DECnet Digital Network Architecture (Phase IV) Token Ring Data Link and Node Product Functional Specification

Order Number: EK-DNAP4-TR-001

Version	V1.0.0
Date	July 1992
Revision/Update Information:	This is a new document

This document specifies the functional requirements for products supporting Digital Network Architecture (DNA) family protocols using the IEEE 802.5 / ISO 8802-5 Token Ring LAN media. This document provides the necessary information to develop implementations that will interoperate with Digital's DECnet products.

digital™

Digital Equipment Corporation
Maynard, Massachusetts

EK-DNAP4-TR-001

First Printing, July 1992

This document was prepared using VAX DOCUMENT, Version 2.0.

Preface

This document is one of a series of specifications which describe Phase IV of the Digital Network Architecture. Other DNA Phase IV specifications may be obtained by ordering one or more of the following documents from your local sales office.

- DECnet Digital Network Architecture (Phase IV) General Description, order number: AA-N149A-TC
- DECnet DNA Digital Data Communications Message Protocol (DDCMP), order number: AA-K175A-TK
- DECnet DNA Phase IV Ethernet Data Link Functional Specification, order number: AA-Y298A-TK
- DECnet DNA Phase IV Ethernet Node Product Architecture, order number: AA-K759B-TK
- DECnet DNA Phase IV Maintenance Operation Protocol, order number: AA-X436A-TK
- DECnet DNA Phase IV Routing Layer Functional Specification, order number: AA-X435A-TK
- DECnet DNA Phase IV Network Services Protocol Functional Specification, order number: AA-X439A-TK
- DECnet DNA Session Control Functional Specification, order number: AA-K182A-TK
- DECnet DNA Phase IV Network Management Functional Specification, order number: AA-X437A-TK
- DECnet DNA Data Access Protocol (DAP) Functional Specification, order number: AA-K177A-TK

Information about DNA Phase V specifications can be obtained from the following document and by contacting with your local sales office.

- Digital Network Architecture (Phase V) General Description, order number: EK-DNAPV-GD

PAGE IV IS A BLANK PAGE

Contents

Contents

Chapter 1

Introduction

The Digital Network Architecture (DNA) is a model upon which DECnet implementations are built. DNA is a layered network model which includes support for many communications media, such as Serial Point-to-Point, CSMA/CD Ethernet, Public Switched Networks, and Token Bus.

This document specifies the key technical material necessary to develop a DNA compliant DECnet Phase IV Node Product on an IEEE 802.5 / ISO 8802-5 Token Ring LAN. This document assumes a working familiarity with Token Ring and the DNA Architectures and does not attempt to duplicate the information defined in other documents.

1.1 References:

- DNA Phase IV General Description. (AA-N149A-TC)
- DNA Ethernet Node Product Arch. V2.0.3 (AA-K759B-TK)
- DNA Ethernet Data Link Func. Spec. (AA-Y298A-TK)
- DNA Phase IV CSMA/CD Data Link Func. Spec. V1.01
- DNA Phase V CSMA/CD Data Link Func. Spec. V2.0 (EK-DNA13-FS-001)
- DNA Phase IV Routing Func. Spec. V2.1 (AA-X435A-TK)
- DNA Maintenance Operation Protocol V3.0 (AA-X436A-TK)
- DNA Maintenance Operation Protocol V4.0 (EK-DNA11-FS-001)
- DNA Phase IV Network Management V4.0 (AA-X437A-TK)

- IEEE 802-1990 Overview
  This includes the requirements for canonical bit representation of MAC station addresses.
- IEEE 802.1D-1989 MAC Bridging
- IEEE 802.2-1989 Data Link Control
- IEEE 802.5-1989 Token Ring Access Method
- IEEE 802.5M-1992 Source Routing Supplement to 802.1D
- IEEE 802.2 Route Determination Entity for LLC

- — RFC 1042, "Standard for the Transmission of IP Datagrams over 802 Networks", Feb 1988, IETF.
- — RFC 1103, "Standard for the Transmission of IP Datagrams over FDDI Networks", June 1989. IETF.
- — IBM Token-Ring Architecture (SC30-3374-02) Third Edition, Sep 1989
  The definitive IBM document that specifies their Token Ring functionality.
- — IBM Local Area Network Technical Reference (SC30-3383-03) Third Edition, Nov 1990
  This document specifies IBM's PC driver interfaces, NETBIOS session interface, NETBIOS transport protocol, and PC adapter commands and registers.
- — Texas Instruments TMS380 User's Guide (SPWU005) Rev A, June 1990
  This document describes the functional interfaces to the TMS380 Token Ring Data Link chip set, and has a good introduction to token ring concepts.
- — Microsoft/3Com LAN Manager, Network Driver Interface Specification (NDIS) V2.0.1, 10/8/90
  This document specifies the industry interface to network device drivers for MS–DOS and OS/2 operating systems.
- — The Ethernet, A Local Area Network, Data Link and Physical Layer Specifications V2.0, Nov 1982, (AA-K759B-TK)
  Besides for the obvious, the 90-00 Loopback protocol is documented here.

1.2 Standards Support

This specification attempts to conform to the relevant published standards, but in the cases of conflict, it will follow the attributes of the most established and compatible industry standards or implementations.

1.3 Goals:

- Specify DECnet Phase IV capabilities for End systems on the Ring
- Specify network connectivity via DECnet Routers
- Specify support for key network application products
- Minimal modification to existing components

1.4 Non-Goals:

- Specification of a general purpose (all protocols) interface for all operating environments
- Specification of an interface to 802.2 LLC Type 2 service
- Excluding the use of other Token Ring interfaces on same adapter (eg: LLC Type 2 or Source Routing)

1.5 Model

Considering the present installed base of Token Ring adapters, it has been decided to support Token Ring under Phase IV with as little architectural change as possible to existing Ethernet-based products.

For the purposes of least change to current design, it is useful to model the Token Ring as an Ethernet, by hiding the differences in the Data Link layer interface. This process will be referred to as **Ethernet emulation**. Emulation can occur at several interfaces, including: packet sending and receiving, multicast reception registration, counter reporting, and network management. The extent of the emulation versus the development of new functions is left to the resources of the system product as long as the minimum requirements are meant.

As DECnet support for Token Ring develops in Phase V, it is expected that full 802.5 frame access will be required and use of the Ethernet emulation will be limited to Phase IV and Ethernet (non-802.3) protocol applications.

Figure 1–1: Data Link Access Model

```
+-----------------+ +-----+
| DECnet Phase IV | |MOP, |
|   NSF           | |LAT, |
|   Routing       | |etc. |  User Interfaces
+----------+------+ ++----+  ----------------
           |         |     /   |  |   |
           |         |    /    |  |   |
           |         |   /     |  |   |
+----------+---------V--V-+    |  |   |
| Ethernet Emulation      |    |  |   |
+------------+------------+    /  |   \
| Source Routing (option) |   /   |    \
+---- -------+------------+  /    |     \
           |                /     |      \
            \              /      |       \
    +--------+------------V--+    |     +--V----------------+
    :  SNAF SAP              !    |     : 802.2 LLC Type 2  |
    +-----------+------------+    |     +-----+-------------+
                 \                |           |
            +-----+---------------V----+      |
            |   802.2  LLC Type 1      !      |
            +--------------------------+------V-----+
            |   802.5  MAC                          |
            +---------------------------------------+
            |   802.5  PHY                          |
            +---------------------------------------+
```

1.6 Overview

In the following chapters, this specification covers four areas of information required for building a compliant DECnet node product:

- Data Link Layer Operation and Interface
- Node Product Requirements
- Routing Layer Media Specific Operation
- Network Management Specifications

An additional feature has been added to the Phase IV Network Routing layer for coexistence support in Token Ring environments. This feature, called **Phase IV-Prime** is described in Chapter 6 and includes changes to DECnet protocols and operation to provide this new capability.

Chapter 2

Data Link Layer Operation

2.1 Overview

DNA operation on Token Rings will be essentially the same as on existing Ethernet/802.3 LANs, with the addition of translating Ethernet frames to an standardized 802.2 format, and the necessity of mapping universal multicast addresses into locally administered functional addresses.

This chapter describes the details of the packet format and translations.

2.2 Packet format

DECnet packets on the Token Ring will use 802.2 format headers. The IEEE 802-1990 Standard SNAP SAP must be used (AA hex). The SSAP will always be the same as the DSAP. The SNAP Protocol Identifier (PID) used shall conform to IEEE 802-1990 and the Internet Task Force (IETF) RFC #1042 and RFC #1103 specifications for mapping Ethernet protocols onto 802.5 and FDDI respectively. Therefore the PID shall consist of three bytes of zero for the Organizationally Unique Identifier (OUI), followed by the two byte Ethernet protocol type.

DECnet will only use 802.2 LLC Type 1, Unnumbered Information (UI) packet types for data transfer. Transmission of TEST and XID frames is not required, but is a recommended option. Received TEST and XID frames must be handled as required by 802.2 specifications (including requests to the Null SAP).

For Ethernet emulation, no information about the SAP, Frame type, or Protocol ID need be passed up on reception of a packet. They are assumed from interface context. No information on the transmit is required either. The data link driver must be able to demultiplex emulated Ethernet users using the SNAP PID field.

The data link driver may (and probably should) recognize other forms of 802.2 frames. However, the exact system specific implementation details are outside the scope of this specification. The operation of the DECnet data link or Ethernet emulation must not interfere with other system data link users of the Token Ring LAN. It must provide for multiple users based on protocol type determined from the SAP or SNAP SAP protocol specifier.

Figure 2–1: Data Link Frame Formats

```
Ethernet Frames:
    Dest   Src  PType   Data
     6      6     2     46-1500

802.3 Frames:
    Dest   Src  Length DSAP SSAP Control Data
     6      6     2      1    1     1-2   44-1496

802.3 SNAP Frames:
    Dest   Src  Length DSAP SSAP Control PID Data
     6      6     2      1    1      1     5  0-1490

802.5 Frames:
    Dest   Src  DSAP SSAP Control   Data
     6      6    1    1     1-2    0-4472/17800

802.5 Frames with Source Routing
    Dest   Src    RI    DSAP SSAP Control    Data
     6      6   2-30     1    1      1-2    0-4472/17800

802.5 SNAP Frames with Source Routing:
    Dest   Src    RI    DSAP SSAP Control PID    Data
     6      6   2-30     1    1       1     5   0-4472/17800

    Where:
        - maximum length of 802.5 frames varies with speed
        - 802 SNAP SAP (AA)   (SSAP==DSAP)
        - Control for SNAP is UI (03)
        - IETF/RFC 1042 and 1103
                PID = OUI(3) + Protocol type(2)
                PID = 00-00-00-60-03  for DECnet
                               80-40  for NETBIOS naming
```

NOTE!

The 00-00-00 OUI PID must not be transmitted on a real Ethernet, as it will cause inter-media bridges to make irreversible translations. Protocols using 802.2 frames on 802.3/Ethernet must use a non-zero OUI in the PID. See Appendix F and Appendix H for further information.

For the purposes of this specification, the Data field is as would be seen from the Ethernet V2.0 point of view. Therefore, any fields included in the DNA Ethernet Data Link architecture (eg: the length field for padded protocols) must be included in the data of the token ring message. However, any optional minimum Ethernet/802.3 length padding data should not be added to 802.5 packets to conserve network bandwidth. This allows direct conversion of packets back into an Ethernet environment by a MAC layer bridge, without DECnet protocol specific processing. It is expected that cross-media bridges will be able to pad under-minimum size packets themselves (based on the length of the message on the wire not the value of the field).

While the 4Mbs Token Ring allows frames up to 4,500 bytes (18,000 on 16Mbs) in length, the Ethernet emulation section of the driver need only transmit or receive packets on the token ring whose data length would also fit on then Ethernet, if routed. (see section on user data size)

Most Token Ring drivers receive their own transmitted packets, particularly for multicast and broadcast. This is not typical behavior for DECnet data link drivers and is not expected at the user interface. The data link driver must disable reception or ignore packets generated by itself.

2.3 MAC Addressing

2.3.1 Address representation

In the following tables the IBM bit-reversed (big-endian) address nomenclature is often used, as it's the form used in Token Ring documentation. IEEE 802 requires that bit order for MAC addresses be consistent for all LANs on the wire and also when represented external to the data link layer. Because the transmission bit order within bytes is reversed for 802.5 from 802.3, 802.5 MAC layer addresses end up bitwise reversed within the same byte order. Within this document, address values separated with "dashes" will be in IEEE 802 **Canonical** order, and those separated by "colons" will be in 802.5 bit-reversed order.

All addresses passed from the data link to the portal user must use 802 Canonical bit order. Eg., 802.5 reversed bit order MAC addresses must not be propagated to data link users. The driver must bit reverse LAN header addresses when transmitting or receiving. No bit reversal should be performed for data. (IEEE 802-1990, Section 5.4)

In general, all interfaces that display the MAC address value must do it in 802 canonical form.

2.3.2 Multicast Address Mapping to Functional Addresses

The current installed base of Token Ring hardware implementations poses a problem, in that they do not support full multicast reception as do normal Ethernet adapters (see Appendix D). The only general purpose multicast reception capability available is that of Functional Addresses. Functional Addresses are an 802.5 specific address form that is in the Locally Administered space.

To support PATHWORKS and most DECnet applications the following multicast addresses need to be supported. These universal assigned addresses will be mapped into 802.5 specific Functional Address bits as shown in the accompanying Table 2–1. (See also Table A–4)

Because of the possibility of conflict in the functional address local administration space these assignments should be considered defaults, and must be implemented so that they can be changed by a field specialist or technical customer on site. Modification of mapping can be either by network management command, configuration procedure, or initialization file.

The status of the mapping must be readable as well as settable. It is incumbent on the local management to coordinate changes from the defaults as this will affect interoperability with other systems.

Table 2–1: Multicast Address to Functional Address Mapping

Description	Ethernet Ptype	Universal Multicast (canonical order)	Functional Addr. (reversed order)
MOP Dump/Load	60-01	AB-00-00-01-00-00	C0:00:40:00:00:00
MOP Console	60-02	AB-00-00-02-00-00	C0:00:20:00:00:00
PhIV All Routers	60-03	AB-00-00-03-00-00	C0:00:10:00:00:00[1]
PhIV All Endnodes	60-03	AB-00-00-04-00-00	C0:00:08:00:00:00[1]
PhIV All L2 Routers	60-03	09-00-2B-02-00-00	C0:00:10:00:00:00[1]
DEC NETBIOS	80-40	09-00-2B-00-00-07	C0:00:04:00:00:00
LAT Advertisement	60-04	09-00-2B-00-00-0F	C0:00:02:00:00:00
LAT Service Solicit	60-04	09-00-2B-02-01-04	C0:00:01:00:00:00
LAT Xwin service Solicit	60-04	09-00-2B-02-01-07	C0:00:00:40:00:00
PhIV-Prime All Routers	60-03	09-00-2B-02-01-0A	C0:00:10:00:00:00[1]
PhIV-Prime Unknown Destination	60-03	09-00-2B-02-01-0B	C0:00:08:00:00:00
LAST (Group 0)	80-41	09-00-2B-04-00-00[2]	C0:00:00:20:00:00
LAST (Other Groups)	80-41	09-00-2B-04-**-**[2]	user assigned
Loopback	90-00	CF-00-00-00-00-00	C0:00:00:10:00:00
SCA	60-07	AB-00-04-01-**-**[3]	user assigned

[1]Routing and Endnode multicasts must be on distinct different values

[2]LAST uses the lower 16 bits as a group code

[3]VAX Clusters derives the lower 16 bits from the cluster id

The use of any other Ethernet multicast addresses is currently undefined. It is acceptable to map them to the Broadcast address, a user settable value, or return an error.

When using Ethernet emulation, multicast addresses will be recognized at the emulation interface and the appropriate Functional Addresses used instead. Thus the caller need not be aware of the multicast differences of Token Ring implementations, and have to change its code.

Other data link users, not using Ethernet emulation, must not have their multicast addresses mapped, or have direct control of the mapping feature.

CAUTION!

Note that all of Frame Format, Address bit reversal and Functional address mapping will affect any Token Ring to Ethernet bridges. For DECnet (or OSI!) to interoperate over such a bridge, the bridge will have to translate the packet into the appropriate format. It is a goal to constrain this problem to the header of the message frame. It also behooves the protocol designer to not propagate media specific problems into the data portion of messages that would transit such bridges. One such problem is the bit order problem discussed in Appendix F.

2.3.3 Multicast and Functional Address packet reception

Given the scarce address space of group addresses implemented in token ring adapters, multiple different protocols are "sharing" functional address values on the wire and in systems. All data link implementations must be able to correctly demultiplex received frames based on DSAP, and SNAP PID and discard frames for a group address which that protocol user on that system has not enabled.

This must not be counted in the Line Counter for Unrecognized Frame Destination (NICE parameter #1063).

Likewise protocols must not depend on the value of the actual multicast address that a packet was received on. This may arise if the protocol has multiple universally administered group addresses that have been mapped to the same functional address. If a packet arrives on a multiply mapped functional address, it will be delivered to the enabled protocol user with the first multicast address that appears in its local mapping table.

2.3.4 User Assigned Functional Address Value

When choosing Functional Addresses values to be assigned, one should be aware of the amount of use that the particular values receive on their particular network. The purpose of multicast addressing is to allow broadcast capabilities to protocols, without burdening all stations on the LAN with reception of packets for protocols not active on that station.

The Network Administrator should choose values not in heavy use by known active protocols on their network. See Table A–1 for information on some known protocol address usage. Poor choices will affect the performance of the station and possibly all stations on the LAN due to reception overhead.

2.3.5 Individual Station Address

The data link driver must be able to set the adapter station address to the DECnet Phase IV required value if ordinary Phase IV is used. The value for the DECnet Phase IV high order is AA-00-04-00 in 802 Canonical order or 55:00:20:00 in 802.5 reversed order. The lower two bytes are the DECnet node address in low byte, high byte order. For example; DECnet node 55.4 would run as station address 55:00:20:00:20:3B on the wire. Also note that this address appears to be in the Local Administered space and will work even on the early revision TI TMS38020 Protocol Handler. (this issue with the TMS380 was corrected in later revisions. See Appendix D)

Chapter 6 specifies a new DECnet capability, Phase IV-Prime, in which the routing layer is run using the manufacturer's unique adapter address or a locally assigned station address.

2.4 User Data Size

An 802.5 4Mbps frame may contain up to 4,442 bytes of data field after allowing for maximal frame overhead and Source Routing data. Likewise, a 16Mbps frame may be up to 17,800 bytes. Minimally, the driver must allow Ethernet size user data (1500 bytes) with the 802 and SNAP frame overhead absorbed in the internal buffering. Use of larger frame sizes is optional for Ethernet emulation. Since DECnet Phase IV routing has no fragmentation method, use of data frames larger than 1500 could not be routed via Ethernets, and would be constrained to connections on the ring.

2.5 Handling of Addressed and Copied Frame Indicators

The Addressed or Copied Frame indicators are for the use of the MAC layer only and are not a reliable technique for determining transmission reception. Particularly in the presence of bridges, these bits become meaningless as they cannot indicate the state of the packet with respect to the off-ring end system. If a bridge sets the bits, then the transmitter does not know if the target system actually got the frame. Likewise if the bridge does not set the bits, the transmitter has no information, and may conclude that target system is not on the LAN when it really is.

Source Routing Bridges that forward, or the destination station will set the Address Recognized Indicator (ARI or A bit) and the Frame Copied Indicator (FCI or C bit) to indicate when it forwards a frame. The A&C behavior of a Transparent (Spanning Tree) Bridge has been specified as optional.

AC	NDIS error	Description
00	NO_SUCH_DESTINATION	Frame not recognized on ring
01	-	Not allowed
10	TRANSMIT_ERROR	Recognized, but not received
11	Success	Recognized, and received

The indication of either ARI or FCI clear, must not be considered a transmit error for an Ethernet emulation user.

An AC=00 indication could be used during Source Routing discovery to indicate that a destination is not on the ring, though a transparent bridge may have forwarded it without setting the ARI or FCI.

In general Digital recommends that the A&C bits not be used for data link application purposes and if used, their values are considered only as hints, not positive indications.

2.6 Source Routing

Source Routing is technically not a standard nor part of 802.5 at the time of this writing. However, it is in use in many IBM token ring installations Efforts are underway in the Source Routing / Transparent (SRT) Bridge working group (802.5M) to define how source routing is to operate and coexist with 802.1D Spanning Tree Bridges.

Source Routing provides an 802.5 unique manner for routing packets within an extended LAN of token rings, that use source routing bridges. As such, it lives below the DECnet routing layer routing, which concerns itself the wide area connectivity across multiple LANs and WANs.

Source Routing support is required for all DECnet products, but can be optionally disabled. All nodes are required, at minimum for Ethernet emulation users, at all times, to receive packets with source routing information, and strip the Routing Information field. Routing Information fields need not be supplied upon transmit. If source routing support is in use, it will be provided in a manner transparent to Ethernet emulation users by default.

Source Routing operation is described in detail in Chapter 5.

Chapter 3

Data Link Interface

The system will provide a data link interface equivalent to the DNA CSMA/CD Architecture Specification. This interface includes the ability to:

- Open and Close data portals based on Ethernet protocol type
- Set the local station address
- Enable and Disable Multicast address reception
- Send and Receive datagrams
- Read network management information

Support for LLC Class 2 operation is outside the scope of this specification. Ethernet emulation must not interfere with LLC Type 2 users. Exact specification of the local system interface is system specific. For Phase IV support it simplifies the effort to emulate a local Ethernet interface as best as possible.

Even though 802.2 data link interfaces may exist they are not used by Phase IV. All 802 differences can be hidden from the data link users.

The DNA Ethernet Data Link Padded protocol option, which is used to include or strip a 16-bit Length field from the first two bytes of the message must be supported, and the length field included in the 802.5 data. However, the driver is not required to add padding bytes to the end of the frame. The byte order of the length field must be the same as on Ethernet (eg: low order, high order).

Note that an Ethernet interface could receive frames only with lengths of 46 to 1500 data bytes. Token Ring can receive frames of 0 to 17,800 data bytes. Use of packet sizes larger than 1500 bytes is allowed, but can lead to problems with bridged mixed LAN media. In such an environment systems must be configured for the maximal common buffer size.

NICE Network Management counters to be returned are listed in section 7.4.3.

Interfaces for access to 802.5 MAC level network management are not specified in this document. These interfaces would be necessary for the implementation of 802.5 specific MAC level monitoring or control.

Chapter 4

MOP Functions

A DNA compliant node product system must have, at least, minimal MOP V3.0 support mapped to an 802.2 environment, as described within this specification and in the MOP V4.0 specification under the appendix on compatibility with previous versions. In general, Phase IV systems would be MOP V3.0 implementations. This support may be implemented at levels above the basic data link driver (system dependent).

The required items are:

- Periodic SYSID multicast (using the FA mapping in Table 2–1)
- Console Server functions:
  - Respond to SYSID request
- Loopback support (using Ethernet Loopback protocol 90-00)

Other higher level functions may be implemented, and the design must include an interface to allow management utilities to send and receive arbitrary client MOP packets using the MOP protocol type and functional address.

Note that a token ring MOP data link counters format message is not defined in V3.0 and therefore the function is not provided in this version of the specification. An implementation should not claim this feature in the system function bit vector.

See Appendix B for value assignments.

Chapter 5

Source Routing

5.1 Scope

The purpose of this specification is to define how Source Routing will be used by all Phase IV DECnet protocols operating on an 802.5 Token Ring LAN extended by Source Routing bridges.

This chapter describes the external system requirements of this operation, and two compliant and recommended implementations. The purpose of offering two implementations, is to allow the developers to make the appropriate tradeoffs with respect to performance and suitability.

Source Routing is a sub-protocol of 802.5. It provides a field within the LAN frame directly below the data link level. The meanings of the various components of that field, and the behavior of a bridge is fairly well specified (though not completely standardized). The intent of this section is to describe how DECnet protocols will use the contents of the source routing information.

However, implementation and protocol issues as to when these fields are used involve tradeoffs in performance and impact to higher level protocols. As long as a system follows the rules within this document, all implementations will interoperate.

CAUTION:

> Given that, at the time of this writing, the specification of Source Routing is still changing, developers are cautioned to test their systems against representative products in the field to guarantee interoperability.

5.1.1 Goals

The primary goals of this specification are as follows:

1. Support DECnet protocols in a interconnected 802.5 Token Ring extended LAN using source routing bridges.
2. Provide an architecture in which source routing support is optional, yet interoperable, on the same ring, with stations that do not implement source routing.

3. Source routing support may be transparent to the layers above the data link layer as the implementation needs. It should be noted that performance impact of source routing depends on the extent that connection-oriented upper layers can provide context for each packet sent or received.
4. It is a goal to minimize the overhead of source routing to the protocols using it, and the complexity of the implementation.
5. Source routing support must attempt to minimize use of bandwidth consuming messages, whenever possible.

5.1.2 Requirements

DECnet Phase IV products are required to fully support source routing, but allow the support to be turned off. In either mode, all nodes are required to be able to receive source routed packets and ignore the source route fields.

The level of source routing support (described later in the chapter) is a system implementation decision. Given product requirements for protocol support, the transparent implementation provides the best support for the most protocols without integration in each. The need for integration and non-transparency must be considered individually by the system product.

5.1.3 Additional References

Technical references on the functioning of source routing bridging were obtained from the following additional draft documents. At this writing some of these are working documents, not standards, and therefore the information is subject to change.

- IEEE 802.5-90/43 Source Routing Tutorial for End System Operation
- IEEE 802.5M/E1 26-Jun-1991 End System Source Route Discovery to IEEE 802.2
- IEEE 802.5M/D5 15-Aug-1991 Source Routing Supplement to IEEE 802.1d (MAC Bridges)

5.1.4 Overview

Source routing is a routing technique that is optimized toward session oriented protocols and topologies that are relatively static.

When a protocol needs a route to a target system, it initiates a route discovery by sending a data link message with an explorer source routing type. That frame is forwarded and modified by the source routing bridges. Each bridge appends its ring number to the route descriptor field within the route information field of that message. When the frame finally reaches the target node, it can use the accumulated route information field as a source route to respond to the originating node.

Performance of a source routing implementation can be enhanced by providing session oriented protocols the ability to indicate the establishment of a new session, and the ongoing context of that session for subsequent transmitted and received messages.

5.1.5 Illegal Configurations

As with other LAN configurations, it is illegal to configure a single DECnet router with multiple direct connections to the same LAN or bridged LAN. If a source routing bridge is placed in parallel to a DECnet router, that bridge must have the forwarding of DECnet Phase IV routing messages blocked or filtered.

5.2 Protocol Operation

5.2.1 Route Discovery

Source route discovery is initiated when a new route is needed. The frame that caused the discovery to be initiated must not be held up awaiting the route, but must be transmitted with a Spanning Tree Explorer (STE) type RI frame. (or TSF, if source routing support is turned off). The source route discovery process will complete asynchronously to the data frame transmission that started it.

Route discovery will send a translated Ethernet Loopback Message. (Protocol type 90-00 in a SNAP SAP UI frame with a OUI of 00-00-00) If possible, the first discovery message should be sent on-ring without a Routing Information (RI) field. If the first discovery message does not receive a response, then another is sent using an All Routes Explorer (ARE) type RI field. This process is reinitiated by the upper level user retransmit. The source routing procedure does not have its own timer for completion of the discovery process.

All DECnet nodes must support the receipt and turnaround of the Loopback message as per DNA Node Product Specifications. A Loopback packet received on the local ring, may be returned with a null RI field or an RI field with no Route Descriptors (RD). When returning the Loopback Explorer packet, the target node's Source Routing support must also change the RI type to Specific Routed, reverse the Direction flag, adjust the Largest Frame size (if the received value is greater than the local receive frame size), and return the packet to originator with the received route descriptors. The target node will consider the route information for its own cache at this time.

Route discovery can be initiated by a non-transparent implementation upon request from the upper layer protocol. The transparent implementation will initiate a route discovery when a new destination address is attempted.

5.2.2 Route Reception

Nodes only need consider a received packet's source route for local selection for caching, if it is an explorer packet or a response to an initiated Loopback message.

5.2.3 Route Selection

Route discovery ends when the originating node stores a source route. In a network without strict tree connectivity, multiple possible source routes will be available to choose from. Some of the possible criteria are:

- First received (shortest latency path)
- Shortest path (fewest hops/length)

- Largest Frame
- a combination of the above

This specification will not limit the implemented selection criteria. First received is considered acceptable.

Any selection process must flush old routes in the case of a topology failure, so that it does not continue to attempt to use a perceived optimal route that is, in fact, currently broken.

Note that this specification does not attempt to establish multiple routes to a node, and therefore cannot utilize parallel bridges deterministicly unless more than one route is stored and traffic is distributed among them. In practice, most extended LANs are configured as trees anyways.

5.2.4 Specifically Routed Frames

Once a route is established, the end stations keep the source route information available to be inserted in each frame as transmitted.

The transparent implementation will insert the RI information itself. A non-transparent implementation may require the upper level to insert its own information.

If a transmit is requested of a transparent implementation when it has not yet established a source route, the message will be sent with an STE type.

5.2.5 Non-Routed Frames

If it is discovered that the peer stations are on the same ring, or reachable via a transparent bridge (this could result from a discovery response without an RI, or an RI with no more than 2 RD fields), then RI information need not be inserted in subsequent packets.

All DECnet 802.5 implementations must be able to receive frames that contain RI fields, even if they optionally support sending them, without error. This allows interoperation between source routing and non-source routing systems.

5.2.6 Multicast Frames

Since multicast messages do not have an explicit destination address or route, they must be treated as a special case in order for these messages to be forwarded by source routing bridges.

If source routing is enabled, multicast messages will be sent with a source routing type of Spanning Tree Explorer. The receiver must not respond to the explorer type, at the source routing level. The higher level protocol must act on the data message as it would upon normally receiving that message.

The caching of STE explored routes by the receiver is optional. However, protocols that use their own multicast data messages to explore by request/response (eg: NETBIOS), benefit by a short time cache of STE multicast routes. Servers that receive query packets need the application to search its database and respond to the query using the reverse route.

5.2.7 Aging and Topology changes

In order that source routes can change when the topology changes, source routes stored in a system will be aged and periodically rediscovered. The aging timer will be reset when an explorer is received from the other system.

A non-transparent implementation could have an interface for the upper level protocol to indicate that a route is suspect and should be rediscovered.

5.3 Abbreviations Glossary

SR - Source Routing
RII - Source Routing Indicator bit: the group/multicast bit of the MAC layer source address is used to indicate that source routing information is present in the frame.
RI - Routing Information: the field that contains the source routing information including the type and list of route descriptors.
TSF - Transparent Spanning tree Frame: a frame without an SR RI field that would follow the spanning tree if transparent bridges were present.
SRF - Specifically Routed Frame: a frame with an RI field that contains the explicit route for the data PDU.
STE - Spanning Tree Explorer: a frame with an RI field that causes the frame to cover the spanning tree of SR bridges.
ARE - All Routes Explorer: a frame with an RI field that causes the frame to flood the SR network covering all routes without repeat.

5.4 Example Flows

The following are example flows of source routing discovery in operation.

5.4.1 Typical Discovery

This example assumes a source route connected LAN with the target node not on the same ring.

Figure 5–1: Typical Discovery

Figure 5–1 (continued on next page)

Figure 5–1 (Cont.): Typical Discovery

```
Xmit Msg --o----STE--->------------------->---- rcv pdu

          Loop(ARE)->------+---+---+---->-+--- (cache route)
                           \   \   \--->-|+
                            \   \------>-||+
                             \--------->-|||+
    (cache route) <------------------<-+||| Loop response (SRF)
                  <------------------<--+||
                  <------------------<---+|
                  <------------------<----+

Xmit Msg--o----SRF--->------------------>---- rcv pdu
     (insert route)
```

5.4.2 Discovery with On-ring detection

This example assumes that the target node is not on the same ring as the originator, and that the originating node uses the optional feature where the first loop message is sent without source routing.

Figure 5–2: Discovery with On-ring detection

```
Xmit Msg--o----STE--->------------------>---- rcv pdu
           \                                      .
          Loop(TSF)->  [not received]             .
                                                  .
                                        (upper level response)
rcv pdu ---<---------------------------<-STE-o- Xmit Msg
                                              /
(cache route) ----<----------+----+----+--<---- Loop(ARE)
                +:-----------/    /    /
               +||--------------/    /
Loop          +|||-------------------
Response      |||+---------------------------> (cache route)
(SRF)         ||+----------------------------->
              |+------------------------------>
              +------------------------------->

Xmit Msg--o----SRF--->------------------>---- rcv pdu
     (insert route)
```

5.4.3 On-Ring Discovery

This example assumes two stations on the same ring where the originating node sends a source routing explorer when the cache entry is initialized.

Figure 5-3: On-Ring Discovery

```
Xmit Msg--o----STE--->------------------>---- rcv pdu

          Loop(ARE)->------------------->-+--  (cache route: On-Ring)
                                          |
      (cache route) <------------------<-+  Loop response (SRF)

Xmit Msg--o----TSF--->------------------>---- rcv pdu
    (insert Null route)
```

5.4.4 On-Ring Discovery, Optimal

This example assumes two stations on the same ring where the originating node uses the optional non-source routed message the first time. This is preferred in that it does not cause explorer messages to be propagated to the rest of the bridged LAN.

Figure 5-4: On-Ring Discovery, Optimal

```
Xmit Msg--o----STE--->------------------>---- rcv pdu
          \
          Loop(TSF)->------------------>-+--  (cache route: On-Ring)
                                          |
      (cache route) <------------------<-+  Loop response (TSF)

Xmit Msg--o----TSF--->------------------>---- rcv pdu
    (insert Null route)
```

5.5 Transparent Implementation

5.5.1 Interface

send_packet(portal, da, data)
received_packet(portal, sa, data)

5.5.2 Operation

This implementation uses a cache table based on data link destination address. If only DECnet is being supported, the addresses may be the two byte node number, otherwise it is the 6 byte data link address.

Cache entries are made for transmit operations when a new address is used. The packet is sent with a Spanning Tree Explorer and discovery is separately initiated and processed.

Cache entries are made for received packets if they have an explorer type, or optionally are a multicast (with or w/o RI).

Cache entries are aged on a timer (default 60 seconds), and rediscovery is initiated when the timer expires. If the route is rediscovered then the entry timer is restarted. If the timer expires again, and the entry is still state, then it will be marked as No-route. Stale entries can be kept until needed to be reused.

5.6 Non-transparent Implementation

5.6.1 Interface

discover_route(da, handle)
read_route(da, handle, buffer)
get_attributes(handle, buffer)
set_attributes(handle, buffer)
reconfirm_route(da, handle)
end_route(da, handle)

send_packet(portal, da, handle, data)
received_packet(portal, sa, handle, data)

5.6.2 Operation

The benefit of a non-transparent implementation is that the session oriented upper level can directly manipulate the discovery process, but does not have to implement the details of the messages.

The upper level can initiate the route discovery, and get a context handle to a stored route (or the route directly) and use that route on subsequent send operations so that the lower level does not need to look up the address. Similarly, if sufficient retries indicate that a route may be failing, a rediscovery should be initiated. When the session is finished, the usage count on the entry can be decremented.

The discover_route function would initiate the discovery procedure and would return a context handle for the status of the route information.
The read_route function would return the status of the source route and the actual routing information.
The get_attributes function would allow the reading of management attributes of the route or the source routing modules as a whole. The set_attributes function would allow the setting of management information.
The reconfirm_route function would initiate a new discovery of the route.
The end_route function would indicate that this user is no longer interested in the route entry.
The send and receive functions would insert (or remove) the route information pointed to, into the packet before transmitting or presenting to the user.

5.7 Source Route cache information

destination address
status

No-route - no path known
On-Ring - no path needed
Have-Route - RI stored

Re-discovering - RI suspect or timeout, discovery in progress
Stale - Have route, but Aging timer expired
Weak-Route - Have STE multicast route

timeout
RI field

type
direction
length
largest frame
route descriptors (0-28)

NOTE

While the 802.5 standard allows for 14 hop routes, current implementations only support 7 hops.

5.7.1 Parameters

Aging timer - timeout for a currently active RI to be rediscovered. The minimum is 5 seconds. The maximum is local infinity. A range of at least several days is best. The recommended default is 60 seconds.

5.7.2 IEEE 802.5 clarifications and notes:

ARE and STEs must have a RI Length = 2 when originated
The x bits of RDType must be zero on xmit, ignore on receipt.
The initial Direction of explorers must be forward
The Largest Frame value should be 1500 bytes for normal Ethernet emulation

5.7.3 Pseudo code

The following code serves as an example, not the standard, of how to implement the source routing process described above.

```
Transmit -

- If Source Routing disabled, send as TSF

- Check if DA is multicast?
    - yes: send as Spanning Tree Explorer

- Check if DA is in cache:
    - yes: set aging in-use flag, and check status
        - on-ring: send data without SR
        - have-route: send data with RI
        - no-route: send data with STE
  - continue discovery: send Loopback msg with ARE

    - no: send data frame with STE
        enter in cache
        mark entry as No-route
        initiate discovery,
            send Loopback msg without RI
--------------------------------

Timeout:
    - Aging:
        - Did we receive a Loopback from the remote?
            Yes: reset aging timer
        - Was entry used to transmit?
            Yes: initiate new discovery
            No: mark as stale

--------------------------------

Receive complete:

    - If Source Routing disabled,
        strip RI field and pass to upper layer

    - Is it a Loopback?
        - yes, continue
        - no:
            - Is it an Explorer?
                - no: strip RI, pass frame to upper layer
                - yes: Is it a multicast and we are caching them?
                        - no: strip RI, pass frame to upper layer
                        - yes: continue

    - check if SA is in cache?
        - no: create entry
            - RI field present?
                - no: mark as On-ring
                - Yes: store route,
                        - if multicast
                              mark as Weak-Route
                          else
                              mark as Have-Route
        - yes: check status?
            - Have-Route:
                - reset aging in-use flag
                - is this route better?
                    -yes: replace with new route
                    -no: discard route
            - No-Route:
                store route,
                if multicast;
                    mark as Weak-Route,
                else
                    mark as Have-Route

            - On-Ring:
                - RI field present?

    - strip RI, pass frame to upper layer
```

5.8 Source Routing Information Field

A summary of the routing information field follows. For more complete information, consult the reference documents.

Figure 5–5: Source Routing Information Field

```
          |<----------------RI Field------------------------->|
          |<-----RC Field------>|<------RD Fields------------>|
          +----------+----------+---+---+---+---+-----+---+----+
          | RT | LTH | D | LF |r|  RD1  |  RD2  | ... |  RDn  |
          +----------+----------+---+---+---+---+-----+---+----+
 Bits       3    5     1   6   1  15       16      ...    16

          |<-----Length represented by LTH------------------->|
```

Table 5–1: Source Routing Information Field Subfields

Subfield	Length (bits)	Description	
RT	3	Routing Type	
		Values are:	
		0xx	Specifically Routed Frame (SRF)
		10x	All Routes Explorer (ARE)
		11x	Spanning Tree Explorer (STE)
LTH	5	Length in bytes (2-30)	
D	1	Direction	
		0=Forward order, 1=Reverse order	
LF	6	Largest Frame	

Table 5–1 (Cont.): Source Routing Information Field Subfields

Subfield	Length (bits)	Description
		Base Values are:

Value	Length (octets)	Description
000	516	ISO 8473, Connectionless Network Protocol
001	1470	ISO 8802-3, CSMA/CD LAN
010	2052	80 by 24 char screen with control
011	4399	ISO 8802-5, FDDI, 4Mb Token-ring, 9314-2
100	8130	ISO 8802-4, token-bus LAN
101	11407	ISO 8802-5, 4 bit burst errors unprotected
110	17749	ISO 8802-5, 16Mb token-ring LAN
111	41600	Base for extended values (See SRT spec.)

Subfield	Length (bits)	Description
r	1	reserved (ignored upon receipt, transmitted as received)
RDn	16	Route Descriptors Subfields:

Bits	Description
4	Bridge Number
12	LAN ID

5.9 Tradeoffs

The following sections discusses issues that were considered but discarded in the design of source routing support.

5.9.1 Route Discovery, A&C bits

The A&C bits could be used to attempt to decide on whether a target node is on the local ring. However, SR bridges will set the A&C bits when forwarding a packet to a destination ring, even if the node does not exist there. TB bridges behavior is optional, and therefore will vary. If a TB bridge does not set the bits, then you may believe the node is not there when it is, and cause unnecessary discoveries. If a TB bridge does set A&C bits, then you may believe that the node is local, when it is not, and you would not attempt route discovery.

5.9.2 Route Discovery, piggybacking on upper level msgs

Because route discovery will result in multiple frames arriving at the target and possibly the originating node, it was decided not to attempt to use upper level messages for carrying session explorer frames. However there is a chance that even STE packets can be duplicated or lost by misconfigured or manual bridges.

5.9.3 Route Discovery, receive processing

To avoid the processing overhead of looking at every packet, receive processing is limited to Explorer frames and Multicast destinations. This however, penalizes the Aging process, since it could benefit from knowledge of working RIs received by this node.

5.9.4 Aging

If routes are not aged, there is no recovery from topology changes. The aging timer must be set to be able to recover before the session disconnect timers expire, preferably half. But not so short as to cause undo overhead.

Aging rediscovery should attempt to minimize overhead. Robustness requires that it be sensitive to unidirectional traffic flows.

5.9.5 Route Discovery, Message used

The original proposal used an 802.2 XID Request message to the Null (00) SAP from a local SAP as the discovery message. This was removed when it was discovered that the IBM 8209 Token Ring to Ethernet Bridge could not support Ethernet nodes that send both Ethernet and 802 frame formats. This also removed the requirement that all nodes support 802 XID server responders, which allows support of Ethernet-only fixed function systems (ie: Terminal Servers).

XID messages will be used for discovery by 802.2 aware applications. Since this specification addresses only Ethernet emulation, this issue will be addressed in DECnet Phase V token ring support.

Chapter 6

Routing using Arbitrary MAC Addresses (Phase IV-Prime)

6.1 Overview

This chapter specifies modifications to the DNA Phase IV Routing and Data Link behavior referred to as **Phase IV-Prime**. These changes are intended to allow the DECnet Phase IV Routing protocol to run on a LAN using an arbitrary MAC Individual Station Address instead of the previously required address using a Digital high-order vendor block and the node address. The benefit of this change is that DECnet nodes can coexist on systems and networks with other protocols that have other station address restrictions.

This capability is new to this specification. It is a required option for any node attached to an 802.5 Token Ring and is the preferred default. Phase IV-Prime may be implemented on other 802 compatible LAN as well, but is not required.

Systems with attachments to both Token Ring and other communications types will have to implement both Phase IV-Prime and regular Phase IV. This is often called a **bilingual** router.

Modifications to the operation of end nodes and routers will be described separately. Changes to the databases, functions, and frame formats will be described for both endnodes and routers.

6.1.1 Transition from Phase IV

Nodes of both Phase IV and Phase IV-Prime can be mixed on a given LAN, if a bilingual router is present. It is recommended that all routers be transitioned to Phase IV-Prime for robustness in the case of failures or outages.

Phase IV-Prime cannot be bridged from Token Rings to Ethernet because the Functional Address mapping cannot be reversed without parsing the message. Routers must be used.

Mixing of Phase IV-Prime nodes with older implementations DECnet Phase IV may cause problems, if the system has not properly implemented message processing for forward compatibility. In general, any message field that has reserved bits or values, must be sent as zeros and ignored on receipt.

6.2 Operation

6.2.1 Address Assignment

These modifications allow the DECnet node to operate over any arbitrarily assigned media layer address. That address may be from the space globally assigned to the manufacturer by the IEEE Standards Office and stored in a ROM on the adapter (the default), or it may be assigned by the system manager out of the Locally Administered space. When addresses are assigned locally, it is the responsibility of the system manager to ensure that the same address is not assigned to more than one LAN attachment point.

6.2.2 Additional Multicast IDs

Two new multicast IDs have been assigned to implement Phase IV-Prime Routing. Since 802.5 Token Ring LAN adapters do not support multicast addressing of this type, for the present time these multicast IDs must be be mapped "Functional Addresses" (see Table 2–1).

When the Functional Addresses found in Table 2–1 are used, these Functional Address values must be considered as defaults only. Phase IV-Prime nodes must support a management mechanism to allow the value of the Functional Addresses used by routing to be changed. This is because Functional Addresses are not guaranteed to be globally unique. On certain LANs, the default Functional Address values assigned for use by DECnet Phase IV-Prime may already be in use. Correct operation of the network may require that Phase IV-Prime nodes use a different value than the default. In any case, it is recommended that the endnodes use a different multicast ID than the routers.

"All Phase IV-Prime Routers" - The multicast ID to which all Phase IV-Prime endnodes transmit end node hello messages. This address is enabled by all Phase IV-Prime routers.

"Unknown Destination" - The multicast ID to which Phase IV-Prime endnodes transmit data packets when the target destination is known to reside on the same LAN, but its data link address is unknown, or when the target destination's data link address is unknown and there is no designated router on the same LAN. This address is enabled by all Phase IV-Prime end nodes.

The actual values assigned to "All Phase IV-Prime Routers" and "Unknown Destination" can be found in Table A–4.

6.2.3 Endnodes

6.2.3.1 Endnode Hello Messages

Phase IV-Prime Endnodes multicast Endnode Hello Messages to the All Phase IV-Prime Routers multicast ID (in the same way that Phase IV endnodes send endnode hello messages), except that the packet header is constructed in a way so as to make the hello messages unintelligible to a Phase IV router. For Phase IV-Prime endnode hello messages, the value of "TYPE" in the FLAGS field = 7 (instead of 6), indicating "EXTENDED TYPE"; and the next higher order 3 bits (that are reserved for all TYPEs other than 7) = 0, indicating "Phase IV-Prime endnode hello" (see "Messages" section).

This type of packet must be ignored by Phase IV routers.

6.2.3.2 Designated Router's Hello Message

Endnodes learn and maintain the ID of the designated router exactly as Phase IV endnodes do, except that Phase IV-Prime endnodes also store the 48 bit data link address of the designated router.

Endnodes can recognize Phase IV-Prime routers from others by the FP bit in the Router Hello message. Phase IV-Prime Endnodes will ignore Designated Router Hello messages from non-Phase IV-Prime routers, if the endnode is not running a DECnet station address.

6.2.3.3 Unknown Destination Multicast

Endnodes listen to the Unknown Destination multicast for Phase IV Long Data Packets. If a packet is received that has the network layer destination address of the local system, then that packet is processed. All other data packets are discarded with an error being logged or counted.

Endnodes send Data packets to the Unknown Destination address only if they do not have a cache entry for the destination routing layer address and they do not have a known Designated Router address.

6.2.3.4 On-LAN Cache Maintenance and Filling in "next hop"

Phase IV-Prime endnodes maintain a cache of destinations with which it is in contact as do Phase IV endnodes, but Phase IV-Prime endnodes also cache the 48 bit data link address of each destination as well. Only packets received with the routing layer destination of the local node will be processed for caching.

The following table describes how entries are made in the cache. as well as which data link address gets associated with the routing layer address of each cache entry.

Table 6–1: On-LAN Caching

Packet Received with:				Cache Entry Resulting	
Routing Layer Source Address	**Datalink Layer Source Address**	**Visit Count**	**Intra-LAN bit**[1]	**Routing Layer Destination Address**	**Datalink Layer Destination Address**
Source RLA	Source DLA	>0	1	Source RLA	Unknown Destination
Source RLA	Source DLA	=0	1	Source RLA	Source DLA
Source RLA	Source DLA	>0	0	Source RLA	Source DLA
Source RLA	Source DLA	=0	0	Source RLA	Source DLA

[1]The Intra-LAN bit was previously known as the On-Ethernet bit

RLA—Routing Layer Address
DLA—Datalink Layer Address

The last case, visit-count=0, intra-LAN-bit=0, should never occur in the presence of correctly implemented endnodes, the case is included here for completeness.

When a Phase IV-Prime endnode needs to send a packet to a RLA that is not in the cache, the packet is sent to the designated router. If the address of the designated router is unknown, the packet is sent to the Unknown Destination multicast ID.

Cache entries are erased according to existing mechanisms. For example, a cache entry for destination A is erased when:

1. the Routing Layer user gives the Routing Layer a packet with destination A and the directive "Tryhard"
2. no traffic is received from node A validating the cache entry for CACHETIMEOUT (a parameter)
3. node A is the least recently used cache entry, and room in the cache is needed for a new entry

6.2.4 Routers

Phase IV-Prime routers implement functions that are a superset of Phase IV routers. This means that Phase IV-Prime routers can forward packets to/from Phase IV-Prime endnodes and routers, and if the router is configured with a valid Phase IV address (consisting of HI_ORD + the network layer address) it can forward packets to and from Phase IV endnodes and routers also.

6.2.4.1 Phase IV and Phase IV-Prime Interoperability

A Phase IV-Prime router with a Phase IV compatible data link address is capable of being a "bilingual" (i.e. Phase IV & Phase IV-Prime) router. One type of bilingual router would be a router with both an Ethernet and a Token Ring interface.

The Ethernet circuit would be Phase IV (with a likewise compatible data link address), and the Token Ring circuit would be Phase IV-Prime (with an arbitrary address). This type of router can forward packets between Phase IV Ethernet nodes and Phase IV-Prime Token Ring nodes.

It is also possible to construct a bilingual router whose only LAN interface(s) is (are) Token Ring. This router would be a Phase IV-Prime compatible router, with a Phase IV compatible data link address, that could receive messages from one type of node and forward to the other type over the same circuit. This type of node will have to send some messages twice, one in each format. (ie: router hellos)

On a given LAN:

1. if there are any Phase IV level 1 routers or endnodes, all Phase IV-Prime level 1 routers must have Phase IV compatible data link addresses.
2. if there are any Phase IV level 2 routers, all Phase IV-Prime level 2 routers must have Phase IV compatible data link addresses.

In sum, a bilingual router must be present on any LAN to which is attached a mixture of Phase IV and Phase IV-Prime nodes.

6.2.4.2 Router Hello Messages

Phase IV-Prime Routers periodically multicast Router Hello messages (as do Phase IV Routers) but to the All Phase IV-Prime Routers multicast ID. The designated router additionally sends Router hello messages to the All-Phase-IV-endnodes multicast ID.

Router hellos messages coming from a Phase IV-Prime router have a newly defined bit set, called the "FP" bit (formerly reserved). This bit is transmitted as zero and ignored on receive by Phase IV routers. Its purpose is to give Phase IV-Prime routers an indication of the presence of any Phase IV routers on the LAN.

If a bilingual router supporting both Phase IV-Prime endnodes and Phase IV endnodes on the same circuit, and becomes the designated router, it must send two Router Hello messages, in each of the respective formats to each of the respective multicast ids.

A Phase IV-Prime router can tell if a router hello message or an endnode hello message originated from a Phase IV node (by observing the packet type in endnode hello messages, and by observing that the FP bit in the flags field of the router hello message).

If a Phase IV-Prime router which DOES NOT have a Phase IV compatible data link address (eg: not running bilingual) hears a Phase IV Router or Endnode Hello message, it must ignore the content of the message, and signal the presence of this potential network topology problem by logging an "adjacency initialization failure" event with a reason of "Phase IV node exists in the absence of bilingual router".

6.2.4.3 Database of Endnodes

In addition to what Phase IV routers store in the adjacency database, Phase IV-Prime routers must store the entire 48 bit DLA of any Phase IV-Prime adjacency. •

Routers learn if a node is Phase IV or Phase IV-Prime by observing its endnode hellos. Phase IV-Prime endnodes send endnode hellos exactly as Phase IV endnodes, except the value of "TYPE" in the FLAGS field = 7 (instead of 6), indicating "EXTENDED TYPE"; and the next higher order 3 bits (that are reserved for all TYPEs other than 7) = 1, indicating "Phase IV-Prime endnode hello".

The node type, Phase IV or Phase IV-Prime, is stored in the adjacency database.

Since Phase IV-Prime endnode hellos are sent to a different multicast ID than are Phase IV endnode hellos, Phase IV-Prime routers must enable both the "All Phase IV-Prime Routers" multicast ID as well as the "ALL-ROUTERS" Multicast ID. In other words, Phase IV-Prime routers "listen to" both Phase IV and Phase IV-Prime endnode hellos.

Phase IV-Prime routers advertise as reachable through it all Phase IV and Phase IV-Prime nodes to which it has an adjacency.

6.2.4.4 Forwarding Process

Phase IV-Prime routers use the stored 48 bit DLA to forward a packet to a Phase IV-Prime adjacency (obviously, the Phase IV compatible destination DLA of Phase IV nodes may be stored and used just like Phase IV-Prime arbitrary DLAs, or they can be constructed out of HI_ORD and the network layer address like Phase IV only Routers do.

Phase IV-Prime Routers clear the "intra-LAN" bit when forwarding between Phase IV and Phase IV-Prime nodes (endnodes or routers).

6.2.4.5 Router Priority

On a given LAN or bridged LAN, if there exist a mixture of Phase IV and Phase IV-Prime routers or endnodes, the DECnet router priorities must be set such that a bilingual router will become the designated router over all other DECnet routers. This is to prevent a partioned routing area, as non-Phase IV-Prime routers will not be able communicate with Phase IV-Prime routers or endnodes that do not have compatible addresses.
To help ensure this, the default value for priority of a bilingual Router should be one greater the default value of a comparable Phase IV router.

6.2.4.6 Multiple Connections to an Extended LAN

If one or more bridges are connected in parallel with a multicircuit router, the parallel bridges must be set to filter out DECnet Phase IV routing layer traffic (Protocol type 60-03).

6.2.5 Messages

The message format notation used here is identical to that used in the Phase IV Routing Layer Functional Specification.

Routing layer 6-byte node ID fields (such as the Long Data Packet D-ID field) still consist of the HI-ORD constant plus the node number. Arbitrary data link addresses only appear in the data link header of a frame.

6.2.5.1 Router Hello Messages

One bit taken from the "reserved" bits in the "Flags" field of the hello messages is defined to indicate the state of the "IV-Prime" field.

```
FLAGS (1) : BM   the Routing Layer control flag, with the following format:

                              +---+---+---+-------+---+---+---+
               Bit:           | 7 | 6 | 5 | 4   3 | 2 | 1 | 0 |
                              +---+---+---+-------+---+---+---+
               Set to:        |PF |  RES  |FP |  TYPE     | 1 |
                              +---+---+---+-------+---+---+---+

                       Bit    Definition
                       ---    ----------
                        0     1 indicates Control Packet
                       1-3    Type = 5
                        4     FP  Four Prime = 1 indicating hello
                              sourced from a Phase IV-Prime router
                       5-6    Reserved
                        7     PF  pad field = 0 indicating no padding
                              follows
```

6.2.5.2 Endnode Hello Message

The FLAGS field in the Phase IV-Prime Endnode hello messages is modified in order to make the hello messages unintelligible to Phase IV routers. This is done by encoding a new message type in the TYPE field. Since there is only one TYPE code left, the last remaining TYPE field indicates the presence of an EXTENDED TYPE field in the next higher order 3 bits.

```
FLAGS (1) : BM  the Routing Layer control flag, with the following format:

                        +---+---+---+---+---+---+---+---+
         Bit:           | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
                        +---+---+---+---+---+---+---+---+
         Set to:        |PF | EXTENDED  |   TYPE    | 1 |
                        |   | TYPE      |           |   |
                        +---+---+---+---+---+---+---+---+

                Bit     Definition
                ---     ----------
                 0      1 indicates Control Packet
                1-3     Type = 7 indicating EXTENDED TYPE field present
                4-6     Extended Type = 0 indicating Phase IV-Prime
                        endnode hello
                 7      PF  pad field = 0 indicating no padding
                        follows
```

Chapter 7

Network Management

7.1 Overview

This chapter specifies how Token Ring data links are managed by DNA Phase IV Network Management. The intent of the description is for implementors of NCP and NICE programs. The final authority on actual definition and behavior of elements of the IEEE 802.5 Standard lies in the standard, not this document.

7.2 DNA Executor Entity

The Executor characteristics and counters are unmodified, except if Phase IV-Prime is implemented, in which case there are three new Executor Type (901) values.

Table 7–1: Executor Type Values

Value	Meaning
6	Area Phase IV-Prime
7	Routing Phase IV-Prime
8	NonRouting Phase IV-Prime

7.3 DNA Circuit Entity

The Circuit characteristics and counters of Token Ring data links can be implemented exactly as for the Ethernet data link circuits. A new value for Circuit Characteristic Protocol (1112) is defined below in Table 7–3.

7.4 DNA Line Entity

7.4.1 Line States

Unlike the Ethernet, Token Ring data links have several intermediate states of operation. These states will be reflected using substates of the Line entity. Any transition into Running will signal the Routing layer to start the circuit. Loss of Ring insertion should signal a line sync loss to Routing.

Table 7–2: Line State Mapping

Token Ring State	NICE Substate
Off (not in ring)	IDLE
Insertion	SYNCHRONIZING
1. Media Lobe test	
2. Physical insertion	
3. Address verification	
4. Neighbor notification	
5. Request Initialization	
Monitor Contention (establishing active monitor)	STARTING
Beaconing (hard error recovery)	REFLECTING
Running	RUNNING
Failed (out of ring)	BROKEN

7.4.1.1 Data Operations while not Running

The recovery modes of the Token Ring can take on the upwards of 20 seconds before totally failing or inducing the failed node to leave the ring. During this time period, requested data transmit requests will either be queued or rejected. The latter being more difficult to implement and detect, but would allow time sensitive applications to try other means.

7.4.1.2 Ring Insertion and Recovery Operation

The driver should wait until opening the first portal to insert itself into the ring. This allows the station address to be set before insertion. Otherwise, the adapter must remove itself, change address and re-insert.

If the adapter senses a ring removal, it should log a line state change. In general, for connectivity failures, the line state must stay On, but the substate may become Synchronizing, while the driver attempts to re-insert.

Figure 7–1: Line State Transition Diagram

```
OFF    <++++ User Command  (or Remote Remove)
 +
 + User Command
 v
SYNC  (Insertion)  <--------------------\
 | \                   ^                |
 |  \-----------------------/           |
 |      Failure     if AutoRemoval      |
 |                    then BROKEN       |
 | Success                              |
 v                                      |
RUNNING    <--------------------\       |
 |   \                          |       |
 |    \ Soft Error              |       |
 |     v                        |       |
 |     STARTING  (Monitor Cont.)  |     |
 |          |        ^    \-------/     |
 | Hard     |         \    Success      |
 | Error    |          \                |
 |        /Failure      \               |
 |       /               \              |
 v      v                 \             |
REFLECTING (Beaconing)     |            |
 | \                       |            |
 |  \-------------------/               |
 |      Success                         |
 |                                      |
 | AutoRemoval                          |
 \--------------------------------------/
```

Possible algorithms would be:

- Driver stays in On-Sync state, until a send request arrives, then attempts re-insertion.
- Driver stays in On-Sync state, and retrys insertion on a timer. (recommended default is 10 seconds)

7.4.1.3 Exception Conditions

The MAC layer may fail to resolve the beaconing process, then auto remove itself from the ring and then possibly fail self test. In which case the line is in the BROKEN state. This event should be logged.

The MAC layer may also receive a Remove from Ring packet from a management station. In this case it will transition to the OFF state, and log the event.

7.4.2 Line Characteristics

The following line characteristics are unique to 802.5 token rings and should be provided via standard network management.

- Station Address - The current individual address in use for receiving packets from the ring. Read only once inserted in ring. Set by management when Line is off.

- Functional Address - The current Functional Address mask in use by the data link for accepting Functional Address group packets from the ring.
- Group Address - The current addresses in use for accepting Group packets from the ring.
- Upstream Neighbor - The address of the next active upstream neighbor station. Learned during periodic (7 second) ring poll. Read only status.
- Ring Number - The administrative ring number. Used by source routing bridges. Learned during ring poll at insertion time. May be zero if not assigned or no bridges on ring. Only bridges or ring parameter servers can set this value.
- Authorized Access Priority - The highest access priority allowed by this station for sending LLC traffic (0-6). Default is 3.
- Ring Speed - The speed of the token ring media in use. (4 Megabits per second or 16Mbs).
- Early Token Release - This indicates whether ETR is in use by the local station. This is only allowed on 16Mbs rings and is the default.
- Source Routing - This specifies if the source routing process is enabled. On is the default.
- Address Type - The type of station address in use. If Phase IV-Prime is active, the MAC layer station address can be arbitrarily set. The default is Hardware, the unique assigned ROM address. "DECnet" indicates a Phase IV compatible constructed address. User assigned indicates that the value of the Station Address parameter is to be used.
- Aging Timer - This is the number of seconds between source route periodic explorations.

Table 7–3: NICE Data Link Line Parameters for Token Ring

Type Number	Data Type	Inf. Type	Set Restrict.	NCP Keywords
Common Parameters				
0	C-1	S*		State
1	C-1	S*	RO	substate
100	C-1	C		Service
110	DU-2	C		Counter Timer
1100	AI-16	C		Device
1105	DU-2	C		Receive Buffers
1112	C-1	C		Protocol
1160	HI-6	C	RO	Hardware Address

Table 7–3 (Cont.): NICE Data Link Line Parameters for Token Ring

Type Number	Data Type	Inf. Type	Set Restrict.	NCP Keywords
Media Specific Parameters				
1170	HI-6	S	RS	Station Address
1171	HI-6	S		Functional Address
1172	HI-6	S		Group Address
1173	HI-6	S	RO	Upstream Neighbor Address
1174	DU-2	C		Ring Number
1175	DU-1	C		Authorized Access Priority (0-6)
1176	C-1	C		Ring Speed (0=4Mbps, 1=16Mbps)
1177	C-1	C		Early Token Release (0=Enabled,1=Disabled)
1178	C-1	C		Source Routing (0=Enabled, 1=Disabled)
1179	C-1	C		Address Type (0=DECnet, 1=Hardware, 2=User)
1400?[1]	DU-2	C		Aging Timer (5-65365 seconds, Default=60)

[1]Contingent on approval of the NICE Registry

The value of PROTOCOL used for CIRCUIT and LINE parameters
(ie: CIRCUIT TYPE (1112) and LINE PROTOCOL (1112))

Value	Meaning
11	802.5/Token Ring

7.4.3 Line Counters

The following section documents the Line counters set for 802.5 data link. The line counters consist of a common set of DECnet counters, counters for line state changes and errors, and the soft counters as in the 802.5 Standard and implemented in the chip set.

The current chip sets implement only the 802.5 counters as one octet quantities. The counter threshold is therefore 255 and an interrupt occurs when a counter is incremented to 255. The chip counters are zeroed upon reading them. The much wider counters below, can be maintained by adding the chip counters to them at appropriate intervals or interrupts, in order not to lose increments. The 64 bit counters should wrap upon reaching maximum.

For Phase IV, NICE management only allows 32 bit counters, therefore this is the greatest width of the NICE structures below. Phase IV counters also latch at their greatest values.

For Phase V, 64 bit counters are standard, and bit map counters are not allowed. To conserve on future work, internal 64 bit counters are recommended for every countable event. These counters can be mapped on the to 32 bit counters and bit maps.

The chip set also maintains a separate set of counters for reporting on the ring at the MAC layer. These counters are reported if non-zero after 2 seconds, then cleared. Interfaces to allow the collection of remote error reports are outside the scope of this specification.

Isolating counters detect errors that have occurred between the detecting station and its upstream neighbor, being a fault with either station or the wire in between. Non-isolating errors indicate a problem in the ring that cannot be isolated to a particular station or lobe.

7.4.3.1 Line Counter Descriptions

DNA Errors

Send Failures - Errors while transmitting a frame
Receive Failures - Errors while receiving a frame
Insertion Failures - Errors while attempting ring insertion
Ring Failures - Errors while ring was running that cause the station to leave the ring
Ring Purges - MAC recovery event to reestablish token
Monitor Contention - MAC recovery event to reestablish Active Monitor
Beaconing Conditions - MAC recovery event to reestablish ring connectivity or integrity.

802.5 Isolating Error Counters:

Line Error - token in data frame, or FCS error
Internal Error - recoverable internal error
Burst Error - lack of proper transitions on wire
AC Error - error with AC bits in ring poll
Abort Delimiter - count of Abort delimiters sent
Private Errors - system specific counter

802.5 Non-Isolating Error Counters:

Lost Frames - Failure to see xmit frame return or complete
Receive Congestion - receive buffer unavailable
Frame Copied - frame received for self with copied bit already set
Frequency Error - bit freq is outside of expectations
Token Error - number of replacement Tokens generated
Private Errors - system specific counter

7.4.3.2 Line Counter Encoding

Table 7–4: NICE Data Link Line Counters for Token Ring

Type Number	Counter Type	Bit Width	Standard Text
Common Counters			
0		16	Seconds since zeroed
1000		32	Bytes Received
1001		32	Bytes Sent
1010		32	Data Blocks Received
1011		32	Data Blocks Sent
1012		32	Multicast Blocks Received
1063		32	Unrecognized frame destination
1064		16	Data Overrun
1065		16	System Buffer Unavailable
1066		16	User Buffer Unavailable

Table 7–4 (Cont.): NICE Data Link Line Counters for Token Ring

Type Number	Counter Type	Bit Width	Standard Text
Media Specific Counters			
1070	BM	32	Send failures 0 - Transmit underrun 1 - Line error 2 - Abort delimiter sent 3 - Lost Frame 4 - Token error 5 - Unrecognized frame (ARI not set) 6 - Remote congestion (FCI not set)
1071	BM	32	Receive failures 0 - Receiver Congestion 1 - Frame Copied error
1072	BM	32	Insertion failures 0 - Lobe wire fault 1 - Signal Loss Error 2 - Timeout 3 - Ring Purge Timeout 4 - Beaconing 5 - Duplicate Address detected 6 - Parameter Server Failure 7 - Remove Received
1073	BM	32	Ring Failures 0 - Lobe wire fault 1 - Single Station detected 2 - Auto Removal Failure 3 - Remove Received
1074		32	Ring Purges
1075		32	Monitor Contention
1076		32	Beaconing Conditions
1080		32	Line Errors
1081		32	Internal Errors
1082		32	Burst Errors
1083		32	Ring Poll AC Errors
1084		32	Abort Delimiters Sent
1085		32	Private Isolating Errors
1086		32	Transmit Lost Frames
1087		32	Receiver Congestion Errors
1088		32	Frame Copied Errors

Table 7–4 (Cont.): NICE Data Link Line Counters for Token Ring

Type Number	Counter Type	Bit Width	Standard Text
Media Specific Counters			
1089		32	Frequency Errors[1]
1090		32	Token Errors
1091		32	Private Non-Isolating Errors

[1] 802.5 defined but not implemented

7.5 Events

The IEEE 802.5-1989 specification defines only SMT events to be reported by the MAC and PHY layer. LLC events are not defined. The SMT events closely mirror the MAC layer event monitoring messages and their content. This information is not necessarily useful to typical DECnet nodes and should primarily be of interest to network management monitors.

The primary events that need to report for Phase IV basic nodes are those surrounding LINE state changes, and unusual conditions.

At this writing no events for Token Ring have been defined for NICE event logging, as no implementation is planning on logging line events. Contact the DECnet Architecture group for current assignments in this space.

State Change Events consistent with NICE:

- Sync failure - Insertion Failure
  Insertion failure reasons:
  - Lobe wire fault
  - Loop Test timeout
  - Duplicate Address detected
  - Insertion timeout
- Line up (->RUN) - Line Up
- Error detected (->START) - Monitor contention started
- Contention Failed (->REFLECT) - Beaconing started
- Line Sync Lost (->SYNC) - Beaconing failed, Auto Removal
- Line Down (->OFF) - Remove Received,or Auto Removal failure

Events defined by SMT:

- Enter Active State (RingNum, AMonAddr, una)
- Active Monitor Error(RingNum, StnAddr, una, error)
- Report Station in Ring(RingNum, iStnAddr, una)
- Configuration Change(RingNum, StnAddr, una)
- Neighbor Notification Incomplete(RingNum, AMonAddr, SrcAddrLS)

- Counter Threshold Reached(RingNum, StnAddr, una, Counters)
- Beaconing Condition (RingNum, StnAddr, una, btype, rstatus)
- Ring Station Removed (RingNum, StnAddr)

Table 7–5: NICE Event Record Definition

Events Class	Type	Entity	Standard Text
4	ROU	?21-31	Circuit events
5	DLL	?21-31	Line events
6	PLL	>5?	Line events

Table 7–6: NICE Routing Layer Events

Class	Type	Entity	Standard Text	Event Parameters and Counters
4	20	circuit	Adjacency Initialization failure Values for Reason are: **Value** — **Meaning** 15 — Phase IV node exists without bilingual router 16 — Phase IV designated router for Ph IV-Prime node	Reason, Packet header, Adjacent node address

Table 7–7: NICE Data Link Layer Events

Class	Type	Entity	Standard Text	Event Parameters and Counters
5	21	datalink	none defined Values for Failure Reason are: **Value** — **Meaning** >12 — /* tbd */	

Appendix A

Address Definitions and Assignments

Functional Addresses are defined in 802.5-1989 Section 3.2.4.1 as Group addresses, in the Local Administered space, with byte 2, bit 0 set to zero. The remaining low order 31 bits are used as separate addresses on a bit-wise basis, not on unique value. The 802.5 standard does not define the upper bits. The IBM Architecture Reference requires the upper bits to be zero, and subdivides the 31 bits into "user" and "reserved" ranges.

Figure A–1: Functional Address bit definitions

```
Summary:
 0        1        2        3        4        5
|01234567|01234567|01234567|01234567|01234567|01234567|
 11zzzzzz zzzzzzzz 0uuuuuuu uuuuurrr iiiiiiia iiiaaiaa
 GL                F
 Group, Local, Functional

z=must be zero, u=user assignable, r=reserved, i=IBM assigned, a=arch assigned
```

Table A–1: Known 802.5 Functional Addresses In Use

802.x Canonical	802.5 Reversed	Description	Ref
03-00-00-00-00-80	C0:00:00:00:00:01	[1]Active Monitor	[1,3]
03-00-00-00-00-40	C0:00:00:00:00:02	[1]Ring Parameter Server	[1,3]
03-00-00-00-00-20	C0:00:00:00:00:04	Network Server Heartbeat	[5]
03-00-00-00-00-10	C0:00:00:00:00:08	[1]Ring Error Monitor	[1,3]
03-00-00-00-00-08	C0:00:00:00:00:10	[1]Configuration Report Server	[1,3]
03-00-00-00-00-04	C0:00:00:00:00:20	Sync Bandwidth Manager	[5]

[1]802.5 Standard assigned

Table A–1 (Cont.): Known 802.5 Functional Addresses in Use

802.x Canonical	802.5 Reversed	Description	Ref
03-00-00-00-00-02	C0:00:00:00:00:40	Locate Directory Server	[5]
03-00-00-00-00-01	C0:00:00:00:00:80	NETBIOS	[3,4]
03-00-00-00-80-00	C0:00:00:00:01:00	[2]SR Bridge PDUs	[2,3]
03-00-00-00-40-00	C0:00:00:00:02:00	IMPL Server	[5]
03-00-00-00-20-00	C0:00:00:00:04:00	Ring Authorization	[5]
03-00-00-00-10-00	C0:00:00:00:08:00	LAN Gateway	[5]
03-00-00-00-08-00	C0:00:00:00:10:00	Ring Wiring Concentrator	[5]
03-00-00-00-04-00	C0:00:00:00:20:00	IBM LAN Network Manager	[3]
03-00-00-00-02-00	C0:00:00:00:40:00	ISO All End Systems	[6]
03-00-00-00-01-00	C0:00:00:00:80:00	ISO All Intermediate Systems	[6]
03-00-00-10-00-00	C0:00:00:08:00:00	User-Defined (12 bits) from	
03-00-02-00-00-00	C0:00:40:00:00:00	-through-	[3]
03-00-00-01-00-00	C0:00:00:80:00:00	Novell NetWare	[9]
03-00-F2-FF-1F-00	C0:00:2F:FF:F8:00	Apple TokenTalk Zone Info (19bits based on Zone ID)	[7]
03-00-02-00-00-00	C0:00:40:00:00:00	Apple TokenTalk Broadcast	[7]
03-00-02-00-00-00	C0:00:40:00:00:00	Remote Program Load (RPL)	[8]

[2]802.1D Standard assigned

Reference Sources:

[1]	IEEE 802.5-1989 Standard, pages 27-28
[2]	IEEE 802.1D-1990 Standard, table 3-6, page 47
[3]	IBM Token-Ring Architecture (SC30-3374-02 Sept 89) page 3-10
[4]	IBM LAN Technical Reference (SC30-3383-03 Dec 90) page 5-4
[5]	IBM Token-Ring Network Manager (LY-30-5595-0 June 86) page 3-82
[6]	ISO/IEC DIS 10589R, sect 8.4.8, table 10.
[7]	Inside AppleTalk, Second Edition 1990, page C-5
[8]	IBM Remote Program Load User's Guide, 2nd Ed. (83X8882 Jun 87) page 3-50
[9]	product inspection

Table A–2: Cross-Company Globally Assigned Multicast Addresses

Universal Multicast (canonical order)	802.5 Equivalent (reversed order)	Description
01-80-C2-00-00-00	80:01:43:00:00:00	IEEE 802.1D Bridge Group addr
01-80-C2-00-00-0x	80:01:43:00:00:x0	IEEE 802.1D Reserved (Bridge filtered)
01-80-C2-00-00-10	80:01:43:00:00:08	IEEE 802.1D All LANs Brg Mngmt Grp Adr
01-80-C2-00-00-11	ud	IEEE 802.1e Load Server Grp Adr
01-80-C2-00-00-12	ud	IEEE 802.1e Loadable Device Grp Adr
01-80-C2-00-00-14	C0:00:00:00:80:00	ISO IS-IS (DIS 10589R) All L1 IS Net Ent
01-80-C2-00-00-15	C0:00:00:00:80:00	ISO IS-IS (DIS 10589R) All L2 IS Net Ent
01-80-C2-00-00-16	C0:00:00:00:40:00	ISO 10030 - All CONS ES
01-80-C2-00-00-17	C0:00:00:00:80:00	ISO 10030 - All CONS SNAREs
01-80-C2-00-01-00	na	ANSI FDDI SMT - RMT Beacon
01-80-C2-00-01-10	na	ANSI FDDI SMT - Status Reports
01-80-C2-00-01-20	na	ANSI FDDI SMT - Root Concentrator
09-00-2B-00-00-04	C0:00:00:00:40:00	ISO 9542 All ES Net Ent
09-00-2B-00-00-05	C0:00:00:00:80:00	ISO 9542 All IS Net Ent
CF-00-00-00-00-00	C0:00:00:10:00:00[1]	Loopback Assistance
FF-FF-FF-FF-FF-FF	FF:FF:FF:FF:FF:FF	Broadcast
na	C0:00:FF:FF:FF:FF	802.5 Local Ring Broadcast

[1]Assigned by this specification

un=Unknown, ud=Undefined, na=Not Applicable.
Source: DECnet DNA LAN Address Registry, 4-Oct-1991

Table A–3: Addresses Assigned to Digital

Address blocks received from Xerox and IEEE	
(canonical order)	(reversed order)
AA-00-00-xx-xx-xx	55:00:00:yy:yy:yy
AA-00-01-xx-xx-xx	55:00:80:yy:yy:yy
AA-00-02-xx-xx-xx	55:00:40:yy:yy:yy
AA-00-03-xx-xx-xx	55:00:C0:yy:yy:yy
AA-00-04-xx-xx-xx	55:00:20:yy:yy:yy
08-00-2B-xx-xx-xx	01:00:D4:yy:yy:yy
00-00-F8-xx-xx-xx	00:00:1F:yy:yy:yy

Source: DECnet DNA LAN Address Registry, 4-Oct-1991

Table A–4: Multicast Addresses and Functional Addresses Defaults in use by Digital

Universal Multicast (canonical order)	Mapped Functional Addr. Equivalent (reversed order)	Protocol type	Description
AB-00-00-01-00-00	C0:00:40:00:00:00	60-01	MOP Dump/Load Assistance
AB-00-00-02-00-00	C0:00:20:00:00:00	60-02	MOP Remote Console
AB-00-00-03-00-00	C0:00:10:00:00:00[1]	60-03	DNA L1 Routers
AB-00-00-04-00-00	C0:00:08:00:00:00[1]	60-03	DNA Endnodes
AB-00-04-00-xx-xx	ua	60-06	Customer Use
AB-00-04-01-**-**[3]	user assigned	60-07	LAN Clusters (SCA)
09-00-2B-00-00-02	ua	80-3B	VAX ELN
09-00-2B-00-00-03	na	80-3F	LAN Traffic Monitor
09-00-2B-00-00-06	na	80-3C	CSMA/CD Encryption
09-00-2B-00-00-07	C0:00:04:00:00:00	80-40	PCSA NETBIOS Emulation
09-00-2B-00-00-0F	C0:00:02:00:00:00	60-04	LAT Service Advertisement
09-00-2B-01-00-00	na	80-38	All Bridges
09-00-2B-01-00-00	na	80-38	All Local Bridges
09-00-2B-02-00-00	C0:00:10:00:00:00[1]	60-03	DNA L2 Routers
09-00-2B-02-01-00	ua	80-3C	DNA Naming Service Advert.
09-00-2B-02-01-01	ua	80-3C	DNA Naming Service Solicit
09-00-2B-02-01-04	C0:00:01:00:00:00	60-04	LAT Service Solicit

[1]Routing and Endnode multicasts must be on distinct different values

[3]VAX Clusters sets the lower 16 bits based on the cluster id

ua=User Assigned, ud=Undefined, na=Not Applicable.
Source: DECnet DNA LAN Address Registry, 4-Oct-1991

Table A–4 (Cont.): Multicast Addresses and Functional Addresses Defaults in use by Digital

Universal Multicast (canonical order)	Mapped Functional Addr. Equivalent (reversed order)	Protocol type	Description
09-00-2B-02-01-07	C0:00:00:40:00:00	60-04	LAT Xwin service Solicit
09-00-2B-02-01-0A	C0:00:10:00:00:00[1]	60-03	Phase IV-Prime Routers
09-00-2B-02-01-0B	C0:00:08:00:00:00[1]	60-03	Phase IV-Prime Unknown Destination
09-00-2B-04-**-**[2]	ua	80-41	LAST
09-00-2B-04-00-00[2]	C0:00:00:20:00:00	80-41	LAST (Group 0)

[1]Routing and Endnode multicasts must be on distinct different values

[2]LAST uses the lower 16 bits as a group code, only Group 0 is assigned a Functional Address

ua=User Assigned, ud=Undefined, na=Not Applicable.
Source: DECnet DNA LAN Address Registry, 4-Oct-1991.

Appendix B

MOP Assignments

(excerpted from MOP V4.0 draft Spec, Appendix A)

This appendix contains the predefined values for various maintenance operation parameters. These values are referenced in the interfaces and in the message definitions. Each parameter has a description to be used in the interface calls and an actual value to be used in protocol messages.

New values are defined on an as needed basis. The list below is as of 4-Sep-1990.

Table B–1: Communication Devices

Value	Name	Device
133	NDI	NDIS Driver on MS–DOS
134	ND2	NDIS Driver on OS/2
135	TRN	DEQRA token ring (802.5) comm link
138	PNT	PROnet-4/16 (802.5) comm link

Table B–2: Data Link Type values

Value	Meaning
1	CSMA/CD
2	DDCMP
3	LAPB (frame level of X.25)
4	HDLC
5	FDDI
6	Token-passing Ring (IEEE 802.5)
7-10	Reserved
11	Token-passing Bus (IEEE 802.4)
12	Z-LAN 4000: Zenith 4 Mbps broadband CSMA/CD LAN

Appendix C

System Specific Data Link Issues

C.1 MS–DOS and OS/2

PATHWORKS for MS–DOS will use the current DEC DLL interface. This allows use of the existing V3.0 & V4.0 software without change. The current DLLNDIS driver should be used as the development base.

PW OS/2 will use a modified DEC DLL interface. The OS/2 V1.1 DNP must be modified to use the new DLL interface that can accommodate the larger frame header. A new DLLMAC driver will be developed.

The driver should startup as it does for Ethernet. The initial parameter source should be DECPARM.DAT or PROTOCOL.INI as appropriate. Any new parameters required can be staticly defaulted at this time. Any unnecessary or inapplicable parameters can be ignored without error.

DNP was changed to support the new data link types, parameters, and counters. However, packet operation did not change. NCP and NML were enhanced.

C.2 VMS

For VMS the existing DECnet Ethernet driver interface will be used. VMS V5.4-3 DECnet components have been modified to allow the use of a TRA0: device with the mnemonic of TRN-0.

The driver should startup as it does for Ethernet. A VMS Ethernet driver is not aware of DECnet databases and can run without installation of DECnet on the system. The DECnet physical (or Station) address value is acquired from the STARTMODE command when DECnet starts the driver. Multicast addresses are learned as the active software opens data link ports.

Token Ring Line characteristics and counters will be supported by the driver and NCP. However, Circuit level characteristics will be unchanged and may reflect Ethernet values. VMS DECnet currently has no mechanism to support driver level network event recording. LAT and LAST support require private internal driver interfaces.

Appendix D

Known Adapter Implementation Limitations

The following are some known limitations in the design or implementation of Token Ring data link devices. The sources of this information are the Technical References or direct communication. For more details, contact the manufacturer.

1. Individual Station Address: All token ring chips allow the setting of the 6 byte station address to an arbitrary value or the unique ROM value. With the following exceptions:
   a. Universal Address Restrictions: The first revisions of the Texas Instruments 4Mbps TMS380 did not allow setting the station address with the Universal/Local bit cleared (eg: Universal). This was fixed in later versions.
   b. Local Administered Address Restrictions: Most early IBM 4Mbps adapters and the first generation TI chip sets compare locally administered addresses differently than just a flat 6 byte compare. The upper two bytes are compared separately, and if the local station address is using the values 40:00 and 7F:FF, the content of the received frame's upper 2 bytes are ignored. Only the lower 4 bytes will be significant on the reception comparison.

      This means that values of 40:00:xx:xx:xx:xx or 7F:FF:xx:xx:xx:xx as station addresses should be avoided, and/or that local administration should insure that all stations on the same ring have unique addresses in the lower 4 bytes. This problem does not exist in IBM 16Mbps adapters or the Second Generation TMS380 chip set.
2. Group Address Value: All current token ring implementations only support the setting of one Group address value for reception of packets on the ring. The later revs of TI chips allow all 6 bytes to be set. All current IBM adapters and early TI 4Mbps chips only allow the lower 4 bytes to be set and the upper 2 bytes to be fixed at C0:00, a locally administered group address. IBM and TI 4Mbps adapters also matched local administered group address according the method mentioned above for station addresses.
3. Functional Address: All token ring implementations provide for setting all 31 bits of a functional address mask.
4. Copy All Packets: All current token ring implementations do not support a Read/Copy all packets mode (also known as "promiscuous mode") in standard products. TI does have such a microcode option available for OEM licensing. The IBM Trace and Performance Adapter also has special modes for performance monitoring, but cannot originate normal traffic while tracing. It also sends out special IBM protocol frames to indicate its presence.

5. Maximum Buffer Size: The original IBM PC Network Adapter (PC bus, 4KB buffer, 4Mbps) can only receive data frames up to 2042 bytes in size. [IBM LAN Tech Ref, page 2-38.]

Appendix E

Data Link Counters, Interrupts, and Statuses

The following information was excerpted from the referenced documents and is here for cross reference information only.

E.1 IEEE 802.5-1989 Spec, Statistics

Page 36, Sect 3.3.2.10, 3.3.2.12 Isolating and Non-isolating Error Counts
Page 42, Sect 3.8 Counters
Page 76, Sect 6.3.2.3 Statistics

Isolating:

- Line Error - token in data frame, or FCS error
- Internal Error - recoverable internal error
- Burst Error - lack of transistions on wire
- A/C Error - error with active and standby monitor frames
- Abort Delimiter - count Abort delimiters sent
- Private Error - vendor specific

Non-Isolating:

- Lost Frame Error - Failure to see frame return within timer
- Receive Congestion - receive buffer unavailable
- Frame Copied - frame received with copied bit already set
- Frequency Error - bit freq is outside of expectations
- Token Error - number of replacement Tokens generated
- Private Error - vendor specific

E.2 IBM Arch, pages 5-19, 5-20

Each is one byte long.

Isolating:

- Line Error
- Internal Error
- Burst Error
- A/C Error

Abort Delimiter Transmitted
reserved

Non-Isolating:

Lost Frame Error
Receiver Congestion
Frame Copied Error
Frequency Error
Token Error
reserved

E.3 NDIS V2.0.1 802.5 Media Specific Statistics

NDIS Appendix D (each are 32 bits long)

FCS or Code violations
Receive Error mask

Receiver congestion
Frame copied error

Burst errors
A/C errors
Abort Frames Sent
Lost Frames Sent
Receive buffer unavailable
Frame copied errors
Frequency errors
Active monitor regenerated
reserved
reserved
reserved
Transmit Error mask

Transmit underrun
Line error
Abort delimiter
Lost Frame
Token error

Number of underruns

E.4 TI Chip set MAC Soft Error Counters

TMS380 UG, page 4-112, fig 4-30 (each 8 bits)

Line error (1 byte)
reserved
Burst error
ARI/FCI error
reserved
reserved

Lost Frames
Receive Congestion
Frame Copied
reserved
Token Error
reserved
DMA Bus
DMA Parity

E.5 Status conditions

Table E-1: Status Interrupts from Chip Sets:

Bit LSB=0	Meaning
15	Signal Loss (Loss of data at receiver)
14	Hard Error (Transmitting or Receiving Beacons)
13	Soft Error (Soft error Report MAC frame sent)
12	Transmit Beacon (Transmitting Beacons)
11	Lobe Wire Fault (open or short in lobe - Adapter closed)
10	Auto-Removal error (Failed beacon process- Adapter closed)
9	reserved
8	Remove Received (MAC requested remove- Adapter closed)
7	Counter Overflow (Counter reached 255)
6	Single Station (No other stations on ring/lobe)
5	Ring Recovery (Monitor contention in progress)
4-0	reserved

- IBM LAN Tech Ref: page B-47, Appendix B, Return Codes, Token-Ring Network Status for All CCBs
- TI TMS380 UG, page 4-63, Table 4-13, RING_STATUS Field Bit Functions
- NDIS V2.0.1, page 5-22, Ring Status

NOTE: Interrupts for these status changes are controlled by the OPEN command option bits 14=Disable Hard Errors, and 13=Disable Soft Errors.

Table E–2: Ring Open Failure Codes:

Value	Description
1	Function Failure - unable loop data on lobe
2	Signal Loss - no data at receiver
5	Timeout - insertion did not complete in time
6	Ring Failure - timeout during initial purge
7	Beaconing - Beacon received
8	Duplicate Node - another station detected
9	Request Init - RPS present, and failed to respond
10	Remove Received - MAC Remove Frame
14	No Monitor Detected - (IBM)
15	Monitor contention failed for RPL (IBM)

- TI TMS380 Page 4-81, Table 4-20
- IBM LAN TR, Page B-49-50,

Appendix F

The Bit Order Problem

F.1 Abstract

Protocols running on both Ethernet and Token Ring LANs will encounter problems communicating with each other via a cross-media bridge for a number of reasons. One reason, which is particularly difficult to fix, is address representation in upper level protocols.

The problem is not immediately obvious to the casual observer, and this appendix attempts to explain it clearly, so that we can get on to attempting to fix it.

F.2 Background

For various reasons, Token Ring MAC level addresses are bit reversed within the bytes of the source and destination frame addresses when viewed in memory. This is easily seen by how it changes the meaning of the Group (or multicast) bit, and the Local/Universal bit in this diagram.

```
          First byte memory layouts:
+-----+-----+----+----+----+----+----+----+
| G/I | L/U |    |    |    |    |    |    |  802.5 Token Ring
+-----+-----+----+----+----+----+----+----+
   0     1    2    3    4    5    6    7     IBM Big-endian bit nomenclature
   |
   +----------------------> 802.5 Wire transmission order

+---+---+---+---+---+---+-----+-----+
|   |   |   |   |   |   | L/U | G/I |  802.3 Ethernet
+---+---+---+---+---+---+-----+-----+
  7   6   5   4   3   2    1     0     Little-endian bit nomenclature
                                 |
       <-------------------------+ Ethernet wire transmission order
```

The remaining 5 bytes of the 48 bit address are likewise inverted between the media. However, it is not true for remaining values in the message frame.

Address blocks are assigned by a standards process to vendors in wire bit order. It is the responsibility of the vendor to implement their values correctly on all 802 media they support.

So, for example, the following are some known address block assignments and how they would be represented if the bytes in memory were printed in hex.

Company	802.3 order	802.5 order
Novell	00-00-1B	00:00:D8
IBM	08-00-5A	10:00:5A
Digital	08-00-2B	10:00:D4
Digital Ph4	AA-00-04	55:00:20

[Nomenclature note: from now on, when I speak of a value in little-endian, or 802.3 order, I will use dashes "-" between the bytes. Big-endian or 802.5 order, I will use colons ":".]

Now, it turns out that the wire order is basically not an issue to worry about from the programmer's viewpoint. He moves data to and from the network in bytes. However the value of the bits when collected into bytes on each network is different. This only becomes a problem when data is moved from one media to another, without conversion. When a bridge is put in place this happens in a critical way. Certainly, one of the bridge's functions would be to maintain the address of any forwarded packet in the correct order, therefore it would reverse the value of address and pass the rest of the data without change.

Let's discuss a simple protocol technique used by some popular network systems. Say that you want to start a conversation with another node, but you do not know their MAC level address. A typical technique, would be to broadcast a query message and hope the target system would receive the message and respond.

Figure F–1: Simple Query Example 1

```
Node: N1                            Node: N2
------------------------            ----------------------------
Query where is "TARGET"?
To: Broadcast From: N1    ---->
                                    Response "TARGET" is here
                          <----     To: N1 From: N2
Session Initiate to TARGET
From: N1  To: N2          ---->
```

Now, this exchange will work well on either LAN, especially if the responding node (N2) picks up the address from the MAC header. But many protocols, pass this information in a data field of the message. Like the following, where node N2 sends tells his data link to use the value of the Source Node data field as the address to send the response message to and does not take into account what kind of media it is on, or where the address came from.

Figure F–2: Simple Query Example 2

```
Node: N1                                Node: N2
------------------------                ---------------------------
Query where is "TARGET"?
To: Broadcast From: N1
Source Node: N1
Dest Node: ??                  ---->
                                        Response "TARGET" is here
                                        To: N1 From: N2
                                        Source Node: N2
                               <----    Dest Node: N1

Session Initiate to TARGET
From: N1  To: N2
Source Node: N1
Dest Node: N2                  ---->
```

So far so, good. Now let's walk through what happens if we attempt to implement this protocol via a cross media bridge. And lets do it with numbers so that we see what happens.

For simplicity, we will shorten the 6 byte MAC layer addresses to 3 bytes. In my examples, "To:" is the MAC Destination address, and "From:" is the MAC Source Address.

Figure F–3: Simple Query Example 3

```
Assignments:  N1 = AA-00-04, N2 = 08-0C-5A

Topology:

    +-------+   Ethernet  +---------+   Token Ring +-------+
    |  N1   |-------------| Bridge  |--------------|  N2   |
    +-------+             +---------+              +-------+

Node: N1 on Ethernet
-------------------------
Query where is "TARGET"?
To: FF-FF-FF From: AA-00-04
Source Node: AA-00-04
Dest Node: 00-00-00         ---->

        Bridge: Ethernet port to Token port
        -------------------------
        Query where is "TARGET"?
        To: FF:FF:FF From: 55:00:20
        Source Node: AA:00:04
        Dest Node: 00:00:00      ---->

                                 Node: N2  on Token Ring
                                 ----------------------------
                                 Response "TARGET" is here
                                 To: AA:00:04  From: 10:00:5A
                                 Source Node: 10:00:5A
                         <----   Dest Node: AA:00:04

                 (Note destination address is a multicast!)

                 Bridge: Token port to Ethernet
                 ----------------------------
                 Response "TARGET" is here
                 To: 55-00-20  From: 08-00-5A
                 Source Node: 10-00-5A
         <----   Dest Node: AA-00-04
```

The Response packet falls on the floor, because there is no node 55-00-20 on the Ethernet! And therefore the connection cannot be established.

The moral of this story is that bridges will take care of the MAC header, but if values are passed in data, they have to be fixed somehow too.

F.3 The crux of the problem

Since the MAC addresses are different, the bridge must reverse them at the MAC layer. The next problem is any time an upper layer uses the MAC address value in a message. That message may potentially go to a different LAN media, and the protocol must be aware of the consequences for this to work.

F.3.1 Possibility 1

It is tempting to build into the bridge, knowledge of the protocol format (e.g.: the offsets of the Source Node and Dest Node fields) and have the bridge bit reverse them as the messages were forwarded. But this has two major long term problems:

1. The bridge becomes protocol dependent.
2. The fixups become a performance burden.

Bridges have been traditionally protocol independent, and this is one of their major features in the LAN market place. Ethernet LAN bridges do not care if you run TCP/IP, NetWare, XNS, DECnet, NETBEUI, or even SNA protocols. However, once you place a protocol dependent bridge, you have locked into supporting only a subset of possible protocols. You lose independence from protocol and version change. Your bridges now become dependent on your vendor's support, and his ability to match his fix-ups against your protocol suites.

Bridges also gain many of their performance features from the simplicity of forwarding packets while only examining the MAC header. Sometimes this is even done with fixed hardware or logic. Introducing the need to parse the packet datastream and modify message fields in variable locations in real time will only introduce a new processing bottleneck in bridges.

F.3.2 Possibility 2

In order to keep the problem simple, a bridge vendor may be tempted to not invert the addresses between the media. However, this is a violation of 802 standards.

IEEE 802 vendor address blocks are assigned across all media. They are assigned as a string of bits in wire order. There are no media specific blocks for just Token Ring or Ethernet.

Address assignments in the globally administered space are for the sole use of that vendor. If another vendor started using addresses in that space, multiple address conflicts would come into play.

Also, addresses on either media could become invalid addresses on the other media. Consider the following:

```
+---+---+---+---+---+---+---+---+
|G/I|L/U|   |   |   |   |   |   |  802.5 Token Ring
+---+---+---+---+---+---+---+---+
  |   |   |   |   |   |   |   |
+---+---+---+---+---+---+---+---+
|   |   |   |   |   |   |L/U|G/I|  802.3 Ethernet
+---+---+---+---+---+---+---+---+
  7   6   5   4   3   2   1   0
```

Token Ring addresses with the low order bit set could become Ethernet multicast addresses or if bit 1 clear infringe on someone's Universal address space. Multicast source addresses are illegal on Ethernet. Conversely, Ethernet addresses with the high order bit set become multicast addresses on the Token Ring, or indicate a source routing field present (perhaps erroneously if the bridge is a transparent bridge). Likewise Global addresses could become local or vice versa, and vendor's address spaces would be violated.

The impact is that anyone with a Ethernet address that begins with a value equal to or greater than 8x, sends packets that become Multicasts or bogus Source Routing frames (depending on whether the address is in the destination or source fields) Likewise for Token Ring vendors that respect their Universally assigned values, if their first byte is odd it will become a multicast on the Ethernet.

Vendors pay fees to IEEE for allocations address spaces so that their products have unique values and not conflict with other products. By not preserving the value of the MAC addresses, such a bridge will cause conflicts between vendor's products whose addresses are bit reversed from each other. This violates the principal of the standard for unique station addresses.

Even if this filtering was done in a protocol specific manner, it would run into problems with systems that support multiple protocol stacks. Given the growth of multiprotocol clients and servers in both DOS and OS/2 markets, this approach is also short-sighted.

F.3.3 Possibility 3

One way to avoid the issue is to not bridge at all. Most of these problems do not become an issue if all connections between Token Ring and Ethernet media are done via Routers (potentially multiprotocol implementations).

In a router, data link differences disappear for the most part, as the MAC frame is stripped and the data link layer deals with the problems. Protocol suites that have network layers, usually come to grips with this problem and can convert addresses or what have you, as they forward packets between the data links. There are many vendors of multiprotocol routers on the market, so this is a fairly well understood technology.

Unfortunately, due to other implementation issue on token rings, some protocols can only be supported by router implementations. Protocols such as Apple's TokenTalk require remapping of multicast addresses into functional addresses and vice versa. It's also not clear how ISO ES-IS multicasts will be converted to functional addresses, but that problem is simpler.

The tradeoff towards routers is often an issue of price performance. But given the improvements in the market, and the other benefits or routers, it would be negligent to not evaluate routers closely.

F.3.4 Better solutions

The best way to handle the problem is for the protocol designer to recognize the issue and deal with it in the node software itself. If upper level protocols were consistent in representing an address value in a single format on both networks, then bridges would not have to concern themselves with this problem. This issue clearly sets up complexity and performance issues for bridges the future.

One approach would be for the node to be cognizant of its local bit order or that of the remote address it is about to use, and perform the necessary conversion (if any) at that time.

Another approach (the one we prefer) is for the device driver to just hide the addressing difference at its interface level. If all addresses are alike, then there is no problem.

This is the approach specified by the IEEE Standard 802-1990 in Section 5.4. 802.5 implementation must bit reverse the MAC address values when communicating with the LLC upper level. So if a token ring device drivers reversed the values of its own ROM address, and the MAC layer addresses in data frames sent and received, the upper level software would not have to worry about this problem.

Clearly, problems due to protocol designer short-sightedness will not go away overnight. There are possibilities beyond what I have described here (and other problems and details as well). It may be that massive software updates are in order. But clearly this issue must be addressed by the vendors of the protocols involved, keeping in mind that concurrent multiple protocols stacks will be the mode of the future.

F.4 Affected Protocols

Two significant protocols suffer from the bit order problem:

1. TCP/IP
2. Novell NetWare

F.4.1 TCP/IP

The original implementations of TCP/IP on Token Ring used the reversed bit-order for MAC addresses in ARP and RARP messages. When the first bridges were introduced, this problem was discovered. Instead of fixing the installed base of software, it has become an unwritten requirement that Token Ring to Ethernet TCP/IP bridge support reverse the bits of MAC addresses in ARP messages.

Since interoperability problems would result if any TCP/IP vendor would correct the bit order problem in their implementation, it is highly unlikely to change at this time. In our opinion, it would be better to stabilize this state, by making it a formal requirement.

F.4.2 NetWare

The Novell NetWare SPX/IPX protocols have protocol data fields in each frame with the MAC addresses of the source and destination nodes. While these fields are at fixed locations, the Service Advertisement Protocol (SAP) also freely exchanges MAC addresses in variable locations in its messages. Therefore, it is extremely difficult to "fixup" the entire protocol suite.

Novell has recently become aware of this problem and is asking its Token Ring adapter vendors to correct their NetWare drivers in the near future. Unfortunately, installing the new bit order will require that all stations migrate at once. A major inconvenience but it would remove all future configuration constraints.

Most NetWare users that have mixed Token Ring and Ethernet configurations are using NetWare Servers to interconnect. Novell calls this feature "Bridging" but it is really *Routing*. When IPX networks are connected together with Routers, LAN addresses are not used on the wrong media, and the bit-order problem does not cause non-operation (though the addresses still have the wrong value).

Appendix G

Known Protocol Interoperability Issues

During the development of this specification, the following characteristics of major protocols became known. They are listed here as summary information. This appendix is the not the authority on the technical details of these problems, that rests with the owners of these protocols specifications.

These issues are primarily of interest to those that wish to attempt to bridge these protocols from Token Ring to other 802 LAN media.

- TCP/IP ARP - bit order & format mapping
- NetWare IPX - bit order & format compatibility
- AppleTalk ZIP - FA mapping vs Mcast mapping
- DECnet Routing - FA to Mcast mapping & format mapping
- ISO ES-IS - FA to Mcast mapping

G.1 TCP/IP

TCP/IP packets have traditionally been in Ethernet data link format. The assignment of SAP 06 is not used by Internet compliant implementations. Instead the mapping described in RFC 1042 and 1103 are used on 802.2 only media.

The ARP request message and response carry media address values and suffer from the bit order problem (described in Appendix F).

G.2 NetWare IPX

NetWare IPX packets contain media specific address values and therefore suffer from the bit order problem (described in Appendix F). Novell will be issuing new drivers and patches to solve this problem.

NetWare also offers users several different packet formats on Ethernet. The default "802.3" format is not usable in multiprotocol Ethernet LANs, as it does not contain a proper 802.2 LLC header as required of any 802 compliant implementation. Users are recommended to use the Ethernet format or the newer "802.2" format.

G.3 AppleTalk

The AppleTalk Phase 2 Zone Information Protocol (ZIP) uses a set of 253 multicasts on Ethernet, and 19 Functional Addresses on Token Ring. For a given Zone ID, the value of the Zone ID is run through a hashing algorithm (see Inside AppleTalk) and a multicast or functional address is selected, modulo table size, for use by ZIP within that Zone.

Given this hashing and folding algorithm, it is impossible to translate functional addresses into their corresponding multicast addresses. It seems that users must use AppleTalk routers to connect a zone across media type or partition zones to have homogeneous LAN media members. The ZIP protocol also has an All Zones multicast which uses a functional address on Token Ring. This would have to be translated between media.

AppleTalk also violates the invertability rules of RFC 1042. They use a zero OUI in their SNAP SAP frames, and expect the frame to continue to be in 802.2 format even on Ethernet media.

G.4 DECnet Routing

As described in this specification, DECnet uses SNAP SAP frames using RFC 1042 mapping on the Token Ring.

DECnet maps several universally administered multicasts into functional addresses for station use on the Token Ring. In the presence of other protocols using those functional addresses, DECnet stations must be able to discard the packets, using the DSAP and SNAP PID as criteria. And any translating bridges must be able to demultiplex and map the functional address packets likewise.

DECnet Phase IV-Prime maps the new multicast assignments on to the same functional addresses used for Phase IV-Classic. This makes it impossible to reverse map token ring frames back onto Ethernet without parsing the message type.

G.5 ISO Network Layer

The ISO standard network layer uses four distinct universally administered addresses. Since these cannot be received by current hardware on the market, alternate functional addresses have been assigned (See Table A–2). Any cross-media bridge must be able to translate these multicasts to functional addresses and vice versa.

Appendix H

Format Conversion in Bridges

Bridges that attempt to cross media types, in particular, from Token Ring to Ethernet, will run into a number of problems due to differences in those media, and how protocol implementations have been developed.

This appendix briefly touches the problems that arise with typical LAN network environments.

H.1 Data Link Frame Format Conversion

Many products exist today on Ethernet LANs that use the Ethernet data link frame format (eg: DECnet, TCP/IP, LAT). This frame format is not supportable directly on 802.5 LANs. A translation format has been worked out in open standards groups and documented in IEEE 802-1990 and IETF RFCs 1042, and 1103.

Bridges that cross media types must adhere to those specifications to insure that open system vendors' products will interoperate in enterprise networks.

The specification involves the mapping of the Ethernet Protocol Type into a SNAP SAP Unnumbered Information frame, and embedding a vendor id (OUI) as well as the original protocol type value in the first five bytes of data. The OUI value should be non-zero, for new protocols that use 802 packet formats on any media, and must be the vendor address block administered by the standards authority to the protocol vendor. The non-zero OUI is reserved for mapped Ethernet protocols only.

A frame must not be originated on an Ethernet/802.3 LAN with a zero OUI, as translating bridges will convert it into its corresponding Ethernet format if it is forwarded onto an Ethernet media. This rule allows bridges to make format conversions based on the information in the frame on a per-packet basis. (as opposed to attempting to remember conversion rules in a limited or difficult to maintain table)

H.2 Multicast to Functional Address Mapping

Because of the limitations of the current Token Ring data link chip implementations, Universal Group Address reception is not available. This necessitates mapping of group addresses on to functional addresses.

Further complications arise because the limited space of functional addresses forces different vendors to use a given functional address value that other vendors are already using. Because of this, it is highly recommended that any functional address mapping also take into account the protocol type as expressed in the DSAP and SNAP PID. Failure to do this will run into problems when a Token Ring Functional Address is used by several protocols, and has to be forwarded onto a LAN media that uses the Universal assignment. That frame may not be translated into the correct address, and the protocol will not operate correctly. (eg: AppleTalk and ISO ES-IS)

H.3 Source Routing to Transparent Spanning Tree

Source Routing is not implemented on Ethernet/802.3 LANs. Token Ring protocols that depend on its use must have a gateway to remember and restore this information, as packets leave and come back.

The development of the Source Routing Transparent (SRT) bridge standard will define how a bridge may implement both bridging techniques in the same network box. However, it does not provide this gateway type function. Instead it allows the non-source routing protocols to span the Token Ring network by using transparent bridging techniques.

H.4 Summary

Because of the differences and the transformations that arise, protocol designers must plan ahead to insure that their protocols will interoperate in a mixed media environment. Particular attention must be paid to; frame format usage and reversible translations, Group Address usage, and Source Routing expectations.

Appendix I

Operation of DECnet over IBM 8209 LAN Bridges

In general, Digital Equipment Corporation cannot be responsible for the operation of its protocols over equipment that has not been fully tested by Digital or is subject to change by the original manufacturer. Digital *does not certify or support* the IBM 8209 or in general recommend its use.

This appendix documents Digital's best attempt to configure an IBM 8209 LAN Bridge. Other configurations may not interoperate.

In the following sections, 8209/SE will be used to denote an IBM 8209 LAN Bridge equipped with the Standard Ethernet Attachment Module and 8209/EE will be to denote an IBM 8209 LAN Bridge equipped with an Enhanced Ethernet Attachment Module.

I.1 Workable Configurations

I.1.1 8209/EE in parallel with a Router (Preferred)

As shown in Figure I–1, the preferred configuration, given that 8209's are present, is to use a multiprotocol router to handle DECnet traffic between 802.5/Token Ring and Ethernet/802.3 LANs. The 8209 is used to bridge all other protocols that cannot be routed.

Figure I–1: 8209/EE in parallel with a router

```
      oooooo                                  TT
    o        o      +--------------+          ||
   o          o-----| IBM 8209/EE  |-----------||
  o            o    +--------------+          ||
  o            o                              ||
  o            o    +----------------+        ||
   o          o-----| Multiprotocol  |--------||
    o        o      |     Router     |        ||
      oooooo        +----------------+        TT

 802.5/Token Ring                           Ethernet
```

I.1.1.1 8209 LAN Bridge Hardware Requirements

This configuration requires the IBM 8209 to be configured with the Enhanced Ethernet Attachment Module (IBM Part # 74F5156)

I.1.1.2 8209 LAN Bridge Management Software Requirements

Version 3.0 or later of the IBM 8209 LAN Bridge Utility Program or Version 1.1 or later of the IBM LAN Network Manager Program are required to correctly configure the 8209/EE.

I.1.1.3 8209/EE Bridge Forwarding Parameters

Table I–1 shows the 8209 forwarding parameter settings to allow it to bridge non-routable DNA-compliant protocols. These settings are the ones that one would use if only DNA protocols were present, however the presence of other non-DNA protocols may call for different values than the ones shown in Table I–1 and the operation of DNA protocols may or may not be affected.

NOTE: Consult the "IBM 8209 LAN Bridge Attachment Module for Ethernet and IEEE 802.3 LANs" (GA27-3891) for details concerning other 8209 parameters.

Table I–1: 8209/EE Bridge Forwarding Parameters

Parameter	Value
Automatic Mode Selection	0 (Disabled)
Mode Priority	1 (Ethernet)
Forward LLC Traffic (Mode 1)	0 (No)
TCP/IP Address Conversion	1 (Enabled)
Dual Mode Multicast Conversion	0 (Disabled)
Use General Broadcast Frames	0 (Disabled)
Broadcast Address Conversion	1 (Enabled)
IPX Support	0 (Disabled)

In this configuration, the 8209 must be configured to filter DECnet traffic. This can be accomplished by defining the proper filter ranges in the 8209. Table I–2 shows the proper ranges for the filtering DECnet traffic. LAT and other non-routable traffic will still be forwarded by the 8209/EE.

Other routable protocols that are being forwarded by the multiprotocol router should also be filtered by the 8209.

NOTE: Since the 8209 filter's cannot check the OUI field of an 802.5 frame as well as the Protocol Type, the following settings may also cause other protocols than DECnet to be filtered in the 802.5 to 802.3/Ethernet direction.

Table I–2: 8209/EE Filter Ranges for DECnet Traffic

	802.5/Token Ring	802.3/Ethernet	
Filter Offset	6	0	(0 - 100)
Range 1 low	0	0	(0 - FFFF)
Range 1 high	6002	6002	(0 - FFFF)
Range 2 low	6004	6004	(0 - FFFF)
Range 2 high	FFFF	FFFF	(0 - FFFF)

I.1.1.4 Default Address Mappings for Non-Routable DECnet Protocols

Table I–3 shows the how the 8209's address mapping table must be configured to bridge non-routable DNA protocols in this configuration.

NOTE: These are the default mappings. It may be necessary to use other functional address values on the 802.5/Token Ring if other protocols are all ready using the default values.

Table I–3: 8209/EE Address Mapping

802.5/Token Ring	802.3/Ethernet	Description
C0:00:40:00:00:00	AB-00-00-01-00-00	MOP Dump/Load
C0:00:20:00:00:00	AB-00-00-02-00-00	MOP Console
C0:00:02:00:00:00	09-00-2B-00-00-0F	LAT Advertisement
C0:00:01:00:00:00	09-00-2B-02-01-04	LAT Solicit
C0:00:00:40:00:00	09-00-2B-02-01-07	LAT Xwin Service Solicit
C0:00:00:20:00:00	09-00-2B-04-00-00[1]	LAST (Group 0)
C0:00:00:10:00:00	CF-00-00-00-00-00	Loopback
user defined	AB-00-04-01-**-**[2]	SCA

[1]LAST uses the lower 16 bits as a group code

[2]VAX Clusters uses the lower 16 bits as the cluster ID

I.1.2 8209/SE in parallel with a Router

As shown in Figure I–2, another possible configuration is to use a multiprotocol router to handle DECnet traffic between 802.5/Token Ring and Ethernet/802.3 LANs and an 8209/SE to handle other protocols such as NETBIOS.

Figure I–2: 8209/SE in parallel with a router

```
      oooooo                                  TT
    o        o      +-------------+          ||
  o            o-----| IBM 8209/SE |----- ||
 o              o    +-------------+          ||
 o              o                             ||
 o              o    +---------------+       ||
  o            o-----| Multiprotocol |---||
    o        o       |    Router     |       ||
      oooooo         +---------------+       TT

 802.5/Token Ring                          Ethernet
```

I.1.2.1 8209 LAN Bridge Hardware Requirements

This configuration requires the IBM 8209 LAN Bridge to be configured with the Standard Ethernet Attachment Module (IBM Part # 55F4785).

I.1.2.2 8209 LAN Bridge Management Software Requirements

Version 1.0 or later of the IBM 8209 LAN Bridge Utility Program is required to configure the 8209/SE.

I.1.2.3 8209 Bridge Forwarding Parameters

Table I–4 shows the required 8209 Bridge forwarding parameter settings. These settings are the ones that one would use if only DNA protocols were present, however the presence of other non-DNA protocols may call for different values than the ones shown in Table I–4 and the operation of DNA protocols may or may not be affected.

NOTE: Consult the "IBM 8209 LAN Bridge Customer Information" (SA21-9994) for details concerning other 8209 parameters.

Table I–4: 8209/SE Bridge Forwarding Parameters

Parameter	Value
Automatic Mode Selection	0 (Disabled)
Mode Priority	1(Ethernet)
Forward LLC Traffic (Mode 1)	No
TCP/IP Address Conversion	1 (Enabled)

Table I–4 (Cont.): 8209/SE Bridge Forwarding Parameters

Parameter	Value
Dual Mode Multicast Conversion	0 (Disabled)

In this configuration, the 8209 must be configured to filter all DNA protocols. This can be accomplished by defining the proper filter ranges in the 8209. Table I–5 shows the proper ranges for filtering DNA traffic.

Other routable protocols that are being forwarded by the multiprotocol router should also be filtered by the 8209.

NOTE: Since the 8209 filter's cannot check the OUI field of an 802.5 frame as well as the Protocol Type, the following settings may also cause protocols other than DNA protocols to be filtered in the 802.5 to 802.3/Ethernet direction.

Table I–5: 8209/SE Filter Ranges for DNA Protocols

	802.5/Token Ring	802.3/Ethernet	
Filter Offset	6	0	(0 - 100)
Range 1 low	0	0	(0 - FFFF)
Range 1 high	5FFF	5FFF	(0 - FFFF)
Range 2 low	6008	6008	(0 - FFFF)
Range 2 high	FFFF	FFFF	(0 - FFFF)

I.1.2.4 Limitations

Only protocols that use DECnet as a transport can be passed by the router. The 8209/SE does not contain the address mapping functionality required to correctly bridge the non-routable DNA protocols. Therefore, the following DNA protocols are **not supported** between 802.5 /Token Ring and 802.3/Ethernet:

- LAT
- MOP
- SCA
- LAST

Appendix J

Example Packets

The following packets are provided as examples of correctly formed packets on a Token Ring.

Example J–1: MOP SYSID

```
SUMMARY  Delta T  Size     Destination   Source        Summary
 16502              56  MOP Console IRON          MOP RC System ID  Receipt=0

MOP:  ----- Maintenance Operation Remote Console Protocol -----
MOP:
MOP:  Data length = 32
MOP:  Code = 7  (System ID)
MOP:  Reserved        = 0
MOP:  Receipt Number = 0
MOP:
MOP:  Information Length =  3, Type = 1 (Maintenance Version)
MOP:     Version Number  = 03
MOP:     ECO Number      = 00
MOP:     User ECO Number = 00
MOP:
MOP:  Information Length =  2, Type = 2 (Functions)
MOP:     Functions Mask (byte 0)  = 4B
MOP:                   0... .... = not console carrier reservation
MOP:                   .1.. .... = data link counters
MOP:                   ..0. .... = not console carrier
MOP:                   ...0 .... = not boot
MOP:                   .... 1... = multi-block loader
MOP:                   .... .0.. = not primary loader
MOP:                   .... ..1. = dump
MOP:                   .... ...1 = loop
MOP:     Functions Mask (byte 1)  = 00
MOP:                   0000 0000 = unused bits
MOP:
MOP:  Information Length =  6, Type = 7 (Hardware Address)
MOP:     Hardware Address = 10005A7512F8
MOP:
MOP:  Information Length =  1, Type = 100 (Communication Device)
MOP:     Communication Device = NDIS on OS/2
MOP:
MOP:  Information Length =  1, Type = 300 (System Processor)
MOP:     System Processor = IBM PC (generic)
```

Example J–1 (continued on next page)

Example J-1 (Cont.): MOP SYSID

```
MOP:

ADDR  HEX                                               ASCII
0000  18 40 C0 00 20 00 00 00  55 00 20 00 48 BB AA AA  .@......U. .H...
0010  03 00 00 00 60 02 20 00  07 00 00 00 01 00 03 03  ....`. .........
0020  00 00 02 00 02 4B 00 07  00 06 10 00 5A 75 12 F8  .....K......Zu..
0030  64 00 01 86 2C 01 01 09                           d...,...

- - - - - - - - - - - - - - - - - - Frame 16647 - - - - - - - - - - - - - - - - - -

SUMMARY  Delta T  Size     Destination   Source        Summary
 16647   19.381     56  MOP Console PKIPSI        MOP RC System ID  Receipt=0

MOP:  ----- Maintenance Operation Remote Console Protocol -----
MOP:
MOP:  Data length = 32
MOP:  Code = 7  (System ID)
MOP:  Reserved       = 0
MOP:  Receipt Number = 0
MOP:
MOP:  Information Length =  3, Type = 1 (Maintenance Version)
MOP:     Version Number  = 03
MOP:     ECO Number      = 00
MOP:     User ECO Number = 00
MOP:
MOP:  Information Length =  2, Type = 2 (Functions)
MOP:     Functions Mask (byte 0)  = 41
MOP:                      0... .... = not console carrier reservation
MOP:                      .1.. .... = data link counters
MOP:                      ..0. .... = not console carrier
MOP:                      ...0 .... = not boot
MOP:                      .... 0... = not multi-block loader
MOP:                      .... .0.. = not primary loader
MOP:                      .... ..0. = not dump
MOP:                      .... ...1 = loop
MOP:     Functions Mask (byte 1)  = 00
MOP:                      0000 0000 = unused bits
MOP:
MOP:  Information Length =  6, Type = 7 (Hardware Address)
MOP:     Hardware Address = 42608C3C0E1D
    (Note: 3Com in wrong order!)
MOP:
MOP:  Information Length =  1, Type = 100 (Communication Device)
MOP:     Communication Device = NDIS on MS-DOS
MOP:
MOP:  Information Length =  1, Type = 300 (System Processor)
MOP:     System Processor = IBM PC (generic)
MOP:

ADDR  HEX                                               ASCII
0000  18 40 C0 00 20 00 00 00  55 00 20 00 22 3B AA AA  .@......U. .";..
0010  03 00 00 00 60 02 20 00  07 00 00 00 01 00 03 03  ....`. .........
0020  00 00 02 00 02 41 00 07  00 06 42 60 8C 3C 0E 1D  .....A....B`.<..
0030  64 00 01 85 2C 01 01 09                           d..+,...
```

Example J–2: Endnode Hello Message

```
SUMMARY  Delta T  Size     Destination   Source        Summary
   862    4.868     57  DRP Routers PKIPSI       DRP ENDNODE Hello  S=55.68
 BLKSZ=1498

DRP:  ----- DECNET Routing Protocol -----
DRP:
DRP:  Data length = 33
DRP:  Control Packet Format = 0D
DRP:               0... .... = no padding
DRP:               .000 .... = reserved
DRP:               .... 110. = Ethernet Endnode Hello Message
DRP:               .... ...1 = Control Packet Format
DRP:  Control Packet Type = 06
DRP:  Version Number  = 02
DRP:  ECO Number      = 00
DRP:  User ECO Number = 00
DRP:  ID of Transmitting Node = 55.68, PKIPSI
DRP:       Information = 03
DRP:         0... .... = reserved
DRP:         .0.. .... = not blocking request
DRP:         ..0. .... = multicast traffic accepted
DRP:         ...0 .... = verification ok
DRP:         .... 0... = do not reject
DRP:         .... .0.. = no verification required
DRP:         .... ..11 = endnode
DRP:  Receive Block Size    = 1498
DRP:  Area (reserved)       = 0
DRP:  Verification Seed     = 0000000000000000
DRP:  Neighbor System ID    = 55.283, ERBIUM
DRP:  Hello timer (seconds) = 30
DRP:  MPD (reserved)        = 0
DRP:  [1 bytes of Data to test the circuit]
DRP:

ADDR  HEX                                              ASCII
0000  18 40 C0 00 10 00 00 00  55 00 20 00 22 3B AA AA  .@......U. .";..
0010  03 00 00 00 60 03 21 00  0D 02 00 00 AA 00 04 00  ....`.!.........
0020  44 DC 03 DA 05 00 00 00  00 00 00 00 00 00 AA 00  D...............
0030  04 00 1B DD 1E 00 00 01  AA                       .........
```

Example J–3: Router Hello Message

```
SUMMARY  Delta T  Size     Destination   Source        Summary
  2219    9.747     51 DRP Routers ERBIUM       DRP ROUTER  Hello  S=55.283
 BLKSZ=2044
```

Example J–3 (continued on next page)

Example J–3 (Cont.): Router Hello Message

```
DRP:  ----- DECNET Routing Protocol -----
DRP:
DRP:  Data length = 27
DRP:  Control Packet Format = 0B
DRP:                 0... .... = no padding
DRP:                 .000 .... = reserved
DRP:                 .... 101. = Ethernet Router Hello Message
DRP:                 .... ...1 = Control Packet Format
DRP:  Control Packet Type = 05
DRP:  Version Number   = 02
DRP:  ECO Number       = 00
DRP:  User ECO Number = 00
DRP:  ID of Transmitting Node = 55.283, ERBIUM
DRP:      Information = 02
DRP:        0... .... = reserved
DRP:        .0.. .... = not blocking request
DRP:        ..0. .... = multicast traffic accepted
DRP:        ...0 .... = verification ok
DRP:        .... 0... = do not reject
DRP:        .... .0.. = no verification required
DRP:        .... ..10 = level 1 router
DRP:  Receive Block Size    = 2044
DRP:  Router's priority     = 64
DRP:  Area (reserved)       = 0
DRP:  Hello timer (seconds) = 30
DRP:  MPD (reserved)        = 30
DRP:  E-List length = 8
DRP:  Ethernet Name, reserved = 00000000000000
DRP:  Router/State length = 0
DRP:

ADDR  HEX                                              ASCII
0000  10 40 C0 00 10 00 00 00  55 00 20 00 D8 BB AA AA  .@......U. .....
0010  03 00 00 00 60 03 1B 00  0B 02 00 00 AA 00 04 00  ....`...........
0020  1B DD 02 FC 07 4C 00 1E  00 1E 08 00 00 00 00 00  .....@..........
0030  00 00 00                                          ...
```

Example J–4: NETBIOS Name Claim

```
SUMMARY  Delta T  Size    Destination   Source       Summary
  2929             183  DEC NETBIOS  BARIUM       SNAP Ethernet Type=8040
      (DEC NetBIOS)

SNAP: ----- SNAP Header -----
SNAP:
SNAP: Type = 8040 (DEC NetBIOS)
SNAP: [161 byte(s) of data]
```

Example J–4 (continued on next page)

Example J–4 (Cont.): NETBIOS Name Claim

```
ADDR  HEX                                               ASCII
0000  10 40 C0 00 04 00 00 00  55 00 20 00 08 BB AA AA   @      U     . . .
0010  03 00 00 00 80 40 02 01  01 00 01 00 42 41 52 49     . @ . .     BARI
0020  55 4D 20 20 20 20 20 20  20 20 20 00 4C 41 4E 47  UM           .LANG
0030  52 4F 55 50 20 20 20 20  20 20 20 00 00 00 76 00  ROUP          . . .v
0040  00 FF 53 4D 42 25 00 00  00 00 00 00 00 00 00 00  . .SMB%. . . . . . . . . .
0050  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   . . . . . . . . . . . . . . .
0060  00 11 00 00 1F 00 00 00  00 00 00 00 00 00 E8 03   . . . . . . . . . . . . . . .
0070  00 00 00 00 00 00 00 00  1F 00 56 00 03 00 01 00   . . . . . . . .V. . . . .
0080  00 00 02 00 30 00 5C 4D  41 49 4C 53 4C 4F 54 5C  . . . .0.\MAILSLOT\
0090  4C 41 4E 4D 41 4E 00 01  00 03 00 00 00 01 01 3C  LANMAN. . . . . . . .<
00A0  00 42 41 52 49 55 4D 00  50 43 53 41 20 66 6F 72   BARIUM.PCSA for
00B0  20 4F 53 2F 32 00 00                               OS/2.

- - - - - - - - - - - - - - - - - - Frame 3841 - - - - - - - - - - - - - - - - - -

SUMMARY  Delta T  Size     Destination  Source       Summary
  3841     6 651    204  DEC NETBIOS  BLACK       SNAP Ethernet Type=8040
      (DEC NetBIOS)

SNAP: ----- SNAP Header -----
SNAP:
SNAP: Type = 8040 (DEC NetBIOS)
SNAP: [182 byte(s) of data]

ADDR  HEX                                               ASCII
0000  1C 40 C0 00 04 00 00 00  55 00 20 00 9B 3B AA AA   .@. . . . . .U.  . .;. .
0010  03 00 00 00 80 40 02 01  01 00 01 00 42 4C 41 43  . . . . .@. . . . . . .BLAC
0020  4B 20 20 20 20 20 20 20  20 20 20 00 4C 41 4E 47  K            .LANG
0030  52 4F 55 50 20 20 20 20  20 20 20 00 00 00 8B 00  ROUP          . . . . .
0040  00 FF 53 4D 42 25 00 00  00 00 00 00 00 00 00 00  . .SMB% . .  . . . . .
0050  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  . . . .  . . . . . . . . . . .
0060  00 11 00 00 34 00 00 00  00 00 00 00 00 00 E8 03  . . . .4. . . . . . . . . . .
0070  00 00 00 00 00 00 00 00  34 00 56 00 03 00 01 00  . . . . . . . . .4.V. . . . .
0080  00 00 02 00 45 00 5C 4D  41 49 4C 53 4C 4F 54 5C  . . . .E.\MAILSLOT\
0090  4C 41 4E 4D 41 4E 00 01  00 03 00 00 00 02 00 3C  LANMAN. . . . . . . . . .<
00A0  00 42 4C 41 43 4B 00 44  45 43 20 4C 61 6E 57 4F  .BLACK.DEC LanWO
00B0  52 4B 53 20 66 6F 72 20  4F 53 2F 32 20 31 2E 31  RKS for OS/2 1.1
00C0  20 46 54 20 55 70 64 61  74 65 00 00               FT Update. .
```