Token passing local area networks:  
A success story for standards

by MARY JANE STROHL  
Concord Data Systems  
Waltham, Massachusetts

ABSTRACT

This paper presents the software architecture and the software development approach used by Concord Data Systems (CDS) for the development of their IEEE 802.4 standard token passing local area network products. The paper also describes the design goals and features of the CDS implementation of the ISO transport protocol. The transport implementation features a flexible interface that supports, transparently, both on-board and off-board session entities. Finally, the CDS services for network management and Layers 5–7 of the ISO/OSI Reference Model are discussed to show the current features and a migration approach for including standard protocols for these layers as they mature and become accepted standards.
INTRODUCTION

Beginning with a design commitment to develop token passing local area network (LAN) products that are compliant with ANSI and ISO standards, Concord Data Systems (CDS) developed a modular software architecture based on the International Standards Organization's Open System Interconnect (ISO/OSI) Reference Model. The architecture supports an incremental approach to building and integrating standard protocols into products, and has worked successfully to produce a versatile set of high-performance LAN communication products.

Token passing network applications vary from factory control applications to interactive computer terminal servers. This diversity in applications dictates communication architecture alternatives for distributing the OSI upper layer software across hardware boundaries. These alternatives provide both LAN network interface units that serve as front end communication processors for an attached host, and LAN units that function as end systems for applications such as the terminal server.

This paper presents the software architecture approach that CDS has used for developing and integrating standard protocols into its LAN products. It is a building block approach with an interface design concept that gives the product set its versatility. The development of the OSI Layer 4 transport protocol, shown in Figure 1, is used as an example to show how the architecture supports independent development testing integration and performance analysis for one of the OSI Layer Protocols.

The role of Layer 5–7 protocols and network management software in token passing LAN products, shown in Figure 2, is discussed next. Standards are just beginning to evolve for these layers. The status of these standards for LANs and an evolutionary development strategy for them are presented.

A BASIC ARCHITECTURE FOR LOCAL AREA NETWORKING SYSTEMS

The CDS architecture approach was required to offer standard IEEE 802.4 high-performance networking products that meet a variety of application requirements. To achieve this, the CDS design focused on interfaces, a modular architecture based on the OSI Reference Model, the use of standard protocols, and performance.

Interface considerations are important to satisfy a wide variety of application requirements. LAN interface units can encompass a range of products, from simple access units where Layers 3–7 are provided by a host computer, to stand alone devices where all seven layers of the ISO/OSI model reside in the LAN unit. An attractive division of functions is one where the LAN unit is truly a communications processor with the communication layers 1–4 in the LAN unit and the application related layers 5–7 in the host (Figure 1). Off-board interfaces to the network layer or possibly network sublayers are other interface possibilities. There are applications to support all of these physical architectures, but to do this effectively in a product line requires well defined and flexible interfaces. The interfaces must be such that the transport layer can service both on-board and off-board session entities.

With a modular design where interfaces are designed correctly for the first release, upgrades to the modules themselves can be made without making interface changes, giving plug-compatible software module versions. Then, with a download capability, product maintenance becomes reasonable. Products can be maintained by downloading new versions of the upgraded modules. This approach was taken with the design of the LLC code in the CDS LAN unit, and worked successfully. The code supports an HDLI (the General Motor's Manufacturing Automation Protocol (MAP) data link protocol that has a flow control option along with the standard HDLC LAP B protocol) interface to provide host access to the LLC module as well as the on-board network and transport interfaces to LLC.

Along with well-designed interfaces, modularity is another critical design requirement necessary to make this architecture work. The OSI Reference Model, with its layering concept, forces a basic modular architecture. Within the layers, the code design must also be modular. To help realize this modularity in the CDS software design, Pascal was the chosen implementation language. Because of its data typing features, Pascal constructs can be used in a design specification and fold directly into an implementation. The First Systems Pascal (by First Systems Corp., Manhattan Beach, CA) that was used by CDS offers ADA-like concepts, such as an equivalent to the ADA Package, and facilitates large project developments. The philosophy was to develop a modular implementation in Pascal that would give a tight, maintainable, and self-documented code architecture. This could then be used to selectively do assembly recodings of small modules where performance was critical.

The incorporation of standards in the CDS architecture closely followed the developments in the standards communities. With LANs, the first standards to evolve were for Layers 1 and 2, and were driven by the IEEE 802 committees. The medium access (MAC) sublayer and physical specifications for CSMA/CD, Token Passing Bus, and Token Ring networks were specified by IEEE 802.3, 802.4, and 802.5, respectively. The logical link (LLC) sublayer of the OSI data link layer was
specifying IEEE 802.2. These early standards for the lower layers of the ISO/OSI Model allowed LAN developers to build their network interface units on a standard Layer 1 and 2 base.

The Layer 4 Transport service matured at about the same time as the lower layer standards. While the military and the ARPANET community have adopted TCP as their transport protocol (with IP as its internetworking companion for Layer 3 services), the rest of the LAN world seems to be a mixture of TCP, OSI Transport, and XNS (the Xerox Protocols). The acceptance of the OSI protocols by General Motors and NBS in the MAP development and the bandwagon they have gathered as a result of their sponsorship of the Multivendor Demonstration at NCC '84 have focused vendor development at least in manufacturing applications on the OSI protocols.

These developments in the standards community allowed the CDS architecture to evolve from an initial prototype product with standards at only Layers 1 and 2, to the product shown in Figure 2. The current product supports the IEEE standard protocols at Layers 1 and 2, a null Layer 3, the ISO transport at Layer 4 and a Serial Port Application at the higher layers. An awareness of and strong participation in the standards work was beneficial to the design phase of the product. ISO Internet and the Connectionless ISO transport, for example, were anticipated, and the design allowed for the eventual support of these protocols. The same is true for the standards work in Network Management where the company has been active in the work of the IEEE 802.1 committee.

The LAN interface units as communications processors or end systems must offer both high performance and low cost. The cost requirement dictates microprocessor architectures. The high performance requirement dictates tight designs that are performance driven. In some cases, the overhead of the multilayers of the OSI model and its large protocols (transport, for example) add an extra and often controversial burden to the performance (throughput) goals. With a good basic modular design, performance enhancements can be gained from the distribution of selected functions to hardware, firmware, or assembler modules. There is also enough richness in the different classes of transport and LLC services to determine, through performance analysis, an optimal configuration given the underlying network's reliability and its application requirements.

To realize the high speeds offered by the LAN access units, the higher layer communication software must also offer high performance. An efficient real-time executive (exec) is an essential ingredient for providing this performance. The CDS exec is a small multi-tasking exec based on MTOS, the Multi-Tasking Operating System. The exec has a simple task scheduling algorithm and service primitives that allow tasks to disable the scheduling mechanism during critical sections of code.

THE CDS TRANSPORT SERVICE

The CDS Transport Features

The primary design goals for the CDS transport implementation, like those for other CDS modules, were modularity, ease-of-use, maintainable interfaces, testability, and performance. Adhering to the OSI Transport Standard, and providing configuration options and counters for network management were critical functional requirements of the design.

The CDS transport implementation is a conformant superset of the ISO DIS 8073 Transport Specification. It offers both the ISO standard connection-oriented service and a connectionless datagram service. The connection-oriented transport classes 2, 3, and 4 are offered in the CDS transport implementation and are fully compatible with the ISO specification. In addition, this transport implementation offers compatibility with the NBS based but ISO compatible MAP transport specification.

The connection oriented transport services provide transparent and reliable data transfer between two transport users. Transport provides a negotiated quality of service by matching the requirements of the user application to the characteristics of the underlying network and providing enhancements, where required, to the network service. The enhancement levels are associated with the transport classes. Five classes, 0 through 4, provide minimal (Class 0) enhancement to increasing levels of error detection and recovery with increasing Class number.

The CDS connection oriented transport implementation supports connection negotiation and expedited data for transport classes 2 and 3, as required by the standard. Connection negotiation allows the transport users to negotiate characteristics such as the transport class, the maximum transport frame size and the use or non use of check sums for their connection. Expedited data allows short messages, up to 16 bytes, to bypass the normal flow control mechanisms of transport and to be delivered ahead of any other non-expedited messages in the transport queues.

<table>
<thead>
<tr>
<th>ISO LAYER</th>
<th>CDS/HOST MODULES</th>
<th>HARDWARE BOUNDARIES</th>
</tr>
</thead>
<tbody>
<tr>
<td>7 APPLICATION</td>
<td>HOST Application</td>
<td>HOST HARDWARE</td>
</tr>
<tr>
<td>6 PRESENTATION</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5 SESSION</td>
<td>CDS Control Unit</td>
<td></td>
</tr>
<tr>
<td>4 TRANSPORT</td>
<td>ISO Transport</td>
<td>CDS CONTROL UNIT</td>
</tr>
<tr>
<td>3 NETWORK</td>
<td>Null or ISO Connectionless Internet</td>
<td>CDS CONTROL UNIT</td>
</tr>
<tr>
<td>2 DATA LINK</td>
<td>IEEE 802.3 Logical Link</td>
<td>CDS INTERFACE</td>
</tr>
<tr>
<td></td>
<td>&amp; IEEE 802.4 Medium Access Sublayers</td>
<td></td>
</tr>
<tr>
<td>1 PHYSICAL</td>
<td>LAN Physical Layer</td>
<td>MODEM</td>
</tr>
</tbody>
</table>

* Interface is in a shared memory or external interface.

Figure 1—Host/CDS transport interface diagram.
The connectionless transport datagram service provides an efficient, low overhead data transport service. The datagram service is not a reliable one because messages are subject to loss and duplication and are not flow controlled at the transmitter. The underlying 802.4 data link services are reliable, however, so that the service is a reliable one from the user's point of view. It is particularly useful for group broadcast services like those required by a name service. The CDS datagram service is based on the current work in the standards community on the connectionless transport service. The implementation will migrate, where necessary, to full compatibility with the ISO connectionless transport service when it reaches a standard status.

**The CDS Transport Interface**

The interface between the LAN transport layer and the application transport user is called the **Communication-Application Interface**. It is designed to provide a communications path between cooperating applications at one or more locations. The key design feature of the interface is its ability to commonly support both applications that reside on the same board as transport and applications that reside offboard. This gives a shared memory transport interface to support host-based applications. The transport service interface can also be ported to a host and carried via an external interface to the LAN unit.

The Communication-Application interface is a shared-data interface whose primary data structures are communications control blocks (CCBs) and request control blocks (RCBs). These structures are linked together on queues that are connected by a CCB base which, together with an event flag mechanism, provide the interface backbone. A set of service routines for control block initialization queue handling and flow control between transport and the application are the tools required to manage this interface.

The design has a flexible buffer allocation strategy that allows the transport user to allocate its buffers to transport according to the session requirements. At the time the CCB is initialized, the transport user allocates buffers to the CCB's free-queue and indicates how these should be used by transport by issuing credits for its receive queues. There are separate queues associated with each CCB for expedited data items and for normal data and control items, and for transport to return issued requests with their status.

Flow control is based on receiver allocation of buffers. The transport layer provides a credit based service that allows transmission over a virtual circuit only if buffers are available at the receiver. A quota field that is shared between transport and the transport user (in ISO, the session layer) is part of the CCB. It provides the credits to the communication layers for received data blocks and is used by transport for credit computations on flow controlled connections. The quota is decremented by transport each time a block is removed from the CCB's free-data queues for a received data frame. The application initializes this field and increments this quota each time a data block is returned to the free-data queue.

**Transport Performance Considerations**

Extensive performance analysis, down to the module level, was done on the CDS transport implementation. The effort identified unreasonable module activity and demonstrated the task dynamics. Assembler recodings of selected modules and adjustments of relative task priorities gave significant performance enhancements. Optimizing transport's use of the exec services also helped to raise the level of the transport performance.

The question of the appropriate transport class and the set of options within a class for LANs, particularly single-hop LANs with no internetworking, has been addressed from a performance point of view. LANs generally offer a communications media characterized by high speeds and low error rates, so the extensive error detection and recovery features of Class 4 Transport may be excessive for some applications. Transport check sums, for example, are costly in terms of performance, and we have demonstrated that they are not necessary to provide reliable end-to-end transport connections on a single-hop LAN. CDS is currently evaluating the performance characteristics of transport with the various classes and options.

**Layers 5–7 and Network Management in LANs**

The CDS port application, called SPA, offers session and presentation services to a terminal user as seen in Figure 2. It interfaces directly to the transport layer and was designed so that the session services in particular could be replaced by a standard session protocol. The SPA session service supports a variety of session types, including fixed or switched circuits, and point-to-point or multipoint connections. Both asynchronous and transparent synchronous data transfers are supported. SPA provides session negotiation and has a rotary

![Figure 2—CDS "Terminal server" architecture](From the collection of the Computer History Museum (www.computerhistory.org))
function that selects the best port for a given call request. In addition to the session service, SPA offers a name service that allows remote ports to be addresses by configurable names. The name service uses the datagram service of the transport protocol. A station management component supports the network management entities' access to internal port application configuration parameters and counters. As an application, SPA can be considered to be a packet assembler/disassembler (PAD) with features similar to those of an X.3 PAD.

The network management facility of the CDS architecture features both local and remote reading and configuration of the parameters of the network interface units. A CDS network control center (NCC), an IBM XT based system, executes the remote procedures with cooperation from the remote station management entities for each software layer, and maintains the network management database with a relational database system. The network management architecture was designed along the guidelines of the IEEE 802.1 work, and will eventually use the management protocols that are being developed in 802.1. Both local load of software and the 802.1 remote load will be supported. The interface, like the transport interface, will be a shared data queued structure for the management transactions.

Activity in the standards community is now focusing on layers 5–7 and network management. It is likely that standards for these layers will emerge and become accepted within the next few years. The need for application support drove early LAN implementations so that an evolutionary plan is required to have LAN products that incorporate standards for the upper layers as they become accepted. Again, a product architecture based on the OSI Model proved to be a wise choice, as early products with non-standard layer implementations can be replaced by standard protocols—thus ultimately achieving a standard product within the original product architecture.

SUMMARY AND CONCLUSION

By adopting a software architecture based on the ISO/OSI Reference Model, an incremental and modular approach to the development of LAN products can be taken. The payoff is products whose base components meet accepted standards, offer interoperability with other vendors' standard products, and provide an evolutionary path for the development of completely standard products as the standards emerge for the higher layers. The architecture then becomes a frame into which the building blocks can be inserted as they are developed.

CDS has used this approach successfully to develop the software architecture for its product base. The ISO transport protocol implementation is one of the standard building blocks that has been incorporated into the original product that was based on the standard IEEE 802.4 token bus access protocols. The approach also provides an evolutionary path for the incorporation of the standard network session and network management protocols.

REFERENCES