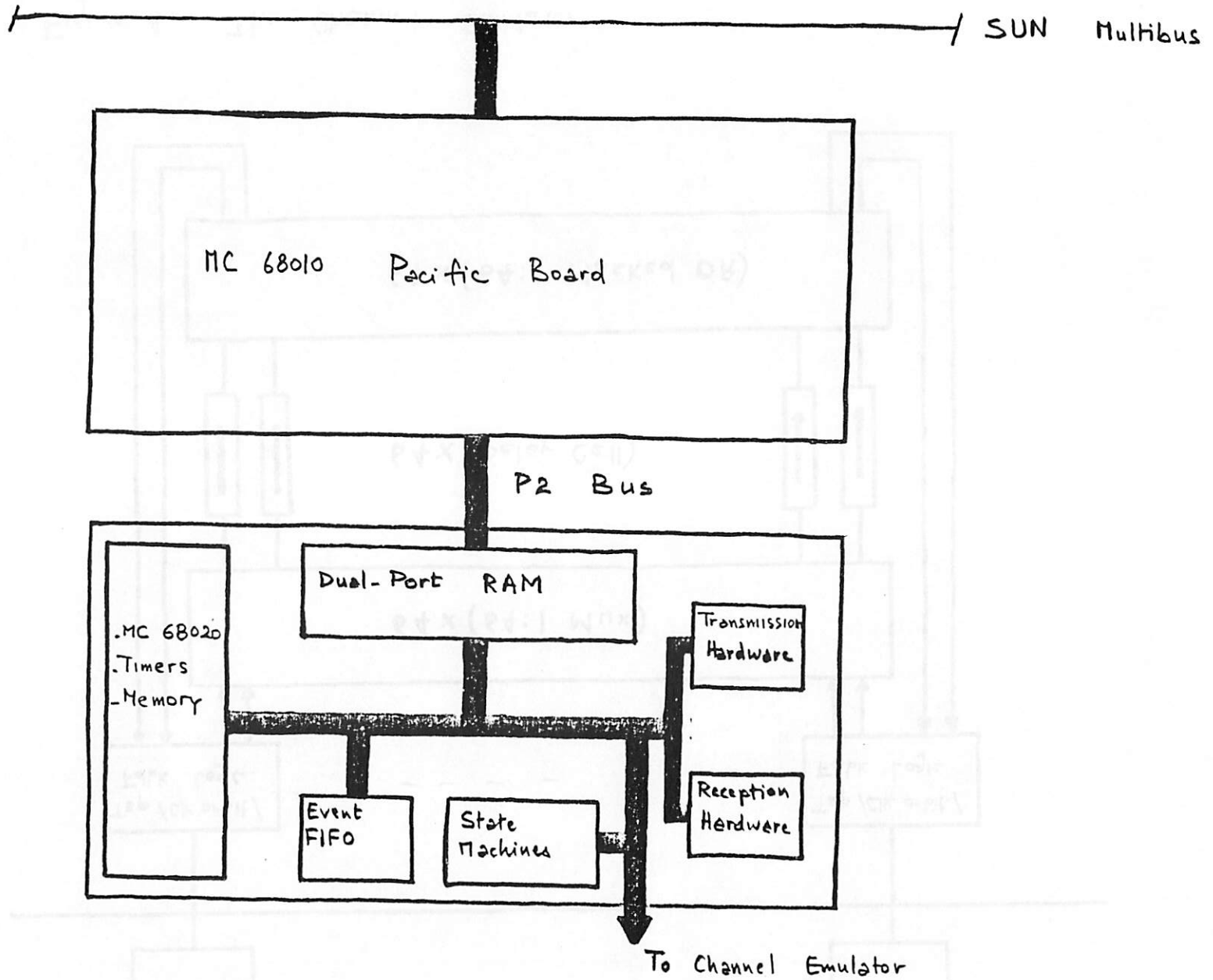


Figure 3. The Node Emulator

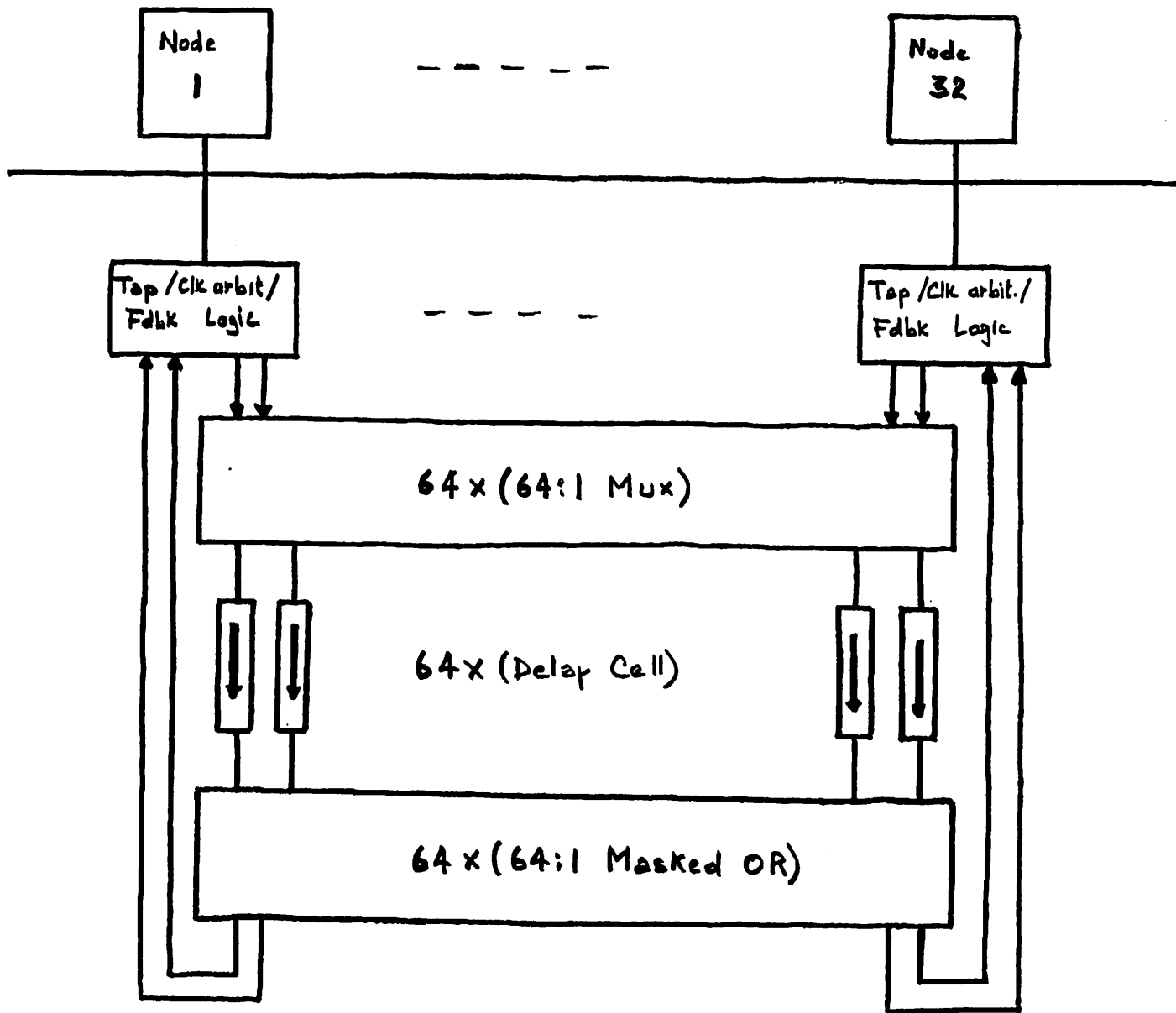


Figure 4. The Channel Emulator

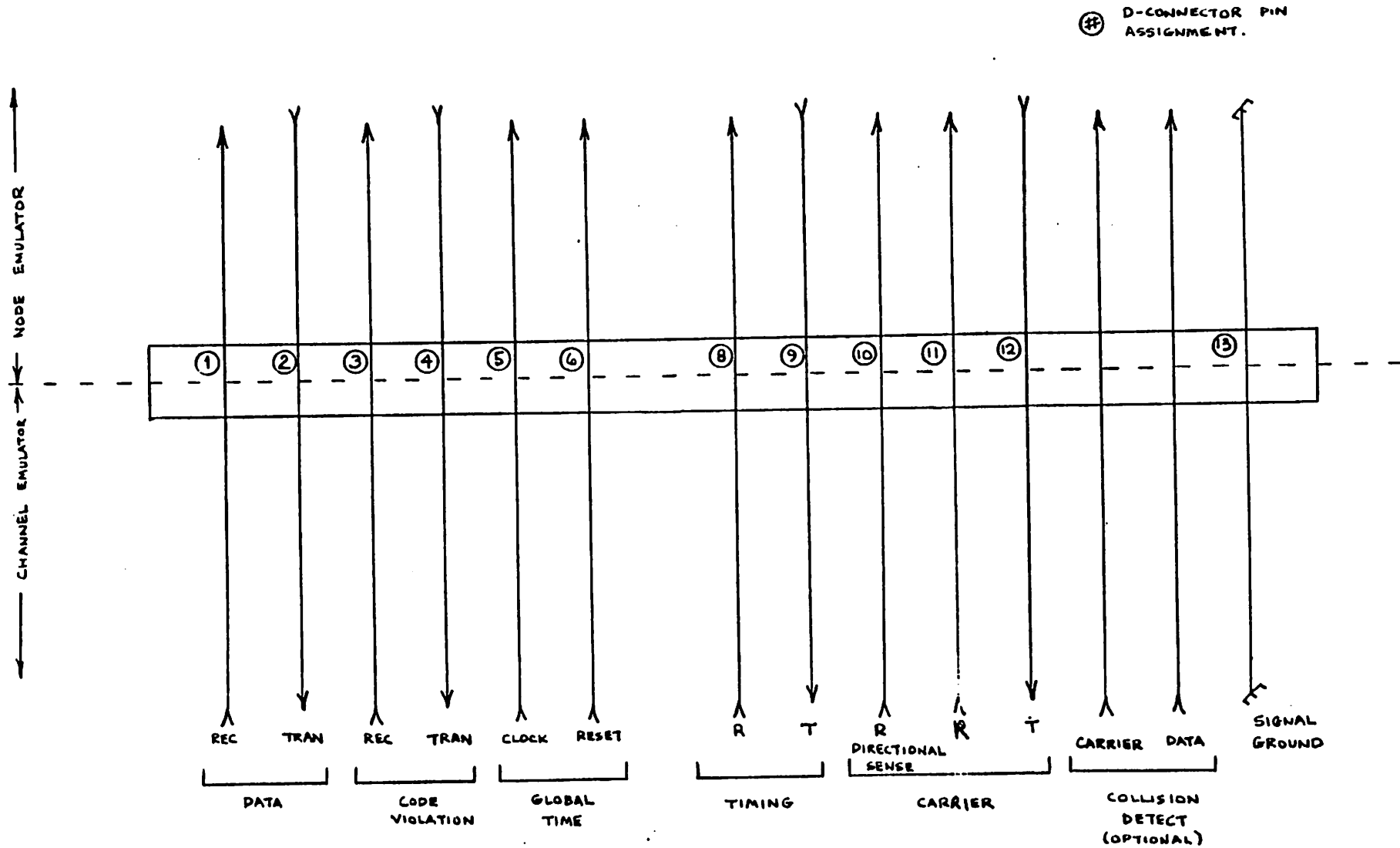


Figure 5. Node Emulator / Channel Emulator Interface

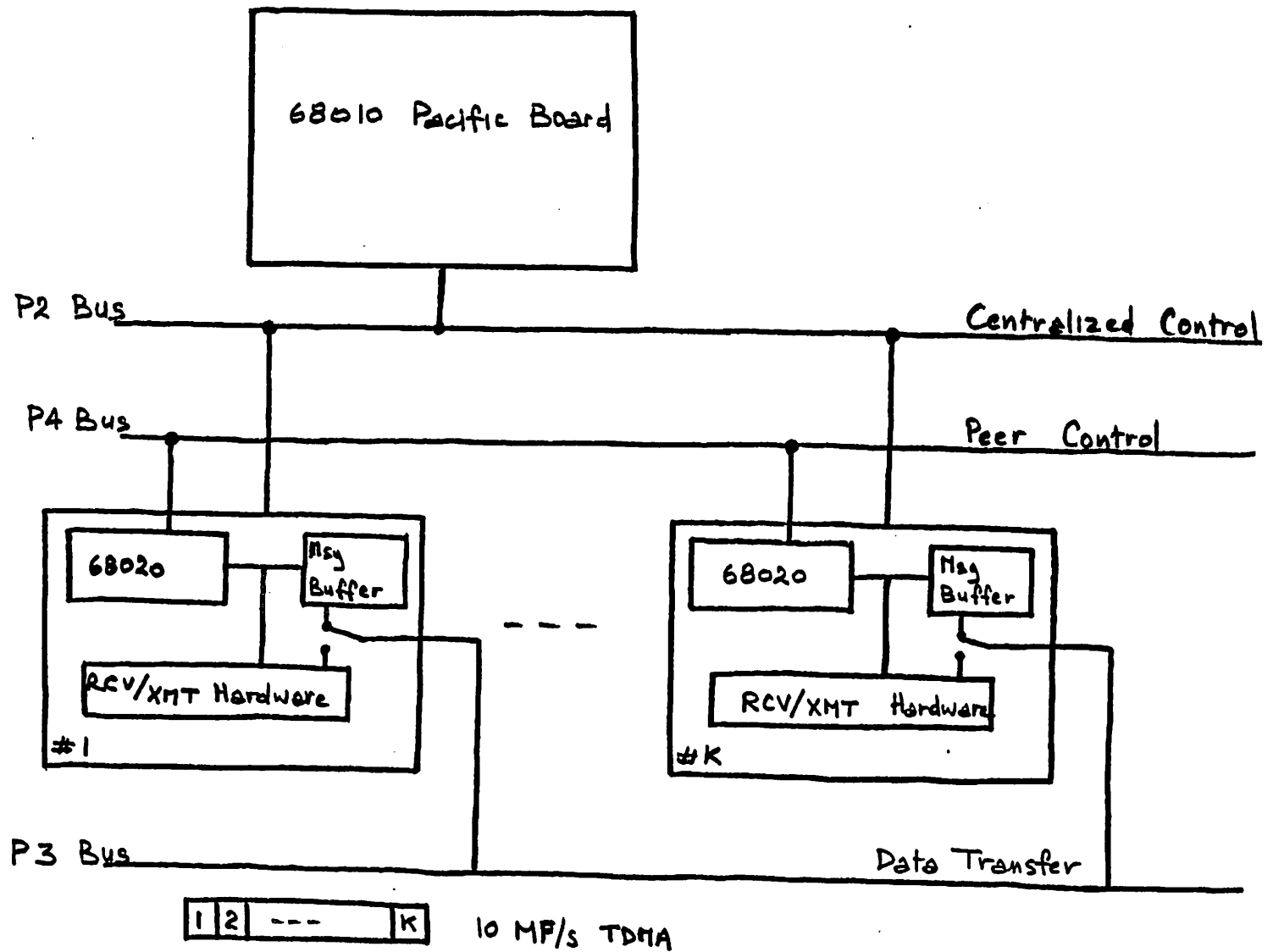


Figure 6. K-port version of the Node Emulator

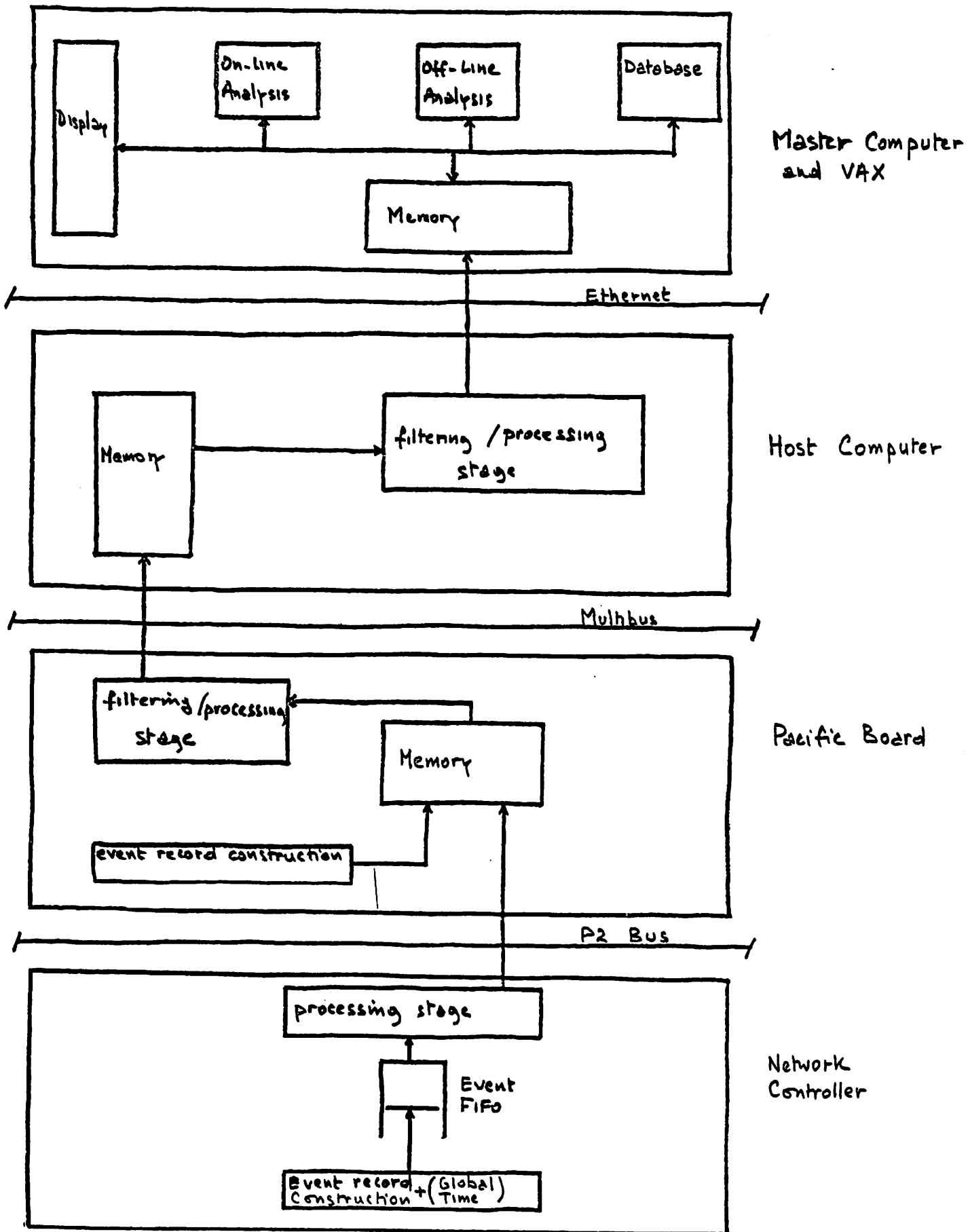


Figure 7. Monitoring System

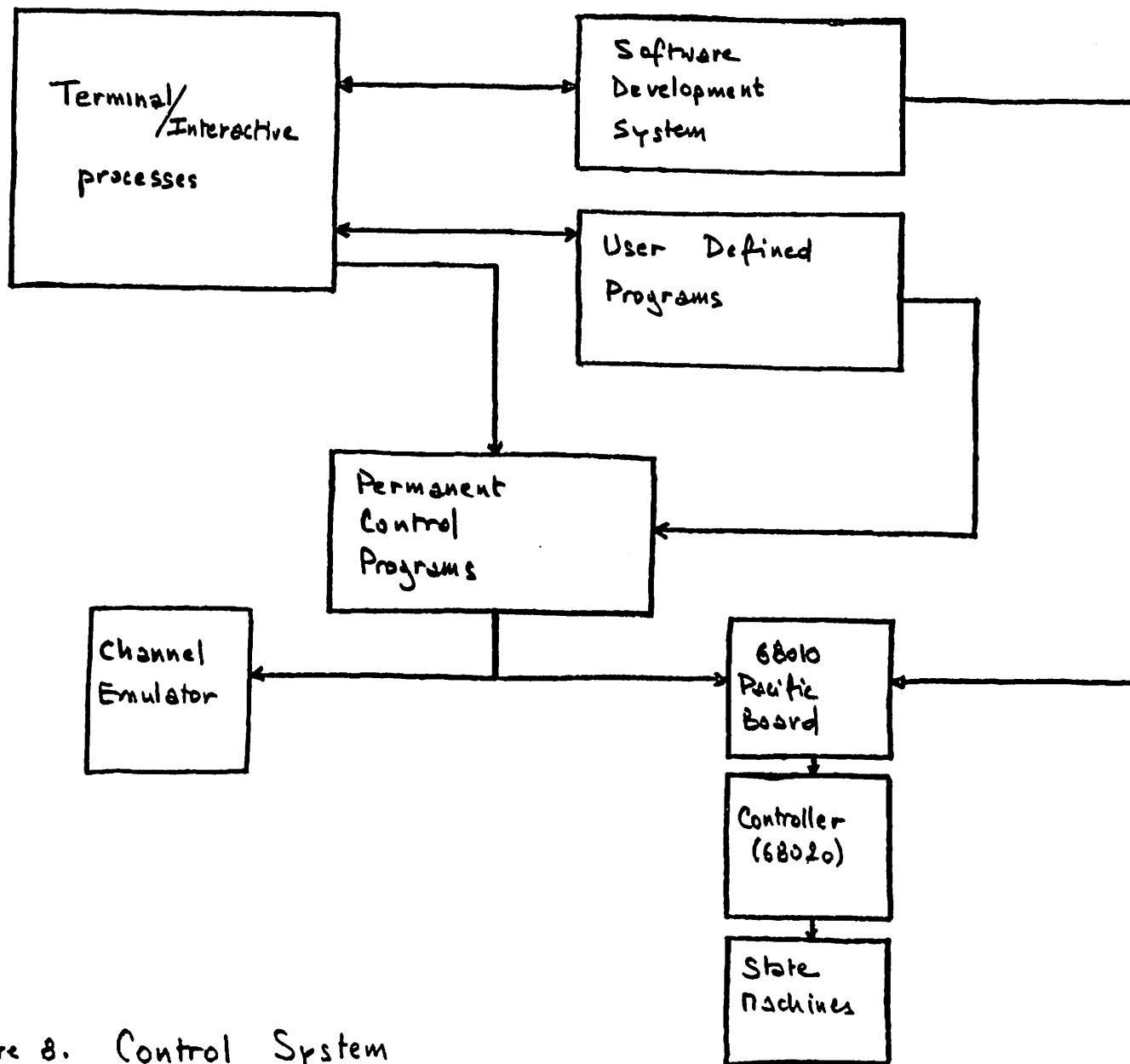


Figure 8. Control System

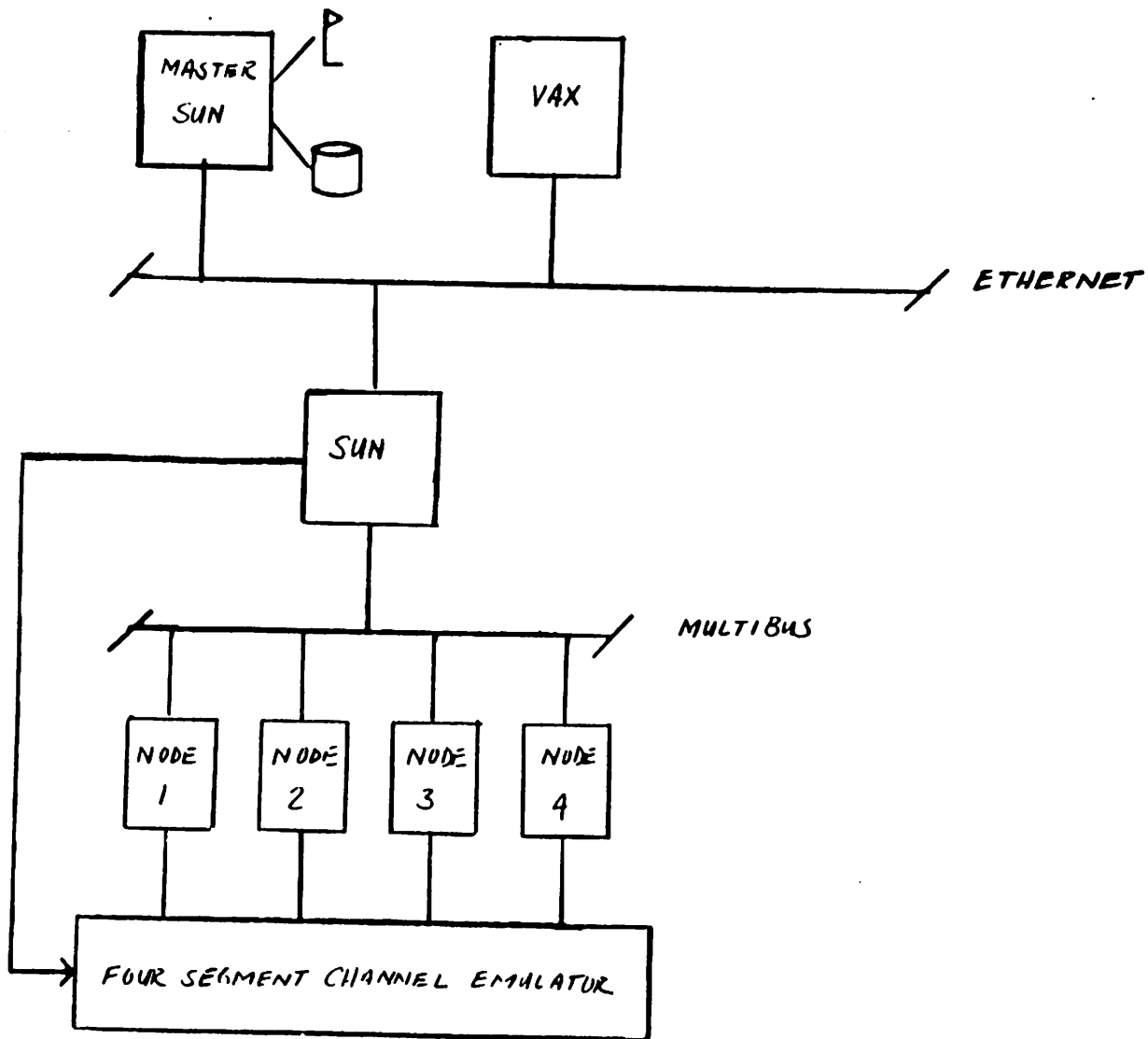


Figure 9. Initial Configuration