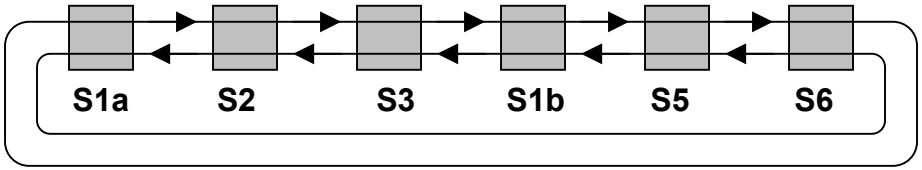
Topology data base - special cases

Leon Bruckman

Detecting duplicate MAC address – Scenario 1



1	macAddressS2	info	ringlet1
1	macAddressS3	info	
0	macAddressS1	info	
1	macAddressS5	info	
1	macAddressS6	info	
0	macAddressS1	info	ringlet0
1	macAddressS2	info	
1	macAddressS3	info	
0	macAddressS1	info	
1	macAddressS5	info	
1	macAddressS6	info	

S1a

1	macAddressS3	info	
0			ringlet1
1	macAddressS5	info	
1	macAddressS6	info	
0	macAddressS1	info	
1	macAddressS2	info	
1	macAddressS3	info	
0	macAddressS1	info	ringlet0
1	macAddressS5	info	
1	macAddressS6	info	
0			
1	macAddressS2	info	
1	macAddressS3	info	

S3

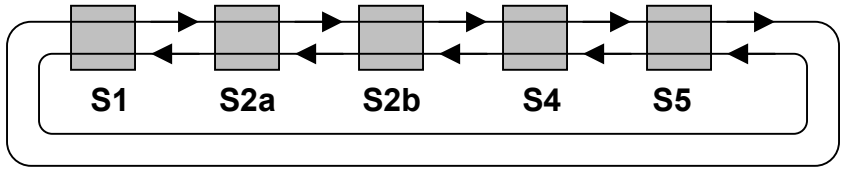
MAC duplication detection Scenario 1

- Duplicated stations
  - No “holes” in topology
  - See themselves in the “middle” of their topology data base in both ringlets
  - Do not see themselves as the farthest station in the topology data base. Looks like open loop, but..
  - Same stations in ringlet0 and ringlet1 and no protection alarms
  - ***Indication***: I am duplicated
  - ***Action***: “Shut-up” or passthrough

MAC duplication detection Scenario 1

- Other station
  - See a “hole” in the topology (since the duplicating stations “steal” one another’s protection frames)
  - Which is the duplicated station ? The one that complies with:
    - *ringlet0hops.duplicatedStation* + *ringlet1hops.duplicatedStation* < *stationsInRing*
  - Is the “hole” created because a station is duplicated, or is there a silent station ? Duplicated station hole if:
    - *ringlet0hops.duplicatedStation* + *ringlet1hops.hole* = *stationsInRing*
    - *ringlet1hops.duplicatedStation* + *ringlet0hops.hole* = *stationsInRing*
  - TLV message can be used to detect which is the duplicated station
  - ***Indication***: Station Sn is duplicated in this ring
  - ***Action***: Don’t talk with duplicated stations

Detecting duplicate MAC address – Scenario 2



0	macAddressS2	info
1	macAddressS4	info
1	macAddressS5	info
1	macAddressS1	info
0	macAddressS2	info
0	macAddressS2	info
1	macAddressS4	info
1	macAddressS5	info
1	macAddressS1	info

ringlet1

ringlet0

S2a

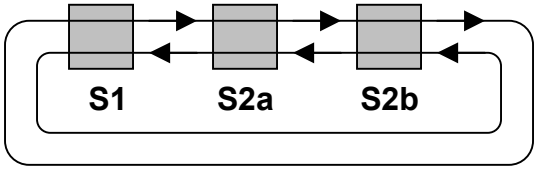
1	macAddressS5	info
1	macAddressS1	info
0		
0	macAddressS2	info
1	macAddressS4	info
1	macAddressS5	info
0	macAddressS1	info
1	macAddressS2	info
0		
1	macAddressS4	info
1	macAddressS5	info

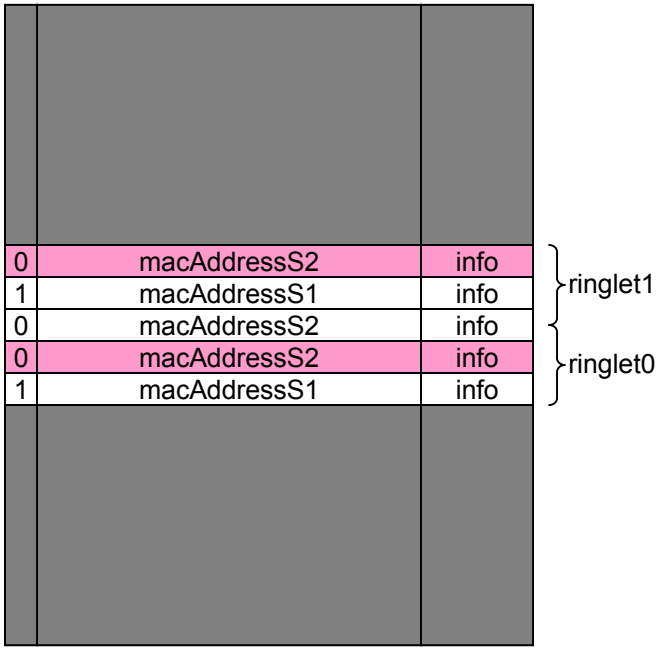
ringlet1

ringlet0

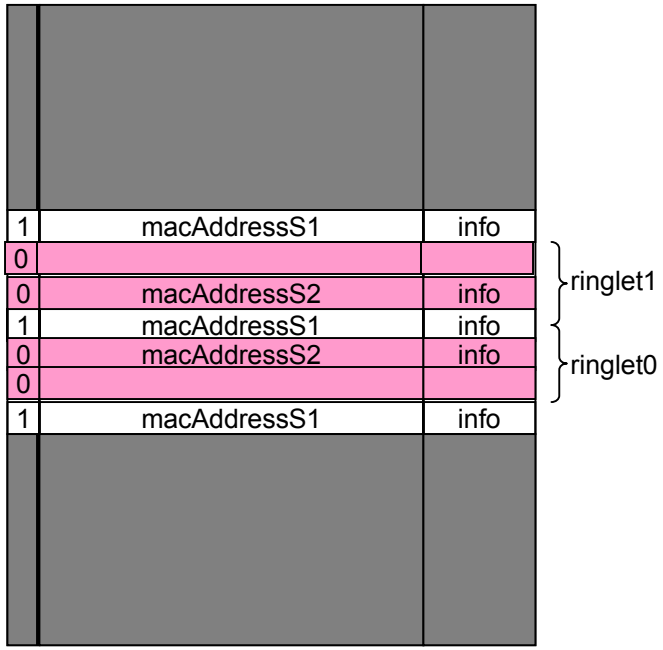
S5

Detecting duplicate MAC address – Scenario 3





S2a



S1

MAC duplication detection Scenarios 2, 3

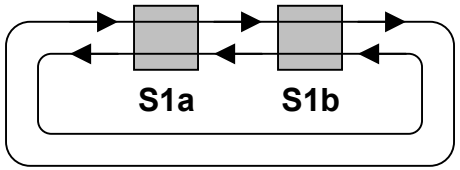
- Duplicated stations
  - No “holes” in topology
  - See themselves in the “middle” of their topology data base in one ringlet and..
  - See themselves as the farthest station in the topology data base in the other ringlet only. Looks like single link failure, but..
  - Same stations in ringlet0 and ringlet1 and no protection alarms
  - ***Indication***: I am duplicated
  - ***Action***: “Shut-up” or passthrough

MAC duplication detection Scenarios 2, 3

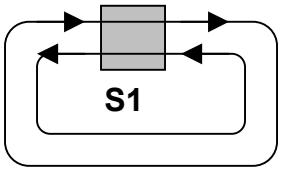
- Other station
  - See a “hole” in the topology (since the duplicating stations “steal” one another’s protection frames)
  - Which is the duplicated station ? The one that complies with:
    - *ringlet0hops.duplicatedStation* + *ringlet1hops.duplicatedStation* < *stationsInRing*
  - Is the “hole” created because a station is duplicated, or is there a silent station ? Duplicated station hole if:
    - *ringlet0hops.duplicatedStation* + *ringlet1hops.hole* = *stationsInRing*
    - *ringlet1hops.duplicatedStation* + *ringlet0hops.hole* = *stationsInRing*
  - TLV message can be used to detect which is the duplicated station
  - ***Indication***: Station Sn is duplicated in this ring
  - ***Action***: Don’t talk with duplicated stations

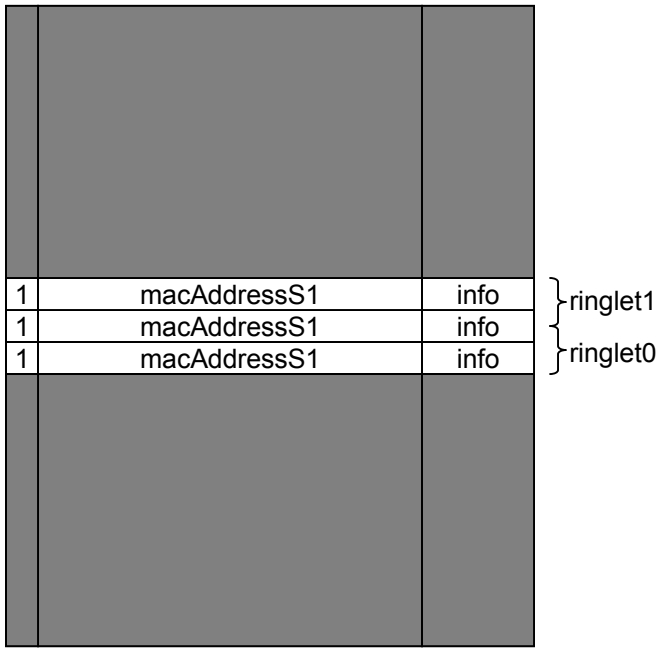
Detecting duplicate MAC address – Scenario 4

4a

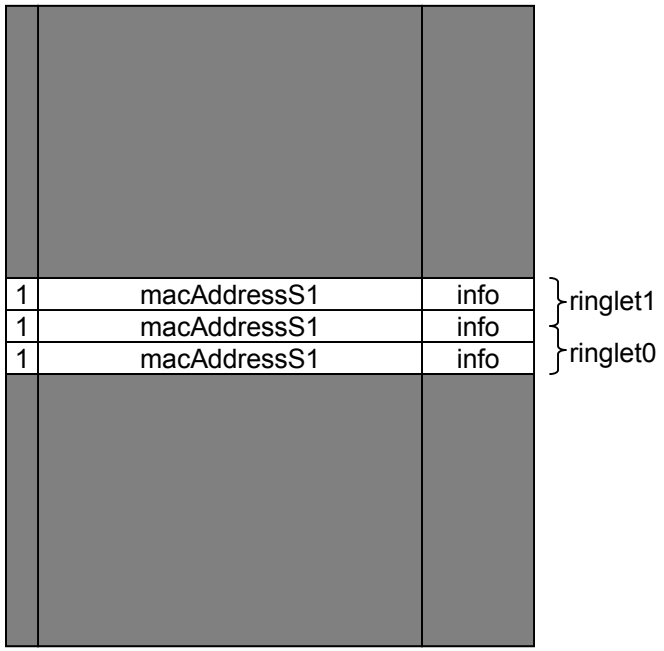


4b





S1a



S1a

MAC duplication detection Scenario 4

- Duplicated stations
  - Stations are not able to differentiate between 4a and 4b.
  - Same will happen if there are N stations in a ring all with the same address
  - The ring has no real value anyway
  - ***Indication***: I am alone.
  - ***Action***: “Shut-up” or passthrough

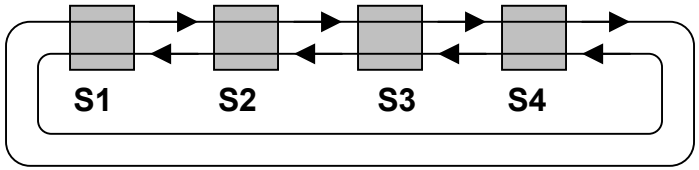
Conclusions “duplicated MAC” - loop

- Fault detection:
  - No alarms
  - Duplicated station sees itself in the middle of one ringlet and no in the edge
  - Other stations see “holes” in the topology data base
  - If all stations have the same MAC address the result emulates a single station in the ring
- Actions:
  - Declare fault
  - Duplicated station stops communication
  - Other station invalidate duplicated stations
  - Stations see different topologies, no conversion to a single ring view – No strict order traffic allowed ?

Too many stations

- For the too many station scenarios we will assume *ttl* is 2 bits
  - MAX_STATIONS = 3
- Conclusions can be extended for the “real” *ttl* (8 bits)
- Only the loop case is presented, the edge case should be analyzed as well

Too many stations – Scenario 1



1	macAddressS2	info	ringlet1
1	macAddressS3	info	
1	macAddressS4	info	
1	macAddressS1	info	ringlet0
1	macAddressS2	info	
1	macAddressS3	info	
1	macAddressS4	info	

S1

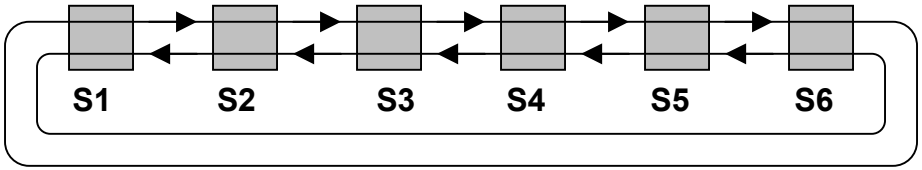
1	macAddressS4	info	ringlet1
1	macAddressS1	info	
1	macAddressS2	info	
1	macAddressS3	info	ringlet0
1	macAddressS4	info	
1	macAddressS1	info	
1	macAddressS2	info	

S3

Too many stations Scenario 1

- Complete topology – no holes
- Number of different stations in topology = MAX_STATIONS + 1
- Do not see themselves as the farthest station in the topology data base. Looks like open loop, but no protection alarms
- ***Indication***: Too many stations
- ***Action***: Operate normally

Too many stations – Scenario 2



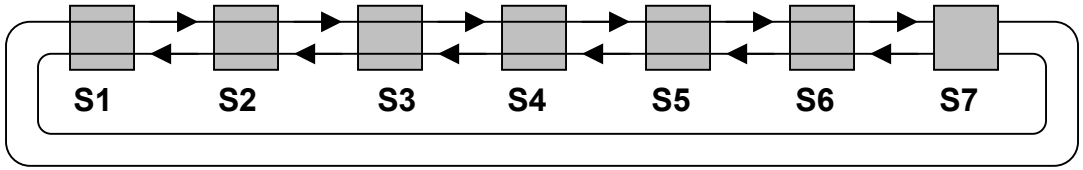
1	macAddressS4	info	ringlet1
1	macAddressS5	info	
1	macAddressS6	info	
1	macAddressS1	info	ringlet0
1	macAddressS2	info	
1	macAddressS3	info	
1	macAddressS4	info	

S1

Too many stations Scenario 2

- Complete topology – no holes
- Some stations appear in both ringlets, some don't
- Number of different stations in topology = MAX_STATIONS + 3
- Do not see themselves as the farthest station in the topology data base. Looks like open loop, but no protection alarms
- ***Indication***: Too many stations
- ***Action***: Communicate with each station according to the ringlet in which it appears as valid

Too many stations – Scenario 3



1	macAddressS5	info
1	macAddressS6	info
1	macAddressS7	info
1	macAddressS1	info
1	macAddressS2	info
1	macAddressS3	info
1	macAddressS4	info

ringlet1

ringlet0

S1

Too many stations Scenario 3

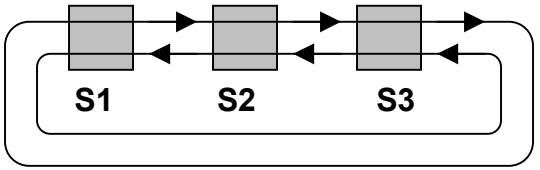
- Complete topology – no holes
- Different stations in each ringlet
- Number of different stations in topology = MAX_STATIONS + 4
- Do not see themselves as the farthest station in the topology data base. Looks like open loop, but no protection alarms
- ***Indication***: Too many stations
- ***Action***: Communicate with each station according to the ringlet in which it appears as valid

Conclusions “too many stations” - loop

- Fault detection:
  - No alarms
  - Station does not see itself at the farthest in any ringlet
  - Number of stations > MAX_STATIONS
- Actions:
  - Declare fault
  - Continue to communicate with stations according to their validity in the topology data base.
  - Stations see different topologies, no conversion to a single ring view – No strict order traffic allowed ?

Going into passthrough – Scenario 1 (Stable)

1a



1	macAddressS1	info
1	macAddressS2	info
1	macAddressS3	info
1	macAddressS1	info
1	macAddressS2	info
1	macAddressS3	info
1	macAddressS1	info

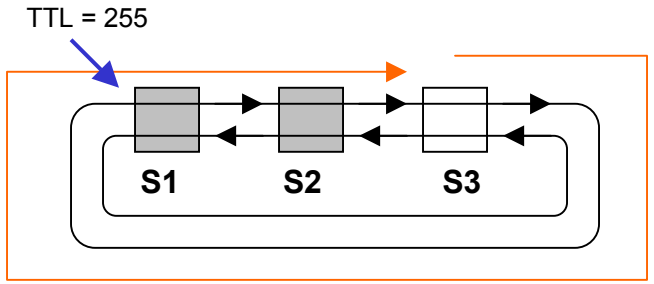
ringlet1

ringlet0

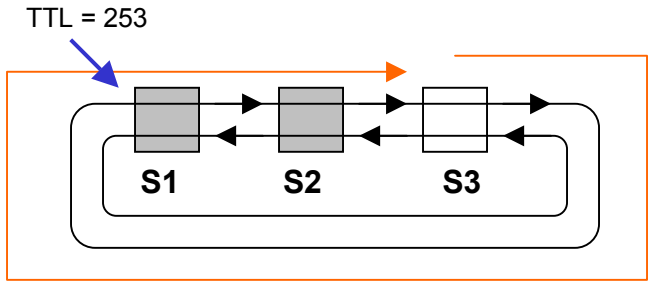
S1

Going into passthrough – Scenario 1 (S6 passthrough)

1b



1c



1	macAddressS1	info	
1	macAddressS2	info	ringlet1
1	macAddressS3	info	
1	macAddressS1	info	
1	macAddressS2	info	ringlet0
1	macAddressS3	info	
1	macAddressS1	info	

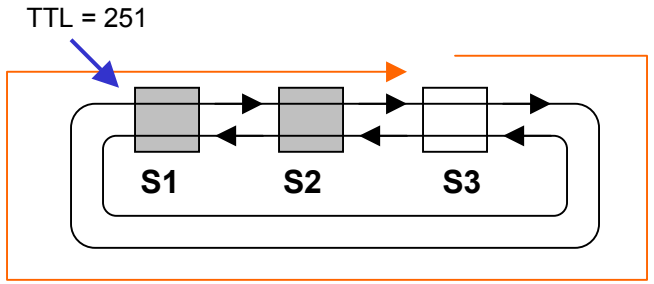
S1

1	macAddressS3	info	
1	macAddressS2	info	ringlet1
0	macAddressS3	info	
1	macAddressS1	info	
1	macAddressS2	info	ringlet0
0	macAddressS3	info	
1	macAddressS1	info	

S1

Going into passthrough – Scenario 1 (S6 message)

1d



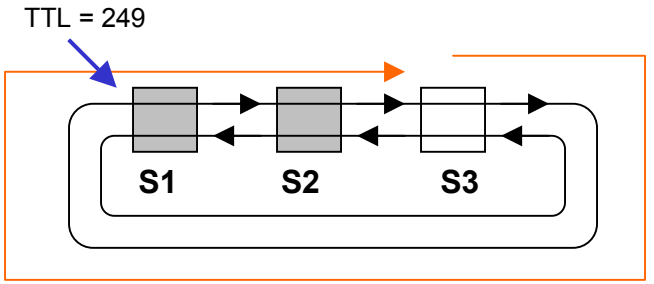
1	macAddressS3	info
0		
0	macAddressS3	info
1	macAddressS2	info
0	macAddressS3	info
1	macAddressS1	info
1	macAddressS2	info
0	macAddressS3	info
1	macAddressS1	info

ringlet1

ringlet0

S1

1e



1	macAddressS3	info
0		
0	macAddressS3	info
0		
0	macAddressS3	info
1	macAddressS2	info
1	macAddressS3	info
0	macAddressS1	info
1	macAddressS2	info
0	macAddressS3	info
1	macAddressS1	info

ringlet1

ringlet0

S1

Going into passthrough scenario 9

- Figure 1a
  - Stable configuration no passthrough, small ring (RTT very low, small number of stations)
- Figure 1b
  - S3 message reaches S1, no problem detected
  - S3 enters passthrough, so its message will not be stripped until TTL expires
- Figure 1c
  - S1 receives stale S3 message with hops count not as expected
  - S1 assumes that there are two stations that went out of passthrough
- Figures 1d, 1e
  - S1 assumes that two new stations go out of passthrough every time it sees S3 message !
- To overcome this scenario we need to indicate that the message is stale
  - Use the toggle flag
    - Transmitter toggles flag every message
    - Receiver ignores messages with bit not toggled

Conclusions “going into passthrough” - loop

- Stale TP frames may cause damage for long periods for passthrough stations
- Need indication that TP frame is stale to ignore it
  - Toggle bit does the job

Summary

- There are several fault scenarios that impact the topology “view”
  - Scenarios should be described in draft
  - Use the presented scenarios, and add more if required
- Define how alarms are declared/cleared
  - Comments filed against D2.2
- Define strict order traffic behavior during stable fault conditions
  - Different stations have a different ring topology view