Annex A.

(normative)

Conformance to 802.1 Packet reorder, duplication, and loss requirement

Editors' Notes (MarcH): *To be removed prior to final publication.*

This write-up represents a proposal for flooding packets with mechanisms to adhere to 802.1 packet reorder, duplication, and loss requirements.

The following terms and definitions are used:

Clockwise (CW) - The RPR ringlet where packets are launched in the clockwise direction.

Counter Clockwise (CCW) - The RPR ringlet where packets are launched in the counter clockwise direction.

Flooding - The act of transmitting a packet such that all nodes on the ring receive the packet once.

Flooding Scope (FS) - The number of hops that a packet can travel (around the ring) from a given source station to a destination station associated with a given ringlet.

Context - The topology image (or steering database) used by a source station to transmit a packet.

Protection switch event - A received protection control packet that causes the local topology image to be updated/changed.

Bidirectional flooding - A frame forwarding transfer involving sending two flooding frames. One on each ringlet (CW and CCW), where each frame is directed to distinct adjacent stations.

Unidirectional flooding - A frame forwarding transfer involving sending a flooding frame in the downstream direction, and that frame is directed to its sending station.

A.1 Flooding

Flooding supports two modes of operation.

1) Strict mode: This mode adheres to the 802.1 packet reorder, duplication, and loss requirements. That is,

   i) There is no guarantee that Service Data Units (SDUs) are delivered.
   ii) Reordering of frames with a given user priority for a given combination of destination address and source address is not permitted.
   iii) Duplication of user data frames is not permitted.

   In general, the complexity associated with supporting this mode is particularly required during station or link failure operations.

2) Relaxed mode: This mode of operation adheres to the 802.1 requirements during normal ring operation. In the advent of a ring failure, a minimal amount of reorder and/or duplication can be encountered.

The flooding mode of operation is a RPR system state. That is, all stations on the ring must operate in the same mode (i.e., strict or relaxed).

Editors' Notes (MarcH): *To be removed prior to final publication.*

This proposal can be extended to support flooding mode indication on a packet basis if so desired.

It is important to note that adherence to these requirements by the RPR MAC is not only applicable to RPR MACs servicing 802.1D/Q compliant clients (i.e. bridges). RPR MACs servicing other clients need to abide by these requirements as well.

50ms service restoration times still need to be adhered to.

A.1.1 Relaxed mode of operation

This mode of operation is currently outlined. For unidirectional flooding, the source address and/or TTL found in the RPR Header is used to scope the travel of the packet. For bidirectional flooding, the TTL field found in the RPR Header is used to scope the travel of the packet in the CW and CCW direction.

MAC stripping rules associated with this mode are outlined in Clause 6.8.

The RPR frame format associated with this mode is outlined in Clause 8.0.

A.1.1.1 Non conformance scenarios overview

Under normal RPR operating conditions, this mode does not introduce any packet reorder or duplication. However, there are protection scenarios where a nominal amount reorder and/or duplication can occur.

Scenarios where packet reorder or duplication can occur include:

a) After ring (link or station) restoration events.
b) Topology images of stations on the ring are not synchronized.
c) Station failure resulting in pass-thru behavior. That is, packets are sent through the transit path unaffected.

d) Compound ring (link or station) failures resulting in segmented chains.
e) Rapid cascading ring failures.

A.1.2 Strict mode of operation

The remaining sections of this clause will outline the mechanisms required to support strict mode.

A.2 Frame format

Three additional fields are required to support flooding (in strict mode). They are TTL_Base, FloodType, and WrapState.

A proposed frame format to support this technique is shown in Figure A.1.

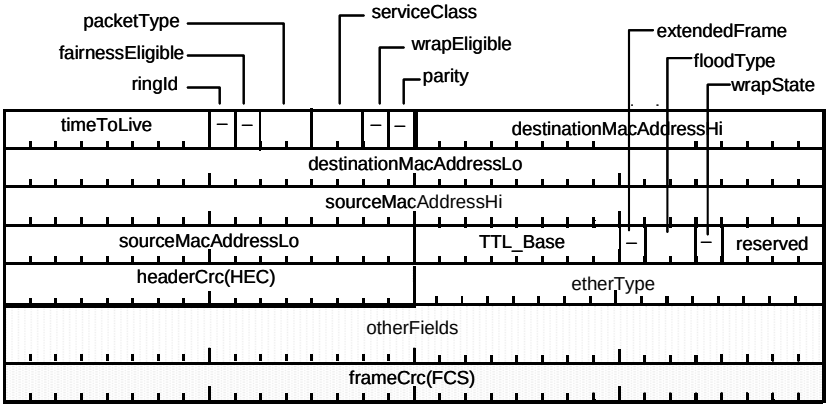

Figure A.1—RPR frame format

The ***Flood_Type*** field is a 2 bit field. It indicates whether the packet is flooded. The values of this field are shown in Table A.1.

Editors' Notes (MarcH): *To be removed prior to final publication.*

May only require a 1 bit floodIndication field. Where bit set to 1 indicates flood, otherwise not flood.

Table A.1—Flood type values

Value	Description
00	no flood
01	unidirectional flood
10	bidirectional flood
11	reserved

The ***WS*** (Wrap State) field is a 1 bit field. It is used by wrapping systems (along with other RPR Control information) to prevent reorder and duplication. It is set to 0 when a packet is first transmitted by a station and set to 1 when a wrapped packet (i.e., packet traveling on secondary ringlet) passes the source station.

The ***Reserved*** field is a 4 bit field. It is available for future use.

The ***TTL_Base*** field is a 1 byte field. It is set to the initial value of the TTL upon transmission of the packet.

The ***extendedFrame*** field is a 1 bit field. It is set to 1 to indicate that a client provided Ethernet packet is contained after the HEC field. Bridging clients will typically set this bit to 1 when relaying packets with remote source and/or destination addresses.

The ***sourceMACAddress*** field is a 48-bit MAC address. It will contain the MAC address of the source transmitting the packet (in strict mode).

The ***destinationMACAddress*** field is a 48-bit MAC address. It can either be a multicast, broadcast, or unicast MAC address.

A.2.1 Format usage

Local transfers refer to locally originated and terminated traffic on the ring. The RPR Control2 field grouping refers to the *TTL_Base*, *WrapState*, and *Flood_Type* information. Local transfers involve prepending of *RPR_Control* and *RPR_Control2* information, to ensure reliable RPR local delivery, as illustrated in Figure A.2.

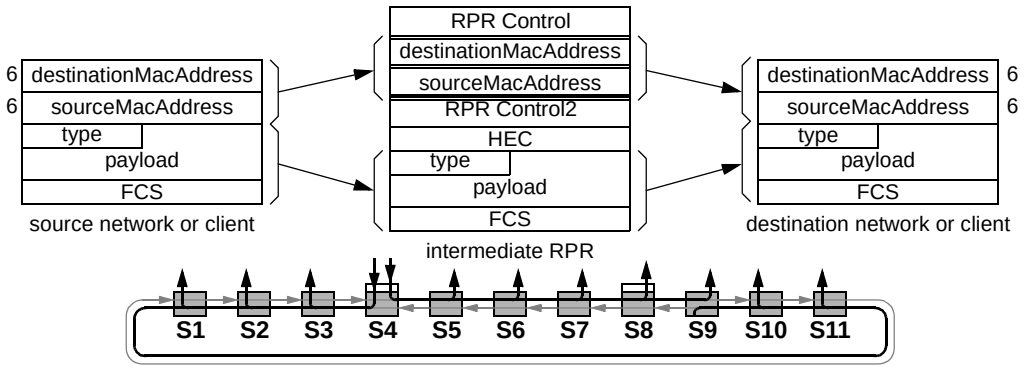

Figure A.2—Local frame forwarding

Remote transfers refer to traffic that is originated from and terminated to a Bridge on the ring. Either the source and/or destination MAC address of the client frame is a remote MAC address. Remotely-sourced transfers involve prepending of *RPR_Control* and *RPR_Control2* information, along with 48-bit *sourceStationID* components to ensure reliable RPR local delivery.

Figure A.3 illustrates a remote transfer that is broadcast over the ring.

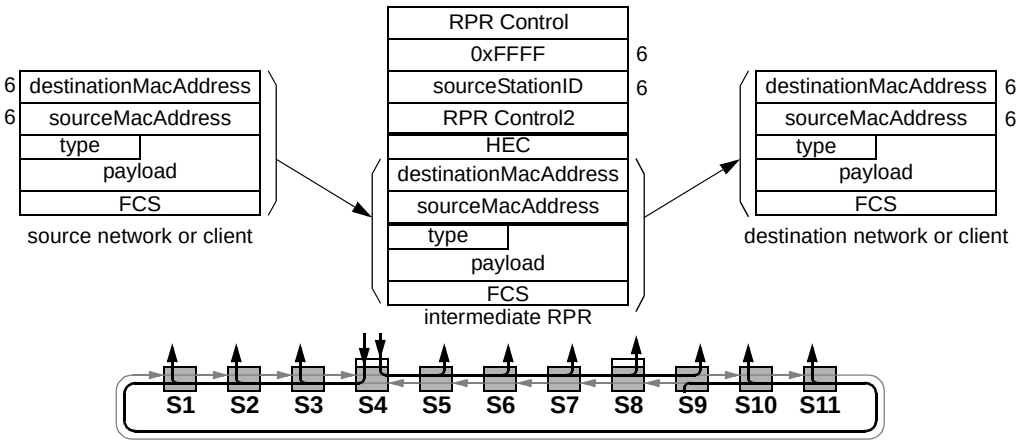

Figure A.3—Remote frame forwarding

Figure A.4 illustrates a remote transfer that will terminate on the ring. This is required to support enhanced bridging applicaitons.

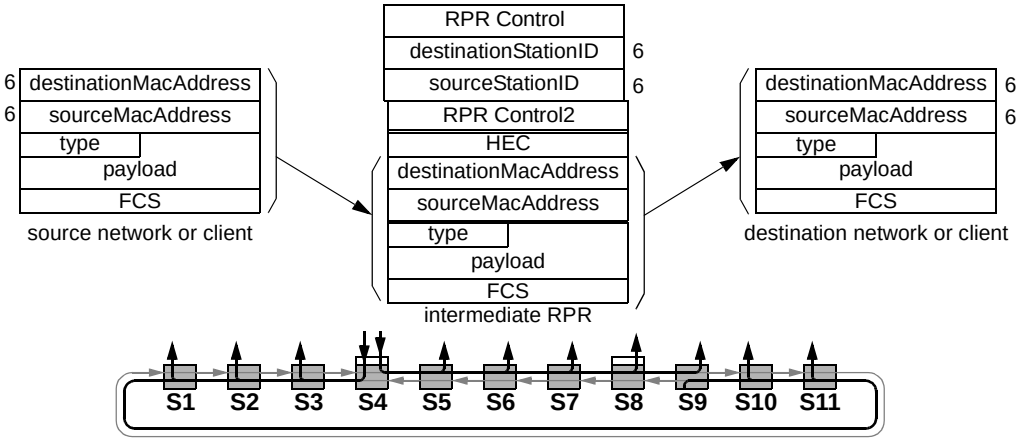

Figure A.4—Remote frame forwarding

A.3 RPR System requirements

The use of the TTL field is to scope the travel of the packet on the ring. The initial value can be set to a system default (e.g., 255) or set to the number of hops that the packet should travel on the ring.

The source address found within the RPR Header is set to the source station address. There is no constraint/ restriction placed on the setting of the destination address (found in the RPR Header).

Editors' Notes (MarcH): *To be removed prior to final publication.*

The destination address could be destinationStationId, remote address, multicast or broadcast address.

Station pass-thru mode is optionally supported. Pass-thru mode can be thought of as being equivalent to an optical regenerator where the MAC operates as a transit node. The MAC enters pass-thru mode when self checking logic determines that the MAC is not sane. In pass-thru mode, TTL is not decremented, and all packets are transited. A station not supporting pass-thru mode will discard all packets (except for IDLE frames if implemented).

Editors' Notes (MarcH): *To be removed prior to final publication.*

This technique could support station pass-thru mode. To accommodate this an addition deletion rule check of *((TTL_Base-TTL)!=hops[SA])* would be required.

A RPR system is either a steering or wrapping system. That is, all stations on the ring steer, or optionally wrap.

The RPR system configuration needs to be bounded in order to guarantee 50ms switch over times. To illustrate this point, consider a ring with ring span of 10 000km. Given that the speed of light is approximately 2E08 meters per seconds, it would take a packet 50ms to travel that ring span. No technique exists that could ensure 50ms protection switching in this case. This technique bounds the maximum distance between any two adjacent stations to be 2 000km.

A.3.1 Steering systems

A key element of this technique is the introduction of a context containment mechanism. This mechanism will ensure that upon detection of a protection switch event, that in-flight data packets that were transmitted by a source using an outdated context gets removed from the ring prior to the transmission of data packets using an updated context. In-flight packets launched using an outdated context are purged from the ring.

A.3.1.1 Context containment

This mechanism is triggered by the reception of a protection control packet (indicating a protection switch is required). Refer to clause 11 for a description of when protection control packets are dispatched and their impact on a station's topology image.

When a station receives a protection switch control packet it will commence to purge all received data packets and purge all data packets currently within the transit buffer(s). This behavior occurs for 15ms. After the 15ms duration has expired, the station may return to normal operations, and dispatches and transits packets as requested.

Editors' Notes (MarcH): *To be removed prior to final publication.*

A 15ms time allocation allows an in-flight packet to travel from one station to an adjacent station and provides a margin to allow the station to remove arriving in-flight packets. A 2000km ring span results in packet delivery from one station to another in 10ms. The additional 5ms is allocated for marginal station processing to remove such packets from the ring and clear the transit buffer(s) of data packets.

Implementation considerations (for single TB systems):

The 15 ms timer can be implemented in SW or HW. For the duration of the timer, the checker entity purges all data packets prior to transfer to PTQ. All data packets leaving the PTQ are also purged.

Implementation considerations (for single TB systems):

This involves storing the current LPTB write pointer and deleting any data packets during read operations while the read pointer has not passed the stored WP. Additionally, any data packets arriving are purged. The 15ms timer can be implemented in SW or HW.

All data packets are purged during this duration. This includes flooded and unicast packets.

The end result of this operation is the removal of data packets in-flight on the ring that were dispatched using a context that is no longer current.

A.3.1.2 Preventing duplication and reorder

The TTL scoping rules used by the source node ensures that no packet duplication occurs. Couple this with station pass-thru being illegal, guarantees that packet duplication is always avoided.

> ***Editors' Notes (MarcH):*** *To be removed prior to final publication.*
>
> NOTE: If station pass-thru mode is to be supported, then the deletion rule check of *(TTL_Base-TTL)!=hops[SA])* can be implemented to guarantee avoidance of packet duplication.

For steering systems, a ring failure breaks the ring and at worst results in packets dropping off the end of the failure point.

The context containment mechanism ensure that there is no reorder of packets. Reorder can occur during a protection switch. Specifically, when a source station transmits packets using one context followed by a different context. Since the flooding scope (FS) of these packets could be different, it is possible for these packets to be delivered to a particular destination station out of order. Context containment removes packets dispatched using an old context before packets are dispatched using a new context. Packet reorder prevention is guaranteed.

A.3.2 Wrapping systems

> ***Editors' Notes (MarcH):*** *To be removed prior to final publication.*
>
> In total alignment with Mike Takefman's proposal for wrapping systems.

A.3.2.1 Preventing duplication and reorder

It is possible that a series of failures can leave the topology significantly different from when the packet was originally launched. A WrapStatus bit in the RPR Header and some eligibility rules for wrapping packets can prevent packet duplication.

The WrapStatus bit is initially cleared when a packet is transmitted by a station. In order to wrap a packet from its primary ring to the secondary ring, the WrapEligibile bit (found in the RPR Control) must be set, and the WrapStatus bit must be cleared and the RI of the packet and station must be the same. When the packet passes the source station on the secondary ring, the WrapStatus bit is set. In order to wrap a packet from the secondary ring to the primary ring, the WrapEligible bit must be set, the WrapStatus bit must be set, and the RI of the packet and station must be different.

The behavior of a wrapping system is illustrated in Figure A.5.

The act of unwrapping the ring will trap packets on the wrong ringlet. Eventually, the TTL deletion mechanism will cause the packets to be deleted. However, in the case of a second fault occurring immediately after the ring recovery, it is likely that not all packets will have been deleted and wrapping these packets onto their primary ringlet will cause reorder.

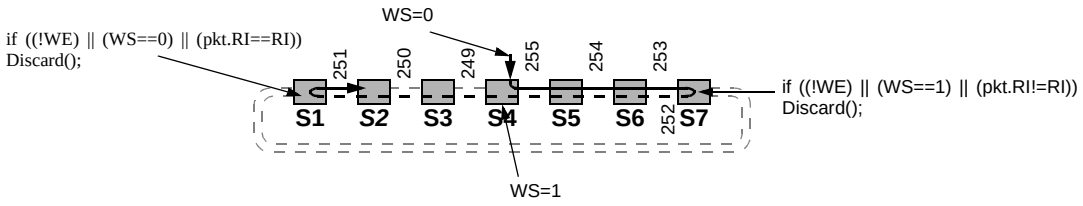

Figure A.5—Wrapping system behavior

Therefore, following the act of unwrapping a ring, all nodes must delete all packets in flight and all packets in the transit buffers whose RI does not match the ringlets RI. This state must persist for 15ms (allowing all packets in flight on a 2000km span to arrive) plus margin.

Editors' Notes (MarcH): *To be removed prior to final publication.*

Implementation considerations (extracted from Mike Takefman's proposal):

Packet deletion from TB on receipt of an unwrap message. This involves storing the current LPTB write pointer and deleting any data packets with the wrong RI during read operations while the read pointer has not passed the stored WP. Additionally, any packets arriving with the wrong RI are purged. The 15ms timer can be implemented in SW or HW.

Note that if a new wrap occurs within this period, some additional packet loss will occur (of the newly wrapped packets) but no reorder will occur.

A.3.3 Reception rules

The duplication-deletion and reorder prevention actions taken by the RPR MAC not currently covered by existing MAC reception rules are shown below. Data packets are discarded if:

a) Station pass-thru
   1) Illegal: Packet received by station are discarded
   2) Supported: *((TTL_Base-TTL)!=hops[SA])*
b) Steering systems:
   1) Protection switch control packet detected. This persist for 15ms.
c) Wrapping systems:
   1) Wrap ingress*: ((WrapState == 1)* || *(WrapEligible == FALSE)* || *(packet.RI!= RI))*
   2) Wrap egress: *((WrapEligible == FALSE)* || *(WrapState == 0)* || *(packet.RI == RI))*
   3) Unwrap: *(packet.RI!= ringlet)*. This persist for 15ms.

A.4 Multicast/broadcast forwarding

Editors' Notes (MarcH): *To be removed prior to final publication.*

Derived from DVJ contribution text.

The most basic multicast/broadcast distribution techniques involves circulating a frame through all stations on the ring. The forwarding techniques for multicast/broadcast transfers are the same as those described for flooded frames.

NOTE—Stations are not expected to optimize the efficiency of multicast forwarding. To reduce complexities, they are expected either support unidirectional multicasts or to forward multicasts and flooded frames in the same fashion.

A.5 Flooding bridge transfers

Editors' Notes (MarcH): *To be removed prior to final publication.*

Derived from DVJ contribution text.

Transparent bridging requires a form of one-to-others distribution called flooding. Flooding protocols (flooding, multicast, and broadcast) require the inclusion of additional information, beyond that included within the client-visible Ethernet frame. That information includes local source station address along with other maintenance/control fields. These addresses assist in scoping the range of the flooding distribution and suppressing undesirable duplicates that might otherwise be generated.

Bridges use the local source station addresses along with the TTL to flood (a flood is a type of broadcast) remote frames for delivery to all bridge clients, as illustrated in the left side of Figure A.6. Flooding involves transmissions between a single source station and all other stations.

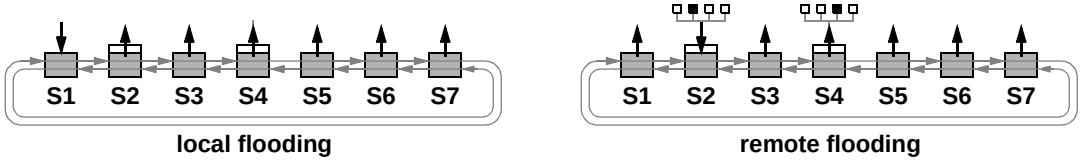

Figure A.6—Basic bridge flooding

With flooding, a frame is placed on the ring by the source, copied by intermediate stations, and stripped at the destination.

Basic-bridging stations maintain simplicity by always flooding, as illustrated in Figure A.6. Although no spatial reuse is possible, this avoids overheads associated with maintaining and utilizing RPR forwarding tables.

A.6 Unicast considerations

Editors' Notes (MarcH): *To be removed prior to final publication.*

Derived from DVJ contribution text.

Local stations improve efficiencies by directing local-unicast traffic to the affected station, rather than flooding this traffic to all others, as illustrated in the left side of Figure A.7. The determination of whether to use flooded or unicast frames is based on a comparison of the frame's destinationMacAddress with the RPR topology database: a unicast is used if a local station matches the same destinationMacAddress; a flood is used otherwise.

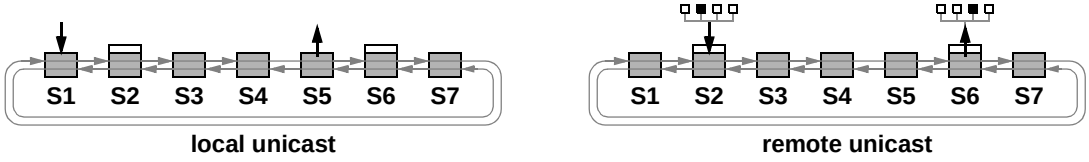

Figure A.7—Enhanced bridge unicasts

Enhanced-bridging stations improve efficiencies by maintaining and utilizing RPR forwarding tables, so that remote frames can also be unicast, as illustrated in the right side of Figure A.7. The learn improves link utilization, due to the frame's unicast (not flooded) and shortest-path nature.

Ordering constraints mandate that flooded and related remote-unicast transfers flow over the same path. The term *related* refers to frames with an identical set of *{sourceMacAddress/destinationMacAddress,VLAN}* identifiers. Flowing over the same path is necessary to maintain ordering, without invoking an inefficient flush between floods and related remote-unicast transfers.

For unidirectional flooding, the potential performance impact of this ordering constraint can be severe, in that the worst case path-length nearly doubles over that associated with bidirectional flooding. To avoid that potential performance impact, enhanced bridges are expected to support bidirectional flooding.

A.7 Flooding alternatives

Two flooding alternatives are provided. They are:

Unidirectional: A frame forwarding transfer involving sending a flooding frame in the downstream direction, and that frame is directed to its sending station. The source address found in the RPR Header is the local source address. The Flood_Type field is set to unidirectional for this flooding alternative.

Bidirectional: A frame forwarding transfer involving sending two flooding frames, one on each ringlet, where each frame is directed to distinct adjacent stations. The scoping of the flooded frames is primarily governed by the TTL within the RPR Header. The Flood_Type field is set to bidirectional for this flooding alternative.

A variety of remote-transfer flows are illustrated in following subclauses, as illustrated in Figure A.8. Downward and upward arrows identify client-to-MAC and MAC-to-client transfers respectively. Downward end-of-flow curves identify locations where frames are stripped. The *x* marker at the end of an error identifies locations where frames are discarded, due to detected inconsistency errors.

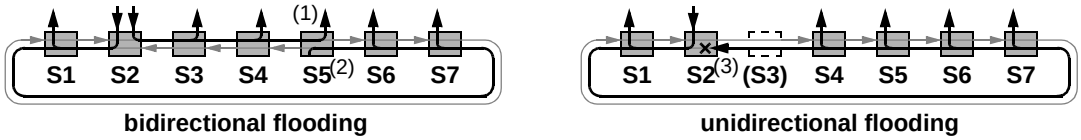

Figure A.8—Flooded receive operations

When a ring is operating with steering based protection, a natural outcome is the absolute need for bidirectional flooding. In fact, there is little need for unidirectional flooding, since any fault forces bidirectional flooding to operate.

When a ring is operating with wrapping protection, bidirectional flooding is not a requirement. In fact, the case concerning the efficiency of bidirectional flooding in support of enhanced bridging argues that supporting bidirectional flooding when wrapping makes little sense. This is due to the inefficiency of sending some of the packets all the way around the opposite ring to be delivered to some of the nodes. It is clearly more bandwidth efficient, to avoid that path entirely and steer the packets instead. Furthermore, given that steering is the baseline of the standard, every station must support bidirectional flooding, therefore the facility is already available.

For completeness, the following subsections elaborate on the various possible flooding and protection combinations.

A.7.1 Flooding with steered protection

Steered flooding involves concurrent transmissions with distinctive nonadjacent station S1 and station S7 failure-point destinations, as illustrated in Figure A.9.

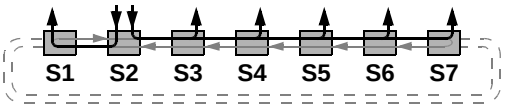

Figure A.9—Steered flooding

From the clients' perspective, flooding on unwrapped and wrapped rings has the same behavior, although the paths of frames changes due to the wrapping at failure points.

A.7.2 Bidirectional flooding

Bidirectional flooding of a ring involves concurrent transmissions on both ringlets, typically directed to a pair of mid-point station, as illustrated in the left and right sides of Figure A.10. A Flood_Type of bidirectional is specified for westside as well as eastside transfers, causing the flooded frame to be passed to the client as each of its removal stations.

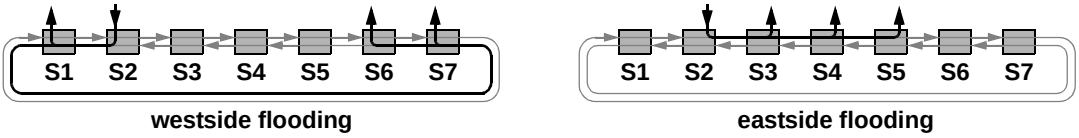

Figure A.10—Bidirectional flooding

Bidirectional flooding with wrap protection involves concurrent transmissions on both ringlets, typically directed to a pair of mid-point station, as illustrated in the left and right sides of Figure A.11. Again, a Flood_type of bidirectional is specified for westside as well as eastside transfers, causing the flooded frame to be passed to the client as each of its removal stations.

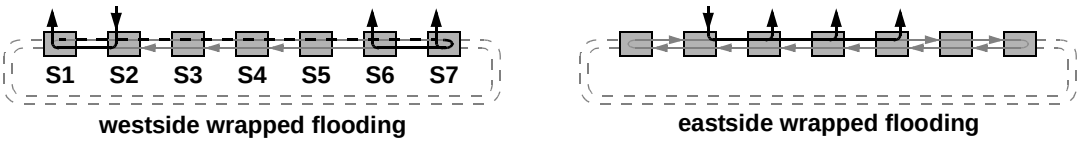

Figure A.11—Bidirectional flooding with wrapped protection

A.7.3 Unidirectional flooding

Unidirectional flooding involves either a westside or eastside transmission directed to the source station, as in the left and right side of Figure A.12 respectively. A Flood_Type of unidirectional is specified, regardless of which direction is selected.

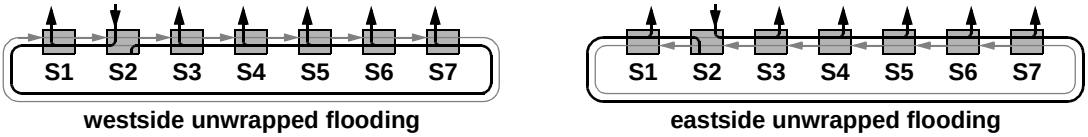

Figure A.12—Unidirectional flooding

Unidirectional flooding with wrapped protection involves either a westside or eastside transmission directed to the source station, as in the left and right side of Figure A.13 respectively. The wrapped flooding opera-

tion relies on the wrap capability at the endpoints. A Flood_Type of unidirectional is specified, regardless of which direction is selected.

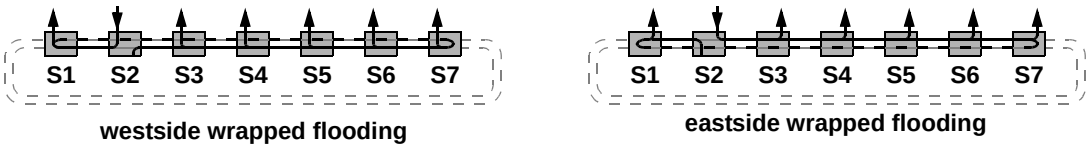

Figure A.13—Unidirectional flooding with wrapped protection

A.7.4 Flood copy rules

Flooding involves selectively copying non-deleted primary-run frames to the client, if a Flood_Type indication of unidirectional or bidirectional is encountered. Packets are stripped from the ring based upon existing packet stripping rules outlined in clause 6.8. Specifically, if source address match or TTL less than 1, the packet is stripped.

A.8 Duplicate scenarios

A.8.1 Duplicate scenarios: Unidirectional source bypass

Unidirectional flooding is susceptible to a source-station-pair loss during flooding, as illustrated in Figure A.14. In this example, source-station *S2* and its upstream neighbor *S3* are both bypassed while the *S2*-sourced frame is circulating. Correct source-bypass processing involves discarding the frame when its recirculates beyond its virtual source, as illustrated by the *x* marks within these Figure A.14.

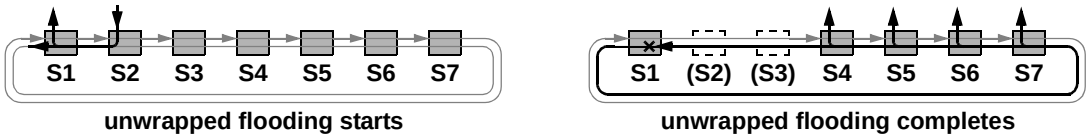

Figure A.14—Duplicate scenarios: Unidirectional source bypass

Cause: The source (that was responsible for packet deletion) disappears before its frame returns.
Problem: The packet passing through stations *S3*&*S2* may be falsely accepted by station *S1* (and others).
Solution: Pass-thru is illegal: Packets will be discarded at S3 (and S2). Pass-thru supported: Station S1 discards packets based on *(TTL_Base-TTL)!=hops[SA]*.

A.8.2 Duplicate scenarios: Unidirectional wrapped source bypass

Unidirectional wrapped flooding is also susceptible to a source-station loss during flooding, as illustrated in Figure A.15. In this example, source-station *S2* and its upstream neighbor *S3* are both bypassed while the *S2*-sourced frame is circulating on the rightside of station *S3*. Correct source-bypass processing involves discarding others' transfers when recirculate beyond the source, as illustrated by the *x* marks within these Figure A.14.

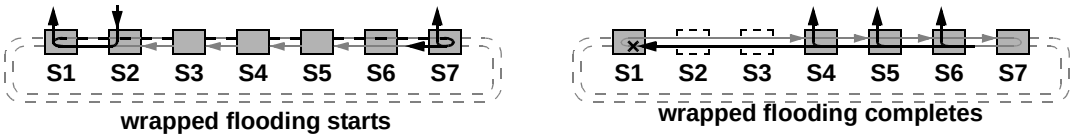

Figure A.15—Duplicate scenarios: Unidirectional wrapped source bypass

Cause: The source (that was responsible for packet deletion) disappears before its frame returns.
Problem: The packet passing through station *S2* may be falsely accepted by station *S1* (and others).
Solution: Pass-thru is illegal: Packets discarded at S2 (and S3). Pass-thru supported: WrapState is not set which prevents wrap exit.

A.8.3 Duplicate scenarios: Bidirectional destination bypass

Bidirectional flooding is susceptible to a destination-station-pair loss during flooding, as illustrated in Figure A.14. In this example, destination stations *S5*&*S6* are bypassed while the *S2*-sourced frame is circulating. Correct destination-bypass processing involves discarding the frame when its circulates beyond its virtual destination, as illustrated by the *x* marks within these Figure A.16.

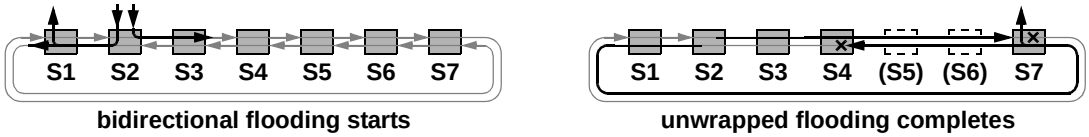

Figure A.16—Duplicate scenarios: Bidirectional destination bypass

Cause: The destination (that was responsible for packet deletion) disappears before its frame arrives.
Problem: The packet passing through stations *S5*&*S6* may be falsely duplicated at station *S4*, *S7*, and others.
Solution: Pass-thru is illegal: Packets would be discarded at station S5 and S6. Pass-thru is supported: Packet discarded at S4 and S7 since *((TTL_Base-TTL)!=hops[SA])* for the received ringlet.

A.8.4 Duplicate scenarios: Bidirectional destination removals

Bidirectional wrapped flooding is susceptible to a destination-station-pair loss during flooding, as illustrated in Figure A.17 In this example, destination stations *S5*&*S6* are removed while the *S2*-sourced frame is circulating. Correct destination-bypass processing involves discarding the frame when its circulates beyond its virtual destination, as illustrated by the *x* marks within these Figure A.17.

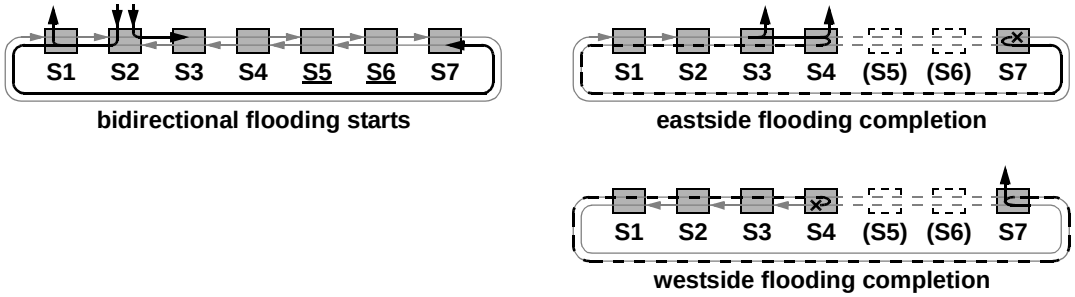

Figure A.17—Duplicate scenarios: Bidirectional destination removals

Cause: The destination (that was responsible for packet deletion) disappears before its frame arrives.
Problem: The packet wrapped before stations *S5*&*S6* may be falsely duplicated at station *S4*, *S7*, and others.
Solution: Pass-thru is illegal: Packets will be discarded at S5 and S6. Pass-thru is supported: S7 and S4 will discard packet based on *((TTL_Base-TTL)!=hops[SA])* for received ringlet check.

A.8.5 Duplicate scenarios: Source&destination removals

Unidirectional flooding could be disrupted when half of the stations (including the source and destination stations) are removed, as illustrated in Figure A.18. In this example, source station *S2* along with stations *S1*,

S7, and *S8* are removed while the *S2*-sourced frame is circulating. Correct processing involves discarding returning frames when their source is missed.

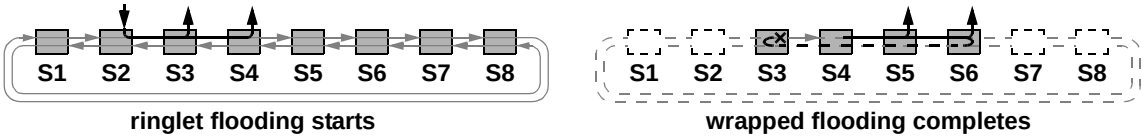

Figure A.18—Duplicate scenarios: Source&destination removals

Cause: The source (that was responsible for packet deletion) disappears before its frame recirculates.
Problem: The packet may be falsely duplicated when recirculated to station *S3* and others.
Solution: Pass-thru is illegal: Packet discarded at S7. If wrap point at S6 is established, packet will not exit wrap at S3 since WrapState bit not set. Pass-thru supported: Packet discarded at S3 the second time around due to *((TTL_Base-TTL)!=hops[SA])* check.

A.9 Reorder scenarios

A.9.1 Reorder scenarios: Protection switch during bidirectional flood

Bidirectional flooding is susceptible to protection switching during flooding, as illustrated in Figure A.18. In this example, while station A is launching bidirectionally flooded packets, a link failure is detected between A and B. When station A updates its steering database (i.e., new context), it will start to launch the bidirectional flooded packets using new flooding scopes derived by the new context. Packet reorder is a concern at station C if in-flight packets launched by station A (using an old context) arrive after packets launched by station A, using the new context.

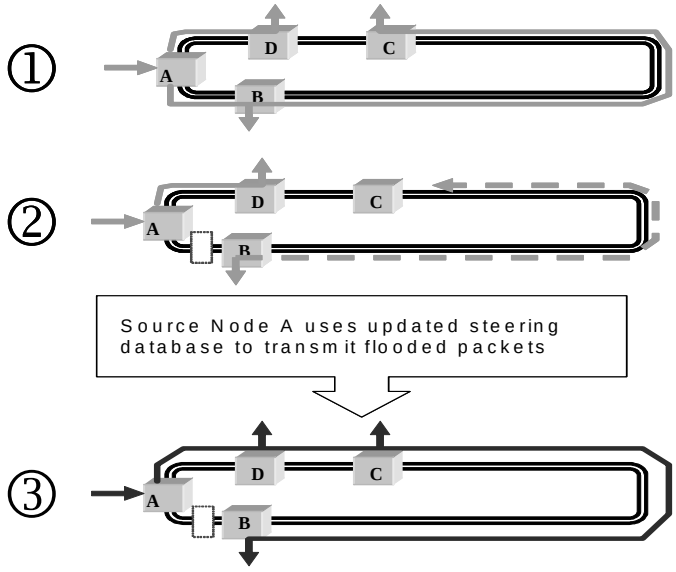

Figure A.19—Reorder scenario: Protection switch during bidirectional flood

Cause: In-flight packets dispatched by source using an old context arrive at a destination after packets dispatched by source using new context.
Problem: The packet received by station C may be received in the wrong order.

Solution: In strict mode, steering systems use the context containment mechanism to ensure in-flight data packet are removed from the ring before packets using a new context are launched

A.9.2 Reorder scenarios: Cascading failures during bidirectional flood

Bidirectional flooding is susceptible to rapid cascading failures occurring during bidirectional transmission, as illustrated in Figure A.19. In this example, while station A is launching bidirectionally flooded packets, the link between station G and E is restored and fails in rapid succession. Packet reorder is a concern at station F and E (in this example) if in-flight packets transmit using the context at step 2 are received before in-flight packets transmitted using the context from step 1.

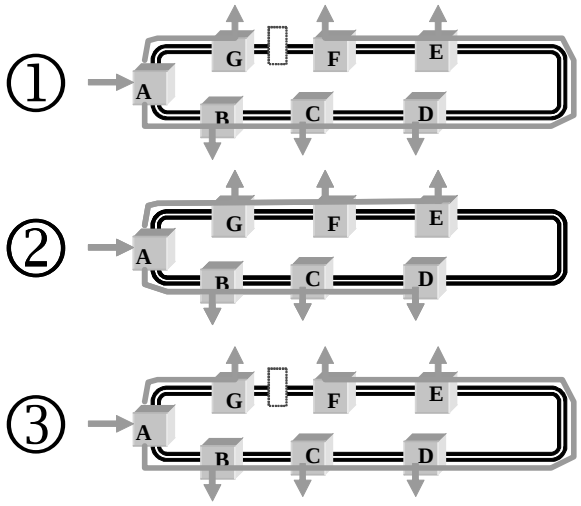

Figure A.20—Reorder scenarios: Cascading failures during bidirectional flood

Cause: At step 1, consider packets in-flight on CCW ringlet (using context #1). At step 2, packets are launched on CW and CCW ringlet (using context #2). Assume station E is now using context #2. That is, station E accepts packets launched from A using context #2. At step 3, station E is using context #3. Assume step 2 and step 3 occur while CCW in-flight packets using context #1 are still in-flight. Station E can accept in-flight packets on the CCW ringlet sourced by A using context #1.
Problem: Packet reorder can occur if station E receives in-flight packets from step 1 after in-flight packets from step 2.
Solution: In strict mode, steering systems use the context containment mechanism to ensure in-flight data packet are removed from the ring before packets using a new context are launched

A.9.3 Reorder scenarios: Protection switch during unicast transmission on steering system

Unicast packet transmission is susceptible to a protection switch event occurring, as illustrated in Figure A.20. In this example, station A is transmitting unicast traffic destined for station C over the CCW ringlet. A link failure occurs between station A and B, causing station A to dispatch the unicast traffic destined to C over the CW ringlet.

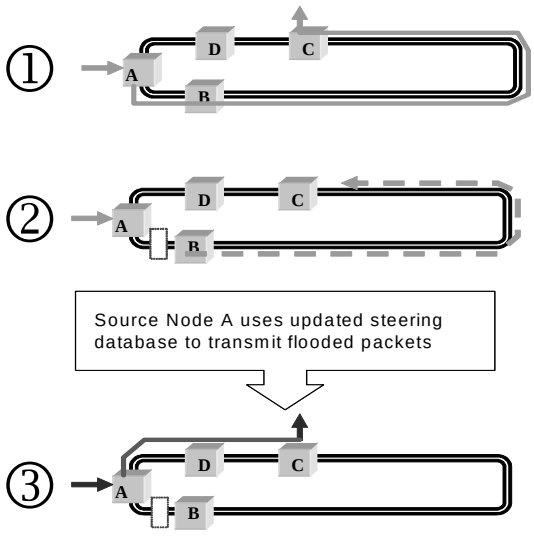

Figure A.21—Protection switching during unicast transmission

Cause: At step 1, consider packets in-flight on CCW ringlet (using context #1). At step 2, station A detects a link failure between station A and B. The context used by station A is updated to reflect configuration shown in step 2. Unicast packets destined to C now are sent over the CW ringlet.
Problem: Packet reorder can occur if station C receive context 2 packets before context 1 packets.
Solution: In strict mode, steering systems use the context containment mechanism to ensure in-flight data packet are removed from the ring before packets using a new context are launched.

A.9.4 Reorder scenarios: Cascading protection switch during unidirectional flood on wrapping system

Unidirectional flooding is susceptible to rapid cascading failures occurring during unidirectional transmission, as illustrated in Figure A.21. In this example, while station A is launching unidirectional flooded packets, the link between station D and E is restored and fails. Packet reorder is a concern at station D (in this example) if packets transmitted at step 3 are received before wrapped packets on secondary ring transmitted at step 1 potentially get unwrapped at station D

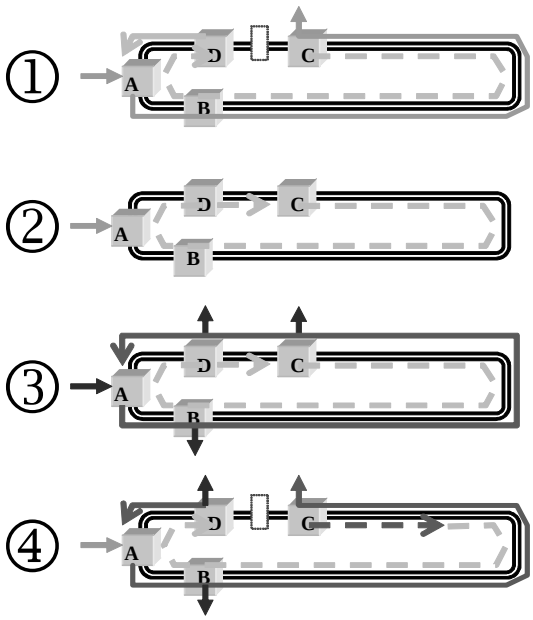

Figure A.22—Reorder scenarios: Cascading failures during unidirectional flood

Cause: At step 1, consider wrapped packets on secondary ringlet. At step 2, the ring heals, however the packets on secondary ringlet continue to circulate until TTL expires. During packet circulation on secondary ringlet, packets are transmitted by station A (as shown in step 3). If another failure occurs (at step 4), prior to circulation of packets on secondary ringlet having their TTL expire, packets launched at step 3 can be received by station D before un-expired packets on the secondary ringlet get exit the wrap condition at station D.
Problem: Packet reorder can occur if station D receive context 3 packets before context 1 packets.
Solution: In strict mode, wrapping systems will purge all packets with packet.RI not equal to the received ringlet for a duration of 15ms. All wrapped packets using an outdated context will be removed from the ring.