

# RPR 802.1 MAC Service Compliance Analysis

---

Marc Holness  
Nortel Networks

# Objective

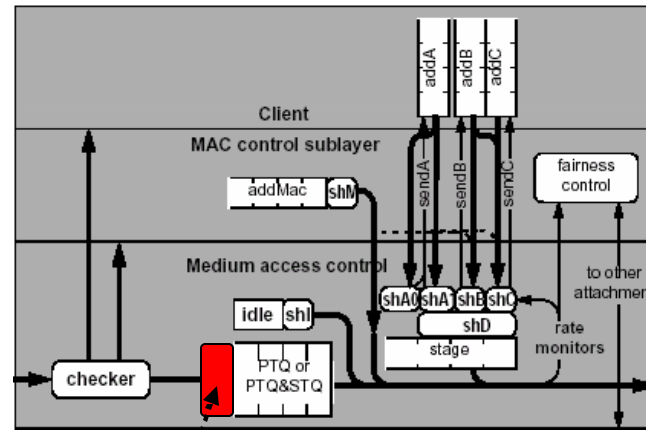
- Analyze P802.17 Draft 2.1 frame Tx/Rx and 802.1 MAC Service adherence consistency checks
- Demonstrate, via example scenarios, deficiencies with current proposal
- Propose extensions to current consistency check rules

# **Summary of RPR**

## **Draft 2.1 consistency checks**



# Data path consistency checks



```
// Wrap checks
if (station.wrap)
    if (!frame.wrapEligible)
        Discard();
```

```
#define numHops (frame.ttlBase-frame.timeToLive)
#define primaryRinglet (frame.RI==myRingletID)

// Consistency checks are performed on strict data frames
if ((frame.frameType==DATA)&&(frame.strictOrder==1)) {

    // Set pastSource when frame transits source station on secondary ringlet
    if ((!primaryRinglet)&&(frame.sourceMacAddress==myStationAddress))
        frame.pastSource=1;

    // Station wrap checks
    if (station.wrap) {
        // Wrap ingress check
        if ((primaryRinglet)&&(frame.pastSource==1))
            Discard();
        // Wrap return check
        if ((!primaryRinglet)&&(frame.pastSource==0))
            Discard();
    }
}
```

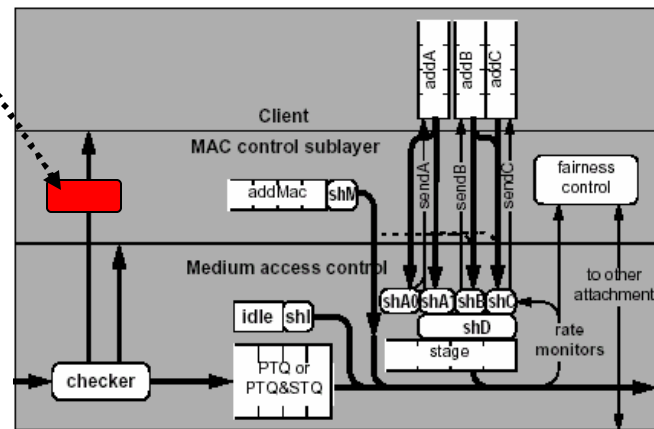
# Receive path consistency checks

```
#define numHops (frame.ttlBase-frame.timeToLive)
#define primaryRinglet (frame.RI==myRingletID)
#define SRC (Topology_Status data base)

// Consistency checks are performed on strict data frames
if ((frame.frameType==DATA)&&(frame.strictOrder==1)) {

    // Source consistency check performed by every station
    if ((primaryRinglet)&&(SRC[numHops, frame.RI]!=frame.sourceMacAddress))
        Discard();

}
```



# D2.1 context containment

- Context containment is a mechanism that purges strictOrdered data frames from the ring
  - Applicable to both wrapping and steering systems, and both mono and dual transit buffers systems
- Steering system: Context containment invoked whenever a change to ring image is detected
- Wrapping system: Context containment invoked only upon unwrap (i.e., ring recovery)

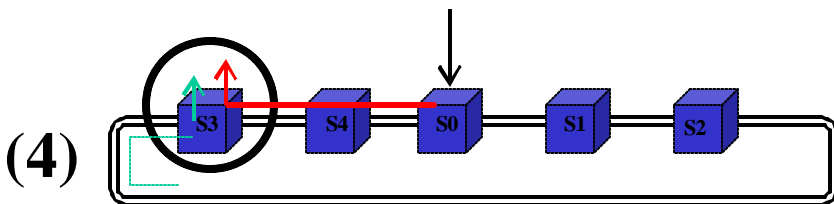
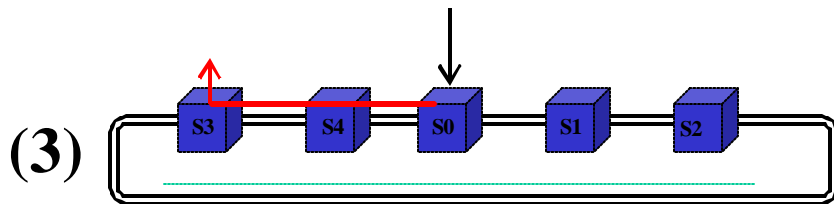
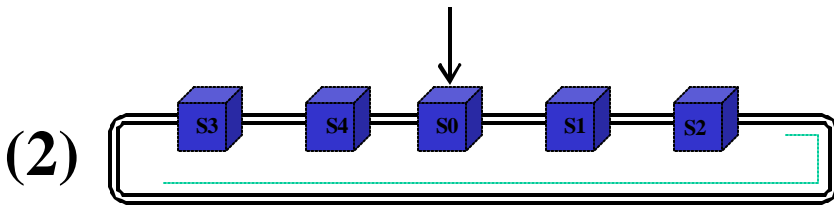
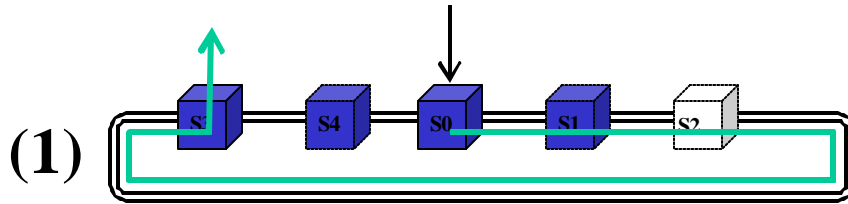
# Existing failure scenarios

# RPR frame transmission scenarios

- Wrapping system non-compliance to 802.1 MAC Service (reorder and duplication) requirements



# Unicast transmission on wrap system



## Step (1)

- S0 is transmitting unicast frame to S3
- S2 is in passthru
- Ringlet0 (i.e., CW ringlet) selected

## Step (2)

- S2 comes out of passthru
- Topology discovery initiated and completed while unicast frames in transit

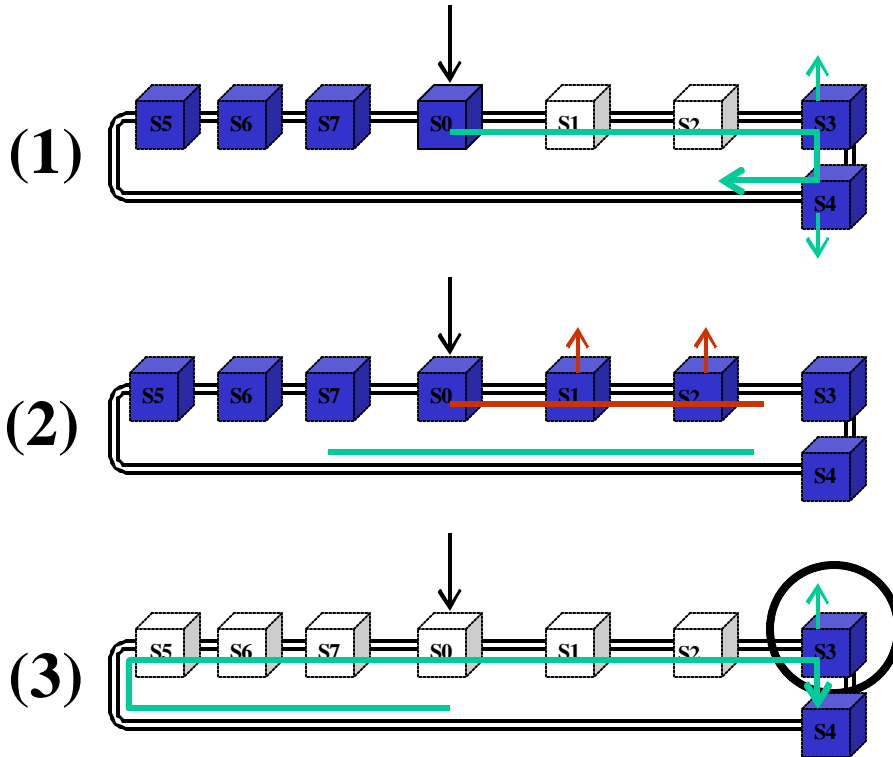
## Step (3)

- S0 transmits unicast frame to S3 via ringlet1 (i.e., CCW ringlet)

## Step (4)

- S3 receives unicast frame via ringlet1
- S3 receives unicast frames via ringlet0 since source consistency check at S3 passes
- Frame reorder occurs at S3!

# Uni-directional flood on wrap system



## Step (1)

- S1 and S2 are in passthru
- Ring is at steady state
- S0 is uni-directionally flooding frames on ringlet0 (i.e., CW ringlet)
- S3 and S4 received flooded frame (but not S5)

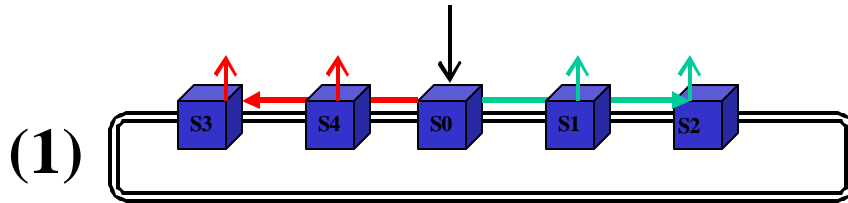
## Step (2)

- S1 and S2 comes out of passthru
- Topology discovery initiated and completed while flooded frames in transit
- Originally sourced in-flight frames from S0 has not arrived at S5 yet

## Step (3)

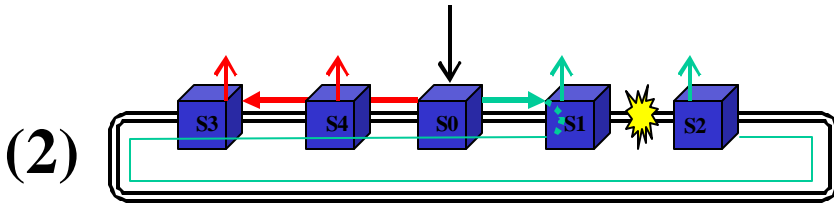
- S5, S6, S7, S0, S1, and S2 enter passthru
- Topology discovery has not been initiated or converged
- Originally sourced in-flight frames from S0 arrive at S3 (again)
- Source consistency check at S3 passes
- Frame duplication occurs at S3!

# Bi-directional flood and wrap



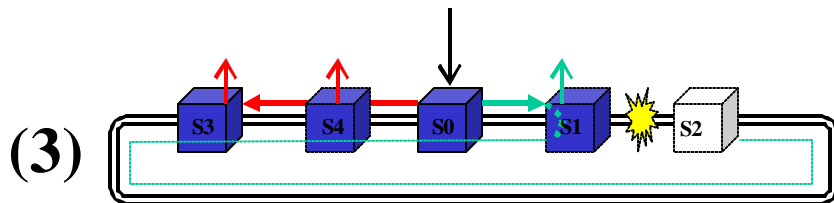
## Step (1)

- S0 is bi-directionally flooding frames on ring



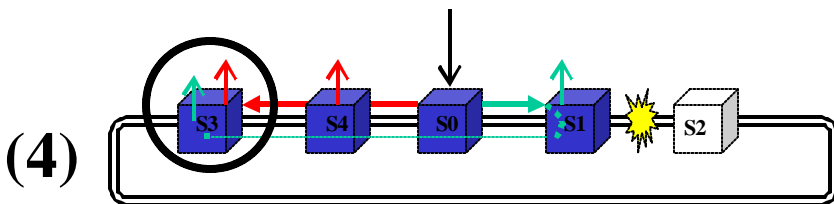
## Step (2)

- Span failure between S1 and S2 occurs



## Step (3)

- S2 goes in passthru
- Passthru detected
- Topology discovery initiated and completed while wrapped frames in-flight



## Step (4)

- S3 return wrap frames on secondary ringlet
- Source consistency check at S3 passes
- Frame duplication occurs at S3!

A vertical teal line on the left side of the slide.

# **Findings and recommendations**

# Findings

- Station passthru (entering and leaving) can alias source consistency check

```
SRC[frame.ttlBase-frame.timeToLive]!=frame.sourceMacAddress
```

which can result in frame reorder  
and/or duplication on wrapping  
systems

# Recommendations

1. For wrapping systems, invoke context containment when station passthru detected or when station added to ring (via coming out of station passthru)
  - Should we generalize to state that context containment should be invoked upon any form of topology rediscovery?

# Recommendations

2. Work with PAH (Protection Ad-Hoc) team to identify mechanism to allow active adjacent station to detect station passthru being entered and exited