The Control Data loosely coupled network lower level protocols

by WILLIAM C. HOHN
Control Data Corporation
Arden Hills, Minnesota

INTRODUCTION

The Control Data Loosely Coupled Network (LCN) provides a form of two-party cooperative communications, similar to that described by Enslow (1). Put another way, LCN acts as an agent for Multi-Mainframing and distributed processing by providing a means for interconnection of, and information exchange amongst, a collection of mainframes or mainframes and peripherals (all referred to as "hosts" in the remainder of this paper).

LCN appears in a host as that software which provides network services to applications. These services include permanent file transfer, queued file transfer, application to (remote) application chit-chat, and shared rotating mass storage.

An equipment called a Network Access Device (NAD), attached to a host using the channel protocol native to that host, is the hardware entry point to the network. NADs in turn are interconnected by bit-serial trunks. Up to 32 NADs can be attached to a trunk; up to four trunks may be attached to a NAD. The combination of NADs and trunks provide the interconnections between hosts, and thus the hardware path for information exchange amongst hosts.

A set of protocols define and control LCN activities. They are shown, in Figure 1, as they relate to the ISO Open System Interconnection Reference model (2). Levels read down the page from the highest (application) to the lowest (physical). The remainder of this paper is limited to the protocols of the lower four levels.

VIRTUAL CHANNEL

Conventionally two hosts have been coupled in one of three ways,

a) common memory storage
b) channel-to-channel (a variation is shared disk)
c) communication lines

in order of descending performance. In addition, operating systems tend to be cognizant of these variations since data rates and response times vary dramatically with the form of coupling. What differentiates LCN from the traditional interconnect schemes are the characteristics of the interconnect.

A host NAD, which interfaces a host to the remainder of LCN, is designed to the channel specification native to the host—including the maximum data rate of the channel.

NADs in turn are interconnected by shared trunks. The amount of trunk bandwidth available to a given NAD to NAD transfer is dependent on the loading of the trunk, varying from a maximum of 50 megabits (minus overhead) down to some minimum but nonzero value as trunk loading increases. Loading refers to how many NADs are attempting to use the trunk simultaneously.

From an operating systems point of view, a NAD to NAD transfer can occur at maximum channel rate, or at some lesser rate down into the realm of communication lines. This effect is analogous to virtual memory in that as the number of jobs mapped onto real memory exceeds the size of real memory, the execution time for each job increases.

A second parallel to virtual memory is that NAD to NAD transfers replace what is otherwise a system bottleneck with a slow transition. For example, coupling synchronous transfer rates of different value. The analogy in virtual memory is the mapping of a 10 million word program onto a half million word real memory without reprogramming for overlays or special I/O techniques.

In summary, LCN appears as neither a channel extension nor as a communications scheme, but rather as a virtual channel.

PROTOCOL DESIGN CONSIDERATIONS

A design decision was made that the NAD should incorporate the lower protocol levels up to and including the transport level, primarily because of the problems associated with

---

From the collection of the Computer History Museum (www.computerhistory.org)
resource contention. Other factors influenced by and therefore favoring this decision included error recovery, performance of the network, simplicity of design, commonality throughout the network, and integration with existing hardware. Inspection of these factors with greater resolution revealed that the implementation had to:

a) meet design goals at low cost
   • connectivity # units, distance, data rates
   • accessibility controlled access, multi-path
   • performance overhead
   • dependability detect errors, graceful degradation
   • maintainability fault trace

b) Which translate into givens
   • serial trunk low cost
   • buffering data rate matching
   • communication like messages vs ready/resume
   • intelligence to manage it all

c) and avoiding these self-inflicted pains
   • single point of failure centralized network management
   • dead box deadlock also called "united we fall"
   • throttling slow host throttles fast host
   • missing message lost data
   • spoofing security breach
   • global autoload load one—load all
   • the "bully" fixed trunk access priority
   • daisy chain NAD to trunk connection
   • resource deadlock no closure
   • error prone transmission requires higher level protocol problems

d) and recognizing the inherent
   • error retries how many
   • resource allocation single path or multi-path
   • host/trunk (real/virtual channel)
   • the "missing ACK" resynchronization in the face of errors

Although the detailed analysis of these attributes is beyond the scope of this paper, they are shown to indicate considerations made in arriving at a working implementation.

**LCN SITE PROTOCOL**

Strictly speaking, Site Protocol is not one of the layers given in the ISO Open System Interconnection document. But it is presented to show that connectivity and accessibility are not the same thing. The configuration of trunk/NAD interconnects, together with assignment of physical addresses and access codes, constitute the site protocol.

**Connectivity**

An example of LCN hardware is shown in Figure 2, where MF = mainframe and P = peripheral. The NAD can interface with one to four trunks, some examples of which are shown in Figure 1. The NAD connects its attached device to the network, but the interconnect pattern of the NADs/trunks defines the connectivity between devices. Note that attaching a device to the network does not imply connectivity with all other devices attached to the network. For example, MF/B cannot connect with P/C, P/D, or MF/E. Another example is MF/I; which can connect to all other devices except P/C.

**Addressing**

The hardware which interfaces a NAD to a trunk is called a Trunk Control Unit (TCU).

Each TCU is identified by an 8-bit physical address. The destination field of a message must match the physical address in order for the message to be accepted by the NAD.

The site protocol requires that all TCU s on a given NAD must have the same physical address.

**Set identifier**

Each NAD/trunk interface includes a 16-bit set identifier (the access code). The access code field of a message must match the switch selectable access code of the TCU in order for the message to be accepted by the NAD. The access code allows sets of NADs to share the same trunk yet be independent. All NADs of a set may access all other NADs within the set, but no others. Messages directed to a NAD outside of the set are not accepted by that NAD. Note that a NAD may be a member of more than one set. Access code transmission and matching are hardware functions.

**Accessibility**

These three characteristics (network connection, physical address, and set identifier or access code) constitute the accessibility of a NAD (and hence its attached host) to all other network NADs.

**TRANSPORT CONTROL PROTOCOL**

The Transport Control Protocol provides the method by which mutually accessible hosts exchange information. This protocol is defined independently of, yet with consideration for, the broad range of hosts considered likely LCN candidates.
Data path

The primary means for information exchange is the data path. A data path is a logical, bidirectional "channel" terminating at each end in a host. Data path creation and deletion is performed by the NAD as directed by its host. A path normally is created between two different hosts, but the protocol allows a host to create a path to itself. Furthermore, some NADs are designed to have multiple attached hosts. When generating a path between hosts on the same NAD, or when a host creates a path to itself, the path is contained solely within the associated NAD since that NAD is attached to both path-end hosts.

More than one data path may exist between a pair of NADs. A NAD can support up to 128 paths. However, since each path requires a small dedicated area in memory for path control (not including data buffers), the maximum number of paths supported by a NAD varies with its hardware configuration.

Information is passed on a path in the form of data, marks, or as a code.

Data path-connect

Path creation is initiated by a host. A successful connect requires acceptance by the local and remote NADs (path control resource allocation), and by the destination host. That means the three intelligent entities—NADs and destination host—all have the opportunity to deny the request. Put another way, the three must cooperate in order to complete the connect. Once connected, a path exists until explicitly disconnected by either of the hosts, or until an unrecoverable error, such as a broken trunk, occurs.

At connect time the host presents the routing parameters to its NAD, or receives them from its NAD, depending on path-end. Both hosts also are given a path ID by their respective NADs. Thence forward, the hosts refer to the path by ID, and the necessary routing is automatically performed by the NAD.

Data path-data exchange

By definition, data transfers are bi-directional on a data path. Host transfers are "blocked" by the NAD before transmission on the trunk in order to (a) decouple the host channel rate from the trunk rate and thereby allow unused trunk time to be used by other NADs, (b) to provide a mechanism whereby the data buffer area of NAD memory can be allocated dynamically across several paths, (c) to provide concurrent bi-directional transfer, and (d) to segment a long transfer for lower probability of induced errors.

The amount of data transferred is unlimited, but the effective rate of transfer is limited by buffer availability (real and path threshold) in the associated NADs, as well as by trunk loading (the "virtual channel" characteristic). A special data transfer mode is provided in which the trunk is not released between data blocks. In this mode only the two NADs involved can use the trunk, all other NADs being locked out. In this mode trunk loading is removed as a factor, and the network appears as a dedicated point-to-point connection. Transfer rates are then limited by the host channel and trunk rates and overhead.

Data path-mark

A mark is a special form of data, analogous to an end of record mark. A mark may be sent at any time. The data and interspersed marks are delivered in the same order as sent.

Data path-code

In addition to data and marks, a very short (32 bit) message may also be transmitted down a path. This message (CODE) is different from data and marks in two ways. First, the data path CODE buffers at each end are permanently allocated and, hence, always available (unlike data buffers which are dynamically allocated). Second, CODE messages are not queued at the receiving end; rather each CODE message "overwrites" the previously received one.
**Data path disconnect**

A path may be disconnected by either path end host. The NAD performing the disconnect first transmits all of its outstanding output data, marks and code; then sends a disconnect message to the other NAD. Residual input is discarded in the NAD initiating the disconnect.

The disconnect is queued at the receiving NAD and is presented to the attached host after all data (and marks) in front of it have been presented.

**Control message**

The control message (DATAGRAM) is an alternate means for information exchange. Control Messages are self-contained, meaning they have fixed length, they contain routing information, and they are not associated with data paths. The host supplies the Control Message to the NAD, and the NAD simply sends it to the destination NAD. A Control Message, which for any reason cannot be delivered to the destination NAD, is returned to the host.

The control message format is shown in Figure 3.

**SUMMARY**

Transport Control Protocol uses data paths and Control Messages for information exchange between mutually accessible hosts. Since data paths exist as logical "channels," a host and its attached NAD may have many paths assigned and serviceable concurrently (but not simultaneously). At the same time the Control Message, independent of data paths, is available for flow control, status, functional requests of a higher level, or just general chit-chat.

**NETWORK CONTROL PROTOCOL**

This protocol level defines information flow control between NADs.

In order for a host with a dedicated NAD to exchange information with any other host, the transfer obviously requires traversing a trunk from NAD to NAD. Hosts sharing a NAD, however, communicate with each other through their shared NAD, but with other hosts across a trunk (and another NAD).

A primary task for the Network Control Protocol is recognizing and processing these routing variations.

**Command/response message modes**

Three modes are defined for communications between NADs. Each mode is tailored to a specific work function, to effectively use the trunk and NAD resources.

In the following three sections, the graphic conventions listed below are used.

The Command Message and Response Message, which always occur in pairs, are illustrated as a pair of boxes connected by a line.

```
: COMMAND : RESPONSE :
: MESSAGE : MESSAGE :
```

The left hand box is always a command message, and the right hand box is always a response message. The interconnecting line represents the interval during which the receiving NAD processor examines the command message and determines the appropriate response message to return. The trunk remains captured until the processor signals the trunk interface to transmit the response message.

**Mode 1—control**

```
: CONTROL : RESPONSE :
```

Mode 1 consists of one command/response message pair. CONTROL is directed to the host, and has meaning defined by a higher level protocol, or is a flow control message directed to and processed by the NAD processor.

**Mode 1 command messages**

* Control Message
* Connect
  * Pathcode
  * Path Mark
  * Disconnect
  * Double Purge
  * Connect Accept
  * Connect Reject
  * Status Change

** There are the only messages (of all three modes) which are generated by the host.
Mode 1 response messages

ACK
WAITNAK
Sequence Error
Illegal Command Message
Illegal Path

Mode 2—data transfer

Mode 2 is a two command/response message set used for transmission of data. The first command, Path Request, identifies the path and the buffer size. The receiving NAD returns an ACK if the data transfer is permissible. The sending NAD then immediately transmits the data, a closing ACK is returned, and the trunk is released.

If the receiving NAD cannot accept the data, it will return a NAK, following which the trunk is released.

Mode 2 command messages

Path Request
Path Data

Mode 2 response messages

ACK
Queue Full NAK
WAITNAK
Illegal Command Message
Illegal Path
Sequence Error

Mode 3—captured trunk (streaming mode) data transfer

Mode 3 is a special form of data transfer in which trunk multiplexing is temporarily halted by capturing the trunk (streaming) for the duration of a multi datablock transfer. Consequently, total trunk bandwidth is allocated to the path capturing the trunk. Two useful effects result. First, the data path transfer rate is maximized since trunk loading has been eliminated and protocol overhead minimized. Second, all other NADs (and their hosts) on the trunk have had their intercommunication momentarily suspended, which implies an interlock capability.

Mode 3 command messages

Enable Stream Path Request
Disable Stream
Path Data
Path Mark
Path Code
Wait

Mode 3 response messages

ACK
BLOCK SEQUENCE ERROR
Illegal command message
Illegal Path
Disconnect
Ready
Nak

Message transmission retry

As explained in the Link Control Protocol section, all transmissions consist of a command and response message pair. A normal transmission consists of a TCU receiving a command message addressed to it, and returning a response to the TCU originating the command.

Transmission abnormality

A transmission abnormality occurs when no response is forthcoming to the command. Listed below are the main reasons for no response occurring:

1. Addressed a nonexistent NAD (actually TCU)
2. Command message garbled on the trunk causing
   a. Destination Field to address a nonexistent NAD
   b. Check sum error at receiver (no answer if check sum error)
3. Command message received correctly but the response message was garbled giving a similar effect as (2).

When a transmission abnormality occurs, the command message is retransmitted (with the same sequence number), up to 256 times, until a response message is received. If no response message is received after 256 retries, a fatal error has occurred. The disposition of the unsuccessfully transmitted command message depends on its type.

During retry the controlware will not attempt to send any command messages other than the one enduring the transmission abnormality. The controlware, however, will accept incoming command messages.

Destination busy

The status of the responding NAD (TCU) is included in a response message. One of these status bits is the "Memory
busyness” bit. The “Memory busy” status signals that, although the command message was received by the destination TCU correctly, it could not be passed on to NAD memory. Hence, the response returned is likewise not from memory but generated by the TCU. Since the NAD processor at the destination did not receive the command, it must be retransmitted.

Provided the remainder of the response status is correct, the command message is queued for retry, and will be retransmitted until accepted by the destination NAD memory (or until a fatal error occurs).

The retry queue is serviced on a fixed interval basis. Other command messages may be sent during this interval.

Destination fatal error

Other NAD status bits in a response message include NAD processor not running, sequence errors, and other indicators of fatal errors at the destination NAD.

If the response status indicates a fatal error, the command message is not retransmitted. Marked “fatal error,” the message is disposed of according to type.

Fatal message errors

Fatal message errors are those which are caused by hardware errors during the transmission of a trunk command/response. These errors can be caused by:

- trunk interface failures at either NAD
- trunk/data set failures
- NAD failures.

LINK CONTROL PROTOCOL

The LCN Link Control Protocol defines a format and sequence of bits impressed upon the trunk to facilitate the transmission of information. The vehicle for all command and response information on the trunk is called a message. Each transmission on the trunk consists of only one message frame. In all cases, communication between two elements \( X \) and \( Y \) consists of a pair of message transmissions: a command message transmitted from \( X \) to \( Y \) and a response message transmitted from \( Y \) to \( X \).

Command message frame structure

A valid command message is a minimum of ten 8-bit bytes in length following the frame synchronization sequence and must conform to the following structures: \( P, F, T, \) FUN, \( A_1, A_2, R_P, S, L_1, L_2, F_C_1, F_C_2, I, F_C_3, F_C_4 \) where,

\[
\begin{align*}
P &= \text{preamble of all ones preceding sync frame} \\
F &= \text{message frame synchronizing byte} \\
T &= \text{destination address byte} \\
\text{FUN} &= \text{function byte} \\
A_1, A_2 &= \text{access code bytes} \\
R_P &= \text{resync parameter byte} \\
S &= \text{source address byte} \\
L_1, L_2 &= \text{length field bytes} \\
F_C_1, F_C_2 &= \text{header frame check sequence bytes} \\
I &= \text{information field, variable length} \\
F_C_3, F_C_4 &= \text{information frame check sequence bytes}
\end{align*}
\]

Frames containing only link control sequences form a special case where no \( I \) field is present.

Response message frame structure

A valid response message is a minimum of ten 8-bit bytes in length following the frame synchronization sequence and must conform to the following structure: \( P, F, T, \) FUN, \( P_1, P_2, P_3, S, L_1, L_2, F_C_1, F_C_2, I, F_C_3, F_C_4 \) where,

\[
\begin{align*}
P_1 &= \text{not used} \\
P_2 &= \text{TCU/TCI status byte} \\
P_3 &= \text{not used}
\end{align*}
\]

and all other elements are identical to the command message elements.

PHYSICAL LEVEL PROTOCOL

The transmission scheme employs carrier modulation of self clocked data, with the NADs attaching to a coaxial trunk via a T-tap. The measured error rate of the combination (including data sets) is 10\(^{-12}\) or better within the configuration limitations.

Access to the trunk is governed by Trunk Reservation and Contention Elimination (TRACE) priority hardware. TRACE is a rotating priority scheme in which every NAD is given access to the trunk in turn.

REFERENCES