

FDDI Topology Mapping

For the FDDI version of the HP Network Advisor protocol analyzer, ring mapping algorithms were devised to provide topological views of FDDI networks. These algorithms are designed to handle many problem situations that are characteristic of emerging LAN technologies.

by Sunil Bhat

Central to most network management products is network topology mapping functionality. A topology map helps in the management of the devices on the network and provides a context for troubleshooting network problems. The FDDI Ring Manager application for the HP Network Advisor protocol analyzer (see article, page 88) is no exception to this rule. The core of the FDDI Ring Manager is the ring mapper, which provides ring topology information that forms a convenient framework for gathering and maintaining connection, configuration, operational, and historical information for all devices on the ring. This article focuses on the processes and algorithms used by the ring mapper for generating topology maps for FDDI rings.

Topologically, an FDDI network forms a dual ring of trees or a subset thereof. This is because all FDDI links are duplex. This results in two separate counterrotating rings. In the event of a fault, the two rings wrap about the fault, forming a single unified ring. The dual ring is referred to as the *trunk* or *backbone* ring.

FDDI network topology can be viewed in two distinct ways. The *logical view* is an ordered sequence of all the active media access control (MAC) elements on the ring that participate in the FDDI protocol. It describes the path of the token through the active MACs. In a fault-free network, there are two separate token paths and therefore two logical rings. These are referred to as the *primary* and *secondary* rings. The *physical view*, on the other hand, describes the way in which the network components like stations, concentrators, and so on are attached to each other with physical links to form the FDDI network.

Basic FDDI Concepts

For the benefit of readers not familiar with the FDDI technology, this section provides a brief introduction to the basic concepts that are relevant to the discussion of topology mapping. Fig. 1 shows a typical FDDI ring. The token path within stations is indicated using dotted lines.

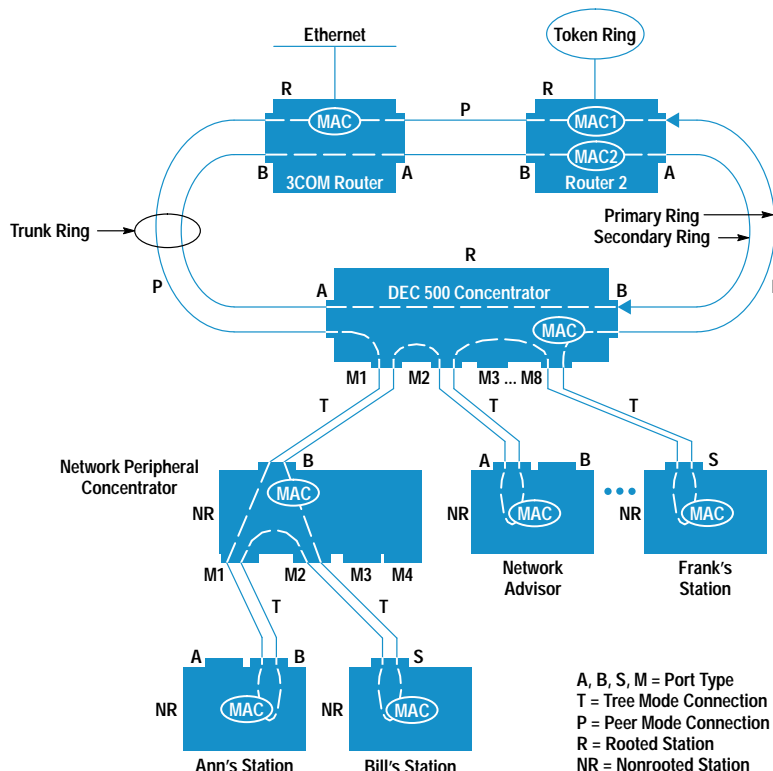


Fig. 1. An example of an FDDI ring.

A node on an FDDI network connects to the media via a *port*, which provides one end of a connection with another node. There are four types of ports: A, B, S, and M. Each connection is supported by a link which consists of a full-duplex fiber or copper cable. It has a mode associated with it based on the types of ports it connects. The valid modes are peer, tree, and none. Peer connections are formed on the trunk ring, which consists of A-B connections. These connections have no end port of type M. Tree connections connect nodes to the M-ports of concentrators. In this mode exactly one end of the connection is a port of type M. M-M connections fall under the none category and are disallowed.

The stations on the trunk ring are called *rooted* while those that connect to it via a concentrator are *nonrooted*. Rooted stations have no active port of type A, B, or S connected in tree mode.

FDDI differs from its preceding technologies like Ethernet and token ring in that the standard mandates that each node support station management (SMT). Station management is responsible for network reliability as well as management. It incorporates a management information base (MIB) that can be queried remotely using station management frame services to get information about the node, such as status, configuration, operation, and so on. This information is invaluable for understanding the topology and the health of the ring. The station management frame services that are used by the ring mapper are neighbor information frames (NIFs) and station information frames (SIFs).

Neighbor information frames are used in the neighbor notification process by every MAC on the ring. Each MAC generates a NIF at least once every 30 seconds, which allows others to monitor topological changes on the ring. A NIF sourced by a MAC contains its upstream neighbor address. This way each MAC can update its upstream neighbor address. NIFs include a description of the station including the type of node, number of MACs, port resources, the status of the station such as wrapped or twisted, and the upstream neighbor address of the logical upstream neighbor MAC.

Station information frames are used to acquire detailed station information in a packaged form. Configure and operation SIF responses contain configuration and operation information for each resource within the station, over and above the basic information contained in NIFs. This information is available only by active querying and includes basic station information contained in the NIFs, a path descriptor for the internal token path within the station, timer values and various frame counts for each MAC resource in the station, and link error information for each connection to the station.

Discovering the Ring Topology

There are both passive and proactive schemes for discovering the ring topology. The passive scheme consists of monitoring the asynchronous NIFs on the ring that are part of the neighbor notification process. A NIF contains quite basic information that lacks physical details, so one can only infer a logical view of the network. By monitoring the ring for a complete round of neighbor notification (at least 30 seconds), the mapper can obtain sufficient information to build the logical map. This map is representative of the ring (primary or secondary) being monitored.

Active schemes involve querying the nodes on the ring for neighbor and other information using NIF, SIF, or get-PMF request frames.[†] By using appropriate requests one can obtain logical, physical, configuration, and operational information. There are two flavors of active querying: broadcast and circular. In either scheme the responses are directed to the agent running the active process.

In broadcast querying the active process can broadcast one of the above request frames, capture all the responses, and build the map from the information contained in the responses. In circular querying the active process queries the nodes on the ring in a sequential manner by sending unicast request frames starting with its upstream neighbor. The neighbor's response to the query should provide the address of the next upstream MAC, which would be queried next. This process continues around the ring until the agent is reached. This process is also referred to as "walking the ring."

Data Structures

The ring mapper of the FDDI Ring Manager uses well-known structures to maintain information on each node in the network and its resources (MACs and ports). Each of these structures can be linked into a list. Sets of relevant structures are linked together in a specific order to represent logical or physical topological information. All of the structures and supporting services like linked lists, hashing services, and so on have been implemented in an object-oriented fashion using the C++ programming language.

The MAC structure contains the following information: this MAC address, upstream neighbor MAC address, downstream neighbor MAC address, and topological resource information. This information is maintained for physical topological mapping purposes only.

The port structure contains the following information: port type, connection state, remote port type, link to the remote port, and topological resource information. This information is maintained for physical topological mapping purposes only.

Topological resource information for a station relates to the token path within the station. The resources within a station are indexed from 1 to n+m, where n is the number of ports and m is the number of MACs physically present in the station. Both the MAC and the port resource structures also contain the following information:

- Resource index. The index of this particular resource among all resources in the station.
- Connection resource index. The resource index of the resource within the station that is the downstream neighbor of this resource in the token path.
- Neighboring resource information. This includes links to neighboring resource structures:
 - Pointer to the upstream resource
 - Pointer to the downstream resource.

The station structure stores basic information regarding a station and maintains links to each of its resource structures. This information includes a description including the type of the station and the number of MAC and port resources, the state of the station such as wrapped or twisted,

[†] A get-PMF request frame requests a parameter value from a MIB on the destination station. The response is a PMF—parameter management frame—containing the parameter value.

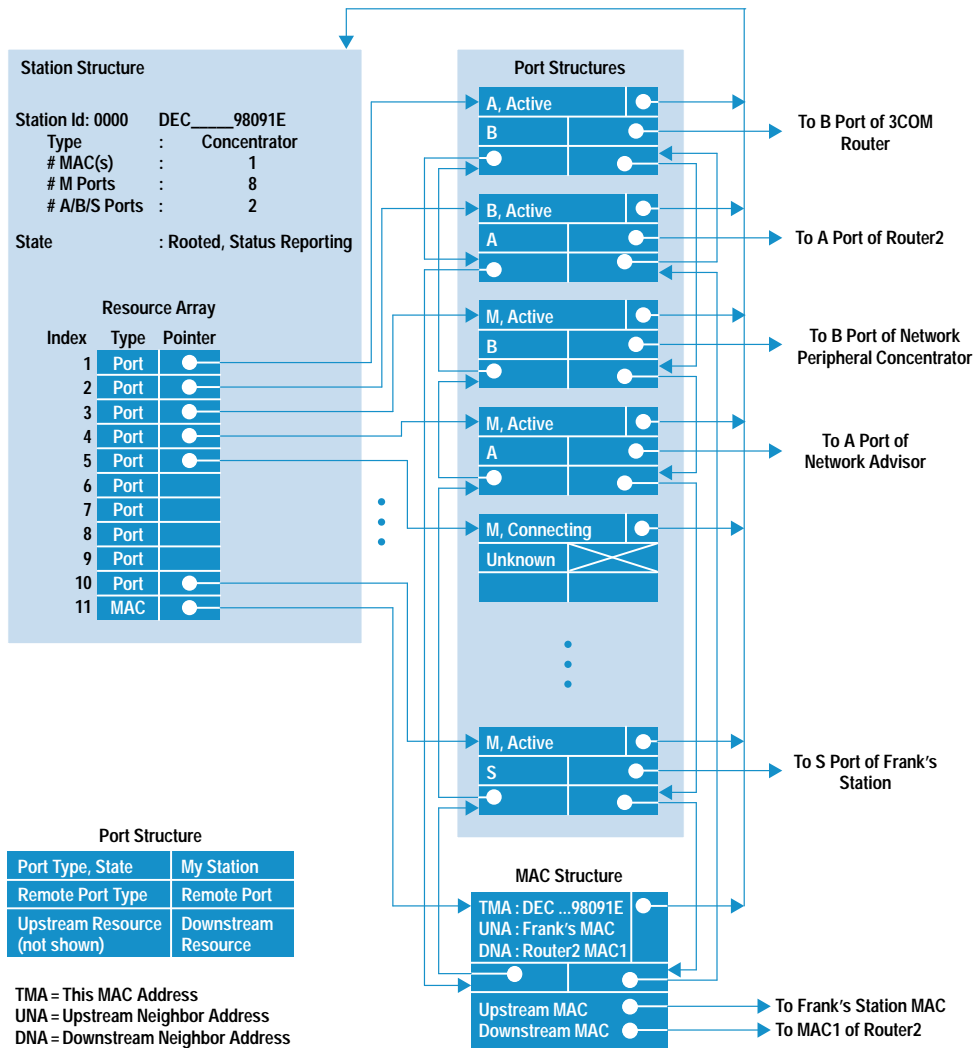


Fig. 2. The interconnection of a station and its resource structures for the DEC 500 concentrator of Fig. 1.

and an array of links to the structure for each resource in the station. Each link consists of a type of resource and a pointer to its structure.

An example of a station structure, along with its resource structures, is shown in Fig. 2. This structure is for a single-MAC dual-attach concentrator with eight M-ports. It also illustrates the internal token paths, which are represented by the upstream and downstream resource links.

Topological information regarding the FDDI ring is maintained by linking the above basic structures on specific lists. Each list provides different semantic information. There is at most one structure for any MAC. A MAC structure, once created, may exist on exactly one of the following three lists based on the current status of the MAC:

- The logical topology list consists of MAC structures, one for each active MAC on the ring in the reverse order of the token flow. In other words, for every MAC structure on the list, the structures to the right and left represent the upstream and downstream neighbor MACs.
- The active MACs list contains the structures for MACs that are active on the ring but have not yet been placed in their appropriate positions on the logical topology list.
- The removed MACs list represents the structures for the MACs that we know of but are currently inactive.

Any MAC structures that reside on any one of the above lists may also be directly accessed by means of a hashing scheme that keys off the least significant bits of its MAC address. The above scheme ensures that collisions in the hashing function are rare.

Timing Elements

The *monitor interval* is the interval of time over which the ring mapper monitors the relevant frames on the ring. The types of frames monitored depend on the scheme being used for topology mapping. For example, for logical mapping using the passive scheme, it suffices to monitor NIFs for an interval of 31 seconds to capture one complete round of the neighbor notification process. The *monitor timer* is used to track the monitor interval.

The *update interval* is the user-specified interval for determining the ring topology. In active schemes, this interval defines the polling frequency of the agent. This is tracked by the *update timer*.

Logical Mapping Process

The topology mapper can be configured to use either a passive or an active scheme to determine the logical view of the ring. This section describes only the passive process, which

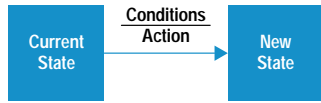


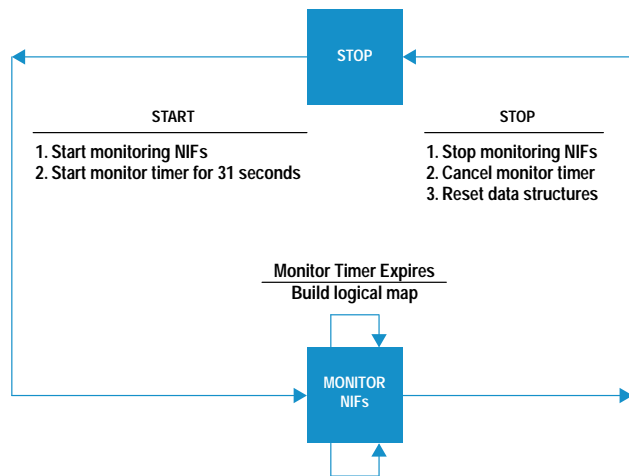
Fig. 3. In Figs. 4 and 6, all transitions from a current state to a new state are represented as shown here. All transition conditions are evaluated from the current state. If the transition conditions are satisfied, the state machine performs the associated transition actions and enters the new state. The new state now becomes the current state.

differs from the active process only in the frames (NIFs instead of SIF or get-PMF responses) it uses for obtaining neighbor information for building the logical topology.

The processes discussed in this article are described using state machines and the individual algorithms are described using simple procedural steps. The convention for the state machines is shown in Fig. 3.

Fig. 4 shows the state diagram for the passive logical mapping process. The mapper is in the STOP state when it is initialized (created). In this state, it waits for a user directive (START) to start the mapping process. On a START directive, it starts monitoring NIFs. For the purpose of knowing the upstream neighbor MAC address for each active MAC on the ring, it is sufficient to monitor only broadcast NIF requests and announcements for a minimum of 31 seconds.

In the MONITOR NIFs state the mapper is monitoring all broadcast NIF requests and announcements on the ring. On receipt of a frame from a sourcing MAC, it accesses (creates if necessary) and updates the structure for that MAC from the information in the frame. The receipt of a NIF from a MAC within a monitor period indicates that the MAC is active for



NIF Received (sa: Source Address, una: Upstream Neighbor Address)

```

1. active_MAC = find MAC structure for sa.
2. if (not found)
  then,
  2.1 active_MAC = create a MAC structure for sa.
  2.2 Append active_MAC to active MACs list.
  else
  2.3 if (active_MAC on removed MACs list)
    then,
    2.3.1 Remove active_MAC from removed MACs list and append to
        active MACs list.
  endif
endif
3. active_MAC.una ← una

```

Fig. 4. State diagram for the logical mapping process.

at least some of the monitor period. Therefore, if the structure for the MAC is on the removed MACs list it is moved to the active MACs list. The mapper exits out of this state when the monitor timer expires and it has finished processing all the received NIFs. At this point the mapper has all the necessary information to build the logical topology map. The mapper can be aborted out of this state by an explicit user STOP directive.

Building the Logical Map

Simply stated, the algorithm for building the logical ring map consists of starting with a MAC that was active (and thereby inserted) on the ring and finding the MAC structure for its upstream neighbor as reported in its upstream neighbor address. Once found it is placed upstream (to the right) of the current MAC. The algorithm then moves to the next MAC and repeats the process until it reaches the start MAC and the ring is complete. Complications to this process arise when there are inconsistencies or incompleteness in the ring information acquired over a monitor period. This causes discontinuities in the ring map which are referred to as *gaps* and are represented using a special MAC structure called a gap structure. Some of the causes for discontinuities are:

- A MAC with an unknown upstream neighbor address
- Multiple MACs reporting the same upstream neighbor address
- A MAC with an upstream neighbor address that is unseen
- Improper NIFs sent by a MAC.

If there are gaps a patching scheme is used to determine the ring map to the extent possible.

The basic steps of the logical mapping algorithm are described below:

1. Remove all MACs from the logical topology list that did not source a NIF over the last monitor period, that is, all inactive MACs. Invariant: A MAC is on the logical topology list or the active MACs list if and only if it sourced at least one NIF (i.e., was active) over the last monitor period.
2. Start with the last active MAC in the monitor period. This MAC has the best chance of being in the ring at the end of the monitor period. Let this be known as the start MAC. Set the current MAC to be the start MAC.
3. While (current MAC's upstream neighbor address differs from the start MAC's address)
 - do
 - 3.1 Find the MAC structure for the MAC corresponding to the upstream neighbor address of the current MAC. This is the new upstream neighbor MAC of the current MAC. The old upstream neighbor MAC was the structure to the right of the current MAC.
 - 3.2 If ((current MAC's upstream neighbor address is unknown) OR (current MAC is one of multiple active MACs reporting same upstream neighbor address) OR (current MAC's upstream neighbor MAC has not sourced any NIF over last monitor period (i.e., is inactive)))
 - then,
 - 3.2.1 Insert a gap structure to the right of the current MAC.

- 3.2.2 Cut the logical topology list past the gap node and append it to the active MACs list.
- 3.2.3 Patch logical map.
- endif
- 3.3 If (new upstream neighbor MAC differs from the old upstream neighbor MAC)
 - then,
 - 3.3.1 If (new upstream neighbor MAC is on the logical topology list)
 - then,
 - 3.3.1.1 Remove MACs from (and including) old upstream neighbor MAC to (but not including) new upstream neighbor MAC and append to active MACs list.
 - else
 - 3.3.1.2 Remove new upstream neighbor MAC from current list and insert to right of the current MAC.
 - endif
 - endif
 - 3.4 Move to the next upstream MAC by setting the current MAC to be the new upstream neighbor MAC.
- endwhile
- 4. Purge all gaps from the active MACs list. We will determine them afresh if needed.
- 5. If (active MACs list is nonempty)
 - then,
 - 5.1 Insert a gap structure to the right of the current MAC in the logical topology list
 - 5.2 Patch logical map
 - endif
- 6. Return SUCCESS

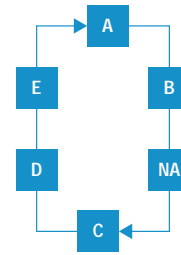


Fig. 5. The primary ring of an FDDI network.

- then,
 - 2.3.1.1 Insert a gap structure to the right of the current MAC.
 - 2.3.1.2 Append entire sequence list to the logical topology list
 - 2.3.1.3 Set sequence complete to TRUE
- endif
- 2.3.2 If (current MAC's upstream neighbor is on the logical topology list and is the first of a sequence)
 - then,
 - 2.3.2.1 Insert entire sequence list to the left of the upstream neighbor MAC.
 - 2.3.2.2 Set sequence complete to TRUE
 - endif
- 2.3.3 If (current MAC's upstream neighbor MAC is on the active MACs list)
 - then,
 - 2.3.3.1 Remove upstream neighbor MAC and append to sequence list
 - 2.3.3.2 Move to next upstream MAC by setting the current MAC to be the upstream neighbor MAC
 - endif
- endif
- endwhile

Patching the Logical Map

This functionality is invoked only if for any reason there is a gap in the logical map information. In the event of multiple gaps the map is broken into sequences such that the logical token flow information is consistent within a sequence. This algorithm uses a temporary list of MAC structures to build and store each sequence. Let us call it the sequence list.

- 1. Purge all gaps from the active MACs list. We will determine them afresh if needed.
- 2. While (active MACs list is nonempty)
 - do
 - 2.1 Remove the MAC at the head of the active MACs list and append it to the sequence list. Let this be the current MAC.
 - 2.2 Set sequence complete to FALSE
 - 2.3 While (sequence complete is FALSE)
 - do
 - 2.3.1 If ((current MAC's upstream neighbor address is unknown) OR (current MAC's upstream neighbor MAC has not sourced a NIF over the last monitor period) OR (current MAC's upstream neighbor MAC is on the logical topology list but is not the first MAC of a sequence))

Logical Mapping Example

Consider the ring in Fig. 5, where the labels on the nodes refer to their MAC addresses. The token flow is as indicated.

The HP Network Advisor protocol analyzer with the FDDI Ring Manager application is the station labeled NA between nodes B and C. Let the sequence of NIFs monitored by the Network Advisor in the first monitor interval (say 31 seconds) be as listed below in increasing order of time.

Source Address	Upstream Neighbor Address
D	C
NA	B
E	?? (unknown)
B	A
A	E

In this list there are two gaps in our information: (1) Node E reports an unknown upstream neighbor address, and (2) There is no NIF from node C (maybe because of a poor implementation) within the entire monitor interval, so we are unable to verify its existence on the ring.

Since the last NIF monitored within the monitor interval was from node A, we shall start building the logical map from it.

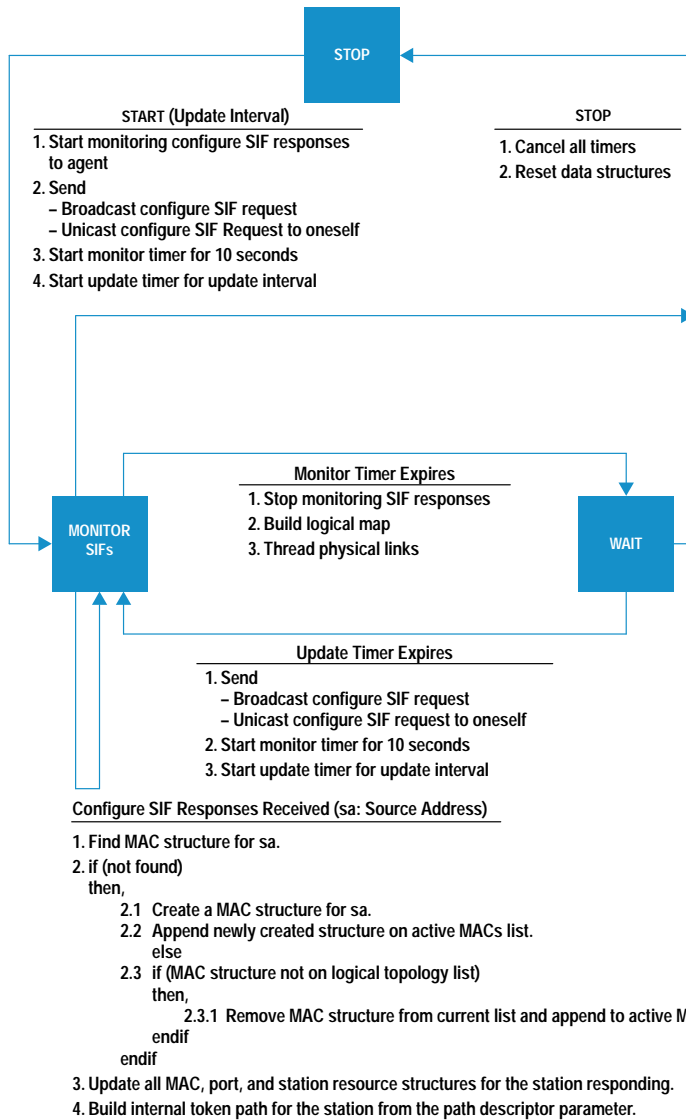
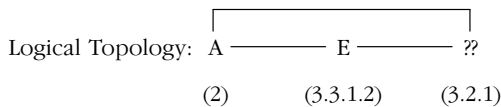
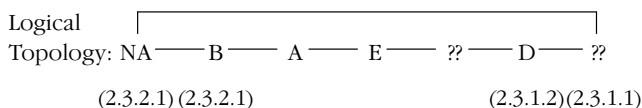


Fig. 6. State diagram for the physical mapping process.

On applying the build logical map algorithm we build a partial sequence before encountering the first gap (represented as ??) in our information. The configurations of the logical topology and active MACs lists at this point are shown below along with the step number of the logical mapping algorithm by which the MAC was added to the logical topology list. Since both lists are double-linked lists, the links are full-duplex.



At this point, the patching algorithm is invoked, which results in the following logical topology. In this case the step numbers reflect the steps of the patching algorithm.



Physical Mapping Process

To build a map of the physical topology of the network, the mapper needs logical as well as internal token path information for all of the active stations on the network. Logical information can be obtained in a passive or an active manner as outlined in prior sections. However, path information can only be obtained by actively querying for it.

The physical mapping process consists of three major steps:

- Obtain the internal token path for each active station on the ring.
- Build the logical topology map of the ring.
- Determine physical links between stations by threading end ports of physical connections using the logical map and the internal token paths of the end stations.

Fig. 6 shows the state diagram for the physical mapping algorithm. The STOP state is the same as for the logical mapping process. On a START directive from a user, the monitor timer is set to an interval (10 seconds) that should be sufficient to receive all responses to the requests. The update monitor is set for a user-defined interval, which needs to be greater than the monitor period.

In the MONITOR SIFs state the mapper is monitoring configure SIF responses destined for it. On receipt of a response, it accesses the station structure for the sourcing MAC and updates the MAC and port resource structures for all resources within it. It transitions to the WAIT state when the monitor timer expires. At this stage it is assumed that the mapper has received all the responses to its requests. The mapper now has all the information to build the logical and physical topology maps. The mapper can be aborted out of this state by a STOP directive.

In the WAIT state the mapper is waiting for the expiration of the update interval at which time it will query the nodes again and transition to the MONITOR SIFs state. The mapper can be aborted out of this state by a STOP directive.

The token flow through the resources of a station, both MACs and ports, is described by the path descriptor parameter of the station management MIB of the station. This information can be requested using configure SIFs or get-PMF request frames. The path descriptor is a mandatory parameter in the configure SIF response. A Network Advisor decode of the path descriptor for a concentrator node is shown in Fig. 7. It contains a record for each of the resources that are physically present in the station, first the ports and then the MACs.

All resources physically present in a station are labeled using sequential indexes. The ports have indexes from 1 to n, where n is the number of ports physically present in the station. The MAC resources are indexed from n+1 to n+m where m is the number of MACs physically present in the station. For a resource in the token path, the connection resource index in its record indicates the index of its downstream neighbor resource within the station. For a resource that is not part of the token path, the connection resource index is its own index. To determine the internal token path, choose a resource that has a connection resource index different from its index. Move downstream to the next neighbor resource by following its connection resource index. By following the connection indexes of the resources in this fashion we can traverse the internal token path, setting both upstream and downstream neighbor resource links appropriately in the process. For the concentrator of Fig. 1 the internal token path and the resource links that reflect it are shown in Fig. 2.

Threading Physical Links

The task of determining the physical links consists of threading through the token path of the entire ring starting from a rooted MAC and moving upstream through the resources on the token path. During this traversal, the algorithm links remote port resources together to reflect actual physical links between these ports. After a successful completion of the algorithm, it is possible to identify the actual physical topology by following the resource and remote port links of the active resources on the ring. The logical topology list provides a basis for the threading process. It represents all active MACs on the ring in the reverse order of token flow. The threading algorithm is as follows:

1. Clear the remote port link for all ports of all stations that we know of. These links will be established afresh as part of the threading process.

```

Frame: 27      Time: Mar 21@ 9:10:59.7015576 Length: 197
***** DETAILED FORMAT *****
Frame Class    2          SIF Configuration
Frame Type     3          Response
Version Id     1          Version 6.2
Transaction Id 303174162
Station Id (User) 0000      User defined section
Station Id (IEEE) DEC----98091E IEEE defined section
Header Pad     0
Info Field Length 160      Total length of station management
parameters present
.
.
.
Parameter     00-08      Path Descriptor
Length         88          Remaining parm octets
Pad/MIB Index (7.2) 0
Port Type     0          Type A
Connect State  3          Active
Remote Port Type 1          Type B
Remote MAC    1          Remote MAC Indicated
Con Resource Index 3          Port Connection Index
Pad/MIB Index (7.2) 0
Port Type     1          Type B
Connect State  3          Active
Remote Port Type 0          Type A
Remote MAC    0          None present
Con Resource Index 1          Port Connection Index
Pad/MIB Index (7.2) 0
Port Type     3          Type M
Connect State  3          Active
Remote Port Type 1          Type B
Remote MAC    1          Remote MAC Indicated
Con Resource Index 4          Port Connection Index
Pad/MIB Index (7.2) 0
Port Type     3          Type M
Connect State  3          Active
Remote Port Type 0          Type A
Remote MAC    1          Remote MAC Indicated
Con Resource Index 10         Port Connection Index
Pad/MIB Index (7.2) 0
Port Type     3          Type M
Connect State  1          Connecting
Remote Port Type 4          Unknown
Remote MAC    0          None present
Con Resource Index 5          Port Connection Index
Pad/MIB Index (7.2) 0
Port Type     3          Type M
Connect State  1          Connecting
Remote Port Type 4          Unknown
Remote MAC    0          None present
Con Resource Index 6          Port Connection Index
Pad/MIB Index (7.2) 0
Port Type     3          Type M
Connect State  1          Connecting
Remote Port Type 4          Unknown
Remote MAC    0          None present
Con Resource Index 7          Port Connection Index
Pad/MIB Index (7.2) 0
Port Type     3          Type M
Connect State  1          Connecting
Remote Port Type 4          Unknown
Remote MAC    0          None Present
Con Resource Index 8          Port Connection Index
Pad/MIB Index (7.2) 0
Port Type     3          Type M
Connect State  1          Connecting
Remote Port Type 4          Unknown
Remote MAC    0          None Present
Con Resource Index 9          Port Connection Index
Pad/MIB Index (7.2) 0
Port Type     3          Type M
Connect State  3          Active
Remote Port Type 2          Type S
Remote MAC    1          Remote MAC Indicated
Con Resource Index 11         Port Connection Index
MAC Address    DEC----98091E
Con Resource Index 2          Mac Connection Index
----End----

```

Fig. 7. Network Advisor decode of the path descriptor parameter in an SIF configuration response.

2. Set the start MAC to be any rooted MAC from the logical topology list. (This choice of the start MAC works fine only if the ring is neither wrapped nor twisted. Those cases are discussed in the next section.) Set the current MAC to be the start MAC.
 3. Set the end MAC to be the first MAC downstream of the start MAC that belongs to a different station than the start MAC. The end MAC can easily be found by traversing the logical topology list in the downstream direction and comparing the station ID for the host station of each traversed MAC with the start MAC.
 4. While (current MAC different from end MAC)
 - do
 - 4.1 Find the first MAC upstream of the current MAC on the logical topology list that belongs to a different station than the current MAC. Let this be the next MAC.
After step 4.1 we have two MACs that belong to neighboring stations. The next step is to find the end ports for the physical connection between these stations.
 - 4.2 Traverse the internal token path from the next MAC in the downstream direction until a port resource is reached. Let this be the next port.
 - 4.3 Traverse the internal token path from the current MAC in the upstream direction until a port resource is reached. Let this be the current port.
 - 4.4 While (current port is already linked to another remote port)
 - do
 - 4.4.1 Current port = current port's remote port
 - 4.4.2 If (current port is same as next port)
 - return SUCCESS
 - This happens in the case of a wrapped station.
 - 4.4.3 Traverse the internal token path from the current port in the upstream direction until a port resource is reached
 - endwhile
- At this point we have two end ports of a physical connection. Set the port structure links to reflect this derived information.
- 4.5 Link the remote port links together:
 - remote port of current port = next port
 - remote port of next port = current port
 - 4.6 Move to the next MAC. It becomes the new current MAC.
- endwhile
5. Return SUCCESS

Problem Situations

The algorithms discussed so far in this article describe the basic framework for generating logical and physical views of FDDI ring topology. They work well for rings having conforming nodes attached to them. Real FDDI installations pose special challenges to the above schemes, especially the one for building the physical maps. By the very nature of the map, a lot of information from all of the active nodes on the ring, which may be from different vendors, needs to fit together somewhat like a jigsaw puzzle to get a consistent

picture of the physical links. For a variety of reasons it may not always fit.

The basic threading algorithm as described above works well if for any two neighboring MACs on the logical map that belong to separate physical nodes, there exists a physical link connecting these nodes. Let this be known as the threading condition. Most problems or complications of the threading process arise when for any reason this condition is violated. It is therefore important to detect all such violations and handle them appropriately. We made significant enhancements to the basic structure of the above algorithms to deal with a variety of such situations, some anticipated and others not. These situations can be broadly classified as topological variations, MACless nodes, incomplete information, and inconsistent information.

Topological Variations. This category of problem situations includes different ring topologies such as a twisted ring, a wrapped dual ring, and a wrapped and twisted ring.

In the event of a twisted ring that is not wrapped, some nodes on the ring disappear from the primary ring and move to the secondary ring such that there is loss of communication between the two sets of nodes. Even though all of the nodes are still physically connected to each other in the form of a ring, the HP Network Advisor will see only a subset of the nodes based on the ring (primary or secondary) it is monitoring. For example, Fig. 8a shows a twisted ring that

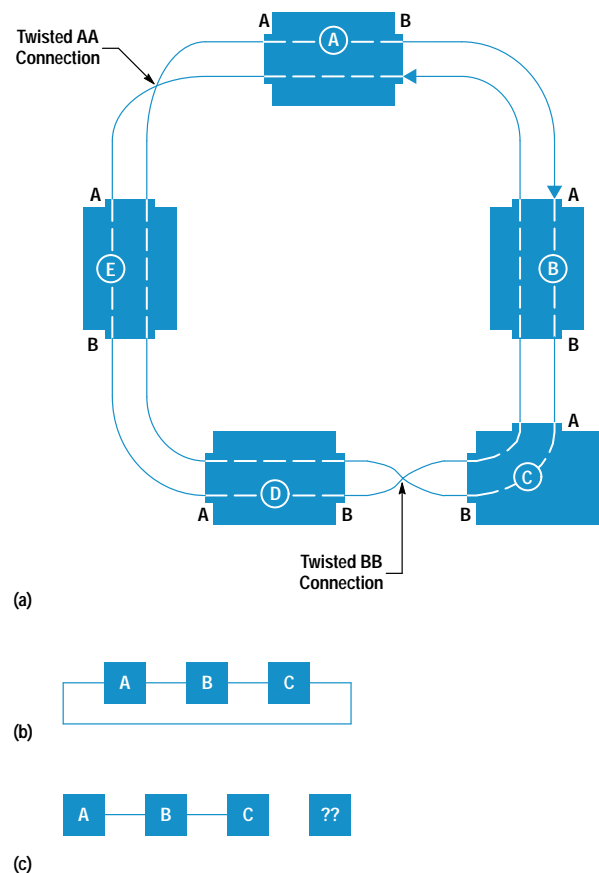


Fig. 8. (a) A twisted dual ring. (b) The output of the basic threading algorithm for the ring in (a). (c) The more accurate physical map generated by the modified threading algorithm.

consists of two disjoint logical rings—A-B-C and E-D, in the order of token flow. If the basic threading process used the first logical ring to determine the physical map, then the result would be as shown in Fig. 8b, which is inaccurate since it shows a direct physical link between nodes C and A, the points of the twist. The ring mapper of the FDDI Ring Manager application of the HP Network Advisor handles this by checking for neighboring stations on the ring that are twisted (it does this by looking at port types) and inserting a gap between them. This results in the map of Fig. 8c, which is as accurate as we can get given the information that we possess.

A wrap on the dual ring causes the primary and secondary rings to coalesce to form one active ring. A wrapped dual ring is handled by choosing the start MAC to be the MAC in the station that is wrapped on its port A and the end MAC to be the first MAC downstream of the start MAC that belongs to a different station. Since the threading process moves upstream, this choice ensures that the traversal of the logical map from the start MAC to the end MAC will not violate the threading condition.

If the wrapped nodes happen to be twisted as well, then it is difficult to use the solution just described for the wrapped dual ring. Typically rings have most if not all of their MACs on the primary ring. Given this knowledge, we need to choose a start MAC that follows the threading condition, that is, there is a physical link to the next upstream node. This can be done by traversing the logical map in the upstream direction from each wrapped MAC to the other, counting the MACs in between. The start MAC chosen is the one that gives the higher count.

MACless Nodes. These nodes pose an especially difficult problem in the sense that they are difficult to detect and even more so to handle. They are invisible to the logical mapping process and are therefore not reflected in the logical map. This causes problems while threading the physical links especially if the MACless node is a rooted concentrator.

For the sake of simplicity, the ring mapper does not handle MACless nodes on the backbone ring. It uses heuristics to place MACless nodes that are connected to a concentrator. It checks the remote port type for each active M-port of the concentrator and tries to match it with the port type of all nodes with one or more MACs connected to them. A mismatch may be an indication of the existence of a MACless device connected to the concentrator. It may also be a result of improper information reported by the concentrator.

Incomplete Information. Lack of responses from any number of active nodes will create a gap in our knowledge about the ring, causing problems in the threading process. These gaps are represented as such in the map. A gap in the map extends from the start of the discontinuity to the next rooted node.

Inconsistent Information. While FDDI is well-defined as a standard, it is still maturing as an implemented LAN technology and there are a large number of semiconforming implementations of station management currently on the market. As a protocol analyzer vendor we need to be able to deal with numerous versions of FDDI products from numerous vendors. The FDDI Ring Manager application is one of the first of its kind that actively queries for management information and pieces it together. It is also the first to deal with the myriad of problems associated with improper responses. Our experience has been that the information in NIFs is more stable and reliable than those in SIF responses. Therefore, the ring mapper's active mode of operation was extended to make monitoring of NIFs optional. However, this does not eliminate inconsistencies, and these are handled much like incomplete information.

The above problem situations get worse when several of them are present on the same ring. As discussed above, the physical mapping process tries to isolate the domain of the problem and uses gaps to signify discontinuities in ring information. Since the physical threading process starts at a MAC and moves upstream, a problem situation at any point in the threading process may cause a gap in information from the current node to the next upstream rooted node.

Conclusion

The topology mapping algorithms discussed in this article provide valuable topology information to the network administrator operator. Being one of the early manufacturers to provide physical mapping capability, we had to deal with a number of problems that stemmed from incomplete implementations of station management. By using the Interoperability Laboratory at the University of New Hampshire and working closely with customers early in the project, we obtained a real understanding of FDDI networks that enabled us to enhance our algorithms to deal with problem situations. This allowed us to deliver a product that our customers could use under adverse conditions. This is viewed as real value by our customers who feel that they have an increased visibility to their FDDI networks.

Acknowledgments

First and foremost, I would like to thank Niel McKee, then working for HP Laboratories Bristol, whose early prototype work with regard to FDDI topology mapping algorithms jump-started this project. Thanks also to Bill Barkley, who designed the FDDI ring status application and the NIF and SIF frame arrival portion of the ring mapper. Many thanks to Ron Pashby and his team at the Interoperability Laboratory at the University of New Hampshire for their feedback and testing efforts. Special thanks to Murali Krishnan, who helped me enormously in isolating and characterizing problems with the software to ensure quality. I also acknowledge the enormous testing effort by all the folks involved. Finally, thanks to Steve Witt who made it all possible.