Annex A.

(normative)

RPR Frame Transmission Conformance to the 802.1 MAC Service

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

This write-up represents a proposal for frame transmission conformance to the 802.1 MAC service requirements (specifically 802.1 frame reorder, duplication, and loss requirements).

The following terms and definitions are used:

ringlet0 - The RPR ringlet where frames are launched in the clockwise direction.

ringlet1 - The RPR ringlet where frames are launched in the counter clockwise direction.

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

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

Context - The topology and status database used by a source station to transmit a frame.

Protection switch event - A received protection control frame that causes the local topology and status database to be updated/changed.

Bidirectional flooding - A frame forwarding transfer involving sending two flooding frames. One on each ringlet (ringlet0 and ringlet1), 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.

.

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

The specifications described by this contribution focus on the support of a transmission mode that adheres to the 802.1 MAC service requirements of:

- **No reordering of frames with a given user priority, for a given combination of destination and source address.**
- **No duplication of user data frames.**
- **No guarantee that user data frames are delivered (i.e., frame loss is acceptable)**

A.1 Frame Transmission.

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

This section (A.1), and related sub-sections, are intended for the "Clause 1. Overview" of the P802.17 D1.1 draft.

Two types of frame transmission are supported.

1) Strict: Adheres to the 802.1 frame 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 strict transmission is particularly required during station or link failure operations.

2) Relaxed: 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, however the chances of delivery is increased.

The RPR MAC shall support both types of the aforementioned frame transmission.

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), but also RPR MACs servicing other clients (e.g., host, router, etc.).

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

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

Need to clearly specify what the 50ms service restoration requirement implies.

A.1.1 Relaxed frame transmission

This type of transmission is currently outlined. For unidirectional flooding, the *sourceMACAddress* and/or *timeToLive* found in the *rprHeader* is used to scope the travel of the frame. For bidirectional flooding, the *timeToLive* field found in the *rprHeader* is used to scope the travel of the frame on ringlet0 and ringlet1.

MAC stripping rules associated with relaxed transmissions are outlined in Clause 6.8. The associated frame format is outlined in Clause 8.0.

A.1.1.1 Non conformance scenarios overview

Under normal RPR operating conditions, relaxed transmissions do not introduce any frame reorder or duplication. However, there are protection scenarios where a negligible amount reorder and/or duplication can occur.

Scenarios where frame reorder or duplication can occur include:

a) After ring (link or station) restoration events.

b) Topology and status database of stations on the ring are not synchronized.
c) Station failure resulting in passthru behavior. That is, frames are sent through the transit path without *timeToLive* decrement or any other frame processing/stripping rules being applied.
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 frame transmissions.

A.2 Frame format.

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

This section (A.2), and related sub-sections, are intended for the "Clause 8. Frame formats" of the P802.17 D1.1 draft.

No changes are being suggested to the P802.17 D1.1 draft specified RPR control frames (e.g., fairness, topology discovery, idle, etc.).

Additional fields are required to support strict transmission of data frames. They include *ttlBase*, *floodType*, *wrapStatus*, and *strictOrder*.

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

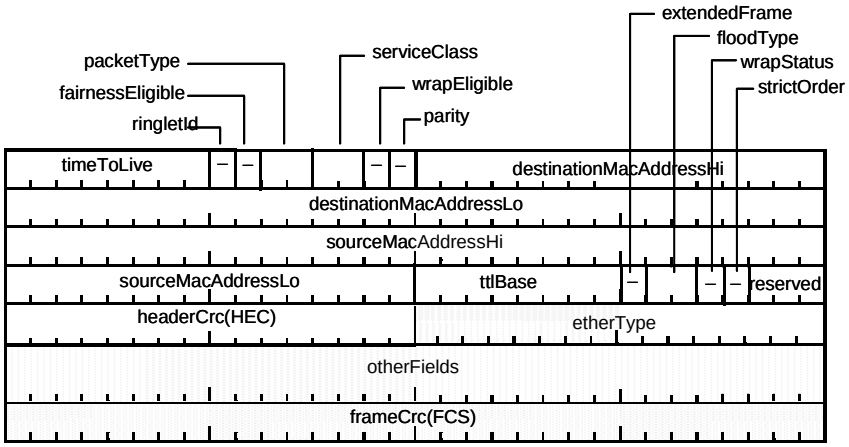

Figure A.1—RPR frame format

The *floodType* (ft) is a 2 bit field. It indicates whether the frame is flooded. The values of this field are shown in Table A.1

Table A.1—Flood type values

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

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

The *strictOrder* (so) is a 1 bit field. It indicates whether the frame should conform to relaxed or strict ordering requirements (as outlined in Section A.1). A value of 0 indicates relaxed, while a value of 1 indicates strict.

The *reserved* field is a 3 bit field. It is available for future use.

The *ttlBase* is a 1 byte field. It is set to the initial value of the *timeToLive* upon transmission of the frame.

The *extendedFrame* (ef) is a 1 bit field. It is set to 1 to indicate that a client provided frame is contained after the HEC field. Bridging clients will typically set this bit to 1 when relaying frames with a remote *sourceMACAddress* and/or *destinationMACAddress*. See Figure A.5 for an illustration.

The *sourceMACAddress* field is a 48-bit MAC address. It will contain the MAC address of the source station transmitting the frame (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

A.2.1.1 Local unicast forwarding

Local unicast forwarding refers to frame transmissions on the ring where the local source station is dispatching a local unicast packet.

The *rprControl2* field depicted in Figure A.2 refers to the fields after the *sourceMACAddress* and before the HEC. Local transfers involve insertion of *rprControl* and *rprControl2* information, to ensure reliable RPR local delivery, as illustrated in Figure A.2.

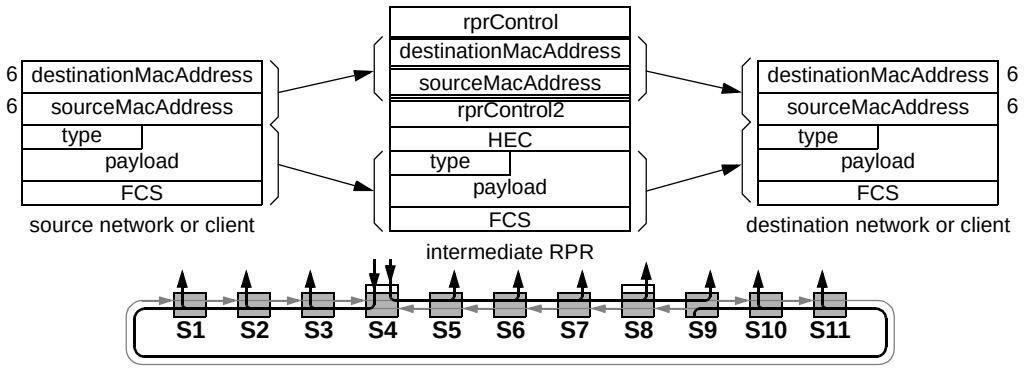

Figure A.2—Local frame forwarding

A.2.1.2 Local multicast, broadcast, and unknown unicast forwarding

Local multicast forwarding refers to frame transmissions on the ring originated by a local station (e.g., host, router) but dispatched to a multicast group. Local broadcast forwarding refers to frame transmission on the ring originated by a local station (e.g., host, router) but dispatched to a broadcast group. Local unknown unicast forwarding refers to frame transmission on the ring originated by a local station (e.g., host, router), where the client provided *destinationMACAddress* is not local to the ring. Local unknown unicast forwarding transmissions are flooded on the ring.

All of the above transmission types are supported by a single frame structure as illustrated in Figure A.3. This is the same frame structure that is used for local unicast forwarding, as illustrated in Figure A.2. Transfers involve insertion of *rprControl* and *rprControl2* information to ensure reliable RPR local delivery.

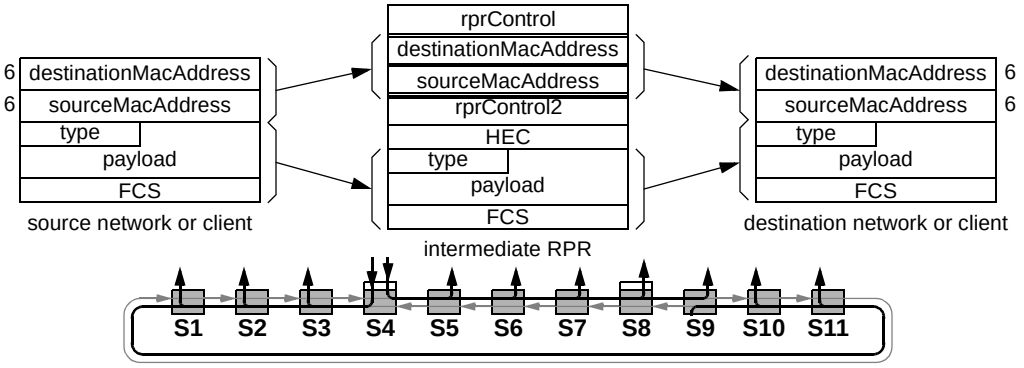

Figure A.3—Local multicast, broadcast, unknown unicast forwarding

A.2.1.3 Remote forwarding

Remote transfers refer to traffic that is originated by a remote device (i.e., an off ring device) and/or terminate to a remote device. Either the *sourceMACAddress* and/or *destinationMACAddress* of the client frame is a remote MAC address. Remotely-sourced transfers involve insertion of *rprControl* and *rprControl2* information, along with setting the 48-bit *sourceStationID* components to ensure reliable RPR local delivery. The *extendedFrame* bit is set to 1 for these transmissions.

Figure A.4 illustrates a remote transfer that is broadcast over the ring.

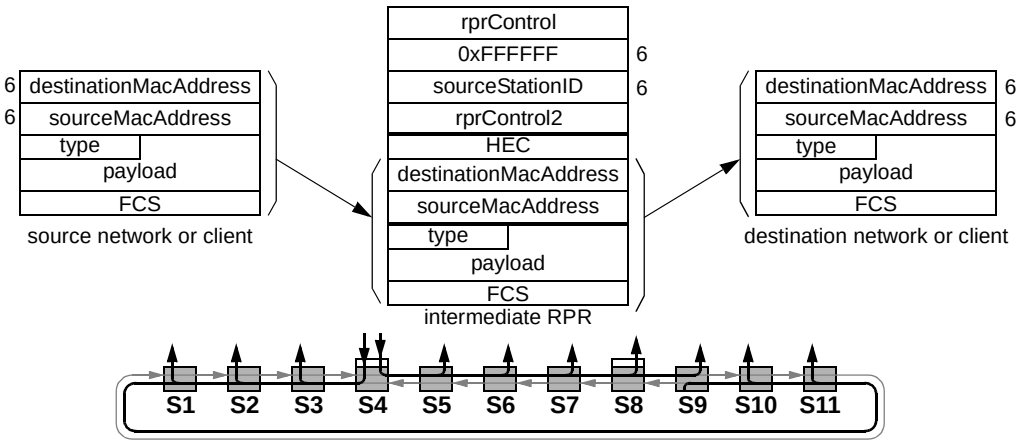

Figure A.4—Example 1: Remote frame forwarding

Figure A.5 illustrates a remote transfer that will terminate on the ring. This is required to support clients such as enhanced bridging.

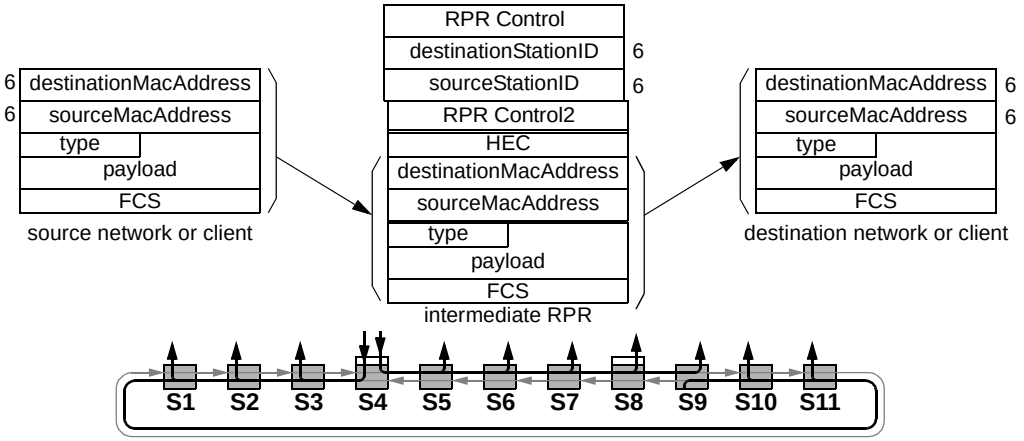

Figure A.5—Example 2: Remote frame forwarding

A.3 RPR System requirements.

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

This section (A.3), is intended for the "Clause 1. Overview" of the P802.17 D1.1 draft.

The use of the *timeToLive* field is to scope the travel of the frame 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 frame should travel on the ring.

The *sourceMACAddress* found within the *rprHeader* is set to the source station address for strict transmissions. There is no constraint/restriction placed on the setting of the *destinationMACAddress* found in the *rprHeader* (i.e., the *destinationMACAddress* could be a destinationStationID, remote address, multicast, or broadcast address).

Station passthru can be thought of as being equivalent to an optical regenerator where the MAC operates as a transit station. The MAC can enter passthru by LME invocation or if self checking logic detects irregularities in the MAC. In passthru mode, *timeToLive* is not decremented, and all frames are transited.

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 service restoration 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 frame 50ms to travel that ring span. No technique exists that could ensure 50ms protection switching in this case. This technique bounds the maximum ring circumference to be 2 000km.

A.3.1 Steering systems

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

This section (A.3.1), and related sub-sections, are intended for the "Clause 6. Medium access control data path" of the P802.17 D1.1 draft. Sections 6.8 and 6.9 are recommended.

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, resulting in a change to the topology and status database, that in-flight strict data frames that were transmitted by a source using an outdated context gets removed from the ring prior to the transmission of strict data frames using an updated context. In-flight strict data frames launched using an outdated context are effectively purged from the ring.

A.3.1.1 Context containment

A protection switch event is a protection control frame. This mechanism is triggered by the reception of a protection control frame (resulting in changes to the local topology and status database). Refer to clause 11 for a description of when protection control frames are dispatched and their impact on a station's topology and status database.

When a station receives a protection switch control frame it will commence to purge all received data frames (with *strictOrder* set to 1) and purge all data frames (with *strictOrder* set to 1) currently within the transit

buffer(s). This behavior occurs for a specified duration[1]. After the duration has expired, the station may return to normal operations, and dispatches and transits frames as requested.

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

The duration of the purge is under investigation.

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

Implementation considerations (for single TB systems):

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

Implementation considerations (for dual TB systems):

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

All data frames (with *strictOrder* set to 1) are purged during this duration. This includes flooded and unicast frames.

Consider the illustration in Figure A.6 below. A link failure occurs between stations G and F. Protection control frames are dispatched by the stations adjacent to the failure (i.e., stations G and F). The context containment mechanism is triggered in stations G and F when the fault is detected and they have to update their local topology and status database. The context containment mechanism is triggered at the interior stations (i.e., stations A, B, C, D, and E), when they receive a protection switch event (i.e., a protection control frame resulting the local topology and status database to be changed).

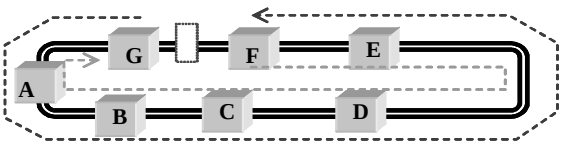

Figure A.6—Protection Event Example

The end result of this operation is the removal of strict data frames on the ring that were dispatched using a context that is no longer current.

A.3.1.2 Preventing duplication and reorder

The *timeToLive* scoping rules used by the source station along with the deletion checking rule of *(SRC[NumberOfHops]!=frame.SA)* ensures that no frame duplication occurs. The linear *SRC[NumberOfHops]* table returns the MAC address of the station, *NumberOfHops* away from the local station (i.e., station that received downstream frame) in the upstream direction.

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

[1]For example, the duration could be specified by a discovery provided purgeNow=1 indication. Alternatively, a duration of 15ms could be generally specified.

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

A.3.2 Wrapping systems

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

This section (A.3.2), and related sub-sections, are intended for the "Clause 6. Medium access control data path" of the P802.17 D1.1 draft. Sections 6.8 and 6.9 are recommended.

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 frame was originally launched. A *wrapStatus* bit in the *rprHeader* and some eligibility rules for wrapping frames can prevent frame duplication.

The *wrapStatus* bit is initially cleared when a frame is transmitted by a station. In order to wrap a frame from its primary ringlet to the secondary ringlet, the *wrapEligibile* bit (found in the *rprControl*) must be set, and the *wrapStatus* bit must be cleared and the *ringletId* of the frame and ringlet traveled must be the same. When the frame passes the source station on the secondary ringlet, the *wrapStatus* bit is set. In order to wrap a frame from the secondary ringlet to the primary ringlet, the *wrapEligible* bit must be set, the *wrapStatus* bit must be set, and the *ringletId* of the frame and ringlet traveled must be different.

The behavior of a wrapping system is illustrated in Figure A.7.

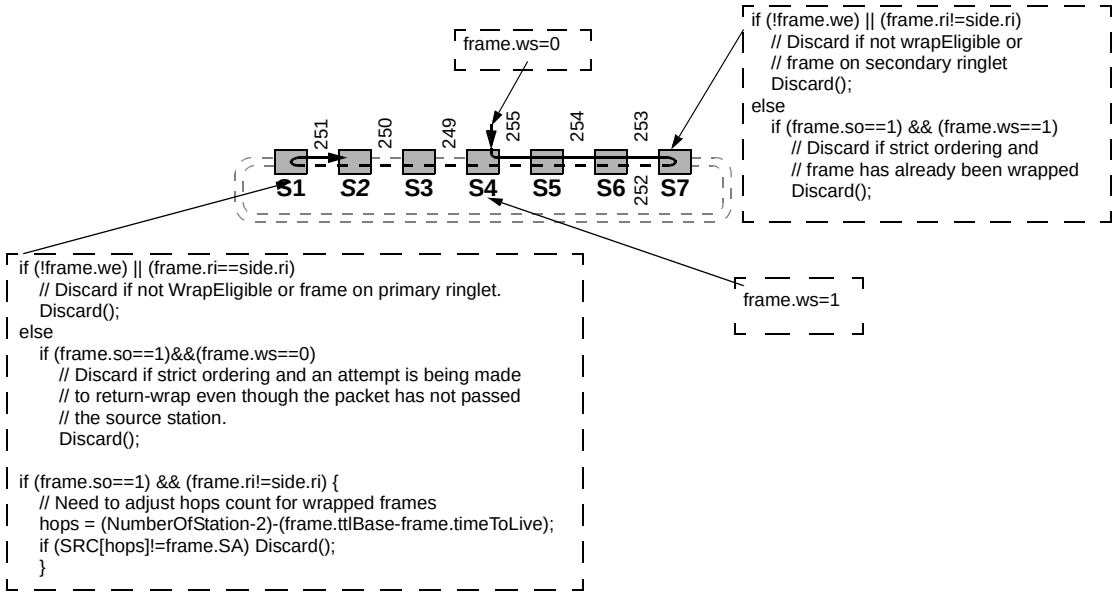

Figure A.7—Wrapping system behavior

The *timeToLive* is decrement when the frame enters the secondary ringlet (i.e., initially wrapped). While the frame travels on the secondary ringlet, the *timeToLive* is no longer decremented. When the frame is wrapped back to the primary ringlet, normal *timeToLive* decrement operation continues.

The act of unwrapping the ring (i.e., ring recovery) will trap frames on the wrong ringlet. These frames will be discarded from the secondary ringlet if no protection event is current.

However, in the case of a second fault occurring immediately after the ring recovery, it is likely that not all frames will have been discarded and wrapping these frames onto their primary ringlet will cause reorder. Therefore, following the act of unwrapping a ring, all stations must delete all strict data frames in flight and all strict data frames in the transit buffers whose *ri* does not match the ringlet traveled. This state must persist for a specified duration (allowing all frames in flight on a 2000km span to arrive).

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

Implementation considerations for the deletion of strict data frames from TB on receipt of an unwrap message.

This involves storing the current LPTB write pointer and deleting any strict data frames with the wrong *ri* during read operations while the read pointer has not passed the stored WP. Additionally, any strict data frames arriving with the wrong *ri* are purged. For example, a 15ms timer can be implemented in SW or HW.

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

A.4 Reception rules

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

This section (A.4), is intended for the "Clause 6. Medium access control data path" of the P802.17 D1.1 draft. Sections 6.8 and 6.9 are recommended.

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

Data frames on the primary ringlet, with a strict indication (i.e., *strictOrder* value of 1) are discarded if the following deletion rule is true (*SRC[frame.ttlBase-frame.timeToLive]!=frame.SA)*. Both wrapping and steering systems perform this check, at all ring stations, on frames traveling on the primary ringlet (i.e., *frame.ri==side.ri*). Figure A.8 provides an illustration.

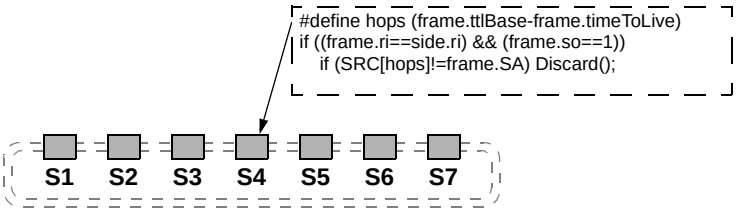

Figure A.8—Source consistency check

In addition:

a) For steering systems, protection switch events will trigger the context containment mechanism (refer to section A.3.1.1).
b) For wrapping systems refer to Figure A.7. Additionally, when a ring unwraps (i.e., ring recovery), the context containment mechanism is applied to the ring for strict data frames where *frame.ri!=side.ri.*

A.5 Transmission Rules

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

This section (A.5), is intended for the "Clause 6. Medium access control data path" of the P802.17 D1.1 draft. Sections 6.8 and 6.9 are recommended.

Frames transmitted by a station should set:

- *ttlBase* to value of *timeToLive.*
- *wrapStatus* to 0.
- *floodType* is set to 01 to indicate the frame will be uni-directional flooded over the ring. It is set to 10 to indicate the frame will be bi-directional flooded over the ring. The value of 01 or 10 is typically set if:
  - A bridging client is requesting the transmission of the frame.
  - A RPR client is requesting the transmission of a frame whose *destinationMACAddress* is not local to the ring.

  Otherwise, this bit is set to 00.
- *extendedFrame* is set to 1 to indicate that a client provided frame is contained after the HEC field. For example, if the frame is transmitted in strict mode (i.e., *strictOrder* is set to 1), then the value is set to 1 if a bridging client is requesting the transmission of the frame.

  NOTE: An enhanced bridging client may want to set the *extendedFrame* to 1 independent of the setting of the *strictOrder* bit.

  Otherwise, this bit is set to 0.
- *sourceMACAddress* is always set to the source station MAC address if transmitting a frame in strict mode (i.e., *strictOrder* is set to 1).
- *strictOrder* is set to 1 to indicate the frame transmission of strict ordering, and set to 0 to indicate the frame transmission is relaxed.

A.6 Unicast considerations

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

This section (A.6), is intended for the "Clause 6. Medium access control data path" of the P802.17 D1.1 draft.

Local stations improve efficiencies by directing local-unicast traffic to the affected station, rather than flooding this traffic to all others, as illustrated in A.9. 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 transmission is used if a local station matches the same *destinationMacAddress*; a flood is used otherwise.

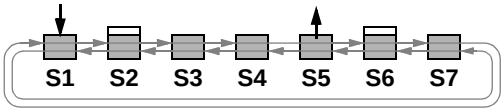

Figure A.9—Local unicasts

A.7 Multicast/broadcast forwarding

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

This section (A.7), is intended for the "Clause 6. Medium access control data path" of the P802.17 D1.1 draft.

The most basic multicast/broadcast distribution techniques involves circulating a frame through all stations on the ring. Refer to section A.2.1.2.

A.8 Flooding alternatives

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

This section (A.8), and related sub-sections, are intended for the "Clause 6. Medium access control data path" of the P802.17 D1.1 draft.

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 *sourceMACAddress* found in the *rprHeader* is the local source address. The *floodType* field is set to unidirectional for this flooding alternative.[2]

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 *timeToLive* within the *rprHeader*. The *floodType* field is set to bidirectional for this flooding alternative.

A variety of remote-transfer flows are illustrated in the following subclauses, as illustrated in Figure A.10. 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.

[2]A potentially more efficient form of unidirectional flooding can also be supported. This is unidirectional transmission that terminates at the source station's upstream neighbor. The *sourceMACAddress* found in the *rprHeader* is the local source address. The *timeToLive* and *ttlBase* fields in the *rprHeader* are set to numberOfStations-1, upon transmission by the source station. The *floodType* field is set to unidirectional.

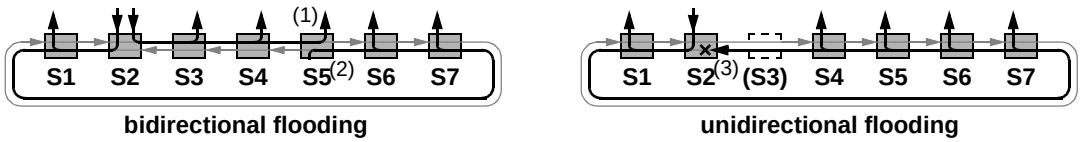

Figure A.10—Flooded receive operations

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 frames all the way around the opposite ringlet to be delivered to some of the stations. It is clearly more bandwidth efficient, to avoid that path entirely and steer the frames 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.8.1 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.11.

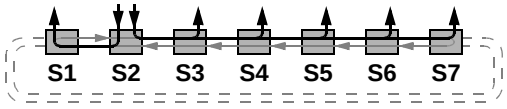

Figure A.11—Steered flooding

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.8.2 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.12. A *floodType* 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.

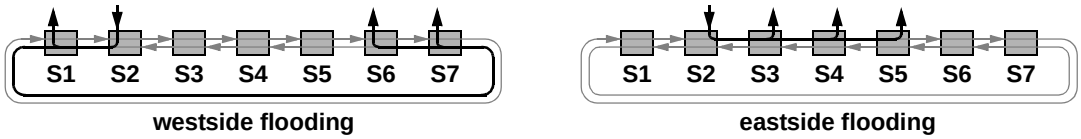

Figure A.12—Bidirectional flooding

Bidirectional floods are ***not*** wrap eligible.

A.8.3 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.13 respectively. A *floodType* of unidirectional is specified, regardless of which direction is selected.

Figure A.13—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.14 respectively. The wrapped flooding operation relies on the wrap capability at the endpoints. A *floodType* of unidirectional is specified, regardless of which direction is selected.

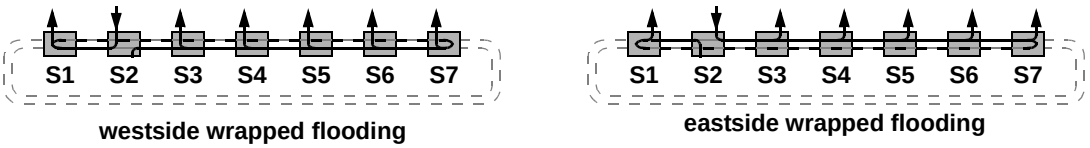

Figure A.14—Unidirectional flooding with wrapped protection

A.8.4 Flood copy rules

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

A.9 Flooding bridge transfers

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

This section (A.9), is intended for the "Annex F. 802.1D and 802.1Q bridging compliance" of the P802.17 D1.1 draft.

Transparent bridging requires a form of one-to-others distribution called flooding. Flooding frames over an RPR requires extra information above and beyond the *sourceMACAddress* and *destinationMACAddress*. The additional information assist in scoping the range of the flooding distribution and suppressing undesirable duplicates that might otherwise be generated.

Bridges flood remote frames for delivery to all bridge clients, as illustrated in the left side of Figure A.15. Flooding involves transmissions between a single source station and all other stations.

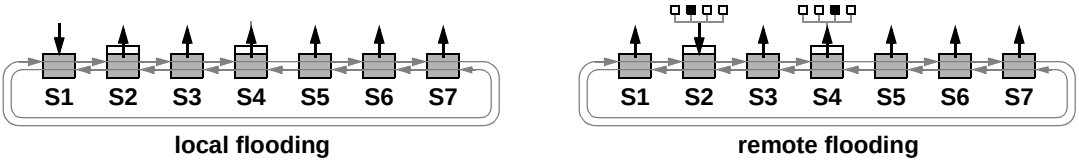

Figure A.15—Basic bridge flooding

Basic-bridging stations maintain simplicity by always flooding, as illustrated in Figure A.15. Although no spatial reuse is possible, this avoids overheads associated with maintaining and utilizing RPR forwarding tables.

Enhanced-bridging stations improve efficiencies by maintaining and utilizing RPR forwarding tables, so that remote frames can also be unicast, as illustrated in Figure A.16. The learn improves link utilization, due to the frame's unicast (not flooded) and shortest-path nature.

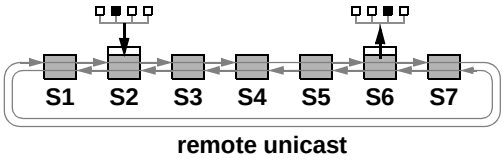

Figure A.16—Enhanced bridging unicasts

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. Alternatively, they can also support flushing before a frame transmission direction change.

A.10 Duplicate scenarios

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

This section (A.10), is intended for the "Annex I. Implementation Guidelines" of the P802.17 D1.1 draft.

A.10.1 Duplicate scenarios: Unidirectional source bypass

Unidirectional flooding is susceptible to a source-station-pair loss during flooding, as illustrated in Figure A.17. 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.17.

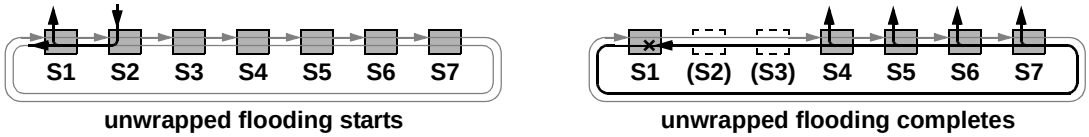

Figure A.17—Duplicate scenarios: Unidirectional source bypass

Cause: The source (that was responsible for frame deletion) disappears before its frame returns.
Problem: The frame passing through stations *S3*&*S2* may be falsely accepted by station *S1* (and others).
Solution: Station S1 discards frames based on *(frame.ws==0)* check passing. We will only return-wrap if the frame has been marked by the source station.

A.10.2 Duplicate scenarios: Unidirectional wrapped source bypass

Unidirectional wrapped flooding is also susceptible to a source-station loss during flooding, as illustrated in Figure A.18. 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.17.

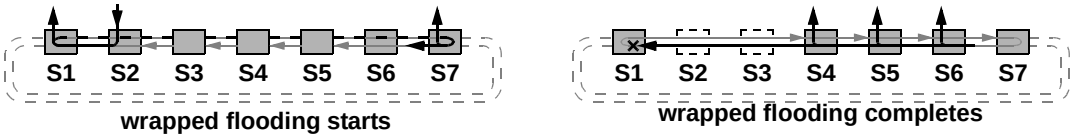

Figure A.18—Duplicate scenarios: Unidirectional wrapped source bypass

Cause: The source (that was responsible for frame deletion) disappears before its frame returns.
Problem: The frame passing through station *S2* may be falsely accepted by station *S1* (and others).
Solution: wrapStatus is not set which prevents wrap exit.

A.10.3 Duplicate scenarios: Bidirectional destination bypass

Bidirectional 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 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.19.

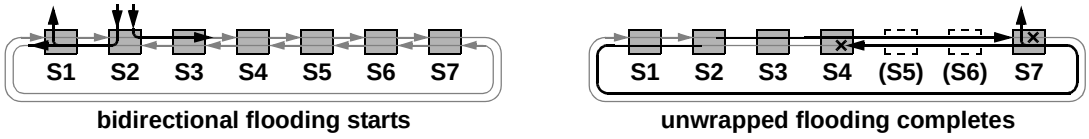

Figure A.19—Duplicate scenarios: Bidirectional destination bypass

Cause: The destination (that was responsible for frame deletion) disappears before its frame arrives.
Problem: The frame passing through stations *S5*&*S6* may be falsely duplicated at station *S4*, *S7*, and others.
Solution: Frame discarded at S4 and S7 since *(SRC[ttlBase-timeToLive]!=frame.SA)* for the received ringlet.

A.10.4 Duplicate scenarios: Bidirectional destination removals

Bidirectional wrapped flooding is susceptible to a destination-station-pair loss during flooding, as illustrated in Figure A.20 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.20.

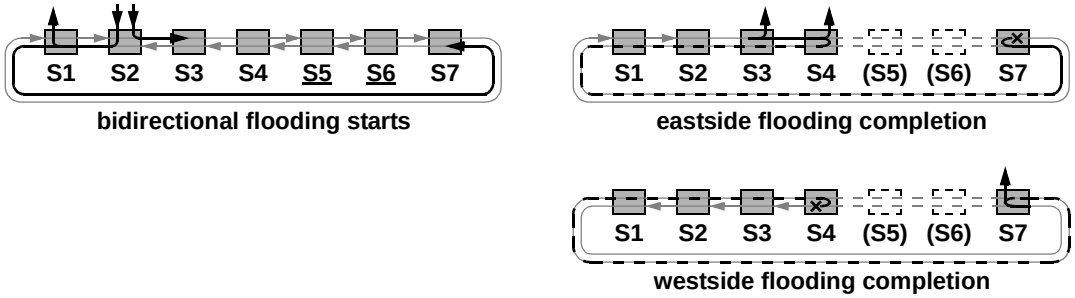

Figure A.20—Duplicate scenarios: Bidirectional destination removals

Cause: The destination (that was responsible for frame deletion) disappears before its frame arrives.
Problem: The frame wrapped before stations *S5*&*S6* may be falsely duplicated at station *S4*, *S7*, and others.

Solution: S7 and S4 will discard frame based on *(SRC[ttlBase-timeToLive]!=frame.SA)* for received ringlet check.

A.10.5 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.21. 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.

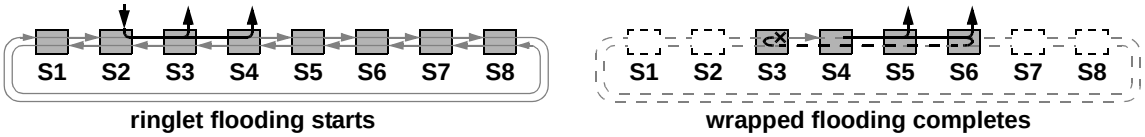

Figure A.21—Duplicate scenarios: Source&destination removals

Cause: The source (that was responsible for frame deletion) disappears before its frame recirculates.
Problem: The frame may be falsely duplicated when recirculated to station *S3* and others.
Solution: Frame discarded at S3 the second time around due to *(SRC[ttlBase-timeToLive]!=frame.SA)* check.

A.11 Reorder scenarios

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

This section (A.11), is intended for the "Annex I. Implementation Guidelines" of the P802.17 D1.1 draft.

A.11.1 Reorder scenarios: Protection switch during bidirectional flood

Bidirectional flooding is susceptible to protection switching during flooding, as illustrated in Figure A.22. In this example, while station A is launching bidirectionally flooded frames, 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 frames using new flooding scopes derived by the new context. Frame reorder is a concern at station C if in-flight frames launched by station A (using an old context) arrive after frames launched by station A, using the new context.

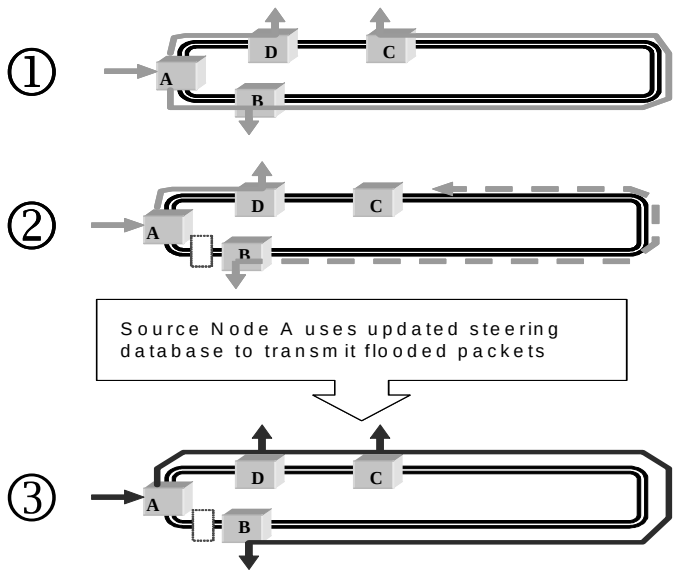

Figure A.22—Reorder scenario: Protection switch during bidirectional flood

Cause: In-flight frames dispatched by source using an old context arrive at a destination after frames dispatched by source using new context.
Problem: The frame received by station C may be received in the wrong order.
Solution: Steering systems use the context containment mechanism to ensure in-flight strict data frame are removed from the ring before strict data frames using a new context are launched

A.11.2 Reorder scenarios: Cascading failures during bidirectional flood

Bidirectional flooding is susceptible to rapid cascading failures occurring during bidirectional transmission, as illustrated in Figure A.23. In this example, while station A is launching bidirectionally flooded frames, the link between station G and E is restored and fails in rapid succession. Frame reorder is a concern at station F and E (in this example) if in-flight frames transmit using the context at step 2 are received before in-flight frames transmitted using the context from step 1.

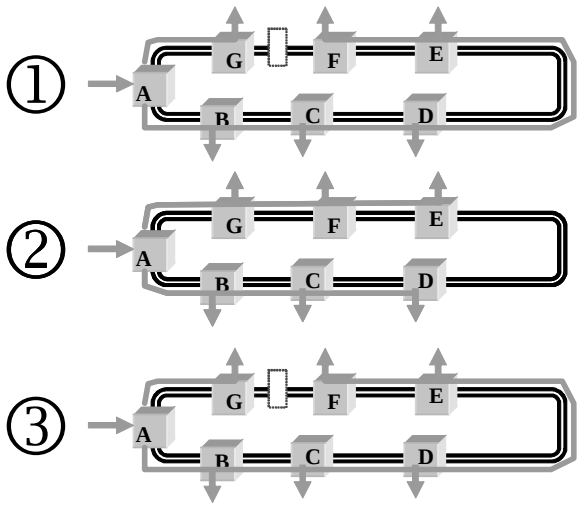

Figure A.23—Reorder scenarios: Cascading failures during bidirectional flood

Cause: At step 1, consider frames in-flight on ringlet1 (using context #1). At step 2, frames are launched on ringlet0 and ringlet1 (using context #2). Assume station E is now using context #2. That is, station E accepts frames launched from A using context #2. At step 3, station E is using context #3. Assume step 2 and step 3 occur while ringlet1 in-flight frames using context #1 are still in-flight. Station E can accept in-flight frames on the ringlet1 ringlet sourced by A using context #1.
Problem: Frame reorder can occur if station E receives in-flight frames from step 1 after in-flight frames from step 2.
Solution: Steering systems use the context containment mechanism to ensure in-flight strict data frame are removed from the ring before strict data frames using a new context are launched

A.11.3 Reorder scenarios: Protection switch during unicast transmission on steering system

Unicast frame transmission is susceptible to a protection switch event occurring, as illustrated in Figure A.24. In this example, station A is transmitting unicast traffic destined for station C over the ringlet1 ringlet. A link failure occurs between station A and B, causing station A to dispatch the unicast traffic destined to C over ringlet0.

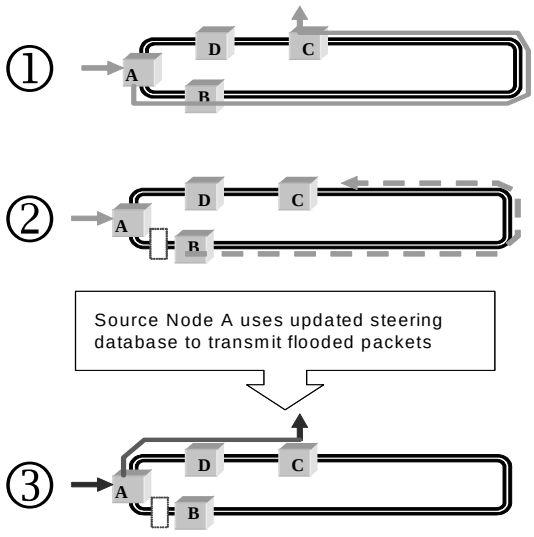

Figure A.24—Protection switching during unicast transmission

Cause: At step 1, consider frames in-flight on ringlet1 (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 frames destined to C now are sent over ringlet0.
Problem: Frame reorder can occur if station C receive context 2 frames before context 1 frames.
Solution: Steering systems use the context containment mechanism to ensure in-flight strict data frame are removed from the ring before strict data frames using a new context are launched.

A.11.4 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.25. In this example, while station A is launching unidirectional flooded frames, the link between station D and E is restored and fails. Frame reorder is a concern at station D (in this example) if frames transmitted at step 3 are received before wrapped frames on secondary ring transmitted at step 1 potentially get unwrapped at station D

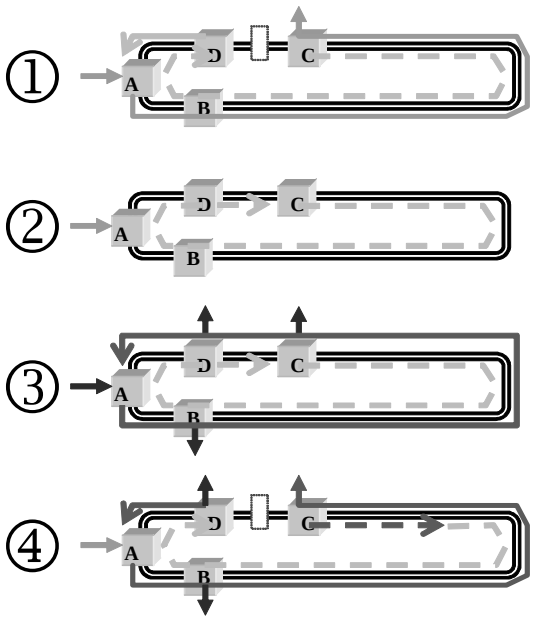

Figure A.25—Reorder scenarios: Cascading failures during unidirectional flood

Cause: At step 1, consider wrapped frames on secondary ringlet. At step 2, the ring heals, however the frames on secondary ringlet continue to circulate until *timeToLive* expires. During frame circulation on secondary ringlet, frames are transmitted by station A (as shown in step 3). If another failure occurs (at step 4), prior to circulation of frames on secondary ringlet having their *timeToLive* expire, frames launched at step 3 can be received by station D before un-expired frames on the secondary ringlet get exit the wrap condition at station D.
Problem: Frame reorder can occur if station D receive context 3 frames before context 1 frames.
Solution: Wrapping systems will purge all strict data frames with *frame.ri!=side.ri* for a specified duration. All wrapped strict data frames using an outdated context will be removed from the ring.