

UC Irvine

ICS Technical Reports

Title

Design of network access arrangements

Permalink

<https://escholarship.org/uc/item/8294c5dm>

Authors

Mockapetris, Paul V.

Lyle, Michael R.

Farber, David J.

Publication Date

1976

Peer reviewed

Design of Network Access Arrangements

Paul V. Mockapetris
Michael R. Lyle
Professor David J. Farber

Notice: This Material
may be protected
by Copyright Law
(Title 17 U.S.C.)

This research is supported by ARPA, through the Information
Processing Technology Office, under contract
N00014-76-C-0954.

ABSTRACT

This paper describes the design parameters and the approaches used in the construction of a Network Access Arrangement (NAA). The NAA connects a network node (host) to the communications medium. The details of the design and implementation are presented, along with some rationale for the design choices. The design is biased towards implementation in Large Scale Integration (LSI), and allows variation in message format and transmission protocol.

This freedom is explored in regard to building both ring networks, such as the Distributed Computer System (DCS) [1], and contention nets, such as Ethernet [2]. The NAA will initially be used to implement access to a ring network, similar to the DCS at the University of California, Irvine. This application is examined in detail.

1. INTRODUCTION

The Distributed Computer System (DCS) is a local network which was implemented at the University of California, Irvine under NSF auspices. DCS provides a general purpose timesharing system based upon 16 bit minicomputers and a ring communications system. The design of the Network Access Arrangement (NAA) is heavily influenced by the successes and shortcomings of the present DCS system and its Ring Interface (RI).

The RI presently used by DCS was implemented in 1972 [3], and consists of approximately 140 TTL packages. The new RI, built using the NAA, will consist of the NAA LSI package, power supply, line drivers, etc. A prototype of the design presented in this paper has been implemented in TTL MSI, and is currently undergoing test. The prototype uses a data rate of 1 MHz. We anticipate a data rate in the range of 1-5 megabits for the LSI version.

The basic structure of the ring is simple: multiple NAAs are connected in a ring. The twisted pair interconnects carry bit serial data in a unidirectional manner. To transmit a message, a NAA sends it around the ring once. The NAA appears to the host to be a high speed, full duplex device.

In the remainder of the paper, we will discuss the implementation and design details of the NAA. First, we describe the logical message format, the transmitted message format, and the transmission system. Next, an overview of the design implementation is presented, followed by a discussion of some implementation considerations. Finally, we discuss some extensions and expansions to the capability of the NAA.

2. LOGICAL MESSAGES

A logical message consists of the data fields used by a process to send a message; it does not include fields added by the hardware during transmission. The logical message format is shown in figure 1A.

The DATA field is the actual message text. The LEN field specifies the length of the DATA field in bytes. The NAA places no restriction on the length of messages other than the maximum length representable in the 16 bits of the LEN field. In practice, software protocols are used to break long messages into packets.

In the present DCS, messages are addressed to processes rather than location; this frees multiple process systems from any restrictions based on process location. We feel that this is a very valuable feature, and have preserved

process addressed messages, while enhancing the addressing options [4].

The Destination Process Name Mask (DPNM) and Destination Process Name (DPN) specify the destination(s) of the message. The Originating Process Name (OPN) is the signature of the transmitting process. Each NAA has a list of process name specifications stored in its Name Table (NT). The NT is composed of a variable number of entries; each entry consists of a 32 bit name and a 32 bit mask. The NT entries are used to identify processes in the attached host. Whenever a message passes a NAA, the NAA compares the message address represented in the message DPN and DPNM fields against the list of address specifications in the name table. The comparison is done in parallel to all names in the name table. If the message is addressed to a process in the attached machine, the comparison will generate a match, and the NAA will attempt to copy the message.

The name table comparison is done one bit at a time, using the corresponding bits from each of the four fields. For a name to match, all 32 bit positions must match. For a bit position to match, either the corresponding bit in the DPNM or NT mask can be on, or the corresponding DPN and NT name bits must be equal. This system has the effect of restricting the name comparison to those portions of the two name fields that are not masked off by either mask. In the

present DCS system, names are broken into encoded components by software convention. Thus the mask fields can be used to override comparison of specified components; the bit mask allows host software to define the partitioning of 32 bit names into components.

3. MESSAGE PACKAGING

When a NAA is instructed by the attached host to transmit a message, the NAA packages the message by adding the information necessary to implement the actual transmission. Figure 1B shows a logical message after packaging.

The connector field is inserted before the message to enable synchronization to the start of the message. The CRC is a conventional cyclic redundancy check over all fields of the logical message.

The match and accept bits are used to provide the sender of a message with an indication about the success or failure of a transmission. These fields are initially set to zero when the message is transmitted. As the message passes each NAA on the ring, the NAA compares the message address against the contents of the NAA's name table. If the message address matches, the message is addressed to a process in the attached host; hence the NAA attempts to copy

the message. The receiving NAA ORs a one into the contents of the accept bit if the message is successfully copied; a one is ORed into the match bit if the copy operation fails. The copy fails if the attached host has not enabled the NAA for input, if the CRC in the message is bad, etc. Note that neither bit is changed if the message is not addressed to the attached host.

As the message circulates around the ring, all NAAs which recognize the message address OR a one into either the match or accept bit. The transmitting NAA records the setting of these bits in the returning message. The four possible outcomes are shown in figure 2. The CRC of the returning message is also checked. If the CRC is bad, the match and accept bits are potentially meaningless; for example, the address specification may be damaged.

The present DCS system uses the match and accept bits to drive the sequencing protocols. This system works reasonably well, but is susceptible to error if the match and accept bits are damaged by transmission error. The match and accept bits cannot be conveniently protected by the message CRC. If this were done, additional inline delay might be necessary. The invariance of the CRC also allows the sending NAA to remember the CRC it used to transmit the message; as the message returns, this stored CRC can identify the returning message. This is a valuable error

check used in the transmission protocol. To increase the reliability of the match and accept bits, a separate Match / Accept Error Check field (MAEC) is added after the match and accept bits. The MAEC field may be expanded to allow the desired level of protection for the match and accept bits.

4. TRANSMISSION SYSTEM

The transmission system used to send packaged messages around the ring is designed to use simple bit serial channels, require a minimum of delay in each NAA, and to coordinate access to the ring. These goals are accomplished by using a unique bit pattern to preface synchronization points, and by allowing only one NAA to add a message to the ring at any time. Note that although only one NAA can transmit at any given time, the number of messages in transit is limited only by ring size.

Two synchronization patterns are used. Both patterns start with a unique bit pattern (8 ones). The bit after the unique pattern differentiates between the two synch marks. The two synch patterns are the connector and token patterns. In order to guarantee the uniqueness of the eight ones pattern, all message fields are encoded using a bit stuffing system similar to IBM's SDLC[5].

When the ring is idle, the token continuously circulates around the ring. When a NAA wishes to transmit, it waits for the token to arrive. As the token passes the NAA that wishes to transmit, the NAA changes the token into a connector. Following the newly created connector, the rest of the packaged message is output. After the message, a new token is output. If two adjacent NAAs transmit, a message train of two packaged messages followed by a token will pass out of the second NAA. This is shown in figure 1C. No NAA can be locked out of the communication medium by heavy traffic load; the system saturates gracefully.

When a NAA outputs a message, it remembers to delete it as it returns around the ring. The NAA does so by deleting the first message on the next train.

Errors in the transmission system may destroy synch patterns. Spurious patterns may also be created. When a host queues a message for output, it also starts a watchdog timer. If the token is destroyed, the NAA never outputs a message, and the timer goes off. The host then requests that the NAA create a new token. Note that several NAAs may timeout simultaneously. If this happens, the tokens either destroy each other, or create multiple message trains. If the tokens destroy each other, the timeouts will occur again, and a new token restart will be attempted. Timeouts are set differently in the different hosts, hence the

recovery will eventually succeed. This system is used in the current DCS system, and works well.

Multiple tokens may be destroyed as a message is sent. When a message is output, the transmitting NAA waits to delete the message as it returns. If two tokens are circulating, the transmitting NAA will attempt to delete the message, if any, in front of the other token. While attempting the delete, the NAA notices that the returning CRC does not match the CRC it remembered transmitting. The NAA responds to this condition by deleting two messages. This policy quickly "crashes" all the running message trains, and a restart results. This scheme seems drastic, but is similar to the behavior of a contention net on every message transmission.

If a connector is destroyed or created, the multiple deletion scheme quickly repairs the damage or crashes the message train.

5. DESIGN OVERVIEW

A block diagram of the NAA is shown in figure 3. The solid lines represent the main data paths in the NAA. All internal data paths are bit serial. The dotted lines separate the NAA into logical sections.

The top section of the NAA is the transmission system. The ring itself is twisted pair, and the LR and LD components are standard TTL current mode line drivers and receivers. The clock extractor constructs a clock from the incoming signal and a local crystal. The output selector is a one bit buffer and associated gating to control the source of the output signal. When the NAA is forwarding a message, the ring input is copied. The output is controlled by the NAA output controller when a message is added to or deleted from the ring. A path also exists from the input side of the NAA to allow setting of the match and accept bits.

Both the input and the output controllers have a Programmed Logic Array (PLA) and a constant table to control sequencing. The two halves function independently. The constant table defines the length of the various fields in the message train. To change the message format, only this table need be changed. For example, if eight bit names and masks were desired, two entries in the each constant table would be changed. The PLA code defines the existence, ordering, and use of the fields in the message train. For example, if a more redundant and reliable MAEC field were desired, only the PLA code would be changed. We are developing methods for loading the PLA code and constant tables under program control.

The input controller uses the synch detector, length counter, and CRC check blocks to process incoming messages. Note that the bit unstuffer restores transparency to the message for processing. The name table is used to recognize messages addressed to the attached host. An 18 byte First-In First-Out buffer (FIFO) is used to buffer the data. This internal buffer can be extended by external buffering. The FIFO output would normally be connected to a host channel, and use DMA transfer. However, the 18 byte FIFO is adequate to hold one complete trivial message. This is done to further the aim of using the NAA to interface terminals and other "stupid" hosts directly to the ring. It also allows the attached host to use programmed data transfer, or a DMA channel with a rate mismatch to the NAA speed.

The output side of the NAA is similar in nature to the input side. The bit stuffer preserves the uniqueness of the synch pattern. The output FIFO, in addition to performing functions analogous to the input FIFO, allows the header of a logical message to be constructed independently of the data section. In a minicomputer environment, this frees the host of unnecessary copy operations into system buffers, as well as reducing the buffer space requirement.

The section in the middle bottom of figure 3 is the interface for command and status sensing purposes. This section also provides host interrupts, a mechanism for

controlling name table contents, etc. Note that this path will in general be completely separate from the input and output data paths.

6. IMPLEMENTATION CONSIDERATIONS

The reliability of a ring system will depend upon the reliability of a NAA and the reliability of the communications lines. Various options exist with regard to communication line failure. These usually consist of some scheme to bypass failed sections of the ring. The use of twisted pair makes this feasible, as there is no complex termination procedure, as is the case with coaxial cable.

A NAA failure must destroy the forwarding section of the transmission logic to impact the ring severely. Hence, the number of components in this section has been minimized. However, an NAA may still fail, or lose power. A relay automatically bypasses the NAA when power fails, and may also be used to take the NAA offline.

Opto-isolators decouple the NAA from the ring. This protects the NAA, as well as dealing with the problem of ground potential differences between NAAs. The line drivers can drive approximately 1 kilometer of twisted pair. The maximum recommended distance is half that, so that any NAA may be bypassed without difficulty. Of course, active repeaters (essentially NAAs) can be used.

7. CONFIGURATION VARIATION

The initial LSI target is two chips: the NAA chip with one name table entry, and a name table expander chip. Including only one name on the NAA chip will minimize chip area. this is important as chip yield (hence cost) falls off rapidly as a function of area. The one entry will allow us to experiment with using terminals, disks, etc. as hosts.

The name table system allows easy expansion of the name table; the use of look ahead logic on/between expander chip(s) will allow a very large number of names. The expander feature will also allow the use of different name table strategies. The attached host need only disable the name table entry on the NAA chip. We have done some preliminary work with regard to network monitoring [6], and in using name variation to enhance net security [7]. Other potential uses [4] include sequencing, construction of inexpensive gateways, and automatic packet reassembly.

We are considering using the NAA to implement a contention system similar to Ethernet. The message formatting components can be adapted by changing the PLA and constant table. A new transmission system is required to drive coaxial cable, as well as a collision detection system. The collision detection may be processed in a

similar manner to unexpected ring tokens. Other parts of the ring protocol, such as the match and accept bits will be eliminated. The changes consist mostly of making external synchronization information available to the PLAs from the outboard transmitter and receiver.

8. CONCLUSION

The trend to distributed systems has created the need for a flexible, low cost communications medium. Conventional MSI TTL approaches result in a interface cost in the \$1500-\$3000 range, but may be difficult to modify. A microprocessor system may be flexible but does not support a high data rate or sophisticated associative addressing. Custom LSI shows promise in this area. The best estimate for the NAA chip is in the \$15-50 range in quantities of one thousand. This yields a total cost estimate for a freestanding network interface of approximately \$300. This design allows protocol experimentation, as well as implementation of network access for low intelligence hosts. These capabilities will prove very valuable in future distributed systems.

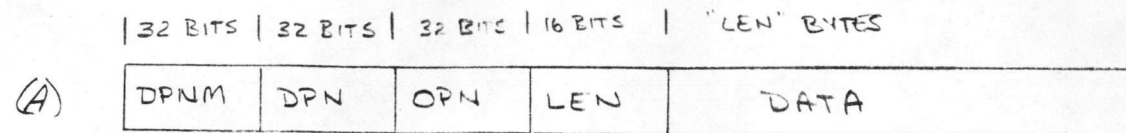
ACKNOWLEDGMENTS

The authors wish to express their great appreciation to Carver Mead and Ivan Sutherland for their technical help in

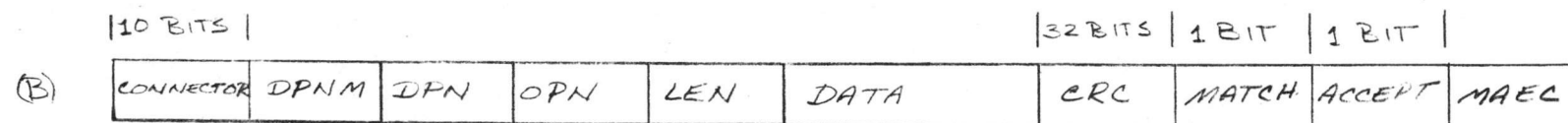
the design of the NAA. The authors also wish to thank Ken Pogran for his helpful comments, and Steve Duff and Paul Tuinenga for their invaluable assistance in the implementation.

REFERENCES

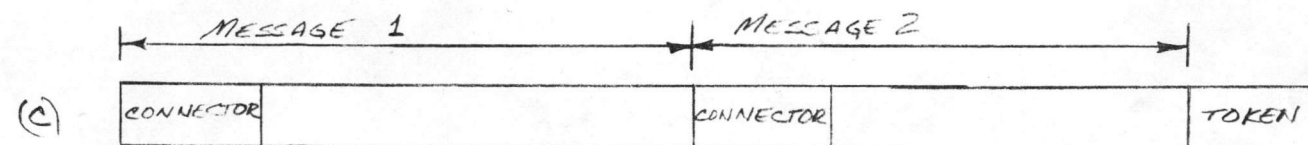
- [1] David J. Farber, The Distributed Computing System IEEE Compcon Proceedings, 1973.
- [2] Robert M. Metcalf, and David R. Boggs, Ethernet: Distributed Packet Switching for Local Computer Networks, Communications of the ACM, vol.19, no.7, July 1976, 395-404.
- [3] Donald C. Loomis, Ring Communication Protocols, Tech Report 26, University of California, Irvine, 1973.
- [4] Paul V. Mockapetris, Design Considerations and Implementation of the ARPA NAA Name Table, Tech Report 92, University of California, Irvine, 1976.
- [5] R.A. Donnan, and J. R. Kersey, Synchronous data link control: A perspective, IBM Systems Journal, vol.13, no.2, 1974, 140-162.
- [6] David J. Farber, and John R. Pickens, The Overseer, International Computer Communications Conference Proceedings, 1976.
- [7] David J. Farber, and Kenneth Larson, Network Security via Dynamic Process Renaming, Proceedings of the 4th Data Communications Symposium, 1976.



LOGICAL MESSAGE



PACKAGED MESSAGE



MESSAGE TRAIN

FIGURE 1 - MESSAGE FORMATS

| MATCH | ACCEPT | MEANING |
|-------|--------|--|
| 0 | 0 | The message was addresssed to a process that does not exist; no RI recognized this message. |
| 0 | 1 | The message was successsfully transmitted to one or more processes; at least one RI recognized and copied the message. |
| 1 | 0 | No process received the message; however, at least one RI recognized the address in the message. |
| 1 | 1 | This message was addressed to processes in at least two hosts; at least one RI was able to copy the message, and at least one RI was unable to copy the message. |

Figure 2 - Match / Accept Results

0-0010-0
2-01

D-9100-SB
5-41

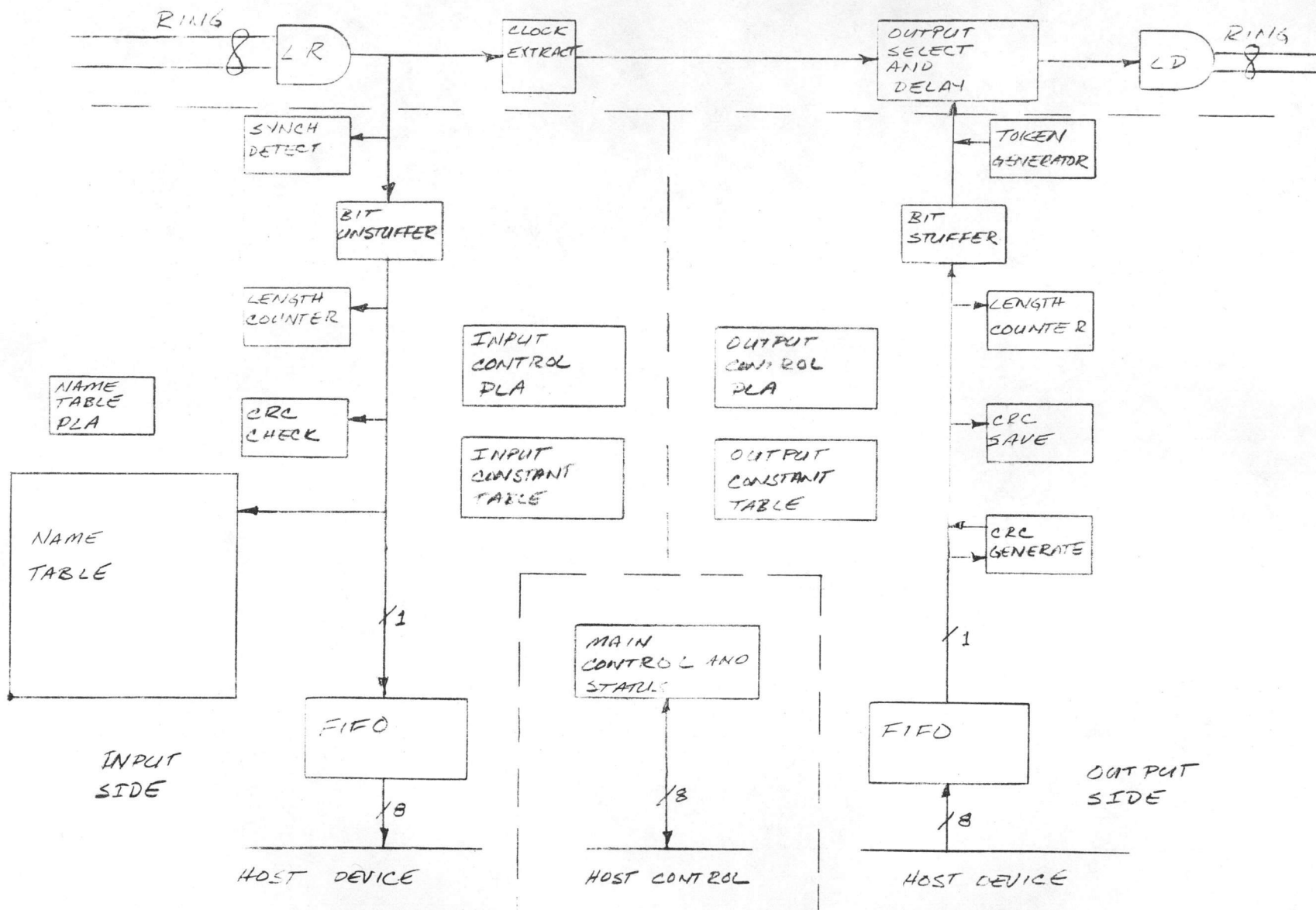


FIGURE 3 - NAA BLOCK DIAGRAM