Multics Technical Bulletin

To: Distribution

From: R.W.Franklin

Subject: VERY DISTANT HOST PROTOCOL for Multics

Date: August 4, 1976

INTRODUCTION

Multics Arpanet software is to be released with MR5.0 as a HIS standard product offering. It is currently supported by MIT's Computer Research Laboratory but will eventually be supported by a Honeywell development group. In order to connect to Arpanet, a Multics system currently must be placed within 2000 feet of an Arpanet IMP; i.e. only the Local and Distant Host protocols have been implemented for Multics. This is not a very tenable marketing position. In order to expand the marketplace for Multics Arpanet connections a more practical and readily available method of connecting to Arpanet must be offered to our customers. This is achieved by implementing the Very Distant Host (VDH) protocol for Multics; A method of connecting a host to an Arpanet IMP over a leased telephone line.

Multics Project internal working documentation. Not to be reproduced or distributed outside the Multics Project.

OVERVIEW

Multics currently interfaces to Arpanet as shown below.

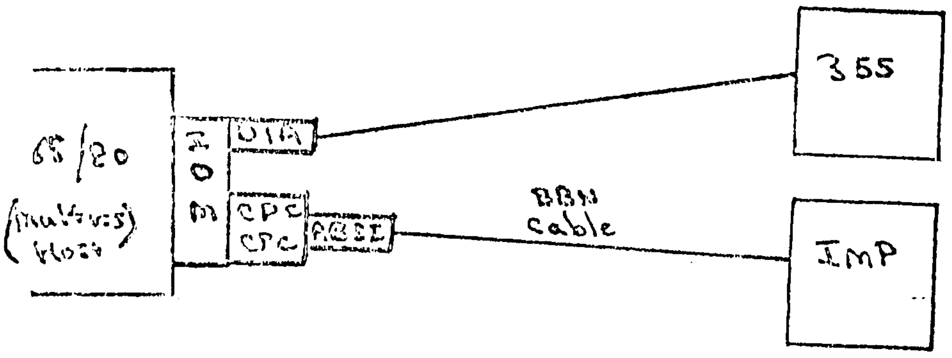

FIGURE 1

A SPecial InterFace (SPIF) board, the Asynchronous Bit Serial Interface board (ABSI), built by MIT, is connected to the IOM using two Common Peripheral Channels (CPC). The ABSI is connected on the other side by a cable to an Arpanet IMP (Interface Message Processor).

There are three possible ways for a host to connect to Arpanet and the picture shown above encompasses two of them. The three ways are:

a. Local Host - The host is connected to an IMP via a cable, less than or equal to 30 feet in length, using BBN (Bolt, Beranek & Newman) specification number 1822.
b. Distant Host - The host is connected to an IMP by a cable, greater than 30 feet but less than 2000 feet, using BBN specification number 1822. I.e. the only difference between the Local and Distant host connections are in the length of the cable and the hardware drivers required to connect a cable over that distance. The software interface for Local and Distant host connections is identical.
c. Very Distant Host - The host is connected to an IMP over a leased synchronous telephone line using an entirely different software and hardware interface. The software components of the Very Distant Host (VDH) interface include the Reliable Transmission Package (RTP) and the Error Detecting Special Host Interface (EDSHI).

The Multics MR5.0 release includes, for the first time, the Arpanet interfacing software. This enables Multics customers to interface to Arpanet if they so desire. However, our present offering of only the Local and Distant host interfaces make this an unrealistic offering. It would be unlikely to have a customer who desires to connect to Arpanet and who is located within 2000 feet of an Arpanet IMP that has an available slot. In order to make our Arpanet offering viable to our customers we must implement the VDH protocol and offer it as a product.

DESIGN GOALS

-The Multics TTYDIM and the Multics Arpanet Network Control Program (NCP) are currently separate and distinct entities; i.e., a new release of one does not affect the other. This is good as they are being written, modified, enhanced, etc., by two separate and distinct software groups. A design goal is to keep it this way.

-Multics communicates with Arpanet through the IOM directly to the network IMP while normal Multics TTY communication uses the Front-End-Processor (FEP). The Multics Communications System (MCS) resides in the FEP and leaves very little room for anything else. A design goal is to keep the Arpanet software removed from the FEP.

-A design goal is to implement VDH in a way that affects the existing NCP as little as possible and hopefully not at all.

-A design goal is to implement VDH in a way that allows two Multics systems to communicate, via the VDH protocol using all existing NCP functions, without requiring a network to exist; i.e., no IMP's required at all.

-A design goal is to implement VDH in a way that allows a Multics system to communicate, via the VDH protocol using all existing NCP functions, with a foreign system that can currently interface to Arpanet. If the foreign system has implemented VDH then it would be a direct VDH to VDH connection. If the foreign system has only implemented the Local or Distant host interface then it would connect to our VDH board on its system via its BBN specified special interface cable. Another design goal in this item is to not affect the foreign systems software at all.

-A design goal is to fit the entire VDH package (RTP and EDSHI) on one MQX universal board plugable into the IOM.

The Decision Making Process

Which way to go

When this project was first conceived, four possible ways to implement the VDH protocol were considered. They were;

CASE 1 Put all of the VDH code in the 355 (MCS). This includes the RTP and the EDSHI (that which handles the synchronous line). To do this, major changes would have to be made to the MCS TTYDIM and the NCP DIM to allow the ARPA-355 portion to communicate with the current ARPA NCP via the DIA.

CASE 2 Put the EDSHI code in the 355 and the RTP code in the 6180. I.e. split the VDH functions between the two pieces of hardware. This still involves major changes to the NCP to allow it to converse with the RTP instead of with the ABSI as it now does.

CASE 3 Put the entire VDH package in a micro computer and directly connect it to the IOM via:
a. the ABSI or
b. a PSI channel or
c. the CPC's directly or
d. a DIA channel

CASE 4 This is the same as case 3 except that a mini computer would be used instead of a micro computer. Only two minis could be given serious consideration and they are the HIS 316 (currently the Arpanet IMP) and the HIS Level 6.

The design goals were then matched with the four possible implementation methods and the below pertinent points were further considered.

-Currently the ARPA software doesn't impact MCS355 software or any portion of the TTYDIM at all. It is maintained entirely separate from MCS. This is true even to the extent that complete rewrites of one do not affect the other. Also ARPA 6180 software is only present when the Network Daemon is up, it is entirely pageable, and it will run regardless of whether the 355 is up. We want to continue to enjoy these relationships. Implementing Case 1 or 2 violates this goal.

-There is a limited market for the VDH interface. In this light alone it doesn't seem reasonable to "kludge" up MCS and the current TTYDIM to include ARPA functions. This would cause problems in maintainability, in fitting them together, and even to finding room in the 355 to include the code and buffers required in cases 1 and 2. Implementing either Case 1 or case 2 requires many changes to the 6180 code to interface ARPA-355 to ARPA-6180 via the TTYDIM. Implementing Case 1 is probably not even possible due to 355 memory constraints alone. Implementing Case 2 still requires changes to MCS355 to accommodate the

50KB lines and a form of binary synchronous communication; In addition, it will probably also encounter buffer allocation/management problems in the 355. Implementing Case 1 or 2 requires two different versions of NCP code based upon whether the VDH or Local/Distant host interface is to be used.

-Points in favor of implementing Case 3 are:

1. It requires <u>NO</u> changes to <u>ANY</u> existing software.
2. It is not dependent on the existence of the front-end processor at all.
3. The difference between VDH and Local/Distant host software reduces down to the presence or absence of the micro computer. Nothing else.
4. It's inexpensive.
5. Existing micro computers are in-house and available for implementation and testing. There are at least four or five other applications, in-house, that are successfully making use of an INTEL 8080 microprocessor chip. Also available is EPROM writing and erasing equipment.
6. Higher level languages are available on the micro computers. E.g. INTEL 8080 offers the PLM compiler while Signetics 2650 offers the PLS compiler (both subsets of PL1). Cross-assemblers, cross-compilers, and simulators are available for the INTEL 8080 on GCOS systems S, X, and T and are being moved to Multics system M. A cross-assembler and a simulator for the Signetics 2650 are available on Multics System M and the PLS cross-compiler will be placed there when it becomes available to us.
7. We could begin design and implementation immediately with no changes to MCS or hardcore required now or in the future.
8. The micro computer will fit onto one MQX Universal board and can plug or be cabled directly into the 6000.

-Some points against implementing Case 3 are;

1. It requires us to purchase someones else's product as a "breadboard" prototype system on which to do some initial development. However this point is minor and we can equate it to the procurement of a terminal which is done with regularity. By the time we market it as a product it will be Honeywells all the way.
2. It will require some hardware design to place it on a board and interface it to either the ABSI, CPC's, DIA, or PSI.

-Some points against implementing Case 4 are:

1. It is more expensive than Case 3.

2. A mini computer cannot be placed on one MUX Universal board as can the micro computer.

-Some points in favor of implementing Case 4 using the HIS 316 are:

1. No hardware changes are required as the 316 currently interfaces to the ABSI in its role as an IMP.
2. It is possible that no software changes would be required if we loaded the 316 with the "Public Domain" software written by BBN for the IMP. It could be put into the 316 and the VDH portion reversed. I.e., the 316 would be operating as part of the host and yet contain the standard IMP software. It would then talk to an IMP using the VDH code (normally it talks to a Host using the VDH code).
3. Both the 316 and the Level 6 have the advantage of using our own hardware exclusively.

-Some points in favor of implementing Case 4 using the HIS Level 6 are:

1. Other HIS systems are swinging towards using the L6 with a DIA interface. The L6 is to have the HDNA communications management functions within it. There is great potential in using the HDNA HIS standard FEP. This FEP is to be used on GCOS III, GCOS 66 and WWMCCS.
2. There are terminal controller products tying in to the L6 which might eventually give Multics a new FEP for HDNA terminal control.
3. Sizeable manpower can be devoted (CII-HB) to L6 implementation of an HDNA FEP. The actual software effort necessary to provide ARPA specific VDH in L6 could be less than the stand-alone effort. By this it is meant that the operating system, its maintainability and other details probably can be provided once for both ARPA VDH and HDNA.
4. The HDNA DIA-6000 interface will be a "letter" interface (corresponds to ARPA messages). The same process of fragmentation and reassembly are done in both, just some different algorithms.

-The implementation of the VDH protocol, using Case 3 or 4, provides us with the capability of connecting two Multics systems together, using the already existing NCP programs, without requiring an actual Arpanet. I.e., a VDH to VDH connection with no network.

-Implementing the VDH protocol using Case 3 or 4 and going directly into the ABSI provides us with the capability of connecting a Multics system to any foreign system that can interface to Arpanet. This allows both systems:

1. to connect without any software changes
2. to connect without requiring an actual Arpanet.
3. To utilize their existing NCP programs as is.

Some Basic Decisions

The first decision made was to implement the VDH protocol on Multics using a micro computer; More specifically, the INTEL 8080 micro computer. Once that decision was made we ordered and have in-house an INTEL SBC-80/10; a Single Board Computer with 4K of PROM and 1K of RAM memory, 48 parallel I/O lines and a USART for serial communications. The SBC-80/10 was ordered in a package (SBC-80P) which also included a rack to mount four boards, an extra prototype board, all cables for power and to interface to teletype or RS232C compatible devices, and a full complement of EPROMS, I/O line drivers and terminators. Separately ordered was the SBC-016, a 16K RAM board addition to the SBC-80P. This combination gives us a 17K RAM, 4K PROM system as a breadboard model on which to implement and test a VDH program. It is connected and running with an INTEL supplied PROM monitor/debug package.

A second decision was to package the micro computer on an MQX Universal board which is plugable into the IOM. This allows a Multics system to communicate with another Multics system without requiring a network (i.e. no IMPs) yet still use all of the NCP functions. e.g...

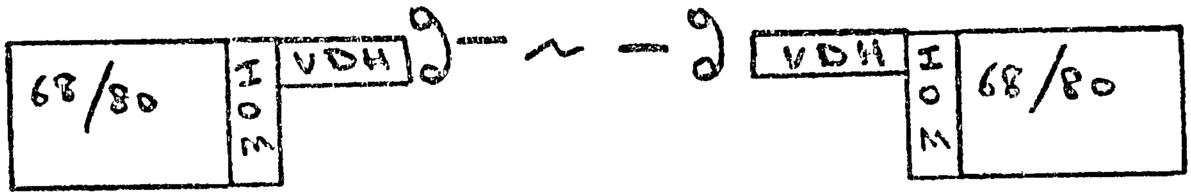

FIGURE 2

The reasons for this decision are primarily lower expense and easier packaging.

A third decision was to interface the micro computer to the ABSI board supplied by MIT. This is a more specialized subset of the second decision. It allows a Multics system to communicate with a Multics system as shown above but it also allows a Multics system to communicate with a foreign system (e.g. an IBM 370) as shown below (SPIF = Special interface; i.e. ABSI for Multics).

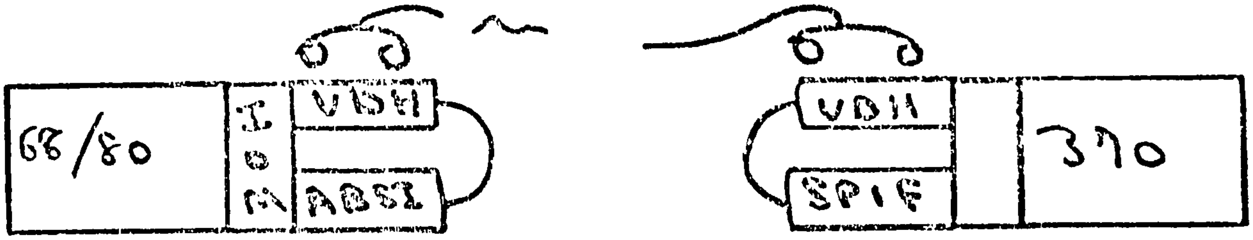

FIGURE 3

This decision is based on a number of reasons which include:

a. There are NO changes to any existing Multics software.
b. it allows Multics to interface to a foreign system (one that can interface to Arpanet) without ANY hardware or software changes to the foreign system. No other method allows this.
c. The hardware logic to interface The micro computer to the ABSI is much simpler than that required to interface it to a PSI, DIA, or CPC channel. The hardware logic will easily fit on less than half of an MQX Universal board which is a requirement if we are to achieve the design goal of one MQX Universal board.

A fourth decision was to pick a current in-house application which uses an INTEL 8080 processor chip and adapt their board to this application. We are currently working with an engineering group that has a prototype board built and running. It is packaged on an MQX Universal board and currently one-half of the board encompasses an INTEL 8080 computer with four USART lines and 4K of RAM memory. The other half of the board is available for logic to interface it to the IOM (in our case, the ABSI).

Other decisions which are pending include:

a. Should the micro computer have a second processor to handle the 50KB line with its associated binary synchronous discipline and 24 bit CRC or should it be a single processor system with a DMA attachment to the USART?
b. Should the timeouts be software loop controlled or should we incorporate an interval timer chip?
c. How many USART's do we actually need? Do we need a different clock to drive the 50KB synchronous USART?
d. Does the logic to drive the ABSI run in a DMA fashion or is it polled or interrupt driven?
e. Should the program be in PROM or RAM memory?

Hardware/Software Overview

For discussion purposes, the below drawn diagram depicts the micro computer as it is functionally laid out on an MQX Universal board.

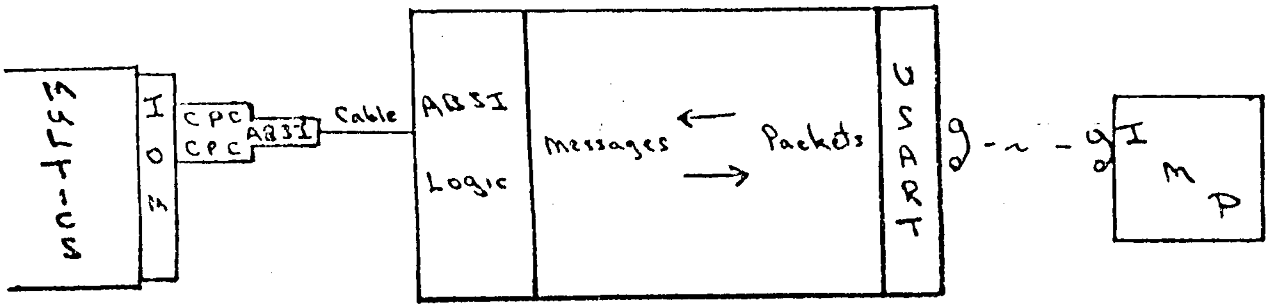

FIGURE 4

The hardware ABSI logic follows BBN specification 1822 and is concerned only with messages. A message is defined to be a bit stream, up to 8096 bits in length, which includes a 32 bit leader. The hardware logic on the micro computer MQX Universal board that deals with the ABSI transmits or receives only messages. The data content of the message that is transmitted or received is not looked at by the hardware ABSI logic.

On the other side of the micro computer MQX Universal board is the USART (Universal Synchronous Asynchronous Receiver Transmitter) chip which is used to communicate over a 50KB synchronous leased line to an Arpanet IMP. Its' interface is defined in BBN specification 1822 Appendix F and is concerned with the transmission and reception of packets using a binary synchronous format. A packet is defined to be a bit stream, up to 1024 bits in length, which includes a 16 bit control word.

Note that the Multics system and the ABSI side of the board deal strictly with messages while the IMP and the USART side of the board deal strictly with packets.

The software in the micro computer concerns itself with the following areas:

a. The handling of Special Packets. The special packets bring the 50KB line to the IMP up, declare it dead, and generally are concerned with knowing the status of this line.
b. The handling of messages to and from the ABSI.

c. The handling of normal packets to and from the IMP.
d. The conversion of a message to packets and vice-versa.
e. The driving of the USART and the driving of the hardware logic to the ABSI.