Annex A. - RPR Frame Transmission Conformance to the 802.1 MACService

(normative)

Editors' Notes (robertC): *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). This proposal is a variation of the "RPR Frame Transmission Conformance to the 802.1 MAC Service", (mh_pkt_transmission_txt_07.pdf) , known as MM proposal. This proposal looks at simplifications to the set of RPR MAC procedures recommended in the MM proposal while maintaining the duplication/misordering requirements needed to achieve 802.1D and 802.1 Q compliance. The proposal utilizes the existing frame format as specified in IEEE Draft P802.17/D1.1. Support of the D1.1 frame format has the primary benefit of minimizing changes to planned RPR silicon development with an added benefit of achievingl the efficiency advantages associated with the contribution "dvj_flooding2002Nov04", proposed by David James and Anoop Ghanwani known as AD proposal. The proposal defines the encoding of strict_mode and relaxed_mode frame types in the FT field of the RPR header. Relaxed mode frames receive all the normal frame processing as defined by draft D1.1. Strict mode frames are processed using similiar techniques recommended in MM proposal. By placing further restrictions on the handling of strict mode traffic, the set of changes imposed on the D1.1 frame format to support the processing of strict mode frames can be achieved within the current D1.1 frame format.

There has been several discussions with 802.1 group regarding the RPR requirements for packet duplication/misordering. Discussions with 802 indicate these protocols are highly in the minority, and are quite rare in networks today. However; since 802.17 is targeted toward MAN service providers, 802 group recommends providing some support in 802.17 that meets the duplication/misordering requirements of these protocols. There may be applications where a network service provider may have a particular customer that requires transport of a duplication/misordering sensitive protocol across the service provider's 802.17 network. Since these protocols are so rare, it is unlikely a service provider will devote one specific 802.17 ring just to service these rare protocols. In all likelihood, the service provider will want to transport these protocols with the rest of their non-sensitive traffic.

Similarly support for duplication/misordering sensitive protocols should not detract detract from the majority of the traffic on the RPR ring. 802 group recommended following a similar strategy as when 802 group defined rapid spanning tree protocol (RSTP). RSTP was developed to improve the network restoration times over the previous STP protocol. In development of RSTP, 802 group identified tradeoffs between maintaining the duplication/misordering robustness of STP vs. the rapid restoration requirements. The final resolution was to introduce the improvements of RSTP while providing the ability within RSTP to provision an STP legacy mode. This allows RSTP fully interoperate and support STP requirements for duplication/misordering sensitive protocols, while providing all the restoration benefits to non-sensitive protocols. Also with the introduction of 802.1S (multiple Spanning Tree protocol), multiple instances of STP and RSTP may run over a single network. 802.17 should not limit support to either STP or RSTP instances when supporting traffic in a mixed environment.

Provisions within 802.17 to support rapid restoration of both non duplication/misordering senstive protocols and sensitive ones provides a distinct advantage for 802.17 vs. other types of carrier networks.

For simplicity and to facilitate understanding of similarities/differences with MH proposal, this proposal uses the "mh_pkt_transmission_txt_07.pdf" as the base text and highlights differences via change bars and editorial comments.

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

The following terms and definitions are used:

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

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

Flooding - The act of transmitting a frame such that all nodes 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 (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 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.

Frame transmission supports two types of operation.

1) Strict: This type 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 this mode is particularly required during station or link failure operations.

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

The RPR MAC shall support both modes 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). RPR MACs servicing other clients need to abide by these requirements as well.

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

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

This proposal assumes the 50ms restoration refers to the time the last failure event is triggered on the ring, to the time transmission of traffic for all stations is restored. In cases where there are multiple successive failures within a 50ms period, restoration occurs within 50ms of the last failure. There are no guarantees of restoring data within a single 50ms period when multiple failures are occuring during that period.

A.1.1 Relaxed mode of operation

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

Note there are no restrictions on the handling of relaxed mode traffic. Relaxed mode traffic is handled by all existing MAC procedures as defined in draft D1.1.

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 frame. For bidirectional flooding, the TTL field found in the RPR Header is used to scope the travel of the frame in the ringlet0 and ringlet1 directions.

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 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 TTL 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 (Robert C): *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.

The frame format uses the existing D1.1 frame format along with frame type code point modifications for the RPR control FT/P sub-fields to support strict/relaxed mode indications, flood / no_flood, and extended frame indications using the RPR frame format. The proposal also looked at tradeoffs in using reserved code points of D1.1 FT/P sub-fields. Working within existing D1.1 code points could work given tradeoffs in how to support the extended frame format.

One alternative for extended frame is to use the PT field with a value designating a bridged frame. Since PT values are adminstered by the IEEE RAC, there's a question as to whether IEEE RAC will allocate a PT field to support a bridged frame format, even though the frame type could be used over other MAC/ PHY interfaces. Another alternative for encoding extended frame format while retaining existing FT/P code points is by designating the MSB of the TTL to encode extended frame format. Using the MSB of TTL reduces the supported number of stations on ring from 255 to 127. Support for 127 stations on the ring is within the RPR requirements, however such a change would need to be agreed upon by the working group.

This proposal recommends FT/P code point modifications identified in this section. If the proposed code point modifications recommended are not considered acceptable, the other alternatives listed above should be explored as well as other possible tradeoffs in frame type encoding.

~~Additional fields are required to support flooding (in strict mode). They include TTL_Base, floodIndicator, wrapState, and strictRelax. The strictRelax bit is encoded in the~~

The RPR frame format is shown in Figure A.1.

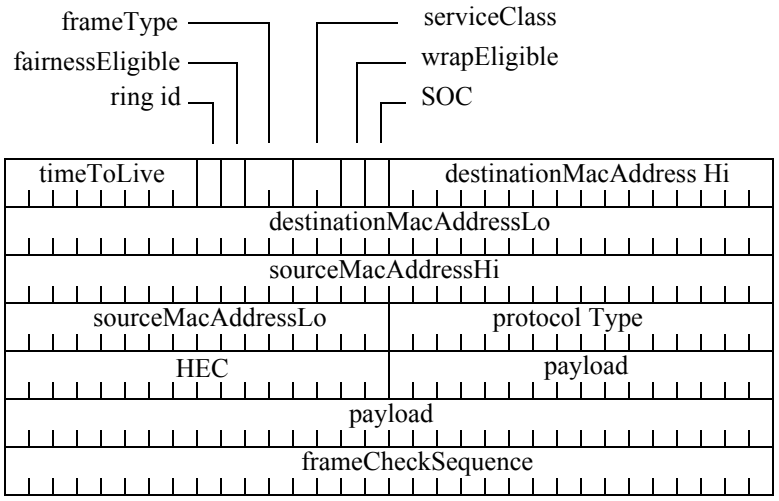


Figure A.1—RPR Local frame format

The *sourceMACAddress* field is a 48-bit MAC address. The sourceMACAddress may either be the MAC address of a local host or remote host, for frames using the local frame format. The sourceMACAddress shall always be the MAC address of a local station on the ring for frames using extended frame format.

The *destinationMACAddress* field is a 48-bit MAC address. It can either be a multicast, broadcast, or unicast MAC address. Frames transmitted using the local frame format, unicast destinationMACAddresses may either be of a local host or remote host. Frames transmitted using the extended frame format, unicast destinationMACAddresses shall always be a local RPR station address.

The use of the TTL 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 TTL is decremented by each station on the ring for frames traveling on the primary ring (frame.RI value matches the incoming ringlet). TTL is decremented by the receive processing function to determine whether a flooded frame will transit to the next station on the ring or not. The TTL of incoming receive frames are first validated by receive logic to determine if the frame should be discarded, or processed by the rest of the receive logic. Frames received with a TTL=0 are discarded. Frames with non-zero TTL are processed by the receive logic. Frames with a TTL = 1 are stripped at the station. Frames with TTL > 1, have their TTL value decremented and are passed to the transit path for transmission to the next station.

The frameType + SOC (formerly parity) sub-fields in the RPR header are combined to form a 3 bit control field defining the RPR frame type. This field is used to encode the various control and data frame types supported on the ring. This field defines relaxed/strict mode frames, extended frame format, and flooded/ non_flooded frames to support basic bridging. The SOC subfield is used to define strict/relaxed mode forms of data frames and control/fairness frame types. The FT sub-field is used to identify control, local user data, extended user data, and flooded. user data. The frame type encodings for these sub-fields are shown in Table A.1

Table A.1—Flood type values

Frame Type	SOC	Description
00	0	Local Data - Flood - Relaxed
00	1	Local Data - Flood - Strict
01	0	Extended Data - Relaxed
01	1	Extended Data - Strict
10	0	Control
10	1	Fairness
11	0	Local Data - No Flood - Relaxed
11	1	Local Data - No Flood - Strict

The SOC bit defines strict ordering or control depending on the frame type value. For data frames (Local Flood, Extended Data, Local Data no_flood frame types), the SOC field defines strict frame ordering. Data frames having SOC=1 (strict ordering mode) define transmission procedures that ensure frames are never duplicated or misordered. Data frames with SOC=0 (relaxed mode) are handled by normal frame transmission procedures, where minimal frame duplication/misordering may occur under certain protection and topology change events. The advantage of relaxed mode traffic is lower traffic loss and faster restore times than strict mode traffic due to the more tolerant data processing procedures. For control frames, the SOC bit identifies whether the frame is a general control frame or a fairness message. SOC = 0 defines a control frame, and SOC=1 defines a fairness message.

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

The FE bit could be associated with the Fairness frame type encoding for use as a parity bit (previous p bit function) if required.

The *flooding indicator* defines when frames are to be flooded onto the RPR ring. Support of the flooding indicator ensures that broadcast, multicast, and bridged unicast frames are flooded on the ring in accordance with basic bridging compliance flooding procedures. The flooding indicator is used to determine whether RPR attached bridges should copy frames to their bridge relay and whether local hosts are to invoke destination stripping rules for receive frames where the destination MAC address in the frame matches the station's local MAC address. Frames where the flooding indicator = *flood*, are copied by bridges to their bridging relay, and may not be destination stripped by hosts. Frames where the flooding indicator = *no_flood*, are not copied by bridges to their bridging relay, and shall be destination stripped (independent of TTL value, for any incoming receive frames with TTL !=0 values) by hosts.

The flooding indicator for broadcast, multicast, and unicast frames are provided in both the local and extended frame formats. The FT field is used to designate the flooding indicator for frames transmitted with the local frame format. An FT value of 0 denotes a flooding indicator of *flood*. An FT value of 3 denotes a flooding indicator of *no_flood*.

The flooding indicator is also provided for frames transmitted using the *extended* frame format. Frames encoded using extended frame format, shall designate the flooding indicator using the *destinationMACaddress* field of the RPR frame header. Frames with the *destinationMACaddress* field set to the MAC broadcast address or MAC multicast address is used to provide the flooding indication value of *flood* for extended frame. Extended frames having the *destinationMACaddress* field set to a local unicast address is used to indicate a flooding indication of *no_flood*. Determination of MAC unicast, multicast, or broadcast can be determined by the I/G bit (LSB of octet 0) of the destinationMACaddress. Extended frames identified as either flood or no_flood follow the same set of flood/no_flood procedures defined for frames using local frame format.

Extended Frame Format (figure A.2) defines a format for encapsulation of 802 MAC frames into RPR payload. Extended frame format is typically used for bridging 802 MAC frames across RPR networks. Extended frame format also contains a 2-byte control field which provides future growth and enhancements to RPR frame format in the future. Some of these control bits can be used to define new RPR data types (non-802) encapsulation such as payloads with protected headers or other types of services in the future.

Encapsulation of 802 MAC frames in RPR extended frame format satsifies two particular 802 bridging applications. The first is providing a mechanism for a bridge that is locally attached to the RPR to encode their station address in the RPR frame for source stripping and station-to-source validation of bridged traffic. Source stripping and station-to-source validation is required for duplication prevention of strict mode traffic. Encapsulation of 802 MAC frames is also useful for enhanced bridging in future, where bridged unicast traffic no longer must be flooded on the RPR.

Extended frame format is designated using an FT subfield value of 3. Extended frame format requires the following semantics within the RPR header. The destinationMACaddress must either be a unicast MAC address corresponding with one of the other stations on the ring, or MAC Multicast, or MAC broadcast address. Frames having the destinationMACaddress set to a unicast station MAC address are not flooded. Frames having the destinationMACaddress set to either MAC Multicast or broadcast address are flooded. The rules for transmitting/receiving flooded/non-flooded extended frames is the same as those for local frames.

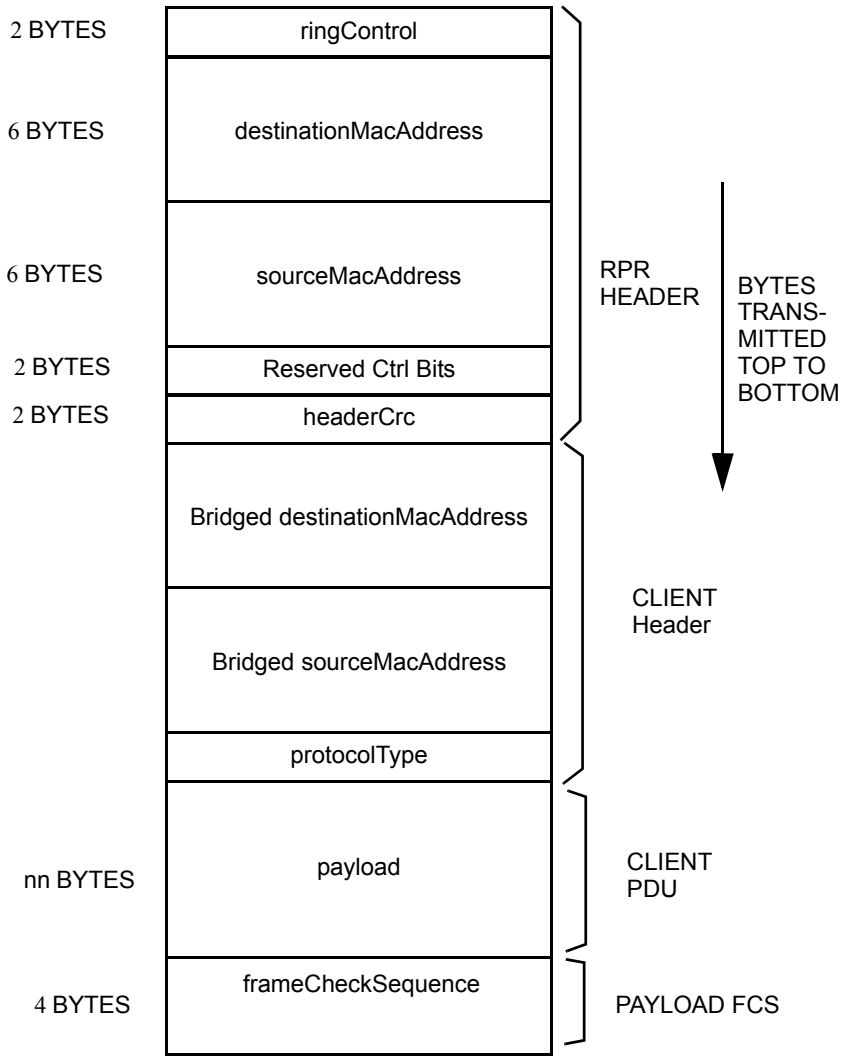


Figure A.2—RPR extended data frame format

~~The *wrapState* 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 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.~~

.

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

One of the simplifications for strict mode traffic is that strict mode traffic is not wrapped. Strict mode traffic is marked steering only (wrap_enable = 0) and thus will only be steered during protection events.

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

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

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

.

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

Another simplication with this proposal is the use of an implicit *TTL_base.* The *TTL_base* in MM proposal is performs a station-to-source frame integrity check which is used to validate the intended number of hops a frame is to travel on the ring. The original intent for TTL_base is that a station may choose to originate traffic with different values of TTL within the station's flood scope and an explicit TTL_base is used to resolve the different TTL scope values of the various frames a station transmits onto the ring. This proposal looks at an implicit method performing the same station-to-source check. The advantage of doing an implicit check, is that the added frame control fields to support TTL_base are not required, and these bytes can retain the original PT value specified in D1.1. The source-to-station integrity check using the implicit TTL_base, provides the same TTL flexibility as the explicit station-to-source check. Station's defining a TTL_base of X, may transmit frames having TTL values of X or less. The implicit TTL_base also supports frame integrity checks of local frames having broadcast, multicast, and unknown unicast (bridged) MAC destination addresses for unidirectional0, unidirectional1 and bidirectional flooding procedures. The station-to-source integrity check works on steered rings for frames being flooded using the local frame format.

Since TTL_base is only needed to control the transmission of strict mode traffic on the ring, all other types of traffic, non-strict, fairness, control, may be transmitted with whatever TTL values are desired. Only strict mode traffic is subject to TTL values that are less than or equal to the implicit TTL_base. The implicit TTL_base is a pair of constants exchanged by all stations during topology updates defining the flood scope of all strict mode traffic transmitted by that station on the ring. Assignment and updates of the implicit TTL_base for strict mode traffic is subject to the context containment algorithm specified in Annex A.3.1.1. Context containment ensures that old contexts of implicit TTL_base are removed from the ring before transmission of strict traffic within a new context. Context containment is the means by which packet duplication / misordering of strict mode traffic is precluded.

A.2.1 Format usage

Local RPR frame format is used in a variety of network scenarios. Local frame format may be used by hosts transmitting either strict or relaxed mode traffic to hosts which are local to the RPR ring, or transmitting to hosts which are not local to the RPR ring (transmitting to remote hosts via a bridge). Local frame format may also be used by bridges transmitting basic bridging relaxed mode traffic on the ring. Local transfers involve insertion of *RPR_Control* ~~and *RPR_Control2* ~~information, to ensure reliable RPR local delivery, as illustrated in Figure A.3.

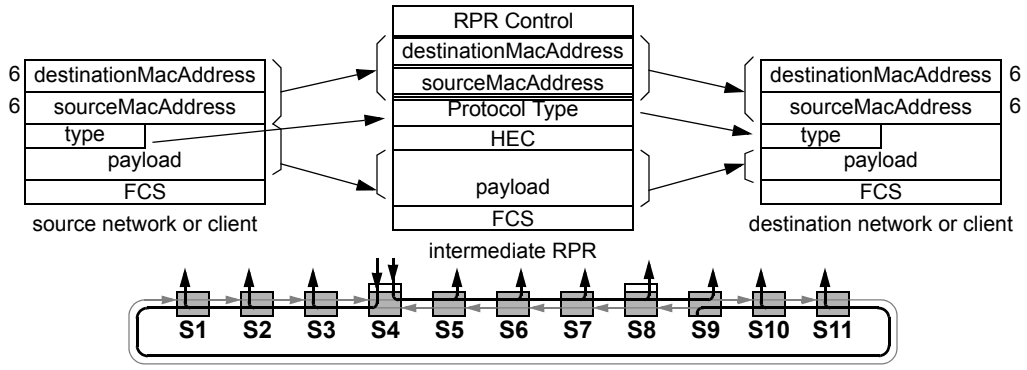


Figure A.3—Local frame forwarding

Extended frame forwarding is may be used by bridges transmitting basic bridging strict_mode bridge traffic on the ring, or transmitting enhanced bridging traffic in the future. Extended frame forwarding is needed when bridges transmit strict mode traffic so that the bridge station's MAC address can be carried in the sourceMACaddress field of the transmitted frame. This allows source stripping and station-to-source validation of frames transmitted by bridges. Extended format frames transmitted by bridges encapsulate the entire bridged 802 MAC frame (Header, type/length, MSDU) following the HEC, along with setting the 48-bit *sourceStationID* to the bridge station's MAC address to ensure reliable RPR local delivery.

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

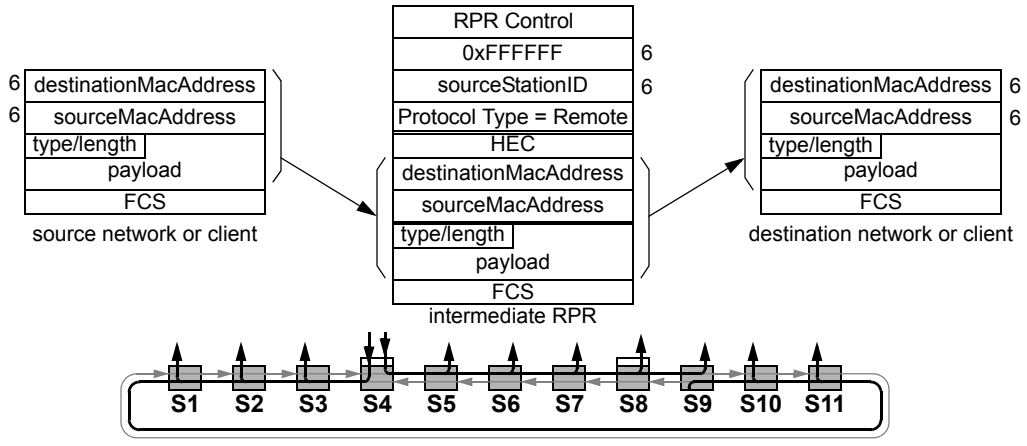


Figure A.4—Extended frame forwarding

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

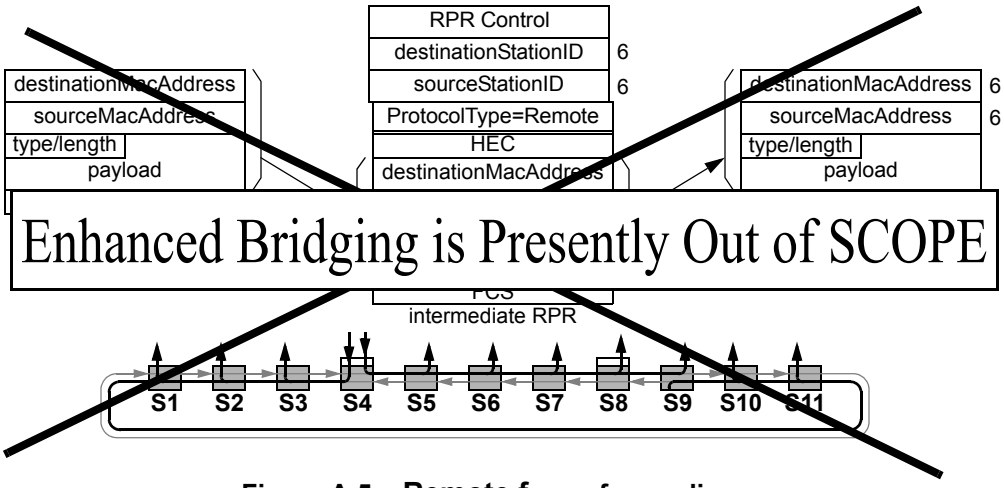


Figure A.5—Remote frame forwarding

A.3 RPR System requirements.

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

This section(A.3), is intended for the "Clause 8. Frame formats" of the P802.17 D1.1 draft.

The destination address for local frame format may be either local unicast, remote unicast, broadcast, or multicast. There is no constraint/restriction placed on the setting of the destination address found in the RPR Header (i.e., the destination address could be a destinationStationID, remote address, multicast, or broadcast address). The destination address for extended frame format may either be local unicast, broadcast, or multicast.

The source address for local frame format may be either a local or remote host address for local frame format, and local station address for extended frame format.

Station bypass (passthru) can be thought of as being equivalent to an optical regenerator where the MAC operates as a transit node. The MAC enters bypass (passthru) mode when invoked by LME. In bypass (passthru) mode, TTL 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. There should be no restrictions in supporting broadcast, multicast traffic on a steered protection ring.

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 distance between any two adjacent stations 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. Section 6.9 is 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 data frames that were transmitted by a source using an outdated context gets removed from the ring prior to the transmission of data frames using an updated context. In-flight 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 and purge all data frames 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 frames as requested.

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

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):

The 15 ms timer can be implemented in SW or HW. For the duration of the timer, the checker entity purges all data frames prior to transfer to PTQ. All 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 data frames during read operations while the read pointer has not passed the stored WP. Additionally, any data frames arriving are purged. The 15ms timer can be implemented in SW or HW.

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

Consider the illustration in Figure A.6 below. A link failure occurs between nodes G and F. Protection control frames are dispatched by the nodes adjacent to the failure (i.e., nodes G and F). The context containment mechanism is triggered in nodes 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 nodes (i.e., nodes 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).

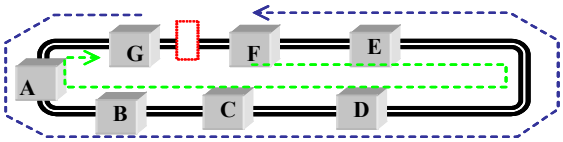


Figure A.6—Protection Event Example

The end result of this operation is the removal of data frames 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 context containment mechanism ensure that there is no reorder of strict mode 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 ensures frames dispatched using an old context are removed before frames are dispatched using a new context. Frame reorder prevention is guaranteed.

With the exception of station bypass, frame duplication of strict mode traffic on a steered protection ring is generally provided through TTL scoping and context containment. Context containment ensures that stations transmitting strict mode traffic is always correctly scoped by TTL. Station bypass presents a particular problem to TTL scoping, in that frames passing through the bypass point (station) without having the TTL decremented are subject to frame duplication.

Station bypass occurs when either a MAC goes into bypass mode, or some other device outside the MAC creates a bypass condition. A bypass condition is where traffic transmitted from one station is received by a station without a corresponding decrement in TTL. An example of a MAC bypass, is where frames received on a particular ringlet is transmitted on the same outgoing ringlet without passing through the transit path of the MAC. In this case, the next suceeding station receives frames that have not had their TTL decremented by the previous station. Another case is where multiple stations are connected through an OXC (optical cross connect). In this case an optical cross connect may connect the output of one station to the receive of a further downstream station bypassing the next adjoining station. This results in in-flight frames having TTL values which are inconsistent with the topology, causing the TTL scoping functions to break down. Topology protocols take a while to discover the new topology induced by the bypass event, resulting in continuous frame duplication for each frame passing through the bypass point that was transmitted under the old topology information. To robustly deal with bypass type scenarios, stations must be able to detect and discard bypassed frames until a new topology context is established.

There are two methods for detecting bypassed frames. One method is to employ a *break-before-make* algorithm. The other is to perform a *station-to-source* integrity check.

For steering systems, a ring failure breaks the ring and at worst results in frames dropping off the end of the failure point. The protection value calulates new flood scope values consitent with the number of ringlet 0/1 hops to the failed span.

A.3.1.2.1 Break Before Make

Break-before-make is a method of ensuring that strict mode frames passing through a bypass station are discarded by the next downstream station upon reception. Break-before-make places a requirement on the entity performing the bypass that it must break the transmission of traffic for some minimum time period (break interval) before the bypass connection is made. Automatic OXC reconfiguration algorithms should also support software configurable options to ensure a break-before-make intervals. This specified break interval is long enough for the next downstream station to detect the break condition and initiate a purge of all strict mode traffic until the station can authenticate its neighbor through the neighbor consistency check. Even without a minimum break interval, an optical cross connect from one RPR station to another shall result in loss_of_light, (probably OOF, LOS, LOF errors) and corrupted HEC, and FCS errors. Detection of minimal break interval causes the station to initiate a local purge until it revalidates its upstream neighbor. If the neighbor consistency check fails, then the station initiates context containment and continues discarding strict mode traffic until the new context containment algorithm has completed and a new context is established.

The break-before-make algorithm is as follows:

- Each station monitors HEC and idle frame monitoring for received frames.
- Entities initiating bypass, will either break connection for the minimum break-before-make interval ensuring the next downstream station begins strict_mode traffic purge. Strict_integrity_check time threshold may also be set low enough to detect change from one station to another. For example, strict_integrity_check threshold could be set low enough to initiate strict mode discards upon detection Loss_of_Light, OOF, LOS, LOF or a single frame HEC or FCS error. Even “instantaneous” OXC optical reconfigurations will minimally result in Loss_of_Light condition resulting in errored RPR receive frames.
- Reception of frames with errored HEC or undetected idle frames exceeding the strict_integrity_check time threshold results in that local station discarding all strict_mode frames until the upstream neighbor is verified through neighbor discovery messages.
- If the neighbor detected in the neighbor discovery message is same as the previous neighbor, discarding of strict_mode traffic ceases.
- If the neighbor detected in the neighbor discovery message is not the same as the previous neighbor, discarding of strict_mode traffic continues, and context containment algorithm is initiated. Context containment initiates a purge of all strict_mode traffic on the ring until all stations relearn the new topology.
- Once the new topology converges, strict_mode traffic is once again transmitted on the ring, using the newly discovered TTL values. The new context ensures that all strict mode traffic from the previous context is purged, ensuring no duplication occurs.

A.3.1.2.2 Station to Source Integrity Check

Another method for detecting alien frames received through a station bypass point is by performing a TTL validation check of all strict mode receive traffic. The station-to-source integrity check validates the TTL received from upstream stations against the expected maximum TTL value that can be received at a given station. Strict frames received with a TTL value that is less or equal to *expected_max_TTL* value will not result in data duplication. These frames are received and processed by the MAC. Strict frames received with a TTL value exceeding this *expected_max_TTL* are discarded.

Each station defines an *implicit_TTL_base* value that defines the flood scope of traffic originating from that station. The flood scope is defined as the maximum number of hops frames may travel in existing topology that does not result in frame duplication. The TTL flood scope is determined once topology converges, and is typically a function of the number of stations on the ring, whether the station employs unidirectional or bidirectional flooding, the form of protection, and whether the ring topology is closed or open, For unidirectional flooding the flood scope for an N station ring is N. A station employing bidirectional flooding on a ring of 6 stations, the flood scope (*implicit_TTL_base*) may be 3/2 or 1/4 for ringlet 0/1 respectively. The *implicit_TTL_base* values (ringlet 0/1) are exchanged through topology messages. Each station relays a single *implicit_TTL_base* value defining its flood scope which is received by all other stations on the ring. These values are used by each of the stations to calculate the *expected_max_TTL* value expected from neighboring sending station on the ring. For an N station ring, each station calculates the *expected_max_TTL* value for each of its N-1 neighbors. These values are stored in a source_validation_database which is indexed by sourceMACaddress. Each receiving station calculates the *expected_max_TTL.source_station* for each sending station from *implicit_TTL_base.sending_station* and the upstream hop count distance of this receiving station to the sending station *upstream_hop.sending_station*.

> expected_MAX_TTL.source(i) = implicit_TTL_base.source(i) - hops_to_source(i) + 1
> (where hops_to_source(i) is the distance from topology database for calculating station to the source. Expected_MAC_TTL value assumes check is performed prior to TTL_decrement of received frame)

Stations perform the following source-to-station validation check of all received strict_mode traffic.

> frame.TTL - Expected_MAX_TTL(frame.sourceMACaddress) > 0, then discard
> frame.TTL - Expected_MAX_TTL(frame.sourceMACaddress) <= 0, then accept

Strict mode frames having a TTL exeeding the expected_MAX_TTL value from a given station are indicative of passing through a bypass point (TTL exceeding the flood scope), are discarded. Frames with TTL less than expected_MAX_TTL are indicative of frames launched within the sending station's flood scope and are accepted for frame transmission. This allows sending stations to transmit valid frames with any TTL value that is less than or equal to *implicit_TTL_base*.

(**editor's note** - instead of discarding frames where frame.TTL exceeds Expected_MAX_TTL, frame.TTL could be overwritten with Expected_MAX_TTL resulting in TTL healing. In this case frames passing through the bypass point having incorrect TTL are updated with the correct TTL, ensuring these frames remain within their predefined flood scope as they traverse the ring.)

Since the above Expected_MAX_TTL validation table is indexed by the 48-bit sourceMACaddress in the receive frame, a 255 entry cam is required.

When the bypass condition first occurs, the station-to-source integrity check ensures that frames that will result in duplication are removed from the ring. Eventually the topology algorithm will detect the change in topology, resulting in the start of the context containment algorithm. Strict mode traffic will be purged from the ring, and each station will recalculate and redistribute iTTL_base. Once the ring converges with new

iTTL_base values, stations resume sending strict mode traffic. This ensures duplication is avoided as the ring transitions from one topology to the next.

A.3.2 Wrapping systems

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

One of the simplifications for strict mode traffic is that strict mode traffic is not wrapped. Strict mode traffic is marked steering only and thus will only be steered during protection events. The assumption discussed with 802 group is that strict mode traffic is so much in minority that restoration performance of strict mode traffic can be traded off in order to ensure absolute no duplication/misordering. Similar tradeoff was made in development of 802.1W rapid spanning tree protocol.

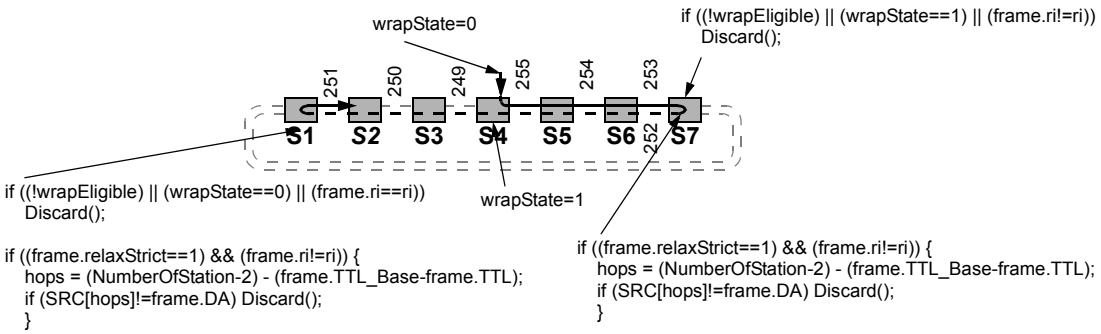
Wrapping mode of protection is fully supported for all non-strict mode traffic and any control or fairness frames.

~~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 RPR Header 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 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 ringlet ID of the frame and station must be the same. When the frame passes the source station on the secondary ring, the *wrapStatus* bit is set. In order to wrap a frame from the secondary ring to the primary ring, the *wrapEligible* bit must be set, the *wrapStatus* bit must be set, and the ringlet ID of the frame and station must be different.~~

~~The behavior of a wrapping system is illustrated in Figure A.1.~~



~~Figure A.1—Wrapping system behavior~~

~~The TTL is decrement when the frame enters the secondary ringlet. While the frame travels on the secondary ringlet, the TTL is no longer decremented. When the frame is wrapped back to the primary ringlet, normal TTL decrement operation continues.~~

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

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

Implementation considerations:

Frame deletion from TB on receipt of an unwrap message. This involves storing the current LPTB write pointer and deleting any data frames with the wrong RI during read operations while the read pointer has not passed the stored WP. Additionally, any frames 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 frame loss will occur (of the newly wrapped frames) but no reorder will occur.~~

A.3.3 Reception rules

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

This section(A.3.3), is intended for the "Clause 6. Medium access control data path" of the P802.17 D1.1 draft. Section 6.9 is 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., *strictRelax* value of 1)~~ are discarded if the following deletion rule is true [frame.TTL - Expected_MAX_TTL(frame.sourceMACaddress) > 0], then discard*)*. Both wrapping and steering systems perform this check, at all ring stations, on strict mode frames traveling on the primary ringlet (i.e., frame.ri==ri). Figure A.2 provides an illustration.

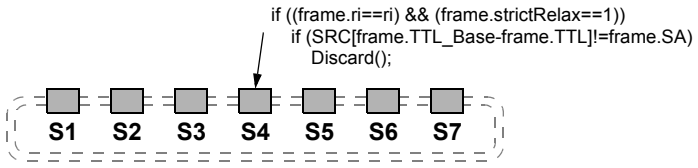


Figure A.2—Source consistency check

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.1. In addition, when a ring unwraps (i.e., ring heal), the context containment mechanism is applied to the ring for data frames where *frame.ri!=ri*~~.

A.3.4 Transmission Rules

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

This section(A.3.4), is intended for the "Clause 6. Medium access control data path" of the P802.17 D1.1 draft. Section 6.9 is recommended.

Frames transmitted by a node should set:

- ~~*TTL_Base* to value of TTL~~
- ~~*wrapState* to 0.~~
- *floodindicator* is to *flood* value for all strict or relaxed mode flooded traffic. ~~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 is set to 01 or 10 if:~~
  - A bridging client is requesting the transmission of the frame.
  - A RPR client is requesting the transmission of a frame whose destinationAddress is not local to the ring.

  Otherwise, the flooding indictor is set to *no_flood* value.
- *extendedFrameFormat* encapsulation is indicated by *frame.FT* = 3 value, indicating a full MAC frame is encapsulated in the MSDU portion of the frame following the HEC field. remoteFrame encapsulation must be used when:
  - A bridging client is transmitting *strict mod*e frames.

(**editorial note** - If break-before-make algorithm is supported to address bypass frames, then bridging clients may transmit all their strict and relaxed mode frames using local frame format).

NOTE: An enhanced bridging client may want to use the *extendedFrameformat* encapsulation independent of the setting of the *strict/Relax* mode setting.

Otherwise, local frame encapsulation is used.

- *sourceMACAddress* is always set to the transmitting station's source station MAC address if transmitting a frame in strict mode (i.e., *SOC* is set to 1).
- *SOC* is used to indicate the frame transmission mode is strict (SOC = 1),or indicate the frame transmission mode is relaxed (SOC = 0).

A.4 Multicast/broadcast forwarding

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.

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. Multicast frames may be transmitted with either the *floodIndicator* set to either the *flood* or *no_flood* value. Multicast frames with the *floodIndicator* set to *flood* are distributed to all bridges and hosts on the bring. Multicast frames with the *floodIndicator* set to *no_flood* are ignored by bridges, and may be stripped by hosts using TTL or the multicat *destinationMACaddress* in the frame.

A.5 Flooding bridge transfers

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.

Transparent bridging requires a form of one-to-others distribution called flooding. Flooding frames over an RPR requires extra information above and beyond the sourceAddress and destinationAddress. The additional information 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.3. Flooding involves transmissions between a single source station and all other stations.

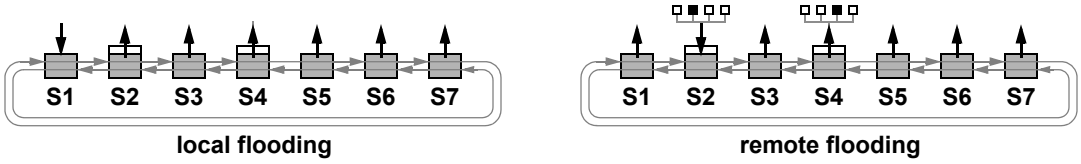


Figure A.3—Basic bridge flooding

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.3. 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.*

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

Local hosts improve efficiencies by transmitting their known unicast traffic to the affected station, rather than flooding this traffic to all others, as illustrated in the left side of Figure A.4. The determination of whether to use *flood* or *no_flood* frame types is based on a comparison of the frame's destinationMacAddress with the RPR topology database: the no_flood type is used if a local station matches the same destinationMacAddress; otherwise the flood type is used.

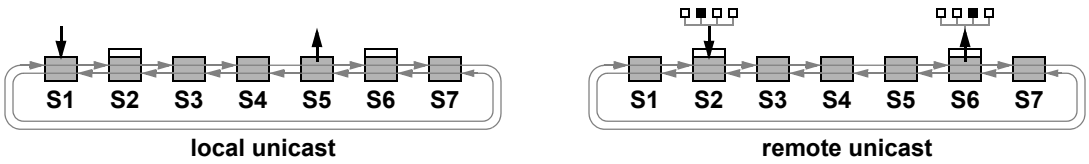


Figure A.4—Enhanced bridge unicasts

~~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.4. The learn improves link utilization, due to the frame's unicast (not flooded) and shortest-path nature~~.

(editorial note - enhanced bridging is outside 802.17 scope).

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.7 Flooding alternatives

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

This section(A.7), 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 source address found in the RPR Header is the local source address. The *floodType* 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 *floodType* 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.5. 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.

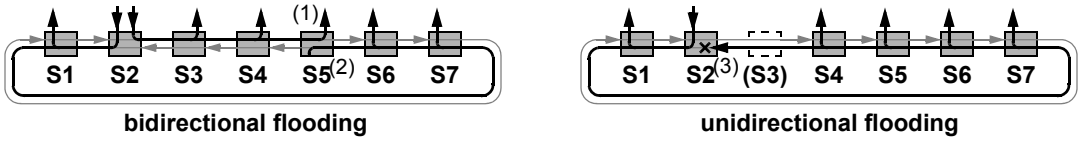


Figure A.5—Flooded receive operations

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 ring to be delivered to some of the nodes. It is clearly more bandwidth efficient, to avoid that path entirely and steer the frames instead. Station's performing bidirectional flooding on a wrapped ring shall mark their bidirectional flood traffic as steer only (WE = 0). Further-

more, 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.6.

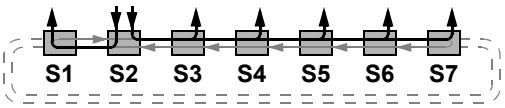
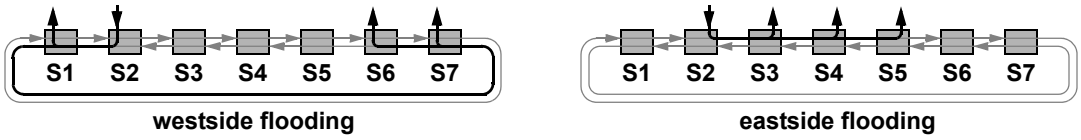


Figure A.6—Steered flooding

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.7. ~~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~~.



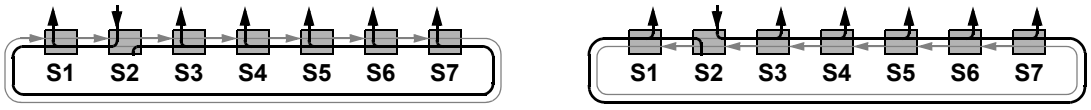
westside flooding **eastside flooding**

Figure A.7—Bidirectional flooding

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

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.8 respectively. ~~A *floodType* of unidirectional is specified, regardless of which direction is selected~~.



westside unwrapped flooding **eastside unwrapped flooding**

Figure A.8—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.9 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.~~

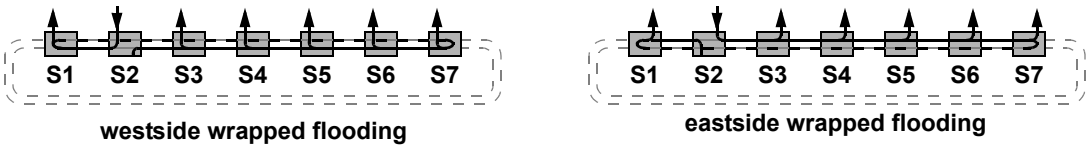


Figure A.9—Unidirectional flooding with wrapped protection

A.7.4 Flood copy rules

Flooding involves selectively copying non-deleted primary-run frames to the client, if a *floodIndicator* indication of value of *flood* 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 TTL equals 1, the frame is stripped. Frames with a TTL equal to 0 are discarded.

A.8 Duplicate scenarios

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

This section(A.8), is intended for the "Clause 6. Medium access control data path" of the P802.17 D1.1 draft.

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.10. 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.10.

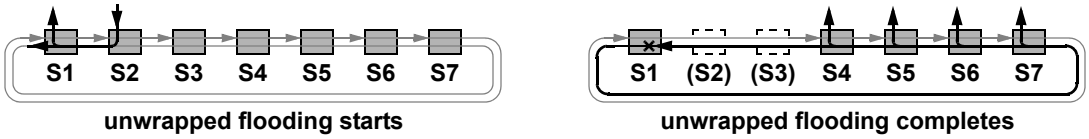


Figure A.10—Duplicate scenarios: Unidirectional source bypass

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 strict mode frames based on *(SRC[TTL_Base-TTL]!=frame.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.11. 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.10.

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: Strict mode frames are not wrapped.

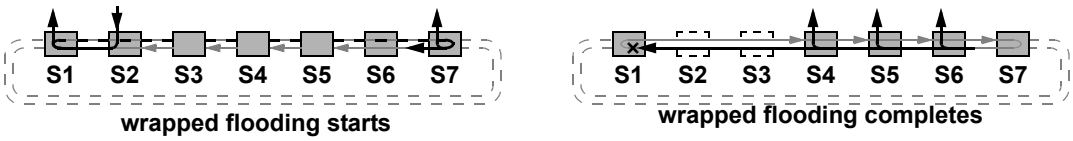


Figure A.11—Duplicate scenarios: Unidirectional wrapped source bypass

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.10. 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.12.

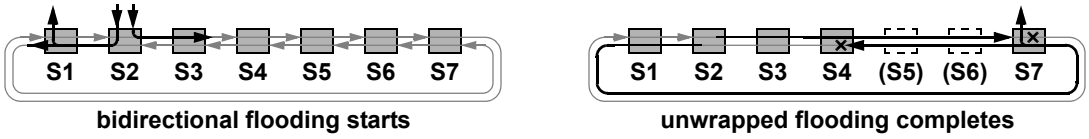


Figure A.12—Duplicate scenarios: Bidirectional destination bypass

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: Station S4 and S7 discards strict mode frames based on *(SRC[TTL_Base-TTL]!=frame.SA)*

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.13 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.13.

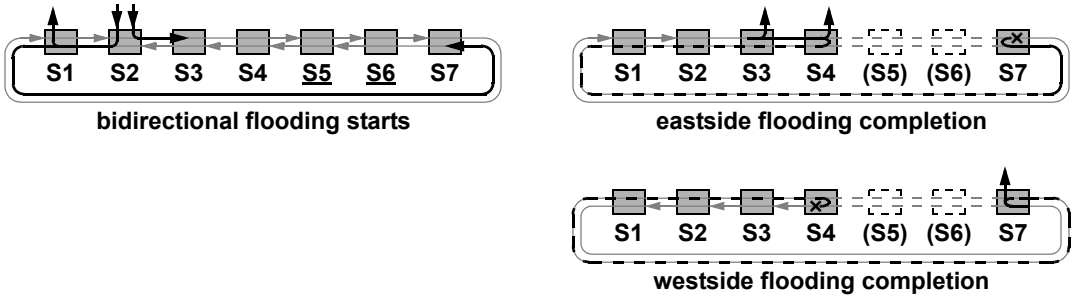


Figure A.13—Duplicate scenarios: Bidirectional destination removals

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: Strict mode frames are not wrapped.

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.14. 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.

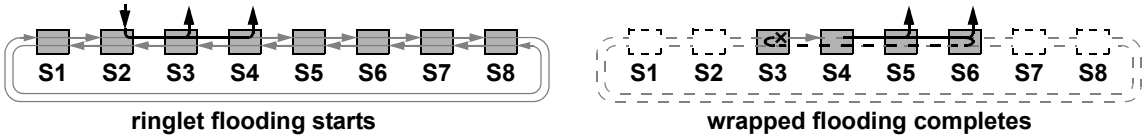


Figure A.14—Duplicate scenarios: Source&destination removals

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: Strict mode frames are not wrapped.

A.9 Reorder scenarios

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

This section(A.9), is intended for the "Clause 6. Medium access control data path" of the P802.17 D1.1 draft.

A.9.1 Reorder scenarios: Protection switch during bidirectional flood

Bidirectional flooding is susceptible to protection switching during flooding, as illustrated in Figure A.15. 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.

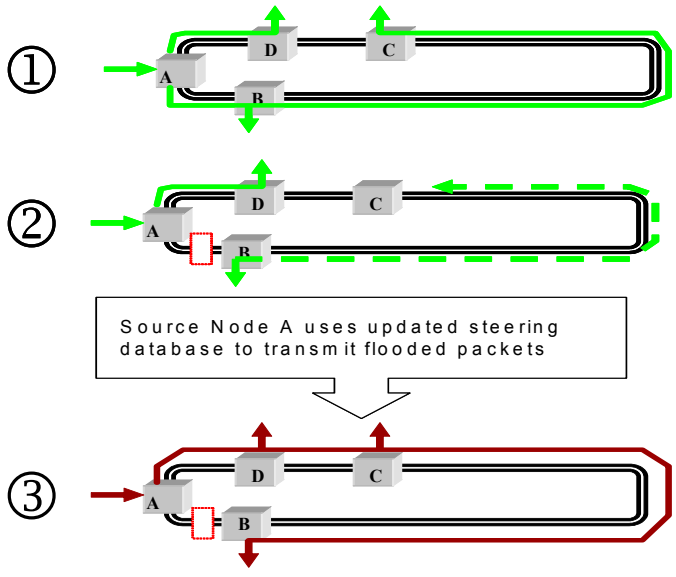


Figure A.15—Reorder scenario: Protection switch during bidirectional flood

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: In strict mode, steering systems use the context containment mechanism to ensure in-flight data frame are removed from the ring before frames 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.16. 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.

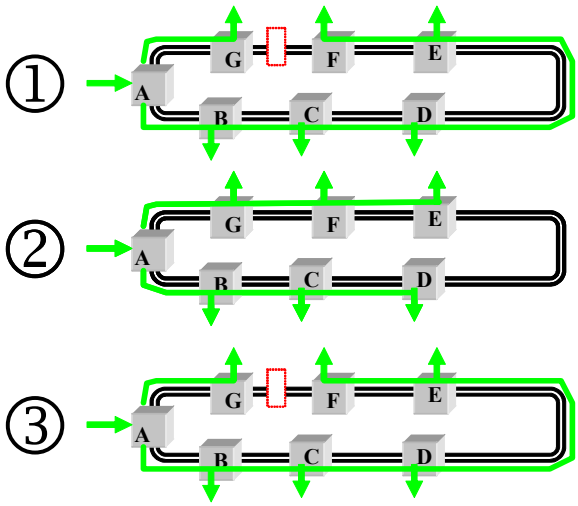


Figure A.16—Reorder scenarios: Cascading failures during bidirectional flood

Cause: At step 1, consider frames in-flight on CCW ringlet (using context #1). At step 2, frames are launched on CW and CCW ringlet (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 CCW in-flight frames using context #1 are still in-flight. Station E can accept in-flight frames on the CCW 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: In strict mode, steering systems use the context containment mechanism to ensure in-flight data frame are removed from the ring before frames using a new context are launched

A.9.3 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.17. 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.

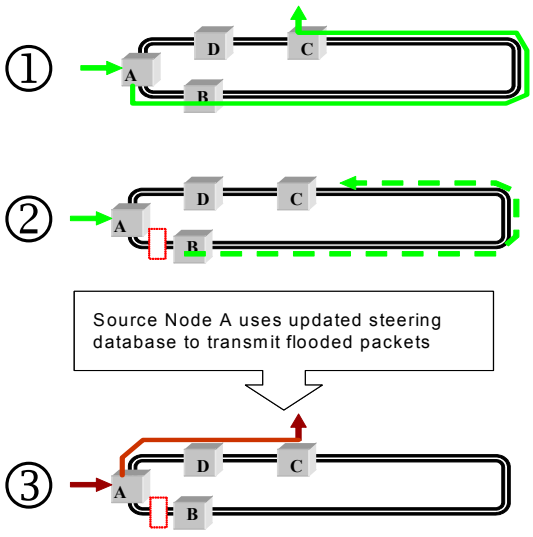


Figure A.17—Protection switching during unicast transmission

Cause: At step 1, consider frames 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 frames destined to C now are sent over the CW ringlet.
Problem: Frame reorder can occur if station C receive context 2 frames before context 1 frames.
Solution: In strict mode, steering systems use the context containment mechanism to ensure in-flight data frame are removed from the ring before frames 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.18. 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

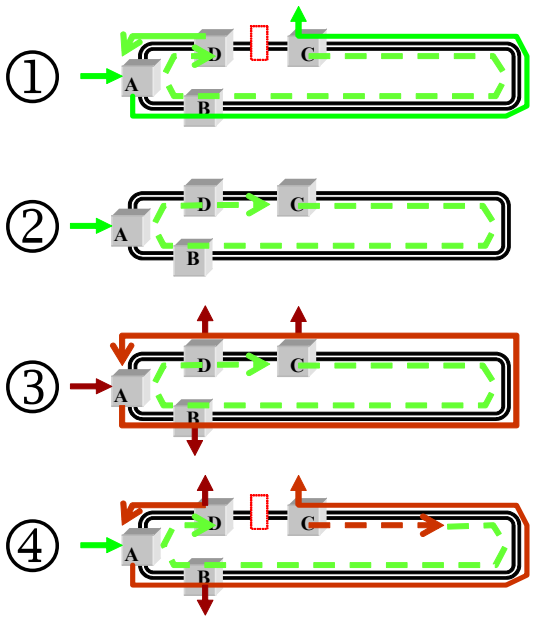


Figure A.18—Reorder scenarios: Cascading failures during unidirectional flood

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 TTL 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 TTL 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: Strict mode frames are not wrapped.