ETHERNET
ARCNET
CAN

Proposed Network Hierarchy for Open Control

By George Thomas
Contemporary Controls

CONTEMPORARY CONTROLS®
Embedded Networking

INTRODUCTION

When we discuss control strategy, the issue of networks is always raised. Since network requirements vary depending upon the complexity of control, different network technologies are usually specified for the various levels of control hierarchy. It is common to identify three different networks when describing a control system. The lowest level is the device network that is used to link sensors and actuators to controllers. Above that is the control level that links the various controllers together. The highest level is the information level used to link the control system to the enterprise-wide information system. With the movement toward open control systems, it is only logical to pick three open networking standards to complete the control system networking hierarchy. What is recommended here is Ethernet for the information network, ARCNET for the control network and CAN for the device network.

NETWORK HIERARCHY

Device Level

Devices such as sensors and actuators share a common communications bus with an I/O scanner located in a controller. Power for the devices might come from the cable. Communication messages are usually short, fast and frequent. Delivery must be dependable.

Control Level

At this level controllers communicate to other controllers. Message length increases as well as data speed. Message delivery must be dependable and predictable to ensure real-time coordination of the control strategy.

Information Level

At this level data from the control system is made available to the enterprise-wide network. Usually the timeliness of delivery of the data is unimportant. What is important is that connectivity to worldwide standards such as the Internet is achieved. Message length can be long.

ETHERNET EVERYWHERE?

There has been much discussion recently regarding the applicability of using Ethernet at all levels of the control hierarchy. Since Ethernet is so prevalent in the office and frequently used as the enterprise network for high-end controllers, it would seem to be a natural to use Ethernet at the control level or even at the device level as proposed by some in our industry. The arguments for its use include low cost, good connectivity and simple migration to higher speed networks. The cry to use "standard" Ethernet ignores the attributes of other open standards such as ARCNET and CAN that are better suited at the lower levels of the network hierarchy.

What is standard Ethernet?

I am not sure what standard Ethernet is but it certainly is not the 2.94 Mbps version that came out of Xerox's Palo Alto Research Center (PARC) in the early 70s. In 1980, Digital Equipment Corporation (DEC), Intel and Xerox published the DIX V1.0 standard which boosted the speed of Ethernet to 10 Mbps while maintaining Ethernet's thick trunk cabling scheme. In 1982 the DIX V2.0 standard was released which is now commonly referred to as Ethernet II. Xerox then relinquished its trademark.

At the time of the first DIX standard, the Institute of Electrical and Electronic Engineers (IEEE) were attempting to develop open network standards through the 802 committee. In 1985 the IEEE 802.3 committee published "IEEE 802.3 Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications." This technology is

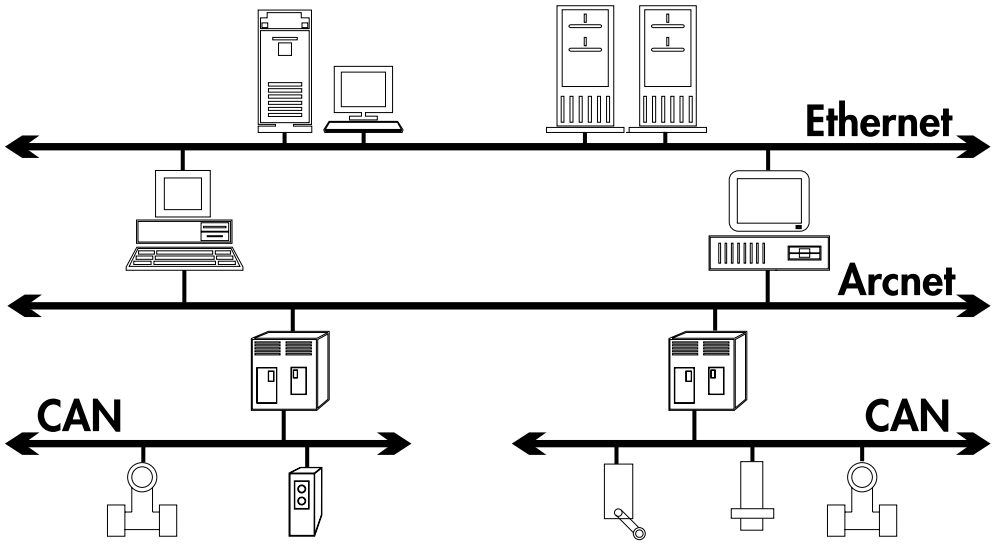

Figure 1 depicts the generalized network model consisting of three separate networks. Although actual systems may not incorporate three separate networks, it is helpful to discuss the services expected at each of the levels.

called 802.3 CSMA/CD and not Ethernet; however, it is frequently referred to as Ethernet even though the frame definition differs from DIX V2.0. Although 802.3 and DIX frames can coexist on the same cable, interoperability is not assured. Therefore, when discussing "Ethernet," it is necessary to clarify 802.3 frames or DIX V2.0 frames.

To further confuse issues, standard Ethernet sometimes means an attached protocol — mainly TCP/IP. Ethernet only defines the data link and physical layers of the Open Systems Interconnect (OSI) Reference Model whereas TCP/IP defines the transport and network layers respectively of the same model. Therefore, when the suggestion is made to use standard Ethernet at the information, control and device levels, does this mean TCP/IP connectivity as well?

ETHERNET FRAMES

The two types of Ethernet frames used in industry are similar. The DIX V2.0 frame, frequently referred to as the Ethernet II frame, consists of an eight-byte preamble, six-byte source and destination addresses, a two-byte type field used to identify higher layer protocols, a variable data byte field followed by a four-byte frame check sequence (FCS) field. The IEEE 802.3 frame divides the preamble into a seven-byte preamble followed by a single byte start of frame delimiter (SFD).The two-byte type field now becomes a two-byte length field. The data field now includes an 802.2 logical link control (LLC) field that precedes the actual data. The FCS remains the same.

Preamble

The DIX preamble consists of 64 bits of alternating "1s" and "0s" but ending with two "1s" to indicate that a valid frame is to begin. This creates a 10 MHz signal that synchronizes the receivers on the network before actual data arrives. Ethernet uses Manchester encoding.

The IEEE redefined the preamble to be seven bytes of preamble, the same as the DIX preamble, followed by a one-byte start of frame delimiter (SFD) which looks like the last byte of the DIX preamble. There is no change in operation between the DIX preamble and the IEEE preamble and SFD byte. Both preambles are not considered part of the frame when calculating the size of the overall frame.

Destination Address

In the DIX standard the first bit of the 48-bit destination address indicates if the address is a multicast address or a physical address. A "0" indicates a unicast transmission to the indicated destination while a "1" indicates a multicast or group address.

The IEEE standard further defines the second bit of the 48-bit destination to indicate if the address is locally administered or globally administered. This bit is a "0" when the address is globally administered; that is, assigned by the Ethernet interface manufacturer.

A 48-bit address of all "1s" is a broadcast address in both DIX and IEEE formats indicating that the transmission is directed to all devices on the network.

Source Address

The 48-bit source address is appended to the transmission as an aid to the higher layer protocols. It is not used for medium access control. To avoid duplicate node IDs for global addresses, the Ethernet adapter manufacturer obtains an Organizationally Unique Identifier (OUI) from the IEEE (for an administration fee).The OUI is 24-bits long and is used as the most significant portion of the 48-bit address. The manufacturer, using good record keeping, will assign sequential numbers to each adapter card he makes thereby creating a worldwide unique address. With 24-bits to work with, a lot of adapters can be produced from a single manufacturer. A list of OUI assignments can be found on the Internet.

Type and Length Field

The original intention of Ethernet was never to use its data link layer as the means for providing guaranteed delivery of data. It was always the intent that a higher layer protocol would do that service. Therefore it was only necessary to identify by number which higher layer protocol was being used through the two-byte field in the DIX frame. Originally, Xerox maintained the assignments and now IEEE provides the administration.

The 802.3 standard does not include the type field but instead defines it as a length field. Per the 802.3 standard, a value in this field of 1518 or less indicates the length of the data field, while values above this may be ignored, discarded or used in a private manner. These out of bound values could then be used to identify higher layer protocols just like DIX frames.

What is important here is that since DIX and IEEE frames are identical in terms of the number of bits and length of fields, both frames can coexist on the same network but may not be able to communicate to one another. Much of the existing TCP/IP software that binds to Ethernet uses DIX frames and not 802.3 frames, so care must be exercised when selecting or developing software or claiming interoperability.

Data Field

A raw Ethernet frame (no encapsulated protocol or LLC) can

be up to 1500 bytes long but no less than 46 bytes. This is the DIX frame.

Although the total available length of the IEEE data field is the same as the DIX frame, the LLC header reduces the amount of field available for actual data or payload as it is sometimes referred to. If the LLC header and actual payload are less than 46 bytes, the data field must be padded to 46 bytes to ensure that the transmission is not interpreted as a runt packet or packet fragment.

Frame Check Sequence

Both the DIX and IEEE standard use four bytes to hold the CRC-32 check on the complete frame from destination address all the way to the end of the data field. The receiving station calculates its own CRC-32, checks on the received data and compares the results with the transmitted CRC-32 value for a match indicating a successful reception. Note that there is no inherent mechanism in the Ethernet data link layer protocol to inform the source node that a reception was accepted or rejected due to a failed CRC-32 check. That task is left to the higher layer protocol.

ETHERNET PHYSICAL LAYERS

Although Ethernet was originally designed as a coaxial bus system, alternate physical layers have evolved since the early 80s. The IEEE 802 committee has defined several physical layers and that is why it is important to specify the correct option when selecting Ethernet.

10BASE5

The original Ethernet was configured as a bus system with a thick coaxial cable as the medium. That is what was specified in the 1980 DIX standard. An external transceiver called a medium attachment unit (MAU) clamps at particular points on the cable marked by stripes every 2.5 meters. From the transceiver, an attachment unit interface (AUI) cable connects to an AUI port on the actual Ethernet adapter that fits into the computer. The AUI port is a DB-15 connector. A coaxial segment can be up to 500 meters long and AUI cables are each restricted to 50 meters in length. A total of 100 transceivers can occupy one trunk segment. Individual trunk segments can be cascaded using repeaters up to 2000 meters. In 1985 the IEEE standardized this configuration as 10BASE5 to signify 10 Mbps baseband signaling up to 500 meters in length.

Thick coaxial cable is indeed bulky and its topology is not always convenient to wire in a plant. Troubleshooting a 100-station segment could be a nightmare, so you do not see new 10BASE5 installations. There is no support for this cable with Fast Ethernet technology.

10BASE2

The answer to the bulkiness of 10BASE5 along with its expense was Thinnet or Cheapernet standardized in 1985 as 10BASE2. Thinnet again was a bus topology but this time with internal transceivers. A thin RG-58/u coaxial cable interconnects up to 30 stations to a maximum length of 185 meters. Segments can be repeated up to 740 meters. BNC style connectors, terminators and taps are used to cable the system. Although easier to install than 10BASE5, the focus on new installations is towards twisted-pair cabling. This cable is likewise not supported by Fast Ethernet.

10BASE-T

In 1990 the IEEE published 10BASE-T after pioneering work was done to introduce twisted-pair cabling and star topology to Ethernet installations. The10BASE-T Ethernet adapters have internal transceivers and RJ-45 connectors. Usually two-pair unshielded cabling is attached to a hub in a point-to-point fashion. Bus connections are not allowed. The connection between an adapter and hub cannot exceed 100 meters in length. Hub-to-hub connection length can vary depending upon the medium used. If another twisted-pair connection is used, the maximum length is again 100 meters. With Thinnet it is 185 meters and with thick coaxial cable 500 meters.

The star topology is much easier to troubleshoot than a bus system; however, the reliability of the hub now must be considered in the overall reliability of the system. Another reason for the focus on twisted-pair is that development of Fast Ethernet is based on twisted-pair and not coaxial cable providing no migration path for installed coaxial cable.

10BASE-F

The 10BASE-F standard is actually a series of fiber optic standards. Fiber optics provides long distance, higher-speed migration, noise immunity and electrical isolation. There are three media standards:

10BASE-FL	*This fiber link standard replaces older FOIRL standard.*
10BASE-FB	*This backbone standard is not very popular.*
10BASE-FP	*This passive hub technology is also not popular.*

The 10BASE-FL standard requires a duplex 62.5/125µm fiber optic cable for each link. Transmission distances of up to 2 km are possible as well as full-duplex operation.

MEDIUM ACCESS CONTROL

What follows is a discussion of the medium access control protocol for a 10 Mbps half-duplex Ethernet network operating with several nodes.

When a station wants to transmit, it first waits for an absence of a carrier, which would indicate that some other station is transmitting. As soon as silence is detected, the station waiting to transmit continues to defer until the Interframe Gap (IFG) time has expired which is 96-bit times (9.6μs). If a carrier still appears to be absent, the station begins to transmit while observing its collision sense circuitry. If no collision is detected, the transmitting station assumes the transmission was sent successfully. If the transmitter detects an early collision, one which occurred during the preamble, the station continues to send the preamble plus 32 bits of data called a jam signal. This ensures that other stations will note the collision as well. After the collision, the transmitting station will backoff from retransmitting based upon a backoff algorithm. If no collisions are detected after 512-bit times (not counting the preamble), the station is assumed to have acquired the channel and no late collisions should occur on a properly working network. The collision counter is cleared. This 512-bit time (51.2 μs) is called the slot time and is critical in the way Ethernet arbitrates access to the cable.

Collision Domain

This slot time defines the upper bound limit of the total propagation delay of a transmitted symbol from one end of the network to the farthest end and back. This includes the time it takes the symbol to travel through cables, repeaters and MAUs and varies with devices used. However, regardless of the path, the resulting propagation delay must be less than the slot time. Therefore the slot time defines Ethernet's maximum network diameter which limits its collision domain. A collision domain that exceeds the maximum network diameter violates Ethernet's medium access control mechanism resulting in unreliable operation.

Collisions can generate runt packets that are less than 512 bits in length. These can be detected by the receiving nodes and discarded accordingly. That is why it is important that a minimum valid Ethernet frame always be sent to distinguish valid packets from packet fragments. A minimum of 46 bytes in the data field ensures that a valid Ethernet frame is 512-bits long.

If the network diameter is small, collision detection is faster and the resulting collision fragments are smaller. As the network diameter increases more time is lost detecting collisions and the collision fragments get larger. Increased network diameter aggravates the collision problem. Silence on the line does not necessarily mean a distant transmitter has not already sent a packet down the cable, which will eventually result in a collision.

Collision Detection

A collision is defined as two stations attempting to transmit at the same time. On coaxial cable transceivers, there is circuitry to detect the DC level of the signal on the cable. This is the indicator of a collision. On fiber optic and twisted-pair interfaces with separate receive and transmit circuitry, a collision is detected by the simultaneous receiving and transmitting of data. Remember that we are discussing half-duplex Ethernet that allows either transmitting or receiving but not at the same time. Only transmitters look for collisions and it is their responsibility to reinforce a collision with a jam signal. Receivers only look for valid packets and automatically discard runt packets that are caused by collisions. Once a collision is detected by simultaneous transmitters, these transmitters will follow a backoff algorithm

Backoff Algorithm

When a collision occurs on the network, the colliding transmitters will backoff from retransmitting for a time determined by a backoff algorithm. This algorithm requires each transmitter to wait an integral number of slot times (51.2 μs) before attempting a new transmission sequence. The integer is determined by the equation:

$$0 < r < 2^k \text{ where } k = \min(n, 10)$$

The variable k is actually the number of collisions capped at a maximum of 10. Therefore, r can range from 0 to 1023 when k = 10. The actual value for r is determined by a random process within each Ethernet node. As the number of

Ethernet II DIX Frame

64 bits	48 bits		48 bits	16 bits	368 to 12000 bits (46 to 1500 bytes)	32 bits
Preamble	Individual/ Group Address Bit	Destination Address	Source Address	Type	Data	Frame Check Sequence

IEEE 802.3 Frame

56 bits	8 bits	48 bits			48 bits	16 bits	368 to 12000 bits (46 to 1500 bytes)	32 bits
Preamble	SFD	Individual/ Group Address Bit	Globally/ Locally Administered Address Bit	Destination Address	Source Address	Length	LLC/Data	Frame Check Sequence

Figure 2—Two types of Ethernet frames are used in industry.

consecutive collisions increases, the range of possible backoff times increases exponentially. The number of possible retries is also capped but at 16.

For example, assume two stations A and B on the network wanting to transmit. They both wait for an absence of carrier and then wait for the IFG time to expire before initiating a transmission. It does not matter if they are 10 meters or 2500 meters apart. They could both be sensing silence and simultaneously begin to transmit causing a collision at some point. They each sense the collision and back off for either 0 or 1 slot time. The odds are 50-50 they will pick the same value and collide again. If they do, they will now back off for either 0, 1, 2 or 3 slot times. The probability of collision is now 25%. Eventually, one will win in which case its collision timer is cleared to zero while the other collision timer continues to increment until a successful transmission.

A high number of retires indicates a busy network with more stations wanting to transmit than originally assumed. That is why the backoff time range is increased exponentially to provide more possible slot times for the additional stations. At ten retries, it is assumed that 1024 simultaneous transmitters exist. This becomes the upper bound limit of stations that can coexist on one Ethernet network. Actually this is the logical limit. Physically it may be impossible to have that many stations on one collision domain without violating cabling rules.

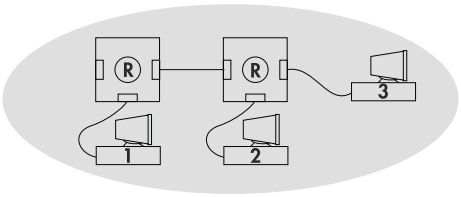

Figure 3—For proper operation, a collision domain must be within the maximum network diameter.

Channel Capture

As shown above, the Ethernet backoff algorithm provides a means for peer stations to each gain access to the network. Access is provided to all but in an unpredictable fashion. The question is if access is fair.

Assume the same two stations A and B as before. This time, however, they both have high amounts of data to send and they attempt to send at the same time and collide on the first attempt. They both back off but this time A was successful. A's collision counter is cleared but B's does not clear. If station A has more data to send and it is quick to assemble another packet to send, it might collide with B again. This time B could be selecting higher and higher backoff times as its collision counter continues to increment. However, station A feels it has only experienced the first collision and will probably select a much lower timeout allowing it to transmit and assemble another packet and could beat station B again in the backoff contest. This phenomenon of channel capture is real and demonstrates that access to the network is neither fair nor predictable. The next time around station B could get the upper hand and limit A's access. If another station C decides to transmit as well, it could beat out station A due to the state of A's collision counter. In actuality a station that was last to arrive could transmit first.

Improving Ethernet's Determinism

There has been much discussion in the literature about implementing methods to improve the determinism of Ethernet. One approach is to incorporate a master/slave protocol such as MODBUS or OPTOMUX on top of Ethernet. In this situation, the slaves only respond to the master's commands thereby controlling the traffic on the cable and thus avoiding collisions. The downside of this approach is that you forfeit the inherent multimaster capability of Ethernet.

BACKOFF RANGE AS A FUNCTION OF COLLISIONS

Collision on Attempt Number	Estimate of Number of Other Stations	Range of Random Numbers	Range of Backoff Times (µs)
1	1	0……..1	0……..51.2
2	3	0……..3	0……153.6
3	7	0……..7	0……358.4
4	15	0……15	0……768.0
5	31	0……31	0…..1587.2
6	63	0……63	0…..3225.6
7	127	0…..127	0…..6502.4
8	255	0…..255	0…13056.0
9	511	0…..511	0…26163.2
10	1023	0…1023	0…52377.6
11	1023	0…1023	0…52377.6
12	1023	0…1023	0…52377.6
13	1032	0…1023	0…52377.6
14	1023	0…1023	0…52377.6
15	1023	0…1023	0…52377.6
16	Too High	N/A	Discard Frame

Table 1—Backoff range increases exponentially with the number of collisions.

Another suggestion is to develop a token-passing protocol that would be implemented in Ethernet's data field. This would have to be developed and its acceptance would have to be sought. The software burden would increase and technologies such as ARCNET already can do this with built-in firmware transparent to the application program requiring no development.

Others suggest simply increasing the data rate to 100 Mbps by using Fast Ethernet technology. By simply using raw horsepower messages will get through with or without collisions. The collision domain decreases by a factor of 10 when migrating to 100 Mbps Ethernet resulting in a maximum network diameter of only 205 meters, which is a small size network. Of course all nodes would need to be capable of communicating at 100 Mbps which could be a burden for under-powered microcontrollers.

One approach is to avoid collisions altogether by using full-duplex technology and switched hubs. In this scheme each node is paired with a port on the hub. Each node/port arrangement creates it own collision domain separate from all others. There are no collisions with a full-duplex link. The switching hub directs messages to other links by observing the destination address within the frame. Switching hubs are more expensive than non-switched hubs and they introduce more latency by their "store and forward" nature. The switching hub now becomes an integral component of the control strategy.

There is an IEEE 802.1p task group studying schemes that would provide higher priorities to the transmission of time-critical data. This activity is mainly addressing the way multicast frames are sent.

Expanding an Ethernet Network

Expanding an Ethernet network is possible by the use of repeaters while maintaining one collision domain. If expansion is required beyond a collision domain, this can only be accomplished by the use of bridges, switches or routers. To maintain one collision domain, a symbol sent from the extreme end of the network must be able to make a complete round trip within the slot time of 512-bits (51.2 µs at 10 Mbps). Calculating the complete propagation delay through adapters, AUI cables, transceivers, trunk cables and repeaters is possible but is also a challenge. Table 2 provides information on the maximum number of MAUs per segment and the maximum segment length. The maximum allowable segment length, as well as the repeaters themselves, has been assigned delay values by the 802.3 specification.

The 802.3 specification discusses ways to interconnect cable segments with repeater sets without exceeding the collision domain. A repeater set is defined as repeater electronics and two or more attached MAUs — one for each segment to be connected. The system designer can use either transmission system model 1 or transmission system model 2. Approach 2 is the detailed approach where exact delay calculations and intergap shrinkage calculations are made. Approach 1 is the simplified approach, which is not as exacting as approach 2. Approach 1 has been further simplified by creating the 5-4-3 rule.

5-4-3 Rule

The 5-4-3 rule states that a system can have up to five segments in series, with up to four repeaters and no more than three mixing segments. The remaining two segments must be link segments. A mixing segment is defined as a segment that may be connected to more than two transceivers. In other words, a bus segment. Only coaxial cable can be used for a bus segment (I am ignoring 10BASE-FP) while fiber optic and twisted-pair cable can be used as link segments. A link segment can only have two transceivers and it must support full-duplex operation (separate transmit and receive channels) to speed up collision detection. This simplified rule does not address all the possible combinations but it does yield some gross network diameters. For example, all five segments cannot be 10BASE5 or 10BASE2. If all five were 10BASE-T then the diameter would be 500 meters. With fiber optics it is different. You cannot use the maximum segment length for all five segments. In the case of 10BASE-F the maximum diameter is 2500 meters. You need to read the standard to understand this restriction.

The 5-4-3 rule does not address the three repeater configuration which yields four segments. In this case, all segments can be mixing providing a network diameter of 2000 meters for 10BASE5 and 740 meters for 10BASE2. For other configurations you need to refer to approach 2.

ETHERNET MAXIMUM MEDIA SEGMENT LENGTH

Media type	Maximum number of MAUs per segment	Maximum segment length (m)
Mixing segment		
10BASE5	100	500 (trunk) 50 (AUI)
10BASE2	30	185
Link segment		
FOIRL	2	1000
10BASE-T	2	100
10BASE-FL	2	2000

Table 2—Expansion rules require that segments be identified as being either mixing or link.

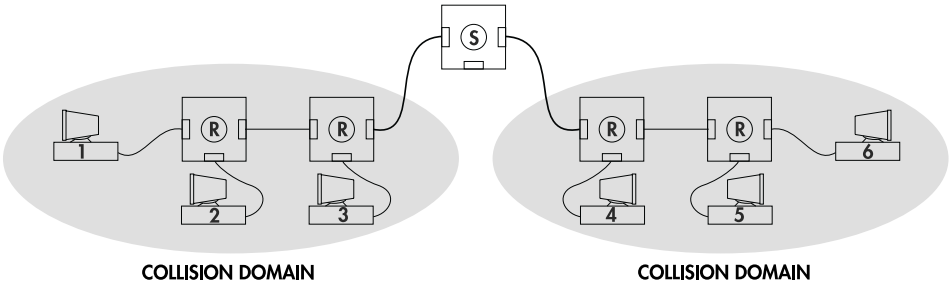

Figure 4—A switching hub, bridge or router is required to interconnect two or more collision domains.

ARCNET AS A CONTROL NETWORK

ARCNET was originally developed as an office automation network in the late 70s. Although ARCNET's use as an office automation network has diminished, ARCNET continues to

find success in the industrial automation industry because its performance characteristics are well suited for control. ARCNET has proven itself to be very robust. ARCNET also is fast, provides deterministic performance and can span long distances making it a suitable control technology. Although not an IEEE standard, ARCNET is defined by ANSI/ATA 878.1.

Unlike office automation networks, a control network must deliver messages in a time predictable fashion. ARCNET's token-passing protocol provides this timeliness. Control network messages are generally short. ARCNET packet lengths are variable from 0 to 507 bytes with little overhead and, coupled with ARCNET's high data rate, typically 2.5 Mbps, yields quick responsiveness to short messages. Control networks must be rugged. ARCNET has built-in CRC-16 (cyclic redundancy check) error checking and supports several physical cabling schemes including fiber optics. Finally there must be low software overhead. ARCNET's data link protocol is self-contained in the ARCNET controller chip. Network functions such as error checking, flow control and network configuration are done automatically without software intervention.

In terms of the OSI Reference Model, ARCNET provides the Physical and Data Link layers of this model. In other words, ARCNET provides for the successful transmission and reception of a data packet between two network nodes. Nodes are assigned addresses and one ARCNET network can have up to 255 uniquely assigned nodes.

Deterministic Performance

The key to ARCNET's performance and its attractiveness as a control network is its token-passing protocol. In a token-passing network, a node can only send a message when it receives the "token." When a node receives the token it becomes the momentary master of the network; however, its mastery is short lived. The length of the message that can be sent is limited and, therefore, no one node can dominate the network since it must relinquish control of the token. Once the message is sent, the token is passed to another node allowing it to become the momentary master. By using token passing as the mechanism for mediating access of the network by any one node, the time performance of the network becomes predictable or deterministic. In fact, the worst case time that a node takes to deliver a message to another node can be calculated. Industrial networks require predictable performance to ensure that controlled events occur when they must. ARCNET provides this predictability.

Logical Ring

A token (ITT — Invitation to Transmit) is a unique signaling sequence that is passed in an orderly fashion among all the active nodes in the network. 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 must be unique and can range from 0 to 255 with 0 reserved for broadcast messages.

For example, assume a network consisting of four nodes addressed 6, 109, 122 and 255. Node assignments are independent upon the physical location of the nodes on the network. Once the network is configured, the token is passed from one node to the node with the next highest node address even though another node is physically closer. All nodes have a logical neighbor and will continue to pass the token to their neighbor in a logical ring fashion regardless of the physical topology of the network.

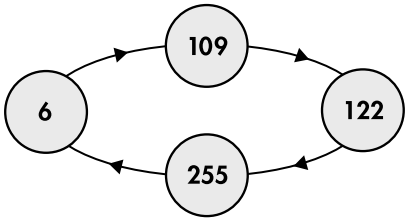

Figure 5 - The logical ring has nothing to do with the physical placement of nodes. The node with the next highest address is that node's logical neighbor. However, logical neighbors could be located at the extreme ends of a physical multi-node network.

Directed Messages

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 by sending out a Free Buffer Enquiry (FBE). The destination node responds by returning an Acknowledgement (ACK) meaning that a buffer is available or by returning a Negative Acknowledgement (NAK) meaning that no buffer is available. Upon an ACK, the source node sends out a data transmission (PAC) with either 0 to 507 bytes of data (PAC). If the data was 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. This is called a fragmented message. The packets are recombined at the destination end to form the entire message.

Broadcast Messages

ARCNET supports a broadcast message, which is an unacknowledged message to all nodes. Instead of sending the same message to individual nodes one message at a time, this message can be sent to all nodes with one transmission. Nodes that have been enabled to receive broadcast messages will receive a message that specifies node 0 as the destination address. Node 0 does not exist on the network and is reserved for this broadcast function. No ACKs or NAKs are sent during a broadcast message making broadcast messaging fast.

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 a node 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 their own node address. The node with the highest address will timeout first and begin a token pass sequence to the node with the next highest 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 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.

If a node leaves the network the reconfiguration sequence is slightly different. When a node releases the token to its logical neighbor, it continues to monitor network activity to ensure that the logical neighbor responded with either a token pass or a start of a transmission sequence. If no activity was sensed, the node that passed the token infers that its logical neighbor has left the network and immediately begins a search for a new logical neighbor by incrementing the node address of its logical neighbor and initiating a token pass. Network activity is again monitored and the incrementing process and resending of the token continues until a new logical neighbor is found. Once found, the network returns to the normal logical ring routine of passing tokens to logical neighbors.

With ARCNET, reconfiguration of the network is automatic and quick without any software intervention.

Several Cabling Options

ARCNET is the most flexibly cabled network. It supports bus, star and distributed star topologies. In a bus topology, all nodes are connected to the same cable. The star topology requires a device called a hub (passive or active) which is used to concentrate the cables from each of the nodes. The distributed star (all nodes connect to an active hub with all hubs cascaded together) offers the greatest flexibility and allows the network to extend to greater than four miles (6.7 km) without the use of extended timeouts. Media support includes coaxial, twisted-pair and glass fiber optics.

ARCNET MAXIMUM MEDIA SEGMENT LENGTH

Media type	Maximum number of MAUs per segment	Maximum segment length (m)
Coaxial star	2	610
Coaxial bus	8	305
Twisted-pair star	2	100
Twisted pair bus	8	122
Fiber optic	2	2000
DC EIA-485	17	274
AC EIA-485	13	213

Table 3—Maximum segment length is based upon 2.5 Mbps operation.

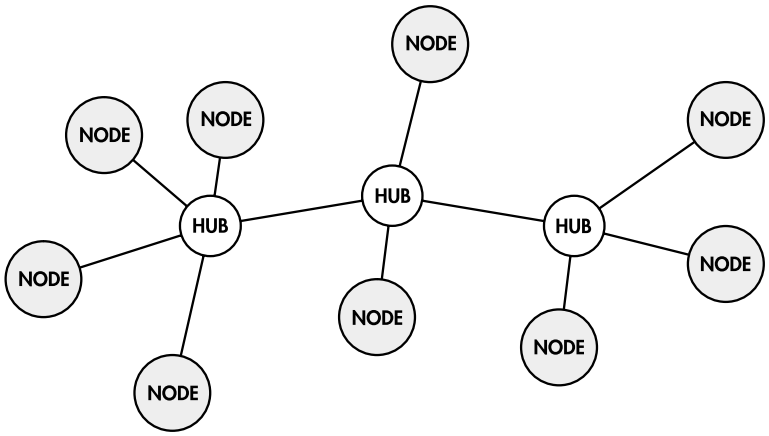

Figure 6 - With ARCNET, a distributed star topology is possible using hubs

ARCNET Frames

ARCNET ANSI/ATA 878.1						
6 bits	11 bits	11 bits	22 bits	8 or 16 bits	11 to 5588 bits (1 to 508 octets)	16 bits
Preamble	SOH	Source Address	Destination Address	Length	SC/Data	Frame Check Sequence

Figure 7—A byte within an ARCNET frame is preceded by a three-bit preamble.

ARCNET utilizes RG-62/u coaxial cable that is similar in size to 10BASE2 Thinnet. In fact, ARCNET's coaxial bus option uses the same style BNC connectors. ARCNET's twisted-pair option is similar to 10BASE-T and ARCNET's fiber optic option has similar performance to 10BASE-FL. Newer ARCNET chips support variable data rates from 19 kbps to 10 Mbps and popular EIA-485 transceivers that utilize twisted-pair cabling. The DC option operates over a wide range of data rates, while the AC option is restricted to a narrow range of data rates. All cabling technologies can be interconnected with active hubs. Unlike Ethernet, ARCNET has more than one frame type. Instead of just a data frame, ARCNET has an ITT, ACK, NAK and FBE (free buffer enquiry) frame. This is because ARCNET has build-in flow control. If a receiver is unable to accept a transmission, it simply informs the transmitter to retry all at another time. Ethernet does not have flow control and it is possible to swamp a receiver thereby losing packets.

Figure 7 shows the ARCNET data packet frame (PAC). Like Ethernet, ARCNET uses both source and destination addresses but they are each only one byte in length. The destination address is repeated to aid the microcode in the ARCNET controller chip. Each byte sent by ARCNET is preceded by two 1s and one 0 yielding 11 bits and is called an octet. The first octet in the data field is the system code that identifies higher layer protocols like the DIX standard. A two-octet length field identifies long packets (less than 508 bytes). Therefore, ARCNET can send only about one-third of what Ethernet can send in its packet.

Expanding an ARCNET network

ARCNET is not a contention based network and therefore does not have collision detection or a collision domain. However, ARCNET has a network diameter that must not be violated. With coaxial cable and ten cascaded hubs, the network diameter is 6700 meters at 2.5 Mbps. This is based upon the fact that the one-way time required for a symbol to travel between the two most distant nodes cannot exceed 31 µs. This diameter can be greatly expanded by invoking extended timeouts without requiring the need for bridges or routers. This makes ARCNET about the easiest technology to configure and expand. Its expansion rules are much simpler than Ethernet's.

CONTROLLER AREA NETWORK (CAN)

An increasing popular device-level network is CAN. Originally designed as an on-vehicle network, CAN borrows some of the attributes of Ethernet. Like Ethernet, CAN is a carrier-sense-multiple-access (CSMA) technology. CAN stations listen for silence and continue to defer for an interframe gap time before initiating a transmission. When a collision occurs, CAN makes the best of the situation. It is designed so that one message will get through even with a collision. Unlike Ethernet, there is no backoff algorithm. Failed transmitters simply try again at the next opportunity. The result is high throughput.

CAN Data Link Layer

CAN was designed by Bosch and is currently described by ISO 11898. In terms of the Open Systems Interconnection model, 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 comp by CAN.

CAN Medium Access Control

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

CAN Standard Frame									
1	12 bits		6 bits			0 or 64 bits	15 bits	2	7 bits
SOF	Identifier	RTR	IDE	Reserved	Length	Data	Frame Check Sequence	Ack	EOF

Figure 8—Standard CAN frames utilize an eleven-bit identifier. The RTR bit is not used with DeviceNet

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 in length at higher speeds. To increase this distance either the data rate is decreased or additional equipment is required.

CAN Frames

CAN transmissions operate using the producer/consumer model. When a CAN device transmits data, 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 8). 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, integral to Ethernet and ARCNET, are not used by CAN. However, higher layer protocols such as DeviceNet and Smart Distributed System define identifier bits in such a way that they become source/destination addresses. Unlike the Manchester encoding used by Ethernet, CAN uses the very efficient non-return to zero (NRZ) encoding with bit stuffing. If at anytime a series of five identical symbols are sent out, the transmitter automatically inserts an opposite symbol to ensure synchronism by the receiver. The receiver automatically rejects the added symbol.

The CAN frame begins with a start of frame (SOF) symbol followed by the identifier and a remote transmission request (RTR) bit. Neither Smart Distributed System nor DeviceNet use this bit. The control field includes an IDE bit used to identify extended 29-bit identifiers. The four-bit length field indicates how many data bytes are included in the frame. Like Ethernet, a frame check sequence is used but incorporates only a CRC-15 check. Unique to CAN is the two-bit ACK field. All CAN receivers are required to acknowledge a successfully received CAN transmission. This is done within the transmission and noted by the transmitter. This is another example of how efficient CAN is.

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.

Expanding a CAN Network

CAN networks are not easy to expand. This is because of its medium arbitration method that requires decisions to be made within one bit time. Repeaters are only useful at the low data rates where distance is limited solely by cable attenuation. At the higher data rates the limit is cable delay and not cable attenuation. Adding more electronics in the form of a repeater introduces more delay further decreasing the length of the CAN network. The only way to extend CAN networks is to use bridging or

CAN(DeviceNet) MAXIMUM MEDIA SEGMENT LENGTH		
Data Rate (kbps)	Maximum number of MAUs per segment	Maximum segment length (m)
500	64	100
250	64	250
125	64	500

Table 4—CAN will operate at data rates besides the ones specified by DeviceNet.

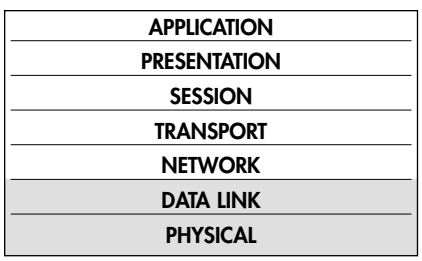

Figure 9 - Ethernet, ARCNET and CAN define the lower two layers of the OSI Reference Model.

routing techniques that break up the CAN collision domains into separate subnets. This approach is similar to the way Ethernet collision domains are separated by using a switching hub.

TCP/IP

So far what has been discussed was the data link performance of Ethernet, ARCNET and CAN. Protocols sit on top of the data link layer and one of the most popular is the TCP/IP suite of protocols. In fact to many people, Ethernet means TCP/IP as well. The transport control protocol (TCP) provides the guaranteed delivery of messages while the internet protocol (IP) provides the means of routing messages between different networks. With the latest version of Microsoft's Windows products, TCP/IP is built-in so it is easy to configure your Ethernet or ARCNET adapter for TCP/IP as long as you have the proper software driver for the adapter you choose. CAN is a different story. Since CAN data packets are so small (only eight bytes), fragmentation is required to encapsulate the IP and TCP headers and data into the data field. This is just not practical. However, with Ethernet and ARCNET there is sufficient room to carry the TCP/IP information even though large messages still require fragmentation. In a non-Windows environment you will need to provide the TCP/IP software which requires about 50K of memory — a burden for microcontrollers.

The other issue is addressing. Each IP node requires a 32-bit address that must be assigned by the node itself or by a host computer attached to the network. This address differs from the physical Ethernet or ARCNET address and care must be exercised when setting up an IP network.
The very nature of IP is to exchange data over several separate networks. The control network is usually just one network and usually private at that. Do we really want the burden of an IP protocol when we are not interconnecting other networks at the control level?

Finally, there is the issue of the real-time performance of TCP/IP. This protocol suite was never designed with speed in mind. It was designed for the reliable transmission of packetized messages over multiple routes. This is important at the information level but not the control level. The recommendation here is to use Ethernet and TCP/IP at the information network level only.

Hubs, Bridges and Routers

Hubs, bridges and routers are used to expand networks. Hubs operate at the physical layer. They provide signal boost, preamble regeneration and signal retiming. Hubs are quite effective in providing these functions for Ethernet and ARCNET but not as effective with CAN because of the increased delay cause by the additional electronics. Even with that said, signal latency is very low with hubs.

Bridges operate at the data link layer and are used to interconnect similar subnets into one logical network. Bridges are effective in expanding Ethernet and CAN networks but are not popular with ARCNET. A switched hub is unique to Ethernet and is actually a special class of bridge. By breaking up a large network into separate collision domains, a larger geographic network can be realized. Bridges introduce data latency because packets must be completely stored and forwarded between subnets. Switches have techniques to reduce this latency.

Routers operate at the network layer and must be "protocol aware." For example, routers respond to IP addressing conventions and either block or forward IP messages. This introduces latency and, therefore, the use of routers should be confined to the information level only. Routers for Ethernet are common while routers for ARCNET and CAN are not.

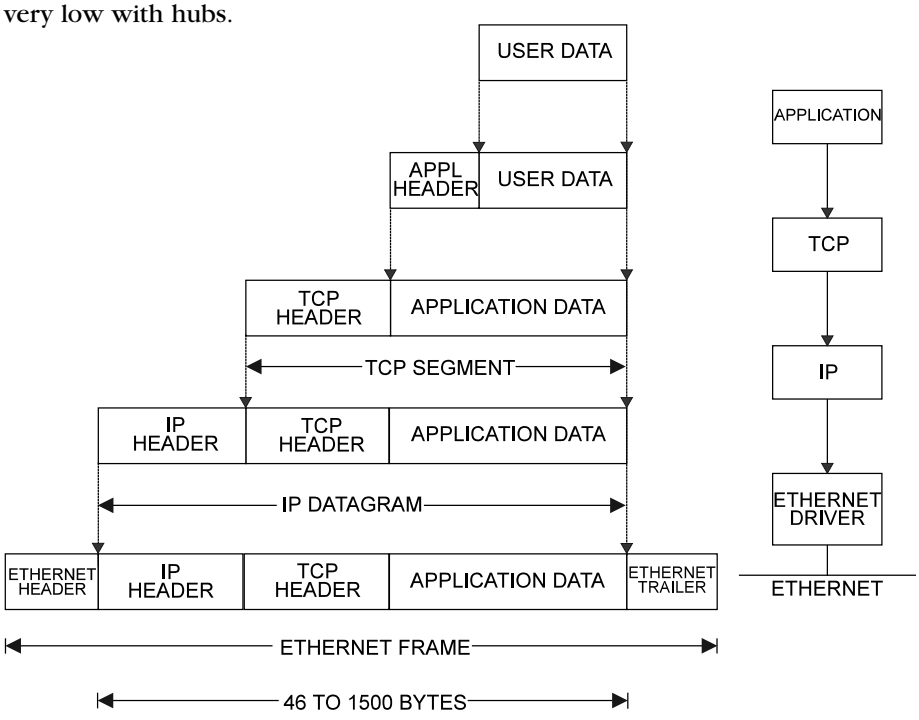

Figure 10—TCP/IP data is encapsulated in Ethernet frames. A similar method is used with ARCNET.

SYSTEM CONSIDERATIONS

What has been discussed is the performance attributes of three networking technologies—Ethernet, ARCNET and CAN. The performance of these technologies differs as well as their suitability at the three levels of the control network hierarchy. We will study the issues involved in making the final determination of what network to use.

Powered Bus

DeviceNet and Smart Distributed System specify a power bus along with a CAN data bus. This makes it convenient to wire up switches and small actuators without the need to provide a separate power cable. Networks such as ARCNET and Ethernet have no provision for powering devices making them less attractive for use as a device network.

Packet size

Ethernet has the largest packet size and is in the best position to send out large amounts of data. ARCNET can do the same but not with the same throughput. CAN with its eight-byte packets is not suitable for sending out large amounts of data. Therefore, Ethernet is best suited for handling the large amount of data sent out on the information level network.

Distance

If you exclude techniques such as bridging, switching and routing, ARCNET provides the largest network diameter and is the easiest to expand with the simplest of expansion rules. Ethernet has the next best distance capability while CAN has the worst. Hubs introduce very little data latency and are the best choice for increasing network diameter. Bridges, switches and routers break the collision domain restrictions but introduce greater data latency. Ethernet's popular twisted-pair wiring further reduces its network diameter. Using ARCNET as a control network allows for simple geographic expansion.

Determinism

CAN provides good determinism even though high priority identifiers are capable of dominating a network. Application layer protocols such as DeviceNet minimize this effect by assigning node IDs within the identifier field. This has the tendency of evening out priorities. ARCNET provides the best determinism due to its token-passing protocol while Ethernet provides the worst due to its contention-based medium access control. Protocol stacks such as TCP/IP further confuse the determinism debate.

Electromagnetic Compatibility (EMC)

DeviceNet provides a CAN cable that includes two shielded pairs covered with an overall shield. ARCNET is predominantly wired with coaxial cable. Although Ethernet has coaxial cable options, the office automation market, which fuels demand for Ethernet, has standardized on unshielded twisted-pair wiring called category 5.

Although twisted-pair wiring is popular in the United States, it is less popular in Europe where there is more sensitivity to EMC issues brought on by the CE marking directive. There is a question if the office grade Ethernet adapters can pass the heavy industry generic immunity standard (EN50082-2) with unshielded twisted-pair cable. Ethernet hubs, switches, bridges and routers may not pass the higher immunity standard as well.

ARCNET's coaxial star transceiver is extremely robust. While transmitting, its output is no less than a 15-volt P-P dipulse. A matched filter tuned to 2.5 Mbps receives this same dipulse providing immunity to noise. Unlike Ethernet, ARCNET does not have sensitive collision detection circuitry vulnerable to outside interference.

CAN has also proven to be quite robust. In-vehicle communications demands high immunity to external noise.

Application Layer Protocols

Ethernet, ARCNET and CAN are termed data link layer protocols and by themselves provide no value. They must be eventually coupled to an application layer protocol in order to perform meaningful control. At the Device level, DeviceNet and Smart Distributed System provide this application layer to CAN. If Ethernet is to be used at this level, a common application layer would have to be developed.

At the information level there are ways for the application to link to the TCP/IP suite of protocols which in turn links to the data link layer. In a UNIX environment, it is Sockets and in a Windows environment, it is WINSOCK. Supporting either of these standards ensures operation on a wide range of computers.

At the control level it is different. The control level is usually considered a private network, and there is no standardized application layer. Control Techniques' CTnet or Eurotherm's ALIN incorporate a private application layer on top of ARCNET.

One, Two or Three Networks

There has been discussion regarding the collapse of the three network hierarchy into one or two networks. For example, could a TCP/IP network be used for both the information network and

control network? Although conceptionally pleasing, there is risk with this approach. Firewalls would have to be put in place to guard the control network from unauthorized access. Routers would need to be configured to ensure that data file transfers at the information level will not impact the determinism required at the control level. Also the control network must be immune to a "PING" request from outside the control network.

One approach to this security problem is to use Ethernet and TCP/IP but have two networks—one for control and one for information. Some machines would then be required to have two Ethernet adapters. This approach certainly provides the desired isolation but if there is a commitment to two networks, why not have one Ethernet and one ARCNET?

At the device level it is hard to imagine having a TCP/IP network. A Motorola 6805 4K microcontroller mounted into a proximity switch can barely execute a DeviceNet slave program. Increasing the performance and memory of the microcontroller just so it can run a TCP/IP stack does not make economic sense.

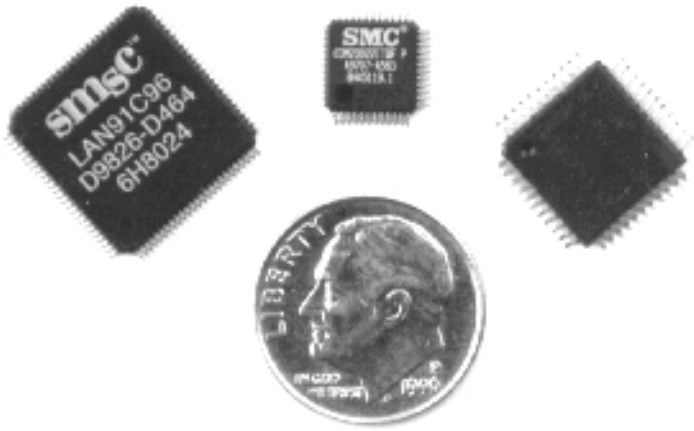

Figure 11 — Ethernet, ARCNET, and CAN controllers (left to right)

CONCLUSION

It has been shown that Ethernet, ARCNET and CAN will provide the necessary networking capability needed at the Information, Control and Device levels, respectively. There has been a lot of discussion of using Ethernet at the Control and Device levels due to its "simplicity" and its perceived low cost. Although Ethernet can perform at these levels, ARCNET and CAN are better suited to the Control and Device levels. As these low cost Ethernet devices are usually office grade devices, they do not receive the same EMC testing that industrial products receive; nor are they designed to work in industrial environments. They are, therefore, less suited to the normally higher noise industrial environment. Also due to Ethernet's twisted-pair low amplitude signaling, Ethernet is more susceptible to electrical noise than ARCNET or CAN. Also, as ARCNET and CAN have built-in determinism, Ethernet must be adapted to provide the normally needed control and device network determinism at the protocol level. There have been a number of suggested protocols; however, in-depth testing will be needed to fully evaluate these protocols. Also setting up a large Ethernet network is not a simple process due to the collision domain rules. ARCNET and CAN provide a much simpler setup process.

AUTHOR

ETHERNET, ARCNET, and CAN—Proposed Network Hierarchy for Open Control, 1999.

George Thomas, Contemporary Controls
gthomas@ccontrols.com

REFERENCES

Practical Networking With Ethernet, Charles E. Spurgeon, 1997, International Thomson Computer Press

CAN System Engineering From Theory to Practical Applications, Wolfhard Lawrenz, 1997, Springer-Verlog

Switched and Fast Ethernet, Second Edition, Robert Breyer and Sean Riley, 1996, Macmillan Computer Publishing USA

TCP/IP Clearly Explained, Second Edition, Pete Loshin, 1997, Academic Press

TCP/IP Illustrated, Volume 1, The Protocols, W. Richard Stevens, 1994, Addison-Wesley Publishing Company

ARCNET Tutorial & Product Guide, Contemporary Controls, 1998

Extending CAN Networks by Incorporating Remote Bridging, George Thomas, 1997, 4th CAN Conference, Berlin, Germany

International Standard ISO/IEC 8802-3 ANSI/IEEE Std 802.3, 1996, The Institute of Electrical and Electronic Engineers, Inc.

FOR MORE INFORMATION:

INDUSTRIAL ETHERNET ASSOCIATION
www.industrialethernet.com

ARCNET® TRADE ASSOCIATION
www.arcnet.com

CiA
www.can-cia.de

CONTEMPORARY CONTROLS®
Embedded Networking
www.ccontrols.com