CONTROL DATA NETWORK ARCHITECTURE
(CDNA)

ROUTING_M=E_ERS

N L Reddy
Bob Woodruff

07/25/84

Version 1

DCS-C {COMMENT} CPD DOCUMENT CONTROL FORM

TO BE FILLED OUT BY SUBMITTER OF COMMENTS

COMMENTS ON DOCUMENT ENTITLED CDCNET DEVICE INTERFACE ROUTING M-E ERS DCS LOG ID NO. ARH 6264

DOCUMENT TYPE ERS FEATURE ____ PHASE ____

COMMENTS SUBMITTED BY J.L NADING / R.K. SEKHAN NO. PAGES 2 DATE 4/27/84

SECTION ADVANCED SYSTEMS DEV'L (NAM/NF) EXTENSION 3542 MAIL STATION ARH254

CONCURRENCE WITH COMPLETENESS AND CORRECTNESS OF THIS DOCUMENT

☒ I CONCUR {INFORMATIONAL}
☐ I DO NOT CONCUR {COMMENT}

COMMENTS ARE:
☐ WRITTEN BELOW
☒ ATTACHED {NO. PAGES 2 }

FOR NON-CONCURRENCE, COMMENTS MUST BE INCLUDED

COMMENTS:

COMMMENTS AGAINST ROUTING M-E ERS.

O following are my comments against routing ME ERS. M..t of these comments are editorial which i think will help in improving the readability of the document. The document in general looks good.

1. Page 1-2 Section 1.1
   - Routing MEs receive RIDUs from other multi-homed systems rather than networks.

2. Page 2-1 Section 2.1.1
   - Restate the sentence 'At a high level, the routing ME may be seen as _ _ _ ' as 'At a high level, the routing ME may be seen as being responsible for computing the best paths to all networks in the catenet from the local system. These paths are recomputed whenever a network in the catenet becomes active or inactive or becomes congested or uncongested'.

3. Page 2-2 Section 2.1.1
   - Change 'maximum number paths' to 'maximum number of paths'.

4. Page 2-2 Section 2.1.2
   - State that the network cost is divided by 4 when the network congestion status changes from congested to uncongested.

5. Page 2-2 Section 2.1.3
   - Restate 'routing around downed networks' as 'routing around inactive networks'.

6. Page 2-6 Section 2.1.5
   - Restate 'In addition, each row has a flag in it _ _ _' as 'In addition, each row has a flag in it, which is true if the network is directly connected to the system'.

7. Page 2-7 Section 2.1.5
   - Routing ME would have to do a transformation from CDNAs XEROX RIDUs rather than CDNAs XEROX routing tables.

8. Page 2-8 Section 2.1.6
   - Restate 'Therefore, it is important to include the 3A commands' as 'Therefore, it is important to send the 3A commands'.

9. Page 2-9 Section 2.1.7
   - Since R1 does not support true xerox networks, the concept of routing info networks should be deleted.

   - The frequency of RIDU generation is determined by changes in the local network titles/address data store in addition to the other two factors mentioned in the 2nd para.

10. Page 2-11 Section 2.2
   - In the description for 'R', mention that routing ME sends and receives RIDUs to/from other systems via generic 3A.

1. Page 2-12 Section 2.3
- Change 'buffer management timer' to 'buffer management, timer'.

2. Page 2-12 Section 2.3.1
- Need to specify that when the changes in the local DCNs are communicated to routing me, it recomputes the routing tables in addition to generating RIDU if multihomed.

3. Page 3-1 Section 3.1.1
- Restate the last para in this section as 'When routing ME is going down, it notifies the internet entity. This is covered in section 10.0 under aborts and recovery'.

4. Page 3-5 Section 3.2.1
- Delete reserved sap id for routing ME as it uses the services of 3A.

5. Page 3-6 Section 3.2.2
- Describe mismatch_user_id as 'The input user_id does not match the user_id in the entry for the input sap_id'.

- The description for sap_already_closed should say 'attempting to close an already closed sap'.

6. Page 3-7 Section 3.3
- Only the broadcast titles are calculated. The broadcast address for remote networks is communicated via RIDUs.

- In the last para, mention that as part of RIDUs, multi-homed systems send community title/address pairs and broadcast address (i.e if the network is a broadcast network).

7. Page 3-7 Section 3.4
- Delete references to routing info network.

- Restate 'when some attribute changes' as 'when some attribute of the local directly connected network changes'.

18. Page 7-2 Section 7.0
- Criteria for rebuilding LCR-DS needs to be clarified. The following points should be highlighted:
  - Criteria for rebuilding LCR-DS is
    - Any change in the local dcn attributes.
    - Any change in remote dcn status.
    - Any change in remote dcn's relay allowed attribute.

19. Page 8-1 Section 8.1
- There should be another flag in the least_cost_routing_ds_entry to reflect the entry status.

20. Page 8-5 Section 8.4
- The 'kind' field in nw_title_addr_ds_entry has a value 'xerox' which is not needed.

- The title_length should be 1..32 instead of 0..41 as the 'kind' field specifies the type of title.

REVISION RECORD			
Revision	Description	Author	Date
A	Original version, Release 1, WBS4.	NLR	01/06/84
B	Update for TDRB comments WBS4.	RW	03/12/84
C	Update for DCS comments.	RW	07/25/84

Table of Contents

1.0 INTRODUCTION

In the OSI environment, the ISO standards define two kinds of management, namely management of the network and management of each layer. The layer management functions in the Internet (3B) layer are grouped together under the name of Routing Management Entity. As such, it has a direct interface with "3A" entities below, specifically, the Generic 3A.

Routing Management is one of the most important of all management functions in the catenet. The stability and robustness of the catenet depends on the accuracy of the routing tables and the speed with which the Routing M-E reacts to failures and congestions in the catenet.

In Release 1, the Routing M-E will support only the Xerox Internet. In subsequent releases, it is expected to manage multiple Internet entities. It is not expected that every system will have more than one Internet, rather that only some systems which interface to different networks such as NI's, will support this feature. This document addresses the one Internet catenet and lays the groundwork for multiple internets.

1.1 PURPOSE

The purpose of the Routing M-E is twofold. The first is to build and maintain the data stores shown below:

- Least Cost Routing Data Store (LCR-DS)
- Alias List
- Local Directly Connected Network Data Store (LDCN-DS)
- Received DCN Data Store (RDCN-DS)
- Network Titles/Addresses Data Store (dedicated and remote)
- 3B SAP (Service Access Point) Data Store

The second purpose is to generate and broadcast Routing Information Data Units (RIDUs) to other Routing M-Es in the catenet. This is true only of systems located on more than

1.0 INTRODUCTION
1.1 PURPOSE

one network, which are called multi-homed systems. However, Routing M-Es in all systems receive RIDUs from multi-homed systems.

In general, the purpose of the Routing M-E is to perform all layer management functions for the Internet layer, so that the internet layer itself can do fast routing without the burden of any overhead.

1.2 REFERENCES

1. CDNA GDS by J H Hart (ARH4243)

2. CDNA Routing M-E Protocol Specification by R R Sekhon

3. Xerox Internet Transport Protocols XSIS 028112

4. Xerox Internet ERS by P Woodruff (ARH6221)

5. Directory M-E ERS by H G Coverston (ARH6384)

6. Intranet 3A ERS by K D Lamar

2.0 FEATURE/SERVICE OVERVIEWS

The services offered by the Routing M-E are:

- Maintaining the Least Cost Routing Data Stores (LCR-DS) and Alias List and the Local and Received DCN Data Stores which are used to build them.
- Generating and broadcasting Routing Information Data Units (RIDU) about local Directly Connected Networks and Community Titles with their associated multicast addresses by Routing M-Es in multi-homed systems. This information is learned from the Generic 3A status interface.
- Receiving, processing and rebroadcasting the above mentioned RIDUs.
- Registering of Network Titles/Addresses with the Directory M-E for use by software modules in the DI; these Titles/Addresses are learned from Generic 3A and RIDUs.
- Opening and closing of 3B SAP's and related maintenance of the Internet SAP Table.

2.1 FEATURES/SERVICES

2.1.1 OVERVIEW

This section explains how the functions of the Routing M-E are perceived by external processes including users. At a high level, the Routing M-E may be seen as responsible for computing the "best" paths to all networks in the catenet from the local system. These pathes are recomputed whenever a network in the catenet becomes active or inactive, or becomes congested or uncongested. Thus, without any human intervention, the catenet reacts quickly to changes in the attributes of the networks or the topology of the catenet.

A "path" from one system or network to another network is defined to be a sequence of links (networks) which are characterized by:

- The first link in the sequence connects to the source and the last link connects to the destination.

2.0 FEATURE/SERVICE OVERVIEWS
2.1.1 OVERVIEW

- There are no duplicates in the sequence
- The maximum number of paths is number of directly connected networks at the source which lead toward the destination.

2.1.2 COST OF PATHS

For Release 1.0, the 'best' path is defined as the one that has the lowest cost. The cost of a path is the sum of the costs of the links. The cost of each link is supplied by Generic 3A to its Routing M-E. The algorithm for cost definition (based on line speed) is given in section 7.1.2.1 of the GDS.

Another factor which influences cost is congestion. The GDS lists two thresholds of congestion but does not define them. To keep it simple for Release 1.0, the network is assumed to be either congested or uncongested. If a network changes to congested, the Routing M-E multiplies the cost by 4. Conversely, the network cost is divided by 4 if the congestion state changes back to uncongested.

2.1.3 RESPONSE TO FAILURES

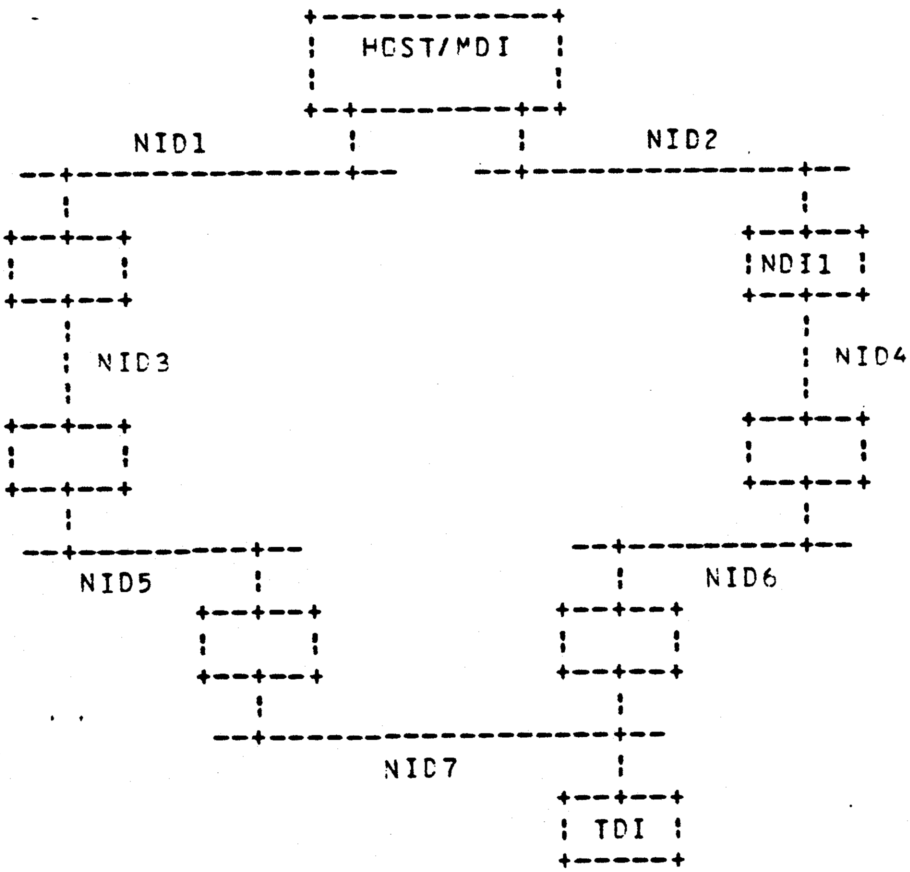
Another major function of Routing M-E is the following: If an application resides in a system which is directly connected to multiple networks, then it really has multiple addresses, one per network. However, it chooses only one among them to register with its title. Routing M-E provides the primary network address which all applications use. A non-trivial problem arises if the primary network fails, and the system can still be reached by means of the remaining networks. The Routing M-E is responsible for the smooth re-routing of messages along the active networks even when the messages are still being addressed to the system via the broken network.

Before addressing broken networks in the destination systems, routing around inactive networks between the source and destination will be examined. The first example, pictured below, shows a trivial case where a network one hop away breaks down:

```
                       +------+
                       !HOST !
                       +--+--+
                          !
  --+---------------------+-----+--
    !        NID=1              !
 +--+--+                    +--+--+
 !     !                    !NDI1 !
 +--+--+                    !--+--!
    !                          !
   PDN                         ! NID=2
    !                          !
 +--+--+                    +--+--+
 !NDI3 !                    !NDI2 !
 +--+--+                    +--+--+
    !                          !
  --+--------------------------+----------------------+--
                                  NID=3               !
                                                   +--+--+
                                                   ! TDI !
                                                   +-----+
```

It is assumed that the best path from the TDI to the host is via NID=2. If NID=2 breaks down, NDI2 would receive the message from the TDI destined for the host and then re-route it via NID=3 through NDI3. A different situation occurs if the link between NDI1 and the network with NID=1 breaks. NDI1 would re-route the message back to NDI2 and it might oscillate back and forth between NDI1 and NDI2. But these are all transient conditions and would go away as soon as NDI1 sends a new RIDU informing the catenet of the failure of its connection to NID1.

The next example is shown in the following diagram:



Assuming that NID2 is the primary network for the HOST/MFI, all applications in that system register their titles using the NID2 and the system ID on that network as part of their address. Assuming that the best path from the TI to the HOST/MFI is via NID6, NID4, and NID2, there are two problems whose solutions are discussed below.

1 - If the best path gets congested, how does the TI use the alternate route NID5, NID3, NID1?

2 - What happens if the NID2 goes down and becomes inactive?

2.0 FEATURE/SERVICE OVERVIEWS
2.1.3 RESPONSE TO FAILURES

1 - Alternate routes

Since the host is multi-homed, it would send RIDUs between NID1 and NID2. As the path to the host via NID2 becomes congested, the TI would send the messages via NID1 assuming that they would be relayed to NID2 unaware that the messages are really destined for the HOST/MFI. When the messages get to the HOST/MFI via NID1, they are delivered to the applications instead of being relayed. This poses another problem: systems in the catenet may use the host as a relay, which is a waste of the precious resources of the host. This is addressed later on.

2 - Alias List

Using this example, assume that the HOST/MFI has two addresses: NID2/SID2 and NID1/SID1 of which NID2/SID2 is the primary address. If NID2 becomes inactive, how do the systems in catenet realize that the host can still be reached by NID1? When NID2 goes down, the RIDUs from the HOST/MFI indicate that NID2/SID2 is inactive. Systems in the catenet create an alias_list entry (described in section 8.2) indicating that NID1/SID1 is equivalent to NID2/SID2; thus, when Internet cannot find an entry for NID2 in the LCR-DS, it checks the alias_list. Since an entry for the destination address NIS2/SID2 is found, the 3B-PDU is sent toward the equivalent address NID1/SID1. Each system which relays this 3B-PDU must go through the same process.

In this example, if the HOST/MFI access NID2 goes down but NID2 remains up for NDI1, an LCR-DS entry will still exist for NID2. Therefore, the LCR-DS contains a boolean field, alias_exists, to indicate that an alias_list entry exist for this network. If this field is TRUE, the alias_list is checked to see if an entry exists for the destination network/system ID. In this example, an entry for the destination NID2/SID2 is found, and the 3B-PDU is sent toward NID1/SID1. As above, this process is repeated for all relays to the final destination.

2.1.4 MULTIHOMED HOSTS

The problem of how a multi-homed system may prevent other systems from using it as a relay is addressed here. The RIDU (described in section 8.3) includes a boolean field called "relay_restricted" for each "dcn_definition_entry" which is used in the following way: The combination of two particular networks in a system cannot be used for relays if both have the "relay_restricted" field set TRUE. When the LCR-DS is generated for a given system, the "relay_restricted" fields of RIDUs are consulted and routes constructed such that no unallowed relays occur unless no other path to a destination exists. If there are, for example, two Ethernets and an LCN connected to this host, the "relay_restricted" fields are set in the following way:

```
Ethernet 1   -  relay_restricted = TRUE
Ethernet 2   -  relay_restricted = TRUE
LCN          -  relay_restricted = FALSE
```

To find out if relay between Ethernet 1 and Ethernet 2 is legal, take the logical "AND" of the two. The result in this case is TRUE which means that the relay is not legal (unless only path to destination). It may be seen further that relays between the LCN and either of the Ethernets are legal. It is to be noted here that the multi-homed system itself, which set the flags in its RIDU, does not enforce it. It leaves it to the Routing M-Es in other systems in the catenet to honor this flag in computing the LCR-DS's.

2.1.5 USE OF LEAST COST ROUTING DATA STORE (LCR-DS)

In CDNA systems, the Least Cost Routing Data Store (LCR-DS, described in section 8.1) contains one row for each network ID in the Catenet that can be reached from the local system. In addition, each row contains a boolean field, "directly_connected", which is set TRUE if the network is directly connected to this system.

The Internet Layer is the main user of the LCR-DS. Its use is described in the Xerox Internet ERS. In general, Internet uses the LCR-DS to find if this system is the final destination or the next hop for a 3B-PDU on its way to the final destination. If it is determined that 3B-PDU is bound for this system, the 3B SAP table (described in section 8.6) is used to locate the Internet user.

2.0 FEATURE/SERVICE OVERVIEWS
2.1.5 USE OF LEAST COST ROUTING DATA STORE (LCR-DS)

After the Routing M-E has calculated a new LCR-DS, a pointer to the current LCR-DS is changed to the new one. Then the old LCR-DS buffer space is released. Because Internet is non-preemptable, tasks using Internet complete their use of the LCR-DS in one time slice. Thus, Internet is unaffected by the change of the LCR-DS.

It should be noted here that this implementation of Xerox Internet is compatible with "true" Xerox only to the extent of using the same Internet header format and SAPs (socket locations in Xerox). The table structures and the structure of the Routing Information Data Units (RIDU) are all different. Consequently, the behavior of this network to failures, congestions, etc. would be different from that of a "true" Xerox Network. Further, for the CDNA Network to be fully connected with a "true" Xerox Network, CDNA Routing M-E would have to do a transformation from CDNA's RIDUs to "true" Xerox RIDUs to send to the Xerox Network; also, a reverse transformation from Xerox RIDU format to CDNA RIDU format would have to be performed for RIDUs received from "true" Xerox systems. This is essentially equivalent to building a gateway between CDNA and a Xerox network at this layer. Release 1.0 does not exchange routing information with "true" Xerox systems.

2.1.6 DIRECTORY M-E REGISTRATION OF TITLES/ADDRESSES

Community titles and corresponding multicast addresses and broadcast titles and addresses for broadcast networks are registered with the Directory M-E by Routing M-E. The community titles/addresses are learned from Generic 3A for directly connected networks and received RIDUs for remote networks. The broadcast title/addresses are calculated from known local and accessible remote networks contained in the RIDUs (as described in section 3.3). The title address information is available to software module in the DI which need a translation of Network Titles.

For example, C170 systems S1 and S11 are located on Network N1 and systems S2 and S21 on Network N2; their multicast address is '123'. A set of configuration commands processed by Generic 3A describe community title/addresses as shown below:

2.0 FEATURE/SERVICE OVERVIEWS
2.1.6 DIRECTORY M-E REGISTRATION OF TITLES/ADDRESSES

```
                      N1                                        N2
--+------+------+------+------+--     --+------+------+------+------+--
  :      :      :      :      :         :      :      :      :      :
+--+--+  :    +--+--+  :      +---------+    +--+--+  :    +--+--+  :
:S1  :   :    :S11:    :      :   NI    :    :S2  :   :    :S21:    :
+----+   :    +----+   :      +---------+    +----+   :    +----+   :
      +--+--+      +--+--+                         +--+--+      +--+--+
      :TI  :       :TI  :                          :TI  :       :TI  :
      +----+       +----+                          +----+       +----+
```

- Systems S1 and S11 are told that they belong to a community whose title is "community_C170 Host", and their multicast address is "123".

- Systems S2 and S21 are told that they belong to a community whose title is "community_C170 Host", and their multicast address is "123". (Note: the two multicast addresses need not be the same).

- The NI is told that the DCN "N1" supports a community called "community_C170 Host" with a multicast address of "123" and that th NI itself is not part of it.

- The NI is told that the DCN "N2" supports a community called "community_C170 Host" with a multicast address of "123" and that the NI itself is not part of the community.

These 3A commands cause the Ethernet SSRs in systems S1, S11, S2 and S21 to enatle the multicast address '123' since they are members of the community. Since the C170 hosts (MFI's) are not multi-homed, they do not generate RIDUs; the NI has the responsibility of including these titles in the RIDUs that it generates. Therefore, it is important to send the 3A commands to all the multi-homed systems in the affected networks.

When a remote network called "N9" gets the RIDU from this NI, the Routing M-Es in all the systems on N9 register the title "community_C170 Host" and the address '123' with their local Directory M-Es. In this way, any application in the remote system desiring to communicate with all C170 hosts, would obtain the translation (2 entries in the translation; one with address NID=1,SID=123 and the other with NID=2,SID=123) from its local Directory M-E and establish communication. The TI's on N1 and N2 would also learn of the title/addresses in the same way.

If the catenet consists of one network, i.e.: one Ethernet with a few TI's and one or more hosts, RIDUs are not automatically generated by any of the systems. Therefore, an Intranet configuration command is provided to force single-homed systems to generate RIDUs. This is the only way for community title information to be broadcast in a single network catenet.

To sum up, the 3A configuration commands to define communities would be sent to:

- All single-homed systems which are part of the community.
- All multi-homed systems in the networks where the communities exist.

The command would have the following information:

- Community name
- Multicast address (NID + SID)
- Whether this system is part of the community

The community name and the multicast address would be saved in a data store and if not previously registered, would be registered with the Directory M-E.

2.1.7 BROADCAST/RE-BROADCAST OF RIDUS

If the Local DCN Data Store contains two or more entries, the Routing M-E on this system generates RIDUs and broadcast them on all active DCN's.

The frequency is determined by:

- once every 30 seconds
- whenever the local DCN Data Store changes (network or community); the timer is reset to zero

The RIDU is generated from information in the Local DCN Data Store and the Network Titles/Addresses Data Store in the manner described in section 7.1.3.4 of GDS.

If the number of defined DCN entries referred to earlier is reduced to one, the generated RIDUs are still broadcast on the remaining DCN for 90 more seconds at the specified interval. Since the "out of date" timer has a value of 90 seconds, it takes that long in the worst case for all remote networks to realize the loss of the DCN. During that time, messages could be directed at this system via the lost DCN.

If a local network access goes down, the RIDU still contains a DCN entry set to "inactive". Receiving systems create alias_list entries as explained in section 2.1.3 in order that 3B-PDUs can still be delivered.

Every received RIDU whose sequence number is greater than previous ones generated by a particular system is re-broadcast on all DCN's in the Local DCN Data Store marked as "CDNA Routing Information Network" and active except the one DCN from which the RIDU was received. Therefore, single-homed systems do not re-broadcast.

Release 1.0 of Routing M-E does not deal with processing requests from and issuing replys for "true" Xerox system.

2.1.8 OPENING/CLOSING OF SAP'S

As the management entity for the Internet Layer, the Routing M-E is responsible for opening and closing Internet SAPs upon request from Internet users. The interface between the Internet user and Routing M-E is by direct call. The interface specifications, concepts and functionality are described in section 3.2.

2.2 FUNCTIONAL RELATIONSHIPS

```
+---------------+-----------------------------------+
!               !                                   !
!               +                                   !
!  Directory  --T->       Internet Users            !
!     M-E       +                                   !
!      ^        !    ^                      ^       !
+-----+ D +-----+-+ S +-+-------------+---+ P +----+
!      !             V   ! Alias List  !      V     !
!                        +             +            !
!     Routing M-E      --M-> LCR-DS --U-> Xerox     !
!                        +             +  Internet  !
!      ^        ^        !3B SAP Table !      ^     !
+----+ 3 +-----+ R +-----+-------------+---+ P +----+
!      V        V                             V     !
!          Generic 3A and Intranet Entities         !
!                                                   !
+---------------------------------------------------+
!                                                   !
!                      SSR's                        !
!                                                   !
```

Figure 1

Interfaces in Figure 1:
- D - Routing M-E registers title/addresses
- T - Internet users request title/address translations
- S - Routing M-E opens 3B SAPs for Internet users
- P - Xerox Internet sends data to/from Internet users and 3A
- M - Routing M-E maintains routing tables
- U - Xerox Internet uses routing tables
- 3 - Routing M-E opens a 3A SAP and receives network status
- R - Routing M-E exchanges RIDUs with other systems via 3A

Figure 1 above shows how Routing M-E relates to Xerox Internet, its users, Generic 3A, Directory M-E and routing tables. Routing M-Es main function is to build the LCR-DS which Xerox Internet uses to determine the route for 3B-PDUs. The Alias list maintains information about inactive network interfaces. Routing M-E uses a direct interface to Generic 3A to open a 3A SAP and receive information about directly connected networks. Also, RIDUs are sent to and received from other systems via 3A. Routing M-E opens and closes 3B SAPs. Community titles learned from Generic 3A and RIDUs and calculated broadcast titles are registered with the Directory M-E.

2.3 UTILIZED EXTERNAL INTERFACES

All the interfaces to Exec and Common Subroutines such as buffer management, timer, etc. are not included here.

2.3.1 GENERIC 3A

Routing M-E uses the services of Generic 3A to open a 3A SAP and to get the status of underlying networks. Community titles/addresses are included with the 3A network information. The interface to Generic 3A and the format of the Network Information Block which describes local networks and community titles is described in the Intranet 3A ERS. Based on status changes, the local DCN data base is updated and the LCR-DS recomputed; and if necessary, new RIDUs are broadcast to the entire catenet. Generic 3A status indications are unsolicited.

The Routing M-E also uses 3A to exchange Routing Information Data Units (RIDUs) with Routing M-Es in other systems.

2.3.2 DIRECTORY M-E

Routing M-E registers all community titles/addresses of both directly connected and remote networks with the Directory M-E. This enables any entity in the network to find out, for instance, the addresses of all C170 hosts (if C170 hosts is a community). In general, community titles are learned from Generic 3A and RIDUs. Interface to the Directory M-E is described in the Directory M-E ERS.

3.0 FEATURE DESCRIPTIONS

The features of the Routing M-E are described in this section. Each subsection discusses a particular interface across which services are offered.

3.1 SERVICE TO INTERNET

3.1.1 FUNCTIONAL CONCEPTS

There is no formal interface between Routing M-E and Internet. In general, it is the Routing M-Es responsibility to manage all the tables and data structures used by the Internet entity, so that the Internet entity can perform its services speedily. When the system is loaded, the Internet is not completely operational until the Routing M-E has gone through the initialization process described in section 7.0. When the process is complete, an Internet user can open a SAP, and a sufficient LCR-DS is present in order for Internet PDUs to reach their destination.

When Routing M-E is going down, it notifys the Internet entity. This is covered in section 10.0 under Aborts and Recovery.

3.1.2 GENERATION OF LCR-DS

The most notable process accomplished by the Routing M-E on behalf of Internet is the generation of the Least Cost Routing Data Store (LCR-DS). In general, the LCR-DS (described in Section 8.1) specifies which system on which directly connected network to send an Internet Protocol Data Unit (3B-PDU) on its way to the final destination. The use of the LCR-DS by Internet is described in the Xerox Internet ERS. The process used to generate the LCR-DS is described in the

CDNA GDS and The Routing M-E IDS. A pointer to the current LCR-DS is kept in the following variable:

```
VAR
  current_LCRDS_ptr: [XREF] ^route_table_row;
```

3.2 INTERFACES TO OPEN/CLOSE 3B SAP'S

On behalf of the Internet entities, the Routing M-E opens and closes 3B SAP's.

3.2.1 OPEN INTERNET SAP

When an Internet user wants to open an Internet SAP, he issues a call to the open_internet_sap procedure. A dedicated well-known or ephemeral (dynamically assigned) SAP can be opened. In general, the user describes the SAP he wants to open and supplies procedure interfaces. The open SAP ID is returned along with the procedure address for Internet. Below is shown the procedure interface to open an Internet SAP and the associated CYBIL parameter records.

```
PROCEDURE [XREF] open_internet_sap ( {
      input_param: ^open_sap_input_parameters; { INPUT
      output_param: ^open_sap_output_parameters; { INPUT
  VAR return_code: open_internet_sap_status); { OUTPUT


{
{  INPUT parameter record for open INTERNET (3B) SAP
{
TYPE
  record = open_sap_input_parameters
    sap_id: sap_id_type; { If <> 0: Requested Dedicated SAP ID
    user_id: ^cell; { user identifier
    destination: destination_3b_sap_if; { Proc for 3B ind
    force_close: force_close_if; { Procedure to close 3B SAP
  recend;


{
{  User procedure interface to receive Internet Indications
{
TYPE
  destination_3b_sap_if = ^procedure ( {
        ind_params: ^internet_ind_if), { INPUT
                                      {  - 3B data ind params
```

```
  internet_ind_if = record { 3B Data Indication parameters          :
    multicast: boolean, { TRUE=multicast, FALSE=datagram            :
    checksum: boolean, { TRUE if message was checksummed            :
    source_address: internet_address,                               :
    destination_address: internet_address,                          :
    control: control_bytes, { hop_count and packet_type             :
    user_id: ^cell, { user_ID for this SAP entry                    :
    data: buf_ptr, { message buffer descriptor address              :
  recend;                                                           :

{
{  User procedure interface
{    for Routing M-E to close an INTERNET (3B) SAP
{
TYPE                                                                :
  force_close_if = ^procedure ( {                                   :
        sap_id: sap_id_type; { INPUT - SAP ID of SAP to close
        user_id: ^cell); { INPUT - user identifier

{
{  OUTPUT parameter record for open INTERNET (3B) SAP
{
TYPE
  record = open_sap_output_parameters                               :
    local_internet_address: internet_address; { assigned SAP ID     :
    internet_request: internet_request_address;
    maximum_request_length: 1 .. max_data_length;                   :
  recend;

{
{  Return codes from open_internet_sap
{
TYPE
  open_internet_sap_status = ( {
    open_sap_successful, { SAP was opened successfully              :
    internet_down); { INTERNET not available                        :
    illegal_dedicated_sap, { This dedicated SAP ID too high         :
    sap_already_opened, { This dedicated SAP is already open        :
    no_sap_entries_available, { All SAP table space in use          :
    internet_down); { INTERNET not available
```

The Routing M-E has a list of well-known SAP's that can be opened. As mentioned above, the user can explicitly open one of these by specifying the SAP value in the open_internet_sap procedure call. The dedicated Internet SAP values are as follows:

3.0 FEATURE DESCRIPTIONS
3.2.1 OPEN INTERNET SAP

```
CONST
{ Interface via Internet (3B) Layer
  Routing_me_sapid           =     1(10), { Reserved for future use
  Echo_me_sapid              =     2(10),
  Error_me_sapid             =     3(10),
  Directory_me_sapid         =    20(10),
  File_Access_me_sapid       =    21(10),
  Command_me_sapid           =    22(10),
  Log_me_sapid               =    23(10),

{ Interface via Transport (4) Layer
  Directory_me_xp_sapid      = 1020(10),
  File_Access_me_xp_sapid    = 1021(10),
  Command_me_xp_sapid        = 1022(10),
  Log_me_xp_sapid            = 1023(10),

{ Ephemeral Internet SAPs
 request_ephemeral_sap       =     0(10),
 first_ephemeral_sap         = 3001(10);
```

The range of SAP's 1 thru 3000(10) are for dedicated SAP's which is in conformity with Xerox's definition for XNS networks. All others are ephemeral, i.e.: they are dynamically assigned and reused. Ephemeral SAP's are assigned on the following basis. A variable <u>in battery-backed RAM</u> holding the next available ephemeral SAP is initialized to 3001(10). Every time a new ephemeral SAP number is needed, the following steps are performed:

Step 1 - A check is made to see if the number in the variable is already assigned and active. If so, the variable is incremented by 1. If less than 65536, repeat Step 1; otherwise, set variable to 3001 and repeat Step 1.

Step 2 - Assign this number and update tables.

Step 3 - Increment variable by 1. If less than 65536, quit. Otherwise set variable to 3001.

This method makes sure that de-assigned SAP's remain unused for a long period of time for analysis and debugging purposes. It also prevents an application from being assigned a newly released SAP and receiving a message meant for the previous application which released that SAP. The variable's location in battery-backed RAM assures that this method will not be disrupted by either system re-loads or power failures.

3.2.2 CLOSE INTERNET SAP

When an Internet user wants to close an Internet SAP, he issues a call to the close_internet_sap procedure. In general, the user specifies the SAP he wants to close. Below is shown the procedure interface to close an Internet SAP and the associated parameters:

```
PROCEDURE [XREF] close_internet_sap ( {
      sap_id: sap_id_type; { INPUT - SAP ID of SAP to close
      user_id: ^cell; { INPUT - user identifier
  VAR return_code: close_internet_sap_status); { OUTPUT

{
{ Return codes from close_internet_sap
{
TYPE
  close_internet_sap_status = ( {
    close_sap_successful, { SAP was closed successfully
    sap_already_closed, { Attempting to close already closed SAP
    mismatch_userid); { Input user_id doesn't match SAP entry
```

3.3 REGISTRATION_OF_COMMUNITY_TITLES

This is not a direct service that Routing M-E provides to applications and M-Es in the DI but occurs automatically. Local community title/addresses are learned from Community Title configuration commands. Remote ones are contained in RIDUs from other systems. Routing M-E registers the local title/addresses with the Directory M-E.

As part of RIDUs, multi-homed systems send community title/addresses pairs throughout the catenet. Receiving systems register the received titles/addresses with the Directory M-F.

3.4 NETWORK_TITLES_DATA_STORE

The Network Titles Data Store is built by Routing M-E for Directory M-E whenever a full update of the LCR-DS is processed. Its format is described in section 8.7. In general, this data store is a list of all accessible networks sorted by the number of hops from this system from lowest to highest. Directly connected networks have a hop_count of 00.

3.5 BROADCAST_OF_RIDUS_TO_OTHER_ROUTING_M-ES

RIDUs are generated by systems that have more than one directly connected network. Every 30 second or when some attribute of a directly connected network or local community title changes, an RIDU describing the directly connected networks and local community titles is broadcast on all active directly connected networks.

Also, when a multi-homed system receives an RIDU from another system, the received RIDU is saved and re-broadcast on all networks that are currently active and are marked as Routing Information Networks, except the one on which it was received.

4.0 PERFORMANCE

The following performance parameters have been established:

The processing of received RICUs		< _ .25 msec
with 10 networks 3 DCN's 20 Relaying System	generation of routing table	< _ .5 msec
Opening a SAP		< _ .2 msec
Closing a SAP		< _ .1 msec
Memory requirements for code		< _ 4K bytes

5.0 FINITE STATE MACHINE

The "Routing M-E Protocol Specification" to describe the Finite State Machine for this ME is in a separate document as an attachment to the GDS.

6.0 LOG_MESSAGES_GENERATED

6.1 LCR-DS_GENERATED

Whenever the LCR-DS is rebuilt in a system, the following log message is generated:

LOG_MESSAGE_ID

r_me_lcrds_rebuilt

DESCRIPTIVE_MESSAGE

LCR-DS Rebuilt

MASK	LOG_MSG_BUFFER		
MASK	LOG_MESSAGE_BUFFER_PTR^ (variable part)		
fixed text	type	value	description
TYPE =	binary octets	4 Chars	Full or partial update

Operator_Display_Format_Example

```
83/08/04  11.00.35      123456789ABC
LCR-DS Rebuilt
TYPE = FULL
```

NOTE: The date, time and originating system address are taken from the PDU header.

6.2 RIDU_WITH_UNKNOWN_ID

Whenever an Routing Information Data Unit (RIDU) is received whose ID field is unkonwn, the following log message is generated:

LOG_MESSAGE_ID

r_me_unknown_ridu_id

DESCRIPTIVE_MESSAGE

RIDU with unknown ID

MASK	LOG_MSG_BUFFER		
MASK	LOG_MESSAGE_BUFFER_PTR^ (variable part)		
fixed text	type	value	description
RIDU =	binary octets	1..512 octets	The RIDU with unknown ID

Operator_Display_Format_Example

```
83/08/04  11.00.35        123456789ABC
RIDU with unknown ID
RIDU=1234567890abcdef0123456
```

NOTE: The date, time and originating system address are taken from the PDU header.

6.3 UNKNOWN_MESSAGE_TO_RIDU_PROCESS

Whenever process_recvd_RIDUs receives an intertask message whose workcode is not "r_me_ridu_msg", the following log message is generated:

LOG_MESSAGE_ID

r_me_unknown_msg_to_RIDU_proc

DESCRIPTIVE_MESSAGE

Unknown intertask message to RIDU process

MASK	LOG_MSG_BUFFER		
MASK	LOG_MESSAGE_BUFFER_PTR^ (variable part)		
fixed text	type	value	description
Message =	binary octets	1..512 octets	The Intertask message

Operator_Display_Format_Example

83/08/04 11.00.35   123456789ABC
Unknown intertask message to RIDU process
Message=1234567890abcdef0123456

NOTE: The date, time and originating system address are taken from the PDU header.

6.4 UNKNOWN_MESSAGE_ID_LCRDS_UPDATE

Whenever update LCRDS receives an intertask message whose workcode is not "r_me_part_update_LCRDS" or "r_me_full_update_LCRDS", the following log message is generated:

LOG_MESSAGE_ID

r_me_unknown_msg_LCRDS_update

DESCRIPTIVE_MESSAGE

Unknown intertask message to LCRDS update

MASK	LOG_MSG_BUFFER		
MASK	LOG_MESSAGE_BUFFER_PTR^ (variable part)		
fixed text	type	value	description
Message =	binary octets	1..512 octets	The Intertask message

Operator_Display_Format_Example

83/08/04 11.00.35 123456789ABC
Unknown intertask message to LCRDS update
Message=1234567890abcdef0123456

NOTE: The date, time and originating system address are taken from the PDU header.

7.0 INITIALIZATION

7.0 INITIALIZATION

After it is loaded, Routing M-E performs the following sequence of steps to initialize itself. It is to be noted that in Release 1.0, only Xerox Internet and Transport are used. Also, the LCR-DS and 3B SAP data stores are all empty.

1) Open a 3A SAP via direct call interface described in the Generic 3A Intranet ERS and wait for unsolicited status about directly connected networks. This 3A SAP is also used for transmitting and receiving RIDUs.

2) Process 3A status information as it is received in a direct call interface described in the Generic 3A ERS. In general, Local DCN-DS entries are created and modified. The following information is supplied:
   - network ID of particular network
   - network status
   - pointer to network information block
   - system ID of receiving system
   - whether this is a broadcast/multicast network

3) Broadcast a request_ridu_info RIDU (described in section 8.3) on the first network described by 3A. The reason for this is to ask for all received RIDU information in order for Internet to become operational sooner. The only systems that would respond to this request are those that have the responsibility for generating and distributing RIDUs. These would include all the NI's on this network and any MDI's configured to perform this function. This prevents a flood of responses from all the systems on the network.

4) Process RIDUs as they arrive in the normal way. Most will probably be the result of the request_ridu_info issued in step 3 above.

5) When the first LCRDS is built as a result of the first network described by a 3A status update, allow Internet users to open 3B SAPs.

It can be seen that once the 3A SAP is opened, Routing M-E operates as it always does: Information is received from Generic 3A and RIDUs from other systems and processed. Local and received DCN entries are created and updated.

INITIALIZATION

Title/addresses are registered with Directory M-E. After the first LCR-DS is built, criteria for rebuilding it becomes any of the following:

1) Any change in local DCN attributes;
2) Any change in remote DCN status;
3) Any change in remote DCN relay allowed status;

After Routing M-E builds it's first local DCN data store entry, it cannot send Routing Information Data Units (RIDU) since it currently considers itself not to be multi-homed. If more local directly connect networks are defined by Generic 3A, more local DCN data store entries are created and the system becomes multi-homed and begins to create and send RIDUs.

8.0 DATA_TYPES

The following CYBIL records are used in the Routing M-E.

8.1 LEAST_COST_ROUTING_DATA_STORE_(LCR-DS)

```
{
{   least_cost_routing data store entries describe the path
{     from this system to a specified network.
{
TYPE
  least_cost_routing_ds_entry = record
    aggregate_cost: integer,
    aggregate_cost_ratio: -8000(16) .. 07fff(16),
    pdu_count: integer,
    relay_count: 0 .. 0ffff(16),
    next_hop_network_id: network_id_type,
    next_hop_system_id: system_id_type,
    parent_network_id: network_id_type,
    congested: boolean,
    relay_restricted: boolean,
    unused: boolean,
    obsolete: boolean,
    local_dcnds_ptr: ^local_dcnds_entry, { If directly_connected
    next_entry: ^least_cost_routing_ds_entry;
  recend;
```

8.0 DATA TYPES
8.1 LEAST COST ROUTING DATA STORE (LCR-DS)

```
{
{ Route Table Row:
{  group of LCR-DS entries for a particular Network ID
{
TYPE
  least_cost_routing_ds_row = record
    network_id: network_id_type,
    changed: boolean,
    directly_connected: boolean,
    alias_exists: boolean,
    multicast: boolean,
    broadcast_address: system_id_type,
    valid_lcrds_entry_count: 0 .. 0ffff(16),
    first_lcrds_entry: ^least_cost_routing_ds_entry,
    next_row: ^least_cost_routing_ds_row,
  recend;

VAR
  current_lcrds_ptr: [XDCL] ^least_cost_routing_ds_row;
```

8.2 ALIAS_LIST

```
{
{  Alias List to relate equivalent NW/system addresses
{    form multi-homed systems
{
TYPE
  alias_list_entry = record
    inactive: system_address,
    equivalent: system_address,
    next_entry: ^alias_list_entry,
  recend,

  system_address = record
   network_id: network_id_type,
    system_id: system_id_type,
  recend;

VAR
  alias_list: [XDCL] ^alias_list_entry;
```

8.3 ROUTING_INFORMATION_DATA_UNIT_(RIDU)

```
{
{ values for id field in RIDUs
{
CONST
  ridu_null = 0,
  normal_ridu = 1,
  request_ridu_info = 2;

{
{ Header for Routing Information Data Unit (RIDU)
{   this header is followed by dcn_definition_entrys
{
TYPE
  routing_info_data_unit_hdr = packed record
    id: 0 .. 0ff(16), { kind of data unit = normal_ridu
    routing_info_changed: boolean,
    title_info_changed: boolean,
    dcn_count: 0 .. 3f(16),
    sequence_no: integer,
  recend;

{
{ Request for all received RIDU information
{   Broadcast by system being initialized
{
TYPE
  request_for_received_ridu_info = id: 0 .. 0ff(16);
                                  { ID = request_ridu_info
```

8.0 DATA TYPES
8.3 ROUTING INFORMATION DATA UNIT (RIDU)

```
{
{  One dcn_definition_entry exists for each directly connected
{   network in the originating system followed by the specified
{   number of community_title entries
{
TYPE
  dcn_definition_entry = packed record
    system_address: system_address_type,
    cost: 0 .. Offff(16),
    community_title_count: 0 .. Off(16),
    filler: 0 .. 01(16),
    routing_info_changed: boolean,
    title_info_changed: boolean,
    network_active: boolean,
    sap_3a_congestion: (none, level1, level2),
    relay_restricted: boolean,
    case broadcast_network: boolean OF
    = TRUE =
      broadcast_address: system_id_type,
    casend,
  recend,

{
{  A corresponding number of community_title_entries exist for
{   community_title_count.  The title_name field only contains
{   the number of characters specified by the title_length field
{
  community_title_entry = packed record
    address: system_id_type,
    filler: 0 .. 3,
    title_length: 1 .. 32, { length
    title_name: string ( * <= 32), { title_length specifies size
  recend;

{
{ The Local RIDU: sequence no, header ptr and data buffer
{
VAR
  local_ridu_sequence_no: integer,
  local_ridu_hdr_ptr: ^routing_info_data_unit_hdr,
  local_ridu: buf_ptr,
  local_ridu_semaphore: lock_type; { To lock Local RIDU
```

8.4 LOCAL DIRECTLY CONNECTED DATA STORE (LDCN-DS)

```
{
{ Entries for local dcn data store
{   these entries describe directly connected networks
{
TYPE
  local_dcnds_entry = record
    cdna_route_info_nw: boolean,
    network_status: network_status_type,
    cdna_routing_addr: system_id_type,
    max_pdu_size: 0 .. 65535,
    dcn_entry: ^dcn_definition_entry,
    next_entry: ^local_dcnds_entry,
    comm_titles: ^array [ * ] of comm_title_entry_id,
  recend;

{
{ pointers to data stores concerning local dcn data stores
{   and related dcn info to broadcast to other systems
{
  VAR
    first_local_dcnds_entry: [XDCL] ^local_dcnds_entry,
    ldcnds_semaphore: [XDCL] lock_type;
```

8.5 RECEIVED DIRECTLY CONNECTED DATA STORE (RDCN-DS)

```
{
{ entries for received dcn data store describing current info
{   about networks connected to relay systems
{
TYPE
  received_dcnds_entry = record
    sequence_no: integer,
    timestamp: integer,
    next_entry: ^received_dcnds_entry,
    dcn_entry: array [ * ] of dcn_information_type,
  recend;

{
{ pointers to data stores concerning received dcn data stores and
{   related dcn info received from other systems
{
  VAR
    received_dcnds_ptr: [XDCL] ^received_dcnds_entry,
    rdcnds_semaphore: [XDCL] lock_type;
```

8.6 COMMUNITY_TITLE_DATA_STORE

```
{
{  Entries describing community title entries.
{    For each comm_title_ds_entry, there exist one or more
{    comm_address_entrys (one for each tranlation)
{
TYPE
  comm_title_ds_entry = record
    first_addr: ^comm_address_entry,
    next_entry: ^comm_title_ds_entry,
    title: string ( * <= 32),
  recend,

  comm_address_entry = record
    id: comm_address_id,
    occurrences: 0 .. Off(16),
    multicast_addr: system_address_type,
    directory_id: dir_id_rec,
    next_entry: ^comm_address_entry,
  recend,

  comm_title_entry_id = record
    entry_ptr: ^comm_title_ds_entry,
    id: comm_address_id,
    member_of_comm: boolean,
  recend,

  comm_address_id = 0 .. Off(16);


VAR
  first_comm_title_ds_entry: ^comm_title_ds_entry.
  comm_titles_semaphore: [XDCL] lock_type;
```

8.0 DATA TYPES
8.7 NETWORK TITLE DATA STORE

8.7 NETWORK_TITLE_DATA_STORE

```
{
{ This record is for "broadcast network rr" titles that are
{  used by Directory M-E.  The data store is created whenever
{  a full update of the LCP-DS occurs.  Entries are created
{  for each accessible network and sorted by hop count
{  ( hop count = 0 for directly connected network).
{
TYPE
  network_title_entry = record
    address: system_address_type,
    hop_count: 0..15,
    next_entry: ^network_title_entry,
  recend;

VAR
  network_titles_ptr: [XDCL] ^network_title_entry,
  rw_titles_semaphore: [XDCL] lock_type;
```

8.8 INTERNET_SAP_TABLE

```
{                                                                        :
{  The INTERNET SAP table maintains Service Access Points for           :
{   INTERNET.  An INTERNET user must open an INTERNET SAP               :
{   (which specifies a procedure address where to deliver               :
{   INTERNET data indications destined for that SAP) in order           :
{   to user INTERNET.                                                    :
{                                                                        :
{   For each open SAP there is one SAP_TABLE_ENTRY and one entry:
{   in the array of INTERNET_SAP_TABLE_TYPE (which points to the:
{   SAP_TABLE_ENTRY).  The array is in ascending order and              :
{   contains only entries for open SAPs.                                 :
{                                                                        :
TYPE                                                                     :
  internet_sap_table_type = record                                       :
    sap_id: sap_id_type,                                                 :
    entry_ptr: ^sap_table_entry,                                         :
  recend,                                                                :
                                                                         :
  sap_table_entry = record                                               :
    user_id: ^cell, { User identifier                                    :
    destination: destination_3b_sap_if, { User indication PROC          :
    force_close: force_close_if, { User force close SAP PROC            :
  recend;                                                                :
                                                                         :
VAR                                                                      :
  internet_sap_table:                                                    :
       [XDCL] ^array [ * ] of internet_sap_table_type;                  :
```

9.0 GLOSSARY

Alias_List: List of network/system ID pairs: Indicates an equivalent network/system address when the network access to a particular system becomes inactive. Internet uses the Alias_List when the specified destination address of a PDU is inactive.

DCN: Directly Connected Network: networks which are directly connect to this system.

LCR-DS: Least Cost Routing Data Store: The routing table maintained by Routing M-E and used by Internet for determining where to send 3B-PDUs on the way to the final destination.

LDCN-DS: Local Directly Connected Data Store: information about networks directly connected to this system.

RDCN-DS: Received Directly Connected Data Store: information about networks directly connected to other systems in the catenet which is learned from RIDUs received from multi-homed systems.

RIDU: Routing Information Data Unit: broadcast from multi-homed systems indicating the status of directly connected networks and multicast titles in order that all systems can build LCR-DS's.

10.0 ABORTS AND RECOVERY

10.1 INTERNET

When Internet is called either by an Internet user or Intranet, it takes over the current task and informs the Exec of an Internet recovery procedure. If Internet fails, an error recovery routine is called. All buffers and stacks related to the failed task are released but other Internet processes are unaffected. Since the Internet datagram service does not guarantee delivery, the sender and receiver are unaffected.

10.2 ROUTING M-E

When the task for Routing M-E is started, it is given some stack space. The initialization within Routing M-E passes the address of the error recovery routine on the stack. If the Routing M-E fails abnormally, the system ancestor is informed and it gives control to this special error recovery routine, which informs 3B entities of its failure and closes all the open 3B SAP's. At the same time, no Internet SAPs can be opened.

After the on-line dump procedure is defined, the memory will be dumped for debugging. As the route tables are not available any more, the DI has to be initialized.