Avoiding Frame Duplication and Misordering in 802.17

David James (dvj@alum.mit.edu)

Anoop Ghanwani (anoop@lanterncom.com)

Introduction

- 802.1D has stringent requirements on frame duplication and reordering
- In the absence of enhanced bridging, all frames with destination MAC addresses that are not ring-local are flooded
- This presentation describes a solution for avoiding frame duplication even when flooding bridged frames

Functional Requirements

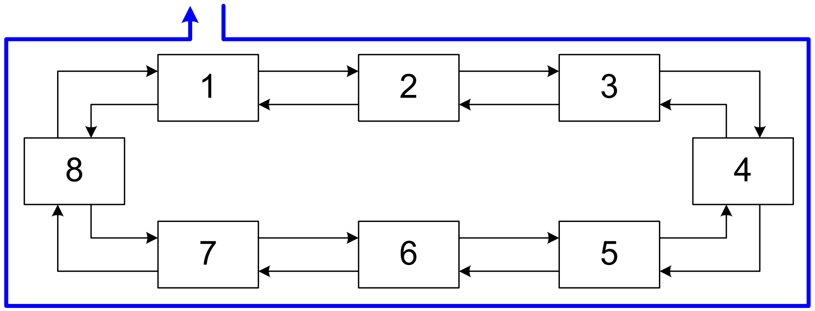
- There are many ways to flood frames
  - The MAC should be able to change flooding direction/method and still maintain packet order
- A node should be able to source packets without knowing that topology maps on all nodes on the ring are in sync
  - However, the packets may only get delivered only when the topology maps are in sync
  - The "topology map" refers to the picture of the ring that the node sees when combining information from both topology and protection protocols
- Must support bridged and non-bridged traffic on the same ring
  - MAC DA and SA may not be local to the ring

Flooding Alternatives

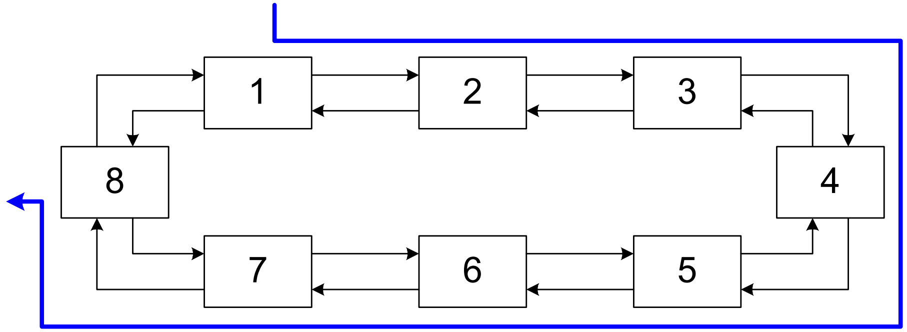
Method	Description
Unidirectional1	Unidirectional flood with destination set to the source.
Unidirectional2	Unidirectional flood with destination set to the node upstream of the source.
Bidirectional1	Flood in both directions with destination as the same station; specify only one of the copies for delivery to the destination.
Bidirectional2	Flood in both directions; in each direction the destination specified is a different node.

Flooding Alternatives

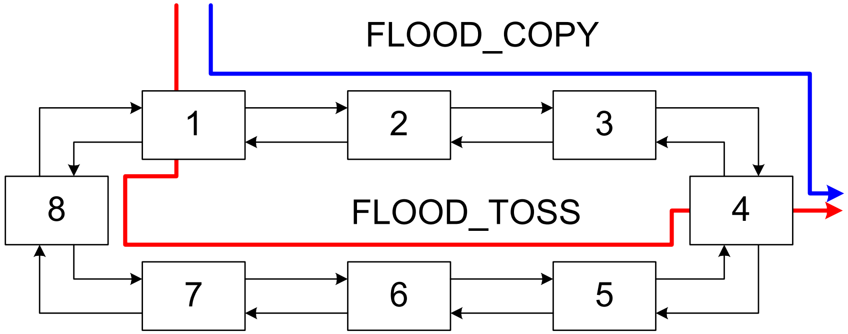
Unidirectional1



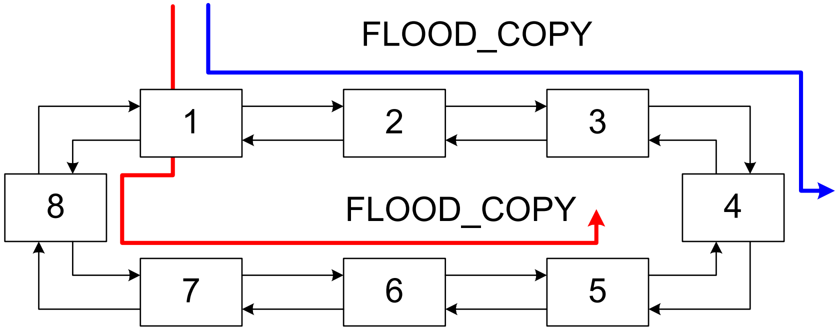
Unidirectional2



Bidirectional1



Bidirectional2



Some Challenging Scenarios

- Bidirectional flooding in use due to a failure and ring heals
- A node sources a multicast frame and dies and the ring uses wrapping
- With bridged packets is there is no way (currently) to identify the ring-local source and destination of the packet
  - Using the TTL for limiting the scope of such frames is insufficient

Basic Elements of the Proposal to Avoid Frame Duplication

- Use of ring-local source and destination identifiers
  - The DSID is always set for flooded frames to a station ID that will strip it

- Use of a bit in the header to indicate FLOOD_TOSS or FLOOD_COPY

- Use of a special procedure for marking and validating wrapped frames

Why Use Station IDs?

- Explicitly identifying the source and destination on the ring helps bridging
- The MAC DA and SA always refer to end stations
- Without station IDs, it is not possible to strip bridged frames based on source / destination address
  - Also increases potential of frame duplication during failure scenarios
- Stations IDs can be 8-bits or fewer depending on max. number of stations supported

Assignment of 7-bit Station IDs

- 7-bits supports a maximum of 128 stations on the ring
- Topology discovery can be enhanced to provide these in an automated fashion
  - Details are provided in a companion presentation and written contribution by David James
- Alternative methods for assignment of station IDs
  - Manual configuration
  - 48-bit MAC addresses (requires a CAM in the transit path)

Frame Sourcing Rules

- frame.timeToLive = 255
- frame.Wrapped = 0
- frame.SSID = MSID (source identifier)
- Setting of frame.Type and frame.DSID as follows:
  - Local or global unicast

    frame.type = DIRECT_DATA, frame.DSID = MSID
  - Unidirectional1 flooding

    frame.type = FLOOD_TOSS, frame.DSID = MSID
  - Bidirectional1 flooding, Direction A

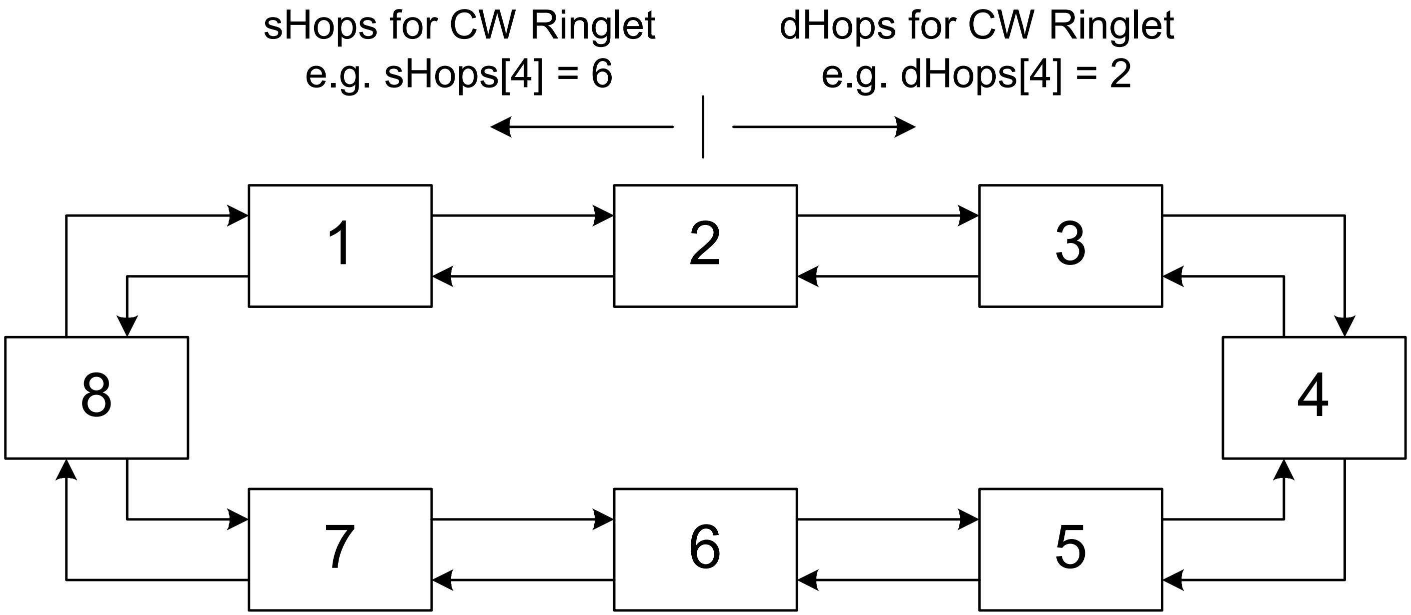
    frame.type = FLOOD_COPY, frame.DSID = HSID (SID of the midpoint station)
  - Bidirectional1 flooding, Direction B

    frame.type = FLOOD_TOSS, frame.DSID = HSID
  - Similarly, the fields can be set for Unidirectional2 and Bidirectional2 flooding methods

Additional Processing for Wrapped Frames

- At the first wrap point, discard if frame.wrapped != 0
- When returning through the source, set frame.wrapped = 1
- No Checks are done on the frame while on the wrapped path, but the TTL is always decremented
- At the second wrap point
  - Discard if frame.wrapped == 0

A Bit of Terminology



- sHops[n]: Distance of node n in the direction of receipt of traffic for a ringlet (i.e in the direction of *source*)
- dHops[n]: Distance of node n in the direction of transmission of traffic for a ringlet (i.e. in the direction of *destination*)

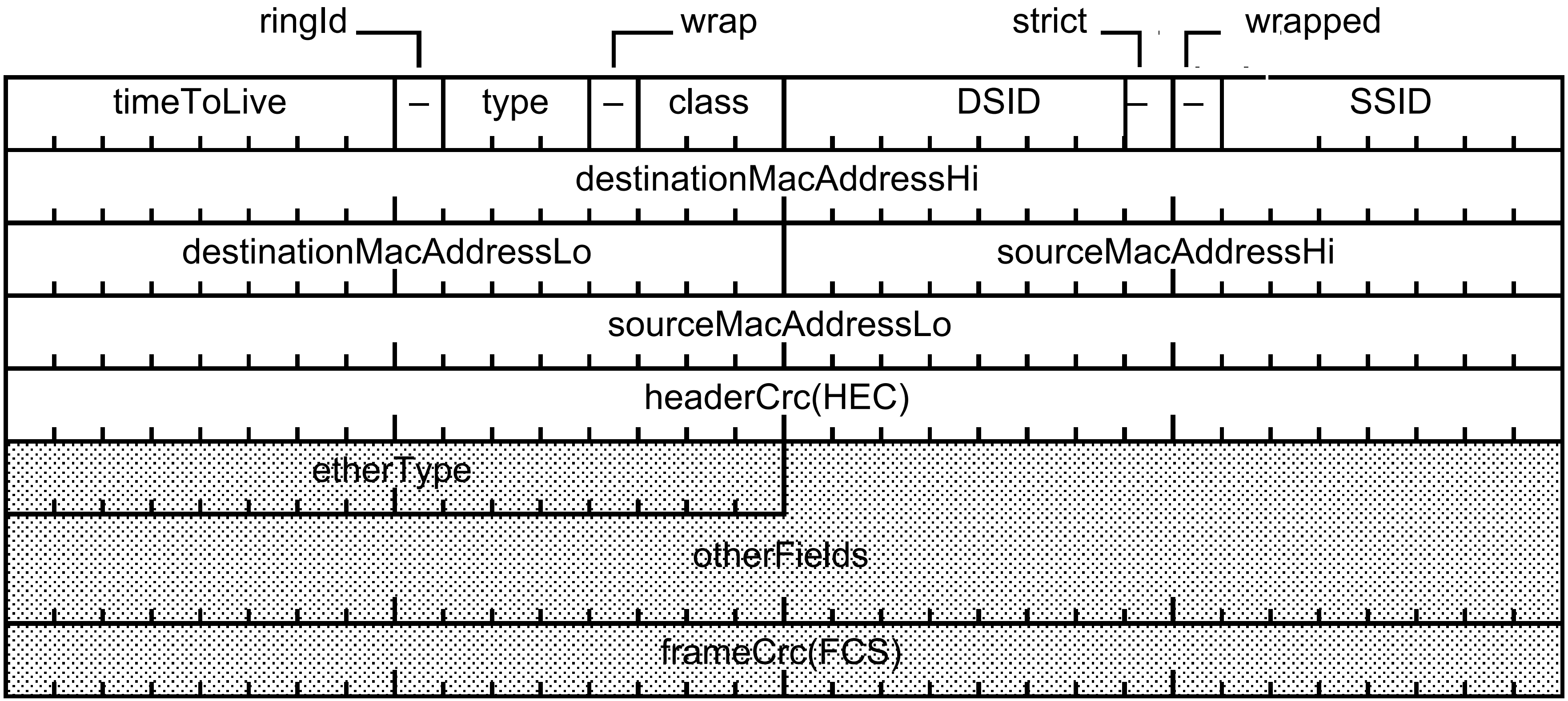
Node Processing Rules

- TTL is always assumed to be set to 255
- Address deletion—*Both* of the following must be satisfied
  - frame.srcMacAddr != frame.srcMacAddr
  - frame.srcMacAddr == database.myMacAddr
- Identifier deletion—*Both* of the following must be satisfied
  - frame.SSID != frame.DSID
  - frame.SSID == database.MSID
- Duplicate squash for unwrapped ring—frame.wrapped == 0 and *any* of the following is satisfied
  - sHops[frame.SSID] != 256 – frame.timeToLive
  - sHops[frame.DSID] == NULL (Destination disappeared)
  - sHops[frame.DSID] < sHops[frame.SSID] (destination appears to have been passed)
- Duplicate squash for wrapped ring (to be done only by nodes supporting wrapping)—frame.wrapped == 1 and *any* of the following is satisfied
  - database.c == 0 (ring is unwrapped)
  - dHops[frame.DSID] == NULL (Destination disappeared)
- If frame is not deleted, frame is copied based on the frame.Type
  - FLOOD_COPY is copied to the client
  - FLOOD_TOSS is copied to the client, unless frame.DSID == MSID
- If frame.DSID == MSID frame is stripped
- The TTL is decremented and the frame is forwarded

Avoiding Frame Misordering

- Requires a ring purge whenever the flooding mechanism changes, e.g. unidirectional to bidirectional, etc.
  - A change is usually required if a protection event happens
- This can be achieved by having each node do the following when it sees a protection status message indicating a change
  - Empty its transit buffers
  - Source no new traffic for a RTT

Proposed Frame Format



ringId

wrap

strict

wrapped

timeToLive

–

type

–

class

DSID

–

–

SSID

destinationMacAddressHi

destinationMacAddressLo

sourceMacAddressHi

sourceMacAddressLo

headerCrc(HEC)

etherType

otherFields

frameCrc(FCS)

Summary

- A robust method for ensuring no duplication was presented
- The mechanism requires only 2 additional bytes in the frame header as part of the RPR control information
- The frame format is fixed regardless of whether or not the frame is a bridged frame