Topology Discovery Proposal

To IEEE 802.17

Edited by

Jeanne De Jaegher, Alcatel

Jason Fan, Luminous

John Lemon, Lantern

Harry Peng, Nortel

Frederic Thepot, Dynarc

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 minimal overhead for ring traffic, and to cause minimal impact 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
- 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 "star" (single 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

The RPR Topology Discovery Protocol is used to discover the static 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.

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 only as needed. Local topology validation eliminates the need for acknowledgements or periodic broadcasts. 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.

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 image is initialized to 0. The station then starts the topology algorithm. All stations running the algorithm continually listen for Topology_Status messages broadcast on the ring, listen for Neighbor_Hello messages from neighbor stations, and send Neighbor_Hello messages to each of their neighbor stations.

These Neighbor_Hello messages allow stations to learn the status of their neighbor links, and to announce their presence to their neighbors. The Neighbor_Hello message is sent to all neighboring stations, both periodically and immediately whenever the local topology image changes. The Neighbor_Hello message contains a summary of the local topology image that is used to validate that neighbors share the same topology image. When a station receives a Neighbor_Hello message from a neighbor, it checks the image information received against its own image. If the 2 images are different, a topology exchange is initiated.

2.2 On change of status

At any point that a station receives a change in status from a neighbor station, or detects that it and a neighbor station are out of synchronization with each other, it initiates a topology exchange. Topology exchange is done by sending a Topology_Status broadcast 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. The combination of Topology_Status messages whenever there is any change in status, and periodic validation checks via Neighbor_Hello messages allows every node to both learn the full ring topology, and to assure that it has the current, correct topology.

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.

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. A (rolling) generation counter allows detection of changes to the topology image.

Note that the topology image is not changed simply because a link is down or a station fails to respond to a hello query. The topology image is changed only when it is

confirmed via a received Neighbor_Hello message that the station ID of a neighboring station has in fact changed to a new value.

2.3 Station_Image_Version

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 inself). 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 link status, or change in neighbor ID. Each station maintains an independent Station_Image_Version.

2.4 Ring_Image_Version

Each station retains the Station_Image_Version sent in the latest Topology_Status message from each other station on the ring. Each time a new Station_Image_Version is received, the receiving station calculates a checksum of all the Station_Image_Version values in its local topology image (including itself). This checksum is called the Ring_Image_Version. The Ring_Image_Version should be the same in all stations on the ring. A mismatch between neighbors indicates a need to update the topology image. However, mismatches are ignored during the time immediately following a change in topology to avoid excessive messages while the topology stabilizes. This period of time is the set by the Topology_Stabilization_Timer.

2.5 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.

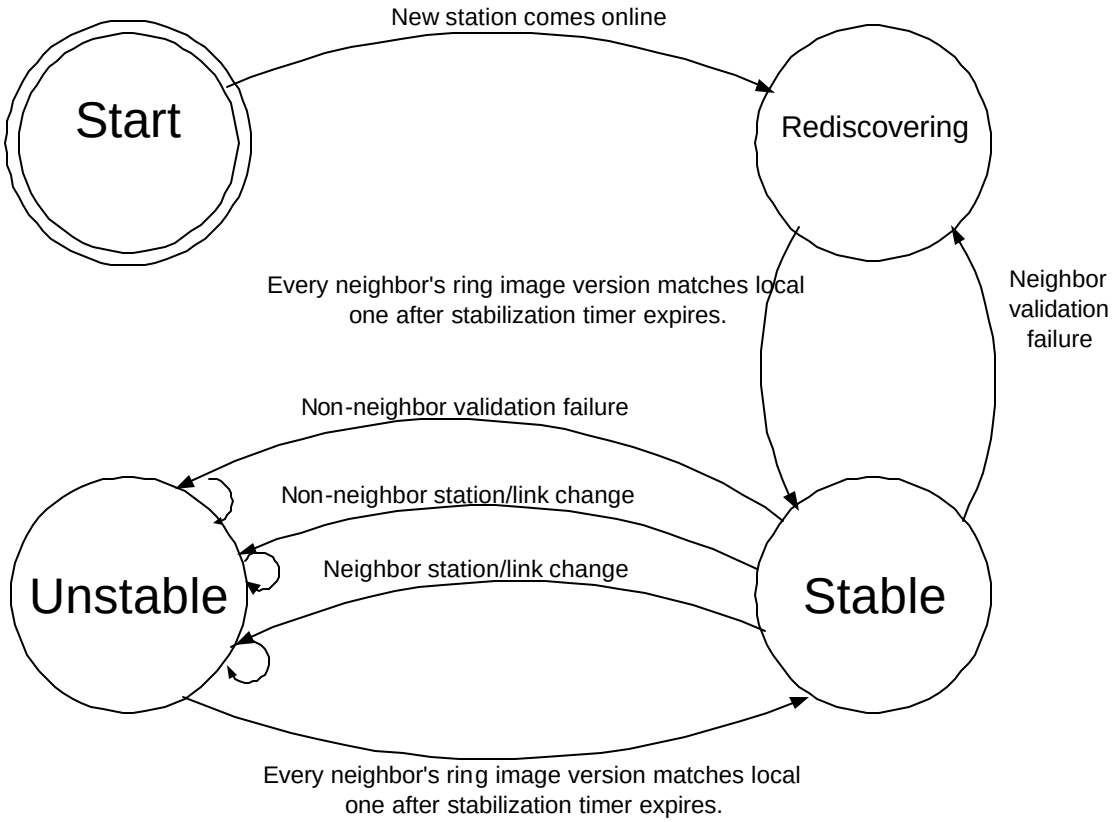
3 Topology Discovery Process

3.1 Topology Discovery Process Description

1. If a neighbor ID or link status changes
    a. Increment the local Station_Image_Version
    b. Broadcast a Topology_Status message
    c. Replace the station information in the local topology image
    d. Update the local Ring_Image_Version
2. If a higher Station_Image_Version is received in a Topology_Status message
    a. Replace the remote station information in the local topology image
    b. Update the remote Station_Image_Version

   c. Update the local Ring_Image_Version
3. If a validation failure is detected due to incorrect Ring_Image_Version (including lack of local topology image)
   a. Set the local copies of the local and all the remote Station_Image_Versions = 0
   b. Send a Topology_Status message
4. If a Station_Change message is received with Station_Image_Version == 0
   a. Update the remote Station_Image_Version to 0
   b. Broadcast a Topology_Status message
   c. Update the local Ring_Image_Version
5. Upon setting the local Station_Image_Version to 0 or updating the Ring_Image_Version
   a. Start the Topology_Stabilization_Timer
   b. While the Topology_Stabilization timer is running, do not compare, and indicate to neighbors not to compare the Ring_Image_Versions

3.2 Topology Discovery Process State Diagram



1. Neighbor station/link change

   - Trigger

No Neighbor_Hello messages in 3 Neighbor_Hello Periods (NHPs) or 2 successive Neighbor_Hellos from a new neighbor in 3 NHPs.

- Action

  a. Increment the local Station_Image_Version.
  b. Broadcast a Topology_Status message.
  c. Replace the station information in the local topology image.
  d. Update the local Ring_Image_Version.

2. Non-neighbor station/link change

   - Trigger

     A higher Station_Image_Version is received in a Topology_Status message.

   - Action

     a. Replace the remote station information in the local topology image.
     b. Update the remote Station_Image_Version.
     c. Update the local Ring_Image_Version.

3. Neighbor validation failure

   - Trigger

     A Ring_Image_Version in a Neighbor_Hello doesn't match the local one, or the local Ring_Image_Version is 0 (a new station).

   - Action

     a. Set the local and all the remote Station_Image_Versions = 0.
     b. Send a Topology_Status message.

4. Non-neighbor validation failure

   - Trigger

     A Topology_Status message with Station_Image_Version = 0.

   - Action

a. Update the remote Station_Image_Version to 0.
b. Broadcast a Topology_Status message.
c. Update the local Ring_Image_Version.

Once in any of the above conditions, start the Topology_Stabilization_Timer. While the Topology_Stabilization timer is running, do not compare, and indicate to neighbors not to compare the Ring_Image_Versions.

4 Topology Discovery Messages

4.1 Topology_Status

Topology_Status messages report changes in neighbor identity or link status. They are sent as control messages, type 0, as a broadcast frame, and TTL of Max_Ring_Size. They are removed by the source station.

The information field of the message is as follows:

Byte 0	Topology_Status opcode
Byte 1	ringlet_id: unsigned 8-bit integer
Bytes 2..5	station_image_version: unsigned 32-bit integer
Byte 6	station_operation_state: {BOOT_STATE = 1, BOOT_CONFIGURED_STATE = 2, RUN_TIME_STATE = 3, RUN_TIME_CONFIGURED_STATE =4, UNKNOWN_STATE =0}
Byte 7	cw_ringlets: unsigned 8-bit integer
Byte 8	ccw_ringlets: unsigned 8-bit integer
Bytes 9..14	cw_station_address[0]: IEEE-48 MAC address
Byte 15	cw_station_in_link_status[0]: status {DISCONNECTED = 1, CONNECTED =2, UNKNOWN =0}
	Above 2 fields repeated as necessary for cw_ringlets
Bytes 16..21	ccw_station_address[0]:
Byte 22	ccw_station_in_link_status[0]: status {DISCONNECTED = 1, CONNECTED =2, UNKNOWN =0}
	Above 2 fields repeated as necessary for ccw_ringlets
Byte 23	private_length
Bytes 24…	private_data

Table 4-1 Topology_Status message format

Parameters (see table above for codings)

(Control message, type 0)

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

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

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.

station_operation_state: The station_operation_state parameter carries the current operational state of the sending station. Valid values are BOOT_STATE, BOOT_CONFIGURED_STATE, RUN_TIME_STATE, RUN_TIME_CONFIGURED_STATE, and UNKNOWN_STATE.

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

cw_station_address: The cw_station_address parameters carry the MAC addresses of the stations clockwise to the sending station.

cw_station_in_link_status: The cw_station_in_link_status parameters carry the current status of the in coming links from the stations clockwise to the sending station. Valid values are DISCONNECTED, CONNECTED, and UNKNOWN.

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

ccw_station_address: The ccw_station_address parameters carry the MAC addresses of the stations counterclockwise to the sending station.

ccw_station_in_link_status: The ccw_station_in_link_status parameters carry the current status of the in coming links from the stations counterclockwise to the sending station. Valid values are DISCONNECTED, CONNECTED, and UNKNOWN.

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

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

4.1.1 When generated

The Topology_Status message is broadcast by the RPR MAC sublayer on initial start of the RPR Topology Discovery, and upon any change in the local copy of the topology image.

4.1.2 Effect of receipt

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

4.2 Neighbor_Hello

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

Neighbor_Hello messages are sent as control messages, type 1. They are sent 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	Neighbor_Hello opcode
Byte 1	ringlet_id: unsigned 8-bit integer
Bytes 2..5	ring_image_version: unsigned 32-bit integer
Byte 5	station_operation_state: {BOOT_STATE = 1, BOOT_CONFIGURED_STATE = 2, RUN_TIME_STATE = 3, RUN_TIME_CONFIGURED_STATE =4, UNKNOWN_STATE =0}
Byte 6	private_length
Bytes 7…	Private_data

Table 4-2 Neighbor_Hello message format

Parameters (see table above for codings)

(Control message, type 1)

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

ring_image_version: The ring_image_version parameter carries the current value of the Ring_Image_Version checksum calculated by the sending station. If there is no current local topology image, Ring_Image_Version shall be set to 0.

station_operation_state: The station_operation_state parameter carries the current operational state of the sending station. Valid values are BOOT_STATE, BOOT_CONFIGURED_STATE, RUN_TIME_STATE, RUN_TIME_CONFIGURED_STATE, and UNKNOWN_STATE.

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

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

4.2.1 When generated

The RTD_HELLO message is generated by the RPR MAC sublayer on initial start of the Topology Discovery, upon expiration of the Neighbor_Hello_Timer, and upon any change in the local copy of the topology image.

4.2.2 Effect of receipt

The receipt of this message causes the MAC sublayer to validate its current local topology image and to broadcast a Topology_Status message if it discovers that the image has changed.