10. Topology discovery

10.1 Scope

This section describes the RPR topology discovery protocol, which implements a reliable and accurate means for all RPR stations on a ring to discover the initial topology of the stations on the ring and any changes to that topology. The protocol is intended to scale up to hundreds of stations, to cause insignificant overhead for ring traffic, and to cause insignificant overhead on software and ASICs.

The services and features provided are:

- Determine/validate connectivity and ordering of stations on the ring
- Ensure all stations on the ring have a uniform and current image of the topology within 1 ringlet circulation time
- Immediate reaction to changes
- Tolerant of message loss
- Operate without any master station on the ring
- Operate independently of and in the absence of any management systems
- Usable with all supported topologies: ring, linear (broken ring), and isolated station
- Support dynamic addition and removal of stations to/from the ring
- Detect mis-cabling between stations
- Provide means of sharing additional information between stations
- Cause minimal overhead

NOTE—Is an isolated station (above) with its clockwise and counterclockwise links attached to each other a supported topology?

NOTE—Detecting mis-cabling (above) assumes static local assignment of ring IDs (as opposed to dynamic discovery).

NOTE—Not detecting dynamic link status is a contentious issue. Some members would like to keep topology and protection quite separate, while some would like to have them linked tightly into one protocol.

The RPR topology discovery protocol is used to discover the physical link configuration between stations. It is not within the scope of the RPR topology discovery protocol to determine the dynamic link status information, i.e. which ringlet links are up or down, ring segment failures, etc. The discovered topology is used by other protocols such as the RPR protection protocol and the RPR congestion avoidance protocol.

10.2 Algorithm overview

The RPR topology discovery protocol provides each station on the ring with knowledge of the number and arrangement of other stations on the ring. This collection of information is referred to as the **topology image**. Each station maintains its own local copy of the topology image for the entire ring. Initially, the station's topology image contains information only about itself.

Ring topology discovery is initiated as needed and periodically. No station acts as a master for the topology image or for the protocol. All ringlet segments that can be discovered are included. A fully connected ring is not needed for the protocol.

In addition to station identifiers and physical connectivity relationships, the topology discovery protocol is also used to propagate additional station information, both that which is used for other parts of this standard, and optionally information beyond the standard.

The messages sent as part of the RPR topology discovery protocol are indicated in the RPR frame header as control frames.

10.2.1 At initialization

At station initialization, the local topology image is initialized to contain only the local station and no links, and the version of the local topology image is initialized to 0. The station starts the topology algorithm by broadcasting a Topology_Status message on all ringlets. Then it continually listens for Topology_Status messages broadcast on the ring, and broadcasts Topology_Status messages periodically and whenever there is a local topology change.

10.2.2 Topology discovery messages

At first bring up, and periodically after that, a station sends a Neighbor_Hello on each of its outgoing ringlets. The Neighbor_Hello message contains the ringlet ID on which it is sent. When a station receives a Neighbor_Hello message, it verifies that it knows who its neighbor is and that the ringlets of both stations are connected with the same value.

At first bring up, and at any point that a station detects a change in status from another station on the ring, it broadcasts a Topology_Status message to all stations on the ring. The Topology_Status message contains all the information about the local station, including its links to its neighbors. When a station receives a Topology_Status message, it updates its local topology image.

10.2.3 Checking topology consistency

It can be easily determined when an image is complete and consistent by examining the image contents. When the contents of the local topology image show station information for each station described in the link information of another station, then the image is complete. When all stations show for every ringlet that all stations on each ringlet have neighbors only in one direction on each ringlet, then the topology image is consistent.

A canonical form for the topology image allows all the stations to eventually arrive at the same image for the topology and to easily compare images.

10.2.4 Topology database and distance determination

The distance to another station can be calculated either through examining the topology image or by using the difference of the received TTL from Max_Ring_Size (255). For each direction, either the topology image must be complete up to the station whose distance is desired, or a Topology_Status message must have been received from the station from the desired direction.

NOTE—The following should probably end up in the OAMP and/or MIB clauses.

For each ringlet, the distance to each station is used as the index into the topology and status database. The values of each entry are: the MAC address of the station, the Mac address of the station clockwise from the station, the MAC address of the station counterclockwise from the station, the protection status of the link immediately after the station, and the available bandwidth of the link immediately after the station. An example table for one ringlet is shown in Table 1.

Table 1—Topology and status database example

values/ distance	local MAC	clockwise neighbor's MAC	counterclockwise neighbor's MAC	next link's availability	next link's fair rate
1	00-10-A4-97-A8-DE	00-10-A4-97-A8-EF	00-10-A4-97-A8-BD	IDLE	3
2	00-10-A4-97-A8-EF	00-10-A4-97-A8-AC	00-10-A4-97-A8-DE	IDLE	4
3	00-10-A4-97-A8-AC	00-10-A4-97-A8-BD	00-10-A4-97-A8-EF	IDLE	3
4	00-10-A4-97-A8-BD	00-10-A4-97-A8-DE	00-10-A4-97-A8-AC	IDLE	3

10.2.5 station_image_version

NOTE—There is not uniform agreement on whether to keep or to remove this section. One good reason to keep it is to use it as a trigger that something has changed on the ring, so a new discovery process should be started. If we keep it, it needs to be added to the above table. If we delete it, we need another mechanism to determine when there have been changes anywhere on the ring. See 10.3.1.

Each station maintains a version number for its local topology image, called the station_image_version. The station_image_version is initialized to 0 to indicate no valid image (other than itself). It is incremented by the local station whenever a change in local status occurs, and sent out in the resulting Topology_Status message. Change in local status is defined by change in neighbor ID. Each station maintains an independent station_image_version.

10.2.6 Determination and validation of ringlet ID

Each station determines which interface is associated with which ringlet and assigns the corresponding ringlet ID either through fixed mapping between hardware locations or through configuration. Each topology control message is sent separately on each ringlet, identifying the ringlet on which it is being sent. Any topology control message received on a ringlet different from the ringlet on which it is identified as being sent shall trigger a mis-configuration alarm.

10.3 Topology discovery process

10.3.1 Topology discovery process description

1) On startup
— Trigger
  At start of state machine.
— Action
  a) Set the local station_image_version to 0.
  b) Send a Neighbor_Hello on each ringlet.
  c) Start the Neighbor_Hello_Timer.
  d) Broadcast a Topology_Status message on each ringlet.

2) Station status received
— Trigger
  A Topology_Status message is received.
— Action
  a) If ringlet_id in message matches ID of ringlet on which message was received, then continue.
  b) If station_image_version is higher than stored version, then continue.
  c) Replace the remote station information in the local topology image.
  d) Update the remote station_image_version.
  e) Broadcast a Topology_Status message on each ringlet (if remote station is new).

3) Neighbor hello received
— Trigger
A Neighbor_Hello message is received.

NOTE—If addition and deletion of ringlets is allowed, then these would also trigger this.

— Action
a) If the SA in the Neighbor_Hello is different from the stored neighbor address, then continue.
b) Increment the local station_image_version.
c) Broadcast a Topology_Status message on each ringlet.

4) Status timer pop
— Trigger
The Topology_Status_Timer expires.
— Action
a) Send a Topology_Status on each ringlet.
b) Start the Topology_Status_Timer.

5) Neighbor timer pop
— Trigger
The Neighbor_Hello_Timer expires.
— Action
a) Send a Neighbor_Hello on each ringlet.
b) Start the Neighbor_Hello_Timer.

10.4 Topology discovery message formats

10.4.1 Topology_Status

Topology_Status messages report changes in neighbor identity. They are sent as MAC Control messages with a control opcode of Topology_Status. They are sent as broadcast frames on all ringlets, with TTL of Max_Ring_Size (255), removed by the source station, and with the source MAC set to the actual MAC of the sending station.

The information field of the message is as follows:

Bytes 0..3	station_image_version: unsigned 32-bit integer
Bytes 4..5	station_capabilities: unsigned 16-bit integer
Byte 6	cw_ringlets: unsigned 8-bit integer
Byte 7	ccw_ringlets: unsigned 8-bit integer
Byte 8	cw_station_ringlet_id[0]: unsigned 8-bit integer
Bytes 9..14	cw_station_address[0]: IEEE-48 MAC address
	Above 3 fields repeated as necessary for cw_ringlets
Byte 15	ccw_station_ringlet_id[0]: unsigned 8-bit integer
Bytes 16..21	ccw_station_address[0]: IEEE-48 MAC address
	Above 3 fields repeated as necessary for ccw_ringlets

Byte 22	private_length
Bytes 23...	private_data

Table 2—Topology_Status message format

Parameters (see Table 2 **above for codings):**

Byte displacement values shown are for 1 clockwise ringlet and 1 counter clockwise ringlet.

topology_status_version: The topology_status_version parameter shall be set to the current value of the version of the Topology_Status message. The initial value, as defined in this standard, is 0.

station_image_version: The station_image_version parameter shall be set to the current value of the station_image_version of the sending station. If there is no current local topology image, station_image_version shall be set to 0.

NOTE—station_image_version obviously needs to be kept only if we keep section 10.2.5.

station_capabilities: The station_capabilities parameter contains the capabilities of the sending station. The contents and format of this field are encoded as shown below in Table 3.

Table 3—station_capabilities format

Bit	Value
0	Single transit buffer(0)/Dual transit buffer
1	Ring identifier (1 or 0)
2	Wrapped node (1) / Unwrapped node (0)
3	Wrap protection capable(1)
4-6	Fairness message version
7-13	Weight
14-15	Reserved

cw_ringlets: The cw_ringlets parameter indicates the number of ringlets in the clockwise direction.

NOTE—The cw_ringlets field will be kept if the WG. decides that a ring can have more than 2 ringlets, else it will be deleted.

ccw_ringlets: The ccw_ringlets parameter indicates the number of ringlets in the counterclockwise direction.

NOTE—The ccw_ringlets field will be kept if the WG. decides that a ring can have more than 2 ringlets, else it will be deleted.

cw_station_ringlet_id: The cw_station_ringlet_id parameter carries the ID of the ringlet on which the corresponding station is connected.

cw_station_address: The cw_station_address parameters carry the MAC addresses of the stations clockwise to the sending station. If the station's MAC address is unknown, it shall be all 0's.

ccw_station_ringlet_id: The ccw_station_ringlet_id parameter carries the ID of the ringlet on which the corresponding station is connected.

ccw_station_address: The ccw_station_address parameters carry the MAC addresses of the stations counterclockwise to the sending station. If the station's MAC address is unknown, it shall be all 0's.

private_length: The private_length parameter carries the length, in bytes, of the private_data parameter.

NOTE—In the interest of keeping this a fixed length message (or least fixed for a given number of ringlets), the private_length field either needs to be dropped or moved into some other message. It's left here only as a placeholder until it gets moved or it is decided to drop it.

private_data: The private_data parameter carries any private data desired beyond the data required by the protocol.

NOTE—In the interest of keeping this a fixed length message (or least fixed for a given number of ringlets), the private_data field either needs to be dropped or moved into some other message. It's left here only as a placeholder until it gets moved or it is decided to drop it.

10.4.1.1 When generated

The Topology_Status message is broadcast on the initial start of the RPR topology discovery, upon change of another station, and periodically. The period between broadcasts of the Topology_Status message starts at a value of 2 ms for the first message sent after a changed message, and increasing by factors of 2 up to a maximum of 1000 ms.

NOTE—The values of 2 ms and 1000 ms are arbitrary, and should be chosen after a decision has been made on whether to combine this algortithm with protection and/or fairness.

10.4.1.2 Effect of receipt

The receipt of this message from another station causes the MAC Control sublayer to update its current local topology image.

10.4.1.3 Neighbor_Hello

The Neighbor_Hello message reports the presence and identity of a source station to a neighbor station. It is resent every time the Neighbor_Hello_Timer expires.

Neighbor_Hello messages are sent as MAC Control messages with a control opcode of Neighbor_Hello, as broadcast frames, and with TTL set to 1. This guarantees that they will be received by any neighbor and removed from the ring immediately. The source MAC address is set to the actual MAC address of the sending station.

The information field of the message is as follows:

Byte 0	ringlet_id: unsigned 8-bit integer

Table 4—Neighbor_Hello message format

Parameters (see Table 4 **above for codings):**

ringlet_id: The ringlet_id parameter carries the ID of the ringlet on which the request is sent.

10.4.1.4 When generated

The Neighbor_Hello message is generated on the initial start of the topology discovery, and upon expiration of the Neighbor_Hello_Timer. The value of the Neighbor_Hello_Timer starts at a value of 2 ms for the first message sent after the start of the state machine, and increases by factors of 2 up to a maximum of 1000 ms.

NOTE—The values of 2 ms and 1000 ms are arbitrary, and should be chosen after a decision has been made on whether to combine this algortithm with protection and/or fairness.

10.4.1.5 Effect of receipt

The receipt of this message causes the MAC Control sublayer to validate and (if needed) update the identity of its neighbor.