# A HIGHLY RELIABLE LAN PROTOCOL

Alfred C. Weaver, Ph.D.

Computer Science Report No. TR-85-25
December 30, 1985

"This paper has been submitted for publication to the IEEE *Journal on Selected Areas in Communications*."

# A HIGHLY RELIABLE LAN PROTOCOL

Alfred C. Weaver
Associate Professor of Computer Science
University of Virginia
Charlottesville, Virginia 22903

## ABSTRACT

As a research project for NASA's Langley Research Center, we developed a variation on MIL-STD-1553B (the military standard for avionics busses) whose goal was increased fault tolerance. The resulting protocol, called Implicit Token Passing (ITP), replaces an explicit token with brief "soundoff" messages from all nodes participating on the LAN. Since every node participates on every "token cycle", bus silence is an error indication and initiates recovery action. By encoding state information in the headers of the transmitted words, nodes are continuously aware of the global state of the network. This local knowledge of the global network state allows the system to continue operation in spite of nodal failures. A station which fails but then recovers can quickly assess the global network state and then safely rejoin the active nodes. ITP features high throughput and bounded message delay, and achieves high reliability through tolerance of failed nodes and automatic resynchronization when failed nodes are revived. The protocol is ideally suited for a bus topology and fiber optic media.

# I. INTRODUCTION

MIL-STD-1553B [1] is the current military standard for avionics busses. It specifies a time division multiplex protocol in which a bus controller (i.e. master node) successively polls all remote terminals (i.e. all nodes other than the bus controller) to initiate and control message transfers. Messages are limited to three types: command words, data words, and status words. Command words specify the address of the remote terminal and the count of data words to be sent or received; data words contain 16 bits of data and one parity bit in a 20-bit frame; status words are used for network control and error reporting. Data messages are limited to 32 successive data words. The bus controller must initiate all message transmissions and thus becomes a crucial element in the network design. All data is transmitted at 1 Mbps over twisted pair media on a bus architecture.

As a research project for NASA's Langley Research Center, the Computer Networks Laboratory at the University of Virginia undertook to redesign the MIL-STD-1553B protocol to improve performance and reliability for a 1-10 Mbps fiber optic environment. We were constrained by hardware considerations to retain the 20-bit word format, but were otherwise free to alter the protocol's operation as necessary to improve reliability. The target implementation environment expected frequent nodal failures as well as unexpected nodal recoveries; thus our goal was to design a robust protocol which could support a highly dynamic nodal membership.

# II. IMPLICIT TOKEN PASSING

We objected to the master/slave design of 1553B because of its reliance on a "master node" — the bus controller. Failure of the bus controller causes total network failure, thereby forcing all practical designs to include some form of redundancy for the bus controller. We preferred a totally distributed protocol, in which each node was aware of

its logical neighbors but not dependent upon them.

Our resulting protocol is called Implicit Token Passing (ITP). At network initialization nodes are assigned a predetermined sequence number which defines the order in which they are to transmit. All nodes are aware of the full transmission sequence. When a node's turn arrives in the sequence, it broadcasts an information packet which contains its logical address; data is optional, and if present the packet includes the destination address, data word count, a checksum on the addresses and word count, the data proper, and finally a checksum on the data. Figure 1 shows the logical message format.

Because of our hardware constraint that transmission must physically occur in 20-bit words, the information packet is divided into two header words and, if data is present, up to 1024 data words. Each word is comprised of a four-bit synch field followed by 16 usable data bits. The synch field contains a transition at the mid-bit time of bit 3; the direction of the change is used to distinguish header words from data words. This distinction is a key feature for network reliability. The logical fields of the message are partitioned into 20-bit header words and data words as shown in Figure 2.

These information packets substitute for the traditional "token" in token passing protocols. Every node identifies its turn in the sequence by hearing his predecessor. Since every node is aware of the full transmission sequence, and since bus silence is an indication of error, a failed node is detected by a simple timeout. If node $n$ transmits at time $t_0$ and finishes transmission at time $t_1$, then node $n+2$ should hear node $n+1$ begin transmission no later than time $t_2 = t_1 + 2 \cdot \rho + \epsilon$ where $\rho$ is the worst case end-to-end propagation delay and $\epsilon$ is the electronic switchover time of the transmitter.

Setting a node's timeout timer to $2 \cdot \rho + \epsilon$ will reliably detect the failure of a single predecessor. If multiple nodes fail in sequence (as would be the case with, say, a common power supply failure), then active nodes will register multiple timeouts and will update

their position in the transmission sequence accordingly. Figure 3 depicts detection of a single failed node.

No attempt is made to reorder the transmission sequence. A failed node is detected anew on each "token cycle"; thus its place in the transmission sequence is reserved and, if it later recovers, it assumes its former place. This repeated detection of failure is acceptable from the point of view of performance because the bus idle time caused by a dead node is typically *less* than the transmission time of the normal information packet. Thus, ITP has the property that, under conditions of partial failure, its throughput improves.

It should be noted that ITP is intended for use in the environment of avionics busses where the short bus length (typically a few hundred feet) leads to short propagation times and where the number of active nodes is small (tens of nodes rather than hundreds). This makes practical the requirement that all nodes know the full transmission sequence, rather than just their immediate predecessor, which in turn enables us to survive the failure of multiple successive nodes.

## III. DESIGN FOR RELIABILITY

When a node has failed but then recovers and reenters an operating network, its proper place in the transmission sequence remains unchanged but it is a non-trivial problem for a newly recovered node to locate it. The node must correctly identify the sequence of nodes transmitting, and then take its proper place within that sequence.

The finite state machine in Figure 4 shows how a newly recovered node finds its proper place. Let node RE (reenter) represent the state of a node which has been dead, but is now alive and wishes to reenter the transmission sequence. The node enters state RC (receive) and begins monitoring transmissions, waiting for a header word. If data words

are heard, the state moves back and forth between NH (non-header) and RC, until finally a header word's synch field signals the arrival of a header word (state HD). States H1 and H2 represent hearing the first or second header words respectively. If the node recovered just before H2 was sent, the second header would be erroneously accepted as the first, but the error would be detected when the next word proved to be data rather than header; in this case the node cycles from expecting a header (HD) to receiving a header (H2) to detecting the error (ER) and returns to the receive state RC where it again awaits a header. Hearing the full two-word header sequence H1 and H2 provides the necessary information on the global network state and leads to state RS in which the node is now resynchronized with the network.

## IV. IMPLEMENTATION AND RESULTS

The ITP protocol was implemented on NASA-Langley's Data Distribution Evaluation System (DDES, [2]). This system, built by Boeing, consisted of three separate processing stations interconnected by four communications busses (two twisted pair busses at 1 Mbps and two fiber optic busses at 1-10 Mbps). Each station on the DDES consisted of a bus control interface unit (BCIU) and a commercial microprocessor development system (MDS) acting as a programming vehicle and system interface. The BCIU was an Intel 3000 series bit-sliced microprogrammable microprocessor which received data from the MDS and controlled the data transceivers. The BCIU was microcoded to execute a Boeing-developed macrocode oriented toward communications tasks. The ITP protocol was described in macrocode and executed by the BCIU. The MDS provided the programming interface and data analysis.

The three DDES stations were partitioned in software to simulate 45 physical nodes. Each MDS provided its BCIU with data to transmit and a destination for it; that BCIU waited its turn in the ITP sequence and then transmitted the header sequence and the proper number of data words needed to complete the message. All transmitted messages

were heard by all nodes; however, only the node to which the message was addressed modified its performance statistics to reflect reception.

Our experimentation focused on two issues: performance and reliability. Detailed performance results are reported in [3] and [4] and show that throughput increases almost linearly with offered load until the bus saturates (i.e., until all available transmission time is being used for header words, data words, and propagation delays). As expected, longer messages are more efficient than shorter messages since they required fewer intervening header words. Also as expected, message delay (the elapsed time between message generation and receipt, including queueing time, network access time, transmission time, and propagation time) increases exponentially as offered load approaches bus capacity. In general, the performance attributes of ITP are quite similar to the performance of an IEEE 802.4 token bus carrying only synchronous traffic when the 802.4's High Priority Token Hold Time has been set to one message time, thereby forcing an 802.4 station to emit at most message per token cycle (see [5]). ITP has a small performance advantage due to its smaller "token" (40 bits vs. a minimum of 96 bits), and also because ITP deals with dynamic membership in an entirely different way from IEEE 802.4.

The reliability experiments judged the protocol's robustness. Failing any one of the 45 nodes had no observable effect. Even failing one-third of the nodes did not cause any errors or synchronization problems among those which remained connected. The protocol likewise proved robust when nodes were physically removed from the network by unplugging and then reconnecting them. While disconnection and reconnection did momentarily increase the network's bit error rate, in all cases the rejoining nodes were resynchronized and the network's bit error rate had returned to zero within one millisecond.

# V. CONCLUSIONS

The goal was to achieve a robust protocol for avionics busses using hardware originally designed to support MIL-STD-1553B. These considerations proved important for performance and reliability:

(1) The application environment of the avionics bus, while still a local area network, allowed us to limit the maximum nodal membership to tens of nodes, rather than hundreds or thousands as in other more general purpose LAN protocols. This permitted us to use the timeout scheme to infer nodal failure from bus silence. The limited network membership also made practical our requirement that each node be aware of the full transmission sequence.

(2) By having each node participate on each "token cycle" by sending its information packet, we replace an explicit token addressed to a specific recipient with a more general "announcement" that the node is alive and well. The 40-bit information packet is less expensive than the explicit 96-bit (minimum) token of IEEE 802.4, which yields a small performance advantage for ITP.

(3) Encoding word identification information (header vs. data) in the synch field of each word provides a simple scheme whereby reentering nodes can reliably determine the current transmitter, and from that determine their position in the transmission sequence. This feature accounted for the network's resistance to confusion when nodes were randomly removed and then reconnected.

(4) Since an avionics bus is typically short (a few hundred feet), our end-to-end propagation time ($\rho$) is short. Detecting a failed node required $2 \cdot \rho + \epsilon$, which was typically 10 microseconds. On a 1 Mbps bus, a normally operating node would require two header words (40 bit times) plus an inter-word gap (2 microseconds) for a total of 42 microseconds. Thus detecting a failed node was faster than the implicit token pass. This

feature prevented serious performance degradation under conditions of partial failure.

## ACKNOWLEDGEMENT

The authors gratefully acknowledge the loan of the Distributed Data Evaluation System made to the Department of Computer Science by the Fault Tolerant Systems Branch of NASA Langley Research Center.

## REFERENCES

[1] U.S. Department of Defense, *Military Standard: Aircraft Internal Time Division Command/Response Multiplex Data Bus*, MIL-STD-1553B, 21 September 1978.

[2] Boeing Aircraft Company, *Data Distribution Evaluation System Reference Manual*, Version 1.0, September 1982.

[3] Alfred C. Weaver and David W. Butler, "A Fault-Tolerant Network Protocol for Real-Time Communications," *IEEE Transactions on Industrial Electronics*, May 1986.

[4] David W. Butler, *Fault-Tolerant Protocols for Real-Time Local Area Networks*, M.S. thesis, Department of Computer Science, University of Virginia, May 1984.

[5] Catherine F. Summers, *Performance Analysis of the IEEE 802.4 Token Bus*, Department of Computer Science technical report TR-85-03, University of Virginia, May 1985.

**Alfred C. Weaver**

Alf Weaver received the B.S. degree in Engineering Science from the University of Tennessee in 1971 and the M.S. and Ph.D. degrees in Computer Science from the University of Illinois in 1973 and 1976 respectively. He joined the Department of Computer Science at the University of Virginia as Assistant Professor in 1977, was promoted to Associate Professor in 1983, and served one year as Chairman in 1984-85.

Dr. Weaver's current research area is protocol design and analysis for local area networks. He has helped various industries and government agencies design high-performance LANs for special purpose applications, with an emphasis on fault tolerance and the use of LANs in real-time control systems. He has given a number of seminars and tutorials on the suitability and performance of the new IEEE LAN standards. He is an ACM National Lecturer on computer networks and currently serves the IEEE Industrial Electronics Society as Vice-President for conferences and as chairman of the technical committee on factory communications.

| SOURCE | DEST | COUNT | CRC1 | DATA1 | — | DATAn | CRC2 |
|---|---|---|---|---|---|---|---|

SOURCE — sending node's address
DEST — destination address
COUNT — number of 16-bit data words in message
CRC1 — cyclic redundancy code over addresses and count
DATAi — data words
CRC2 — cyclic redundancy code over data words

Figure 1.
Logical Message Format

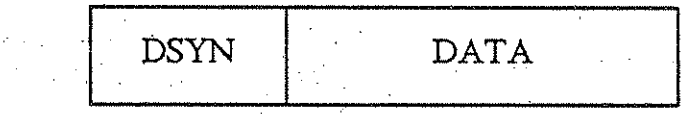
First header word:


bits: 4 8 8

Second header word:


bits: 4 10 6

All data words:




bits: 4 16

HSYN — header word synch field
DSYN — data word synch field

Figure 2.
Header Words and Data Words

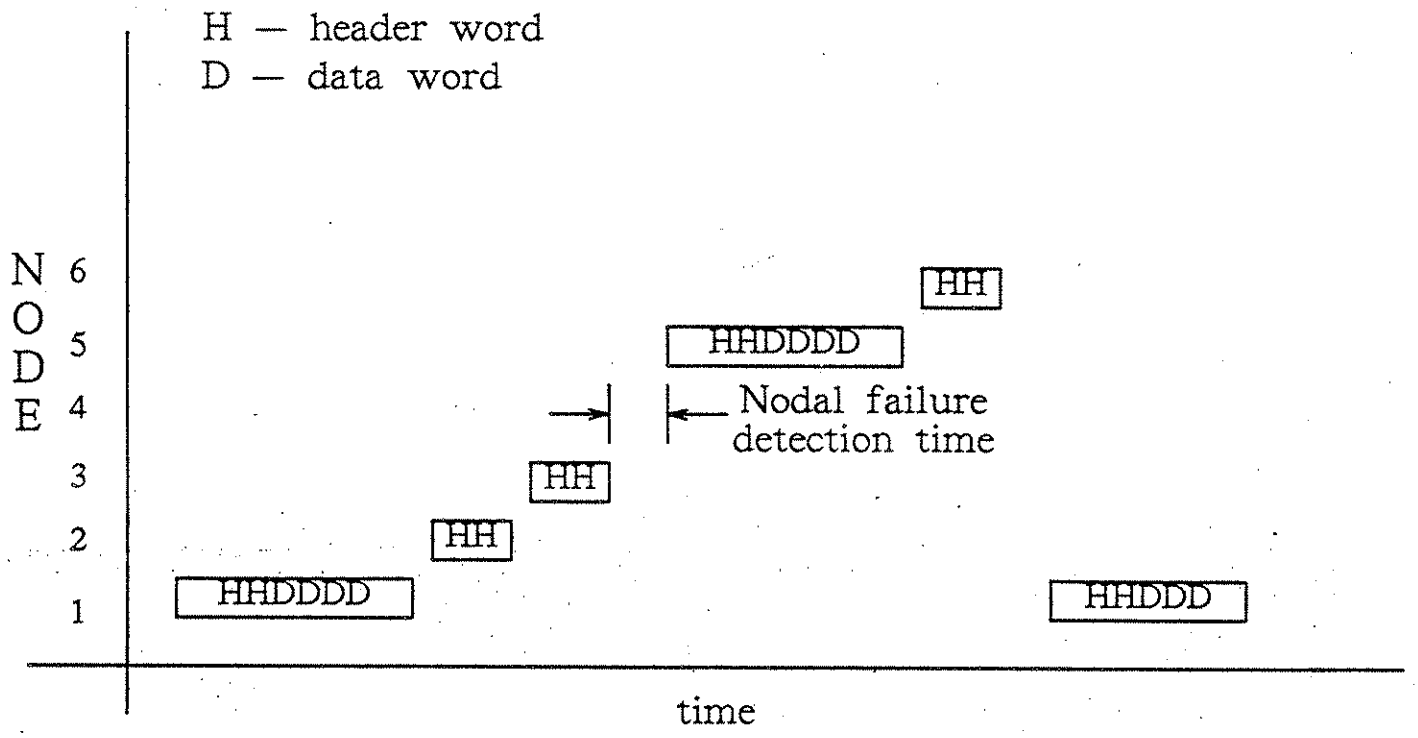
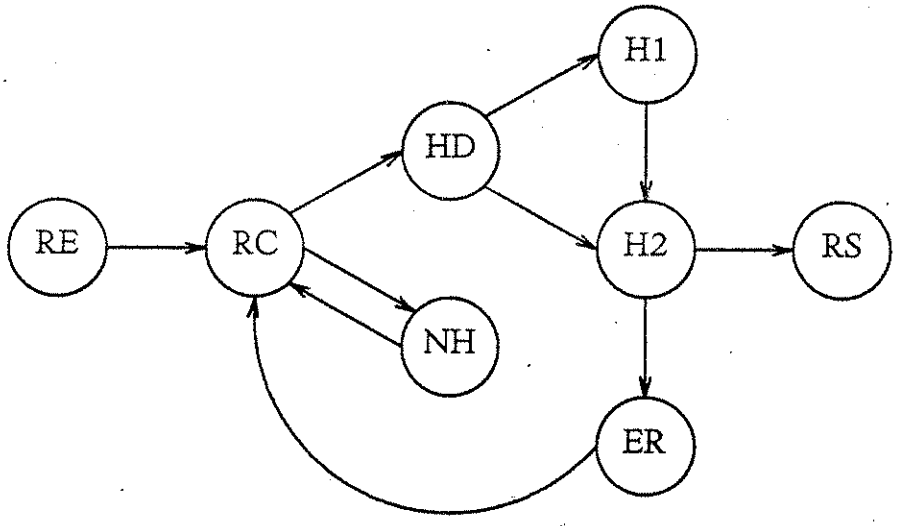



Figure 3.
Detection of Failure in Node 4




RE - reenter RC - receive HD - header

H1 - header 1 H2 - header 2 RS - resynchronized

NH - non-header ER - error detected

Figure 4.
State Diagram for Nodal Resynchronization