iCC 1997

4th international CAN Conference

in Berlin (Germany)

Sponsored by

Motorola Semiconductor
NEC Electronics (Europe)
Siemens Semiconductors

Organized by

CAN in Automation (CiA)
international users and manufacturers group
Am Weichselgarten 26
D-91058 Erlangen
Phone +49-9131-69086-0
Fax +49-9131-69086-79
Email:headquarters@can-cia.de
URL: http://www.can-cia.de

George Thomas
Contemporary Control Systems, Inc.
2512 Wisconsin Avenue
Downers Grove, IL 60515 USA
gthomas@ccontrol.com

EXTENDING CAN NETWORKS BY INCORPORATING REMOTE BRIDGING

ABSTRACT

Controller Area Network (CAN) technology has become increasingly more popular for factory floor applications due to its small size, low cost and high speed. This technology utilizes a clever bit-wise arbitration scheme for medium access control resulting in the non-destructive transmission of the highest CAN identifier. Although this scheme provides high throughput, it places a distance limitation on node separation since nodes must all respond within a fraction of a bit-time. For factory floor applications, this distance constraint could be a problem.

This paper discusses the use of remote bridges interconnected with a high-speed deterministic network. Each bridge has two ports—one for a CAN segment and the other for the high-speed deterministic network. Messages received on the CAN segment are encapsulated into the data frame of the high-speed deterministic network and sent to all other bridges. Each bridge extracts the data and converts the data back to a CAN format for rebroadcast to all other CAN segments. Bridges can be separated up to four miles and can be cabled with either coaxial or fiber optic cable. These bridges operate at the data link layer and, therefore, all higher layer protocols such as DeviceNet, Smart Distributed System, CAL, CAN Kingdom and CANopen pass without modification making bridging applicable to several commercial systems.

BACKGROUND

CAN was designed by Bosch and is currently described by ISO 11898[1]. In terms of the Open Systems Interconnection model (OSI), CAN partially defines the services for layer 1 (physical) and layer 2 (data link). Other standards such as DeviceNet, Smart Distributed System, CAL, CAN Kingdom and CANopen (collectively called higher layer protocols) build upon the basic CAN specification and define additional services of the seven layer OSI model. Since all of these protocols utilize CAN integrated circuits, they therefore all comply with the data link layer defined by CAN.

CAN specifies the medium access control (MAC) and physical layer signaling (PLS) as it applies to layers 1 and 2 of the OSI model. Medium access control is accomplished using a technique called non-destructive bit-wise arbitration. As stations apply their unique identifier to the network, they observe if their data are being faithfully produced. If it is not, the station assumes that a higher priority message is being sent and, therefore, halts transmission and reverts to receiving mode. The highest priority message gets through and the lower priority messages are resent at another time. The advantage of this approach is that collisions on the network do not destroy data and eventually all stations gain access to the network. The problem with this approach is that the arbitration is done on a bit by bit basis requiring all stations to hear one another within a bit-time (actually less than a bit-time). At a 500 Kbps bit-rate, a bit-time is 2000 ns which does not allow much time for transceiver and cable delays. The result is that CAN networks are usually quite short and frequently less than 100 meters at higher speeds. To increase this distance either the data rate is decreased or additional equipment is required.

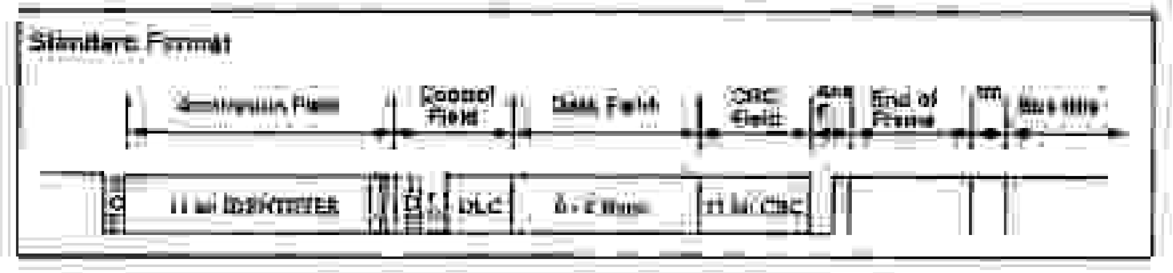

Figure 1-An 11-bit identifier is used in standard format.

CAN DATA LINK LAYER

CAN transmissions operate using the producer/consumer model. When data are transmitted by a CAN device, no other devices are addressed but instead the content of the message is designated by an identifier field. This identifier field, which must be unique within the network, not only provides content but the priority of the message as well. All other CAN devices listen to the sender and accept only those messages of interest. This filtering of the data is accomplished using an acceptance filter which is an integral component of the CAN controller chip. Data which fail the acceptance criteria are rejected. Therefore, receiving devices consume only that data of interest from the producer.

A CAN frame consists mainly of an identifier field, a control field and a data field (figure 1). The control field is six bits long, the data field is zero to eight bytes long and the identifier field is 11 bits long for standard frames (CAN specification 2.0A) or 29 bits long for extended frames (CAN specification 2.0B). Source and destination node addresses have no meaning using the CAN data link layer protocol.

Bus arbitration is accomplished using a non-destructive bit-wise arbitration scheme. It is possible that more than one device may begin transmitting a message at the same time. Using a "wired AND" mechanism, a dominant state (logic 0) overwrites the recessive state (logic 1). As the various transmitters send their data out on the bus, they simultaneously listen for the faithful transmission of their data on a bit by bit basis until it is discovered that someone's dominant bit overwrote their recessive bit. This indicates that a device with a higher priority message, one with an identifier of lower binary value, is present and the loser of the arbitration immediately reverts to receiving mode and completes the reception of the message. With this approach no data are destroyed and, therefore, throughput is enhanced. The losers simply try again during their next opportunity. The problem with this scheme is that all devices must assert their data within the same bit-time and before the sampling point otherwise data will be falsely received or even destroyed. Therefore, a timing constraint has been introduced which impacts cabling distance.

PROPAGATION DELAY

In a Philips application note[2], the author does an in-depth study on the maximum allowable propagation delay as a function of various controller chip parameters. The propagation delay (figure 2) is due to the input/output delays of the CAN controller chip (t_{sd}), transmission delay of the transceiver (t_{tx}), reception delay of the transceiver (t_{rx}) and the signal delay due to the cable (t_{cbl}). The total propagation delay (t_p) experienced is basically the round trip delay from a CAN node located at the end of a cable segment communicating to the furthest node and is expressed as follows:

$$t_p = 2(t_{sd}+t_{tx}+t_{rx}+t_{cbl})$$

All delays are constant except the cable delay (t_{cbl}) which depends upon the length of the cable and the propagation delay factor of the cable (Pc). The author provides a chart of maximum allowable propagation delays (t_{pm}) for various data rates and CAN chip timing parameters. The actual propagation delay must not exceed the maximum allowable propagation delay. By making the appropriate substitutions we can determine the maximum allowable cable length (L).

$$L < \frac{t_{pm}-t_{sd}-t_{rx}-t_{tx}}{Pc}$$

Using appendix A.1 of the application note and the most favorable parameters for long

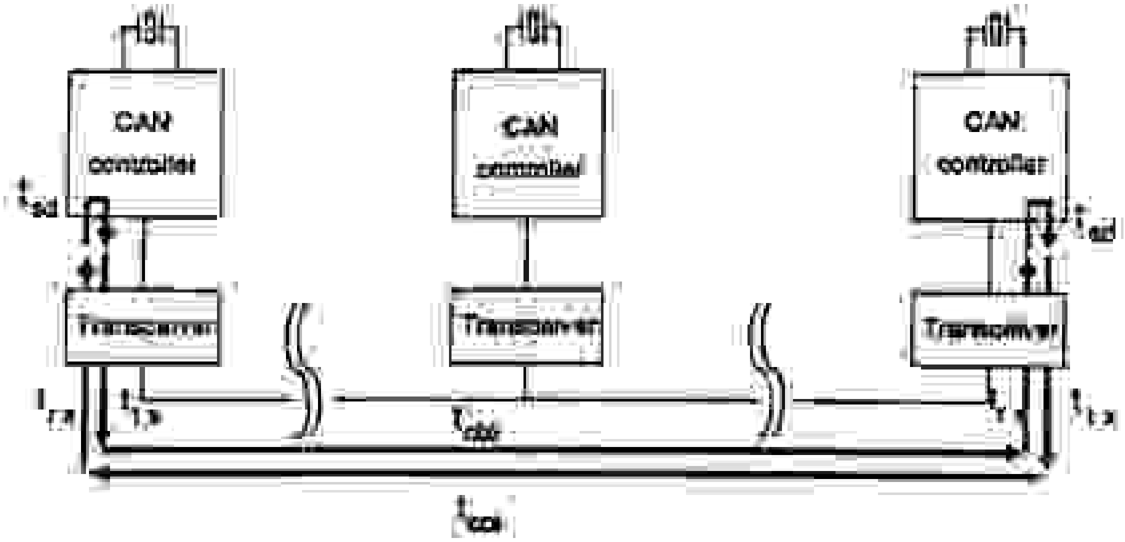

Figure 2-Use the longest path when calculating propagation delay.

distance, at 500 Kbps tpm equal 1626 ns. Assuming transceiver delays of 100 ns each, chip delay of 62.5 ns and a cable propagation factor of 5.5 ns/m, the maximum cable length is 100 meters which is the value used in the DeviceNet specification. Doing the same calculation at 250 Kbps yields 248 meters and at 100 Kbps 680 meters. These values can be improved with better cable and faster transceivers.

The point here is that CAN's bit-wise arbitration scheme inherently limits the maximum length of a CAN segment. Increasing the distance requires a reduction in data rate; however, there might be some benefit to incorporating repeaters or bridges.

REPEATERS

The usual approach to increasing network distance is to use repeaters. Repeaters provide signal boost to make up the loss of signal strength on a long segment. However, the problem with long CAN segments is not lack of signal strength but excessive signal latency. This latency is due to the propagation delay introduced by the transceivers and twisted-pair wiring. If this latency approaches one bit-time, the non-destructive bit-wise arbitration mechanism fails. Repeaters actually introduce more delay due to the additional electronics and are not effective in increasing the overall length of CAN networks. Repeaters can be used to increase the effective length of drop cables from CAN trunk lines. Repeaters operate on the physical layer and are ignorant of the data link layer.

BRIDGES

Bridges are defined as devices that link two similar networks[3]. A local bridge stands by itself connecting adjacent wiring segments together as in the case of a repeater. Therefore, a local CAN bridge would have two CAN chips, one for one segment and one for the other. A microprocessor would pass messages between the two CAN chips. Using this approach, the effective length of the complete network is doubled while requiring only one bridge. Remote bridging interconnects two physically separated but similar networks together using a different interconnecting medium. Therefore, a pair of bridges are required to interconnect two networks the way two modems are used on leased phone lines. Sometimes bridges block network traffic by restricting data only to stations specified in the transmission that resided on the network controlled by the bridge. This blocking is difficult to implement in broadcast networks such as CAN and, therefore, not recommended. Bridges operate at the data link layer and, therefore, are ignorant of the higher level protocols sent over CAN. As with the local bridge, two ports are required. However, instead of two CAN ports, one CAN port is replaced with a port compatible with the technology of the bridging connection. The technology chosen should be fast, deterministic, robust and capable of extending

CAN networks without introducing excessive delay that would jeopardize the operation of the CAN system. One possible technology would be ARCNET.

ARCNET

ARCNET is a local area network technology which is described in ANSI/ATA 878.1[4]. Like CAN, ARCNET only defines the data link and physical layers (figure 3). ARCNET is attractive for use in bridging CAN segments because it is faster than CAN (2.5 Mbps), it supports many nodes (255), it can send large packets (507 bytes), it can communicate over long distances (4 miles) and it provides deterministic performance due to its token-passing medium access control. All these elements are important if CAN messages are to be transferred with the lowest possible delay.

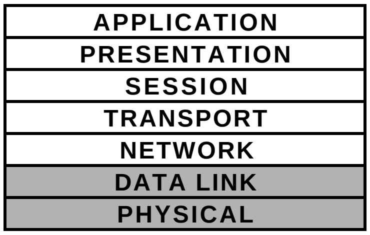

Figure 3-Like CAN, ARCNET defines the lower two layers of the OSI Reference Model.

Logical Ring

A token is a unique signaling sequence that is passed in an orderly fashion among all the active nodes in the network[5]. When a particular node receives the token, it has the sole right to initiate a transmission sequence or it must pass the token to its logical neighbor. This neighbor, which can be physically located anywhere on the network, has the next highest address to the node with the token. Once the token is passed, the recipient (likewise) has the right to initiate a transmission. This token-passing sequence continues in a logical ring fashion serving all nodes equally. Node addresses range from 0 to 255 with 0 reserved for broadcast messages.

In figure 4, the highest address is 255 and potentially its logical neighbor is 1. However, in this example its logical neighbor is 6.

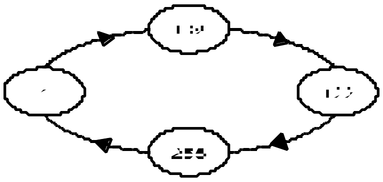

Figure 4-The logical ring has nothing to do with the physical placement of nodes. The node with the next highest address is that node's logical neighbor. However, logical neighbors could be located at the extreme ends of a physical multi-node network

Secure Message

In a transmission sequence, the node with the token becomes the source node and any other node selected by the source node for communication becomes the destination node. First the source node inquires if the destination node is in a position to accept a transmission (FBE). The destination node responds with either a yes (ACK) or a no (NAK). Upon an ACK, the source node sends out a transmission from either 0 to 507 bytes of data (PAC). If the data were properly received by the destination node as evidenced by a successful CRC test, the destination node sends another ACK. If the transmission was unsuccessful, the destination node does nothing, causing the source node to timeout. The source node will, therefore, infer that the transmission failed and will retry after it receives the token on the next token pass. The transmission sequence terminates and the token is passed to the next node. If the desired message exceeds 507 bytes, the message is sent as a series of packets—one packet every token pass. The packets are recombined at the destination end to form the message. This process is called fragmentation.

ARCNET supports a broadcast message which is an unacknowledged message to all nodes. Nodes which have been enabled to receive broadcast messages will receive a message that specifies node 0 as the destination address.

Automatic Reconfigurations

Another feature of ARCNET is its ability to reconfigure the network automatically if a node is either added or deleted from the network. If a node joins the network, it does not automatically participate in the token-passing sequence. Once it notices that it is never granted the token, it will jam the network with a reconfiguration burst that destroys the token-passing sequence. Once the token is lost, all nodes will cease transmitting and begin a timeout sequence based upon its node address. The node with the highest address will timeout first and begin a token pass sequence to the node with the next highest responding node is noted as the logical neighbor of the originating node. The sequence is repeated by all nodes until each node learns its logical neighbor. At that time the token passes from neighbor to neighbor without wasting time on absent addresses.

Cabling Flexibility

ARCNET is regarded as one of the most flexible industrial networking technologies to wire due to the many cabling options available to the user. ARCNET can be configured as a star or distributed star network using hubs (figure 5) or a bus without hubs. Cabling can be coaxial, twisted-pair or fiber optics. With the proper selection of hubs, repeaters and links, cabling and topology can be mixed to a distance of four miles.

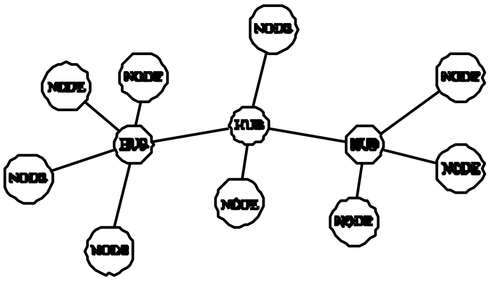

Figure 5-With ARCNET, a distributed star topology is possible using hubs.

A PROPOSED CAN-BRIDGE

The CAN-BRIDGE is classified as a remote bridge that is used to extend CAN-based device networks. On the device side of the bridge, a CAN segment is connected complying to the distance limitation for the bit-rate used. On the other side is ARCNET that captures the CAN traffic and replicates it to another bridge. The receiving bridge converts the data to its attached CAN segment. A minimum of two bridges (figure 6) is required but in the general case, many bridges can be used since ARCNET supports star and distributed star topologies. The bridges are protocol neutral. They do not understand higher layer protocols. The bridges simply capture the data transmitted on its CAN segment and encapsulate the data into

address. If that node does not respond, it is assumed not to exist. The destination node address is incremented and the token resent. This sequence is repeated until a node responds. At that time, the token is released to the responding node and the address of the

ARCNET frames for retransmission to the other bridges on the network. The receiving bridges remove the CAN data from ARCNET packets and rebroadcast the data to its CAN segment. The bridges do not filter out MAC addresses or CAN identifiers. CAN messages originating on a particular CAN segment are rebroadcast to all other CAN segments without modification. Therefore, it is important that all CAN compliant devices on the complete network incorporate a unique CAN identifier which would be the case for a CAN network without bridges.

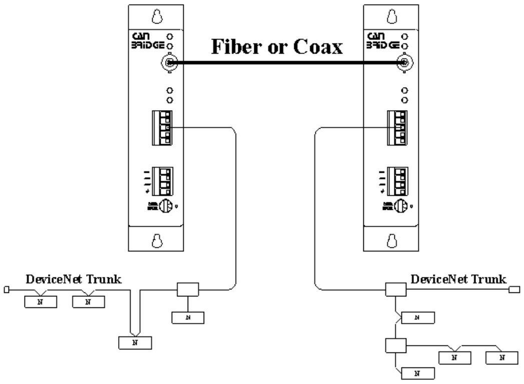

Figure 6-Two remote bridges are required to link two distant DeviceNet segments using either a fiber optic or coaxial cable backbone.

CAN-BRIDGE Engine

A high-speed 80C188 microprocessor provides the computing power for the CAN-BRIDGE. The ARCNET port consists of a 20020 controller chip and either a coaxial cable or fiber optic transceiver. The CAN port, which consists of an Intel CAN controller and isolated transceiver, is capable of generating interrupts at a high speed since the CAN-BRIDGE must listen to all CAN traffic. Back to back CAN data frames can generate an interrupt every 94 µs at 500 Kbps. The ARCNET buffers will also generate interrupts making interrupt handling a challenge for the CAN-BRIDGE.

CAN Port

One electrically isolated CAN port has been provided capable of operating to the DeviceNet[6] physical layer specification. This was done to minimize ground loop problems while providing isolation to the ARCNET backbone.

When a CAN port is receiving data from the CAN segment, its acceptance filter is wide open since all messages must be received. When a CAN port is transmitting to the CAN segment, the port replicates the CAN message originating from a remote CAN segment as if that that CAN chip was present locally.

ARCNET Port

The ARCNET port operates at 2.5 Mbps. Each CAN-BRIDGE requires a unique ARCNET node ID which has no meaning to the CAN segments. Node ID's are automatically assigned by the CAN-BRIDGES themselves using an arbitration scheme upon power up eliminating the need to make switch assignments in the field.

ARCNET'S DATA LINK PROTOCOL

The ARCNET data-link level protocol is comprised of five basic transmissions[7]. Each transmission is preceded by an alert burst (SD) which consists of six consecutive intervals of a logic "1" or mark condition. Each of the transmissions consists of a combination of bytes including ASCII characters, source address (SID), destination address (DID), continuation pointer (CP), system code (SC), data and cyclic redundancy check (CRC). Each byte has appended a preamble consisting of two intervals of mark and one interval of space. Therefore, eleven bits are required to send one byte. The transmissions and the time to execute a transmission at 2.5 Mbps data rate are shown below in figure 7.

The length of data packets varies with the number (n) of data bytes. The above data packet frame is for short packet mode in which the number of data bytes can vary from 0 to 252 bytes. There is a long packet mode as well in which the number of data packets can vary from 256 to 507 bytes. Messages from 253 to 255 bytes long must be padded to at least 256 bytes for proper handling. The total time it takes to send a message can be determined by knowing the time required to execute each of the transmissions.

CAN FRAMES

CAN transmissions exist as either standard frames or extended frames. The standard frame includes an eleven-bit identifier while newer CAN controller chips are also capable of producing an extended frame with a 29-bit identifier. While most higher level protocols support only standard frames, the CAN-BRIDGE recognizes either and transmits either.

The CAN-BRIDGE listens to all CAN frames on its CAN port and if a successful acknowledgment is noted, stores the identifier, control and data fields into a buffer. The bridge continues to listen for additional transmissions while the buffer is flushed by sending the data over the ARCNET backbone.

If the CAN-BRIDGE receives a message from the ARCNET backbone, it must transmit the message to its CAN port. The CAN-BRIDGE will have the identifier, control and data fields for the transmission it needs to produce and conditions the CAN port accordingly. This time, however, the CAN port on the CAN-

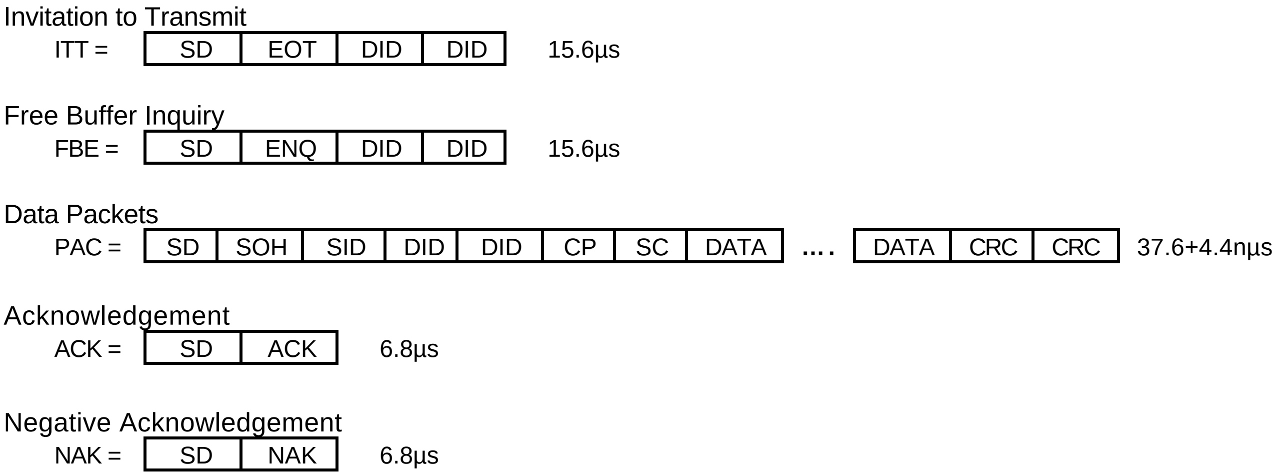

Figure 7-There are five basic ARCNET transmissions

BRIDGE must transmit an identifier which might have a low priority and could experience difficulty gaining bus access in order to transmit this message. During this time the CAN-BRIDGE continues to receive CAN port data while attempting to transmit onto the CAN segment. Data is not lost during this time but queued in the CAN-BRIDGE. Once the CAN transmission is initiated, the acknowledgment is monitored for success. If unsuccessful, the transmission sequence is repeated until successful. CAN messages are queued on a first come-first serve basis so that fragmented CAN messages are not missequenced.

ARCNET FRAMES

ARCNET frames are longer than CAN frames. Instead of eight data bytes for CAN, ARCNET can accommodate up to 252 bytes in short packet mode, 507 in long packet mode. Short packet handling is more efficient and, therefore, was chosen for the CAN-BRIDGE. The CAN frames are encapsulated into ARCNET frames, and it is possible that more than one CAN frame could be stored within one ARCNET frame. For each CAN message, space must be reserved in the ARCNET frame for identifier, control and data fields.

ARCNET HEADER

As mentioned before, ARCNET data packets can vary from 0 to 507 bytes in one ARCNET frame. There is an ATA standard (ATA 878.2)[8] which provides a mechanism for transmitting much longer packets by fragmenting the longer message into manageable ARCNET frames. Although sending longer messages is not of interest here, the standard introduces the concept of sequence numbers (FSN) which are used to check against the reception of duplicate packets which can occur when a transmitting node fails to receive a successful ACK from a receiving node. This check has been included in the CAN-BRIDGE implementation.

A third ATA standard (ATA 878.3)[9] allows for encapsulating other protocols within ARCNET frames. Two additional bytes are defined in the message field to identify the encapsulated protocol ID and to signify if the message is a command or a response from or to a master. With the CAN-BRIDGE, this latter byte is meaningless and is used for another purpose.

By supporting the upper two standards, message handling is more robust and multiple protocols can be sent simultaneously on the same ARCNET network. The cost is five additional bytes of overhead—three for the fragmentation standard and two for the encapsulated protocol standard. These five bytes plus the other bytes used to identify source and destination of the ARCNET message constitute the ARCNET header.

ENCAPSULATING CAN DATA

The CAN data that needs to be encapsulated includes the identifier field, the control field and the data field. It is easier to simply reserve fixed locations in the ARCNET frame for each CAN message. Therefore, four bytes are reserved for the identifier field—one for the control field and eight for data yielding a total of thirteen bytes. If we choose short packet

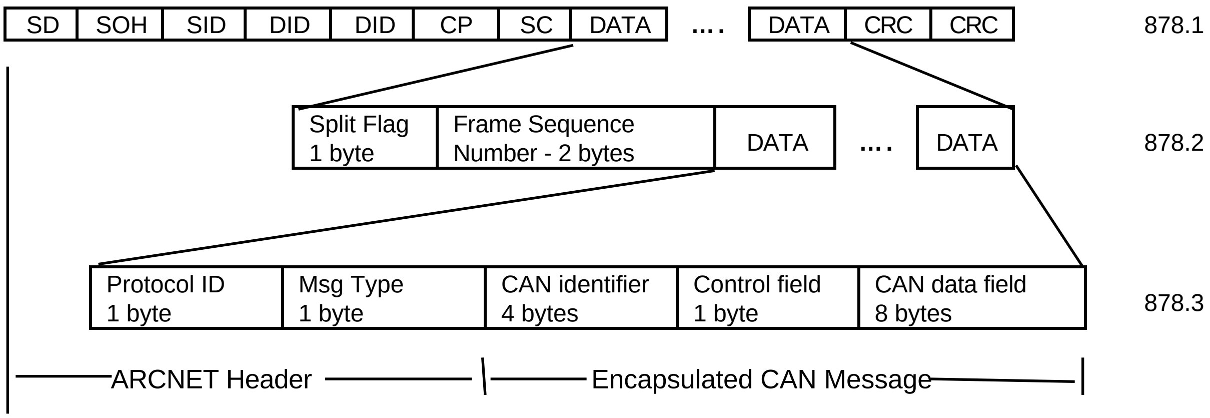

Figure 8-CAN messages are encapsulated into an ARCNET frame complying to the ARCNET Trade Association fragmentation (878.2) and encapsulation (878.3) standards.

mode for ARCNET, a maximum of 252 bytes can be sent. However, we must reserve an additional five bytes for the fragmentation and encapsulation standards. Therefore, a total of 247 bytes are available for CAN messages. A total of 19 CAN messages can be sent within one ARCNET short packet. However, this would only be done if CAN messages are being queued. In general, only one CAN message would be sent per ARCNET frame and the frame length would be shortened accordingly as shown in figure 8. Variable packet lengths are a feature of ARCNET and this feature is used in the CAN-BRIDGE.

CALCULATING LATENCY

In order to calculate the time it takes to send a message over ARCNET, a few delay constants must be known. First there is a turnaround delay (Tta) of 12.6µs due to the ARCNET controller chip. This is the time it takes from the end of a received transmission to the start of a response. Next is the propagation delay (Tpt) due to the cable. This can range from 0 to 31µs. If the delay exceeds 31µs, then the ARCNET data link protocol will fail. A maximum delay of 31µs exceeds the time required to send a signal down 4 miles of coaxial cable and through ten interposing hubs.

For sake of discussion consider two ARCNET nodes separated 2000 feet (610m) of coaxial cable. The propagation delay factor for coaxial cable is typically 4 ns/m. Therefore, the propagation delay (Tpt) would be 2.44µs one way. For a token pass and the successful delivery of a packet, the following calculation can be made:

ITT	15.6 µs		
Tta	12.6	+	Tpt
FBE	15.6		
Tta	12.6	+	Tpt
ACK	6.8		
Tta	12.6	+	Tpt
PAC	37.6	+	4.4n
Tta	12.6	+	Tpt
ACK	6.8		
Tta	12.6	+	Tpt

The value of n in this example would be 18, five for overhead and 13 for CAN data. Therefore, the delay experienced over ARCNET for one CAN message would be 236.8µs.

If the cable distance was increased to 22,000 feet (6710m) which is slightly more than four miles, a total of ten hubs would need to be added. Each hub introduces a delay of 320ns. The total delay would then be 30µs which approaches the 31µs limit. The impact on latency would be an additional 150µs of delay for a total of 386.8µs.

There are other sources of delay. The time it takes for the microprocessor to capture a CAN message and send it to the ARCNET controller is about 300µs which is the same time it takes to reverse the process. Therefore, the total latency from one CAN segment to a remote CAN segment would be 836.8µs at 2000 ft. and 986.8µs at 4 miles. Therefore, the longer distance does not impact latency significantly, and the total latency, which is less than 1 ms, would not impact most systems that are handling input/output (I/O) data.

SYSTEM CONSIDERATIONS

There are some design considerations when implementing a remote bridging system.

By its very nature, the CAN-BRIDGE system introduces additional signal latency which may disturb CAN systems with tight timing constraints. With the DeviceNet protocol, there has not been any evidence of any timing problems. However, the potential exists for a system to erroneously signal a failed response to an action when short cabling delays are assumed.

Within a CAN segment, at least one device must acknowledge the valid receipt of another device's transmission. That acknowledgment, however, does not extend across the CAN-BRIDGE. Even though a successful transmission occurred on a CAN segment, that transmission must be replicated on all other CAN segments generating additional acknowledgments. Therefore, it is possible that a replicated transmission on one CAN segment may fail due to a cabling problem resulting in no acknowledgment while all other CAN segments view the transmission successful. Therefore, upper layer protocols should not rely upon the CAN data link acknowledgment as sole indication of a successful transmission. Additional error checking should be incorporated in the upper layer protocol.

Single nodes can operate on an individual CAN segment with remote bridging. Since each CAN-BRIDGE has one internal CAN chip, this CAN chip acknowledges the single node's message. Without remote bridges, a single node will fail to hear an acknowledgment and will continuously retry.

Some CAN upper layer protocols support autobauding which is difficult for the CAN-BRIDGES to implement. The CAN-BRIDGE must appear to the complete system as an extension cord working all the time. If the CAN-BRIDGES invoke an autobauding algorithm, they will at one point fail to function as a communications link thereby confusing the devices on each side of the remote bridges which may also be undergoing an autobaud algorithm. It is best that the CAN-BRIDGE baud rates be settable by way of a switch. There is, however, no inherent reason why individual CAN segments cannot be set to different baud rates.

Using the same extension cord analogy, it would appear that a remote bridging system must be powered before or at the same time as the CAN devices or host controller in order that all devices can execute initialization routines such as duplicate MAC ID tests as in the case of DeviceNet. However, if a remote bridge loses power while all other devices remain powered, the failure mode should be no different than cutting the cable in the middle of a CAN segment. When power is restored to the remote bridges, the restart sequence should be the same as if the maintenance person reconnected a disconnected cable.

Actually, ARCNET is capable of spanning distances much greater than four miles if extended timeouts are invoked. There has been no testing done to verify that the increased latency would not disrupt the CAN messaging. If extended timeouts are used, it is critically important that all ARCNET nodes be set to the same timeout otherwise the reconfiguration algorithm will fail to operate.

CAN networks are usually configured in a bus or multidrop topology while ARCNET can be configured as a bus, star or distributed star topology. Therefore CAN implementations can take advantage of the more flexible ARCNET cabling options.

Implementing fiber optics over any reasonable distance with CAN is difficult due to the increased delays caused by the additional circuitry. However, fiber optic ARCNET solutions are readily available. Therefore, the benefits of fiber optics can be gained simply by adding remote bridges. Note that the propagation delay of fiber optic cable (5ns/m) is 25% more than that of coaxial cable. This is important when calculating ARCNET delay margin.

CONCLUSION

Implementing remote bridging in order to link distant CAN segments is feasible. By using ARCNET as the interconnecting means, additional benefits such as star topology and fiber optics are achieved. The CAN-BRIDGE demonstrates how two very different fieldbus technologies can be integrated into one cohesive network.

ABOUT THE AUTHOR

Mr. Thomas received his BSEE and MSEE from the Illinois Institute of Technology in Chicago, Illinois. He has held engineering positions at Motorola, Johnson and Johnson and Datalogics before founding Contemporary Control Systems. He is a senior member of the Institute of Electrical and Electronics Engineers (IEEE), a member of International Society of Measurement and Control (ISA) and a registered professional engineer in the State of Illinois.

BIBLIOGRAPHY

1. Controller Area Network-A Serial Bus System-Not Just for Vehicles, CAN in Automation (CiA)

2. Application Note, Bit Timing Parameters for CAN Networks, Report No. KIE 07/91 ME, Philips Components, Buehring, Peter, 1991

3. PC Magazine Guide to Connectivity Second Edition, Duerfler, Frank J., Jr., Ziff-Davis Press, 1992

4. Local Area Network: Token Bus (2.5Mbps), ANSI/ATA 878.1, ARCNET Trade Association, 1992

5. Guide to Configuring an ARCNET Network with Contemporary Control Systems, Contemporary Control Systems, Inc., 1994

6. DeviceNet Specification, Volume 1, Release 2.0, Open DeviceNet Vendors Association, 1997

7. ARCNET Designer's Handbook 61610, Edition 02, Datapoint Corporation, 1988

8. ARCNET Packet Fragmentation Standard, ATA 878.2, ARCNET Trade Association, 1992

9. ARCNET Protocol Encapsulation Standard, ATA 878.3, ARCNET Trade Association, 1993