BROCADE

Reaction Point Identifier for 802.1Qau

Is it really needed?

au-ghanwani-rpid-0908-v1

Anoop Ghanwani (anoop@brocade.com)

Asif Hazarika (ahazarik@fma.fujitsu.com)

September 2008

Overview

- Motivation
- Problems solved by RPID
- Problems not solved by RPID
- Problems introduced by RPID
- Deployment considerations
- Recommendation for the WG

Motivation

- RPIDs were introduced in P802.1Qau/D1.2
  - Based on <au-nfinn-RPID-0508-v03.pdf>
  - Mainly needed for dealing with LAGs
    - Avoiding fate sharing in the network
    - Processing of CNMs at the RP
  - If we find problems with it, we revisit the decision
- QCN was particularly attractive because it didn’t require any frame format changes
  - But now we’re revisiting that assumption
- Issues and arguments
  - What are some of the challenges with getting RPIDs to do what they are being advertised for?
  - Are RPIDs absolutely necessary?

Problems solved by the RPID

- Fate sharing when using LAG
- Reaction time when using LAG across multiple NICs in an end station
  - In the absence of RPID and cooperation between bridges and NICs, software would need to be involved in processing of CNMs adding extra processing delay
- Association of CNM to RL without having to parse the SDU that may have added headers from the network

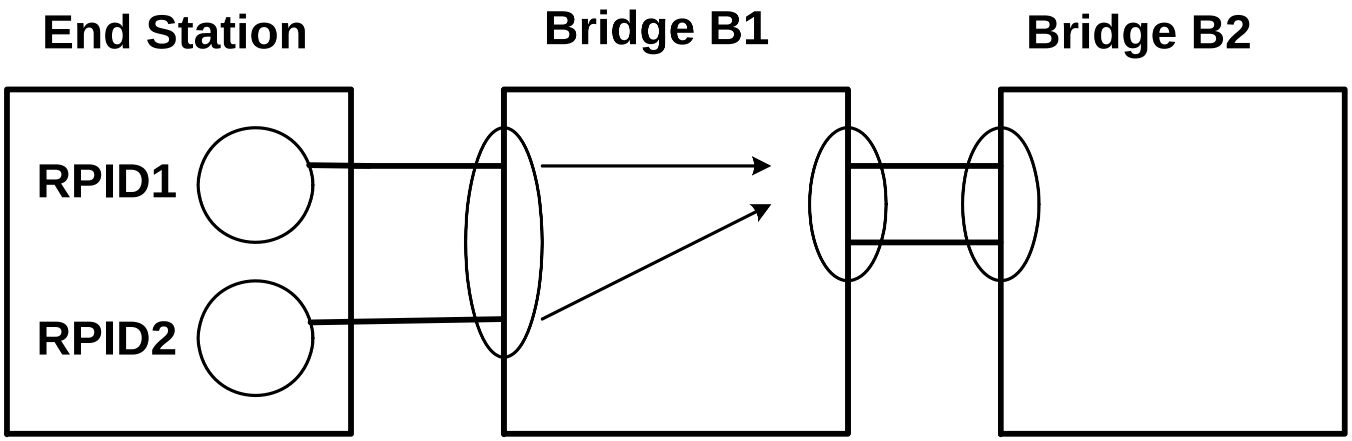
Problems not solved by the RPID

- LAGs across NICs when stateful offload is being performed
  - The requirement here is that forward and reverse traffic need to use the same member of the LAG
  - RPID doesn’t help with this
  - NIC teaming is used for HA and is more common than LAG
  - Weakens the argument for needing multi-NIC LAGs
- Abstracts out flow information
  - Current proposal doesn’t send the SDU
    - End station is worse off than without RPID with respect to knowing which “flow” is the problem one
  - Alternative is to send the SDU
    - But then we lose the advantage of not having to parse the SDU
  - Yet another alternative is to use a Flow ID
    - See <au-bestler-flowidoptions-0808-01.pdf>
    - But then we lose the ability to manage LAGs

Problems <u>introduced</u> by the RPID

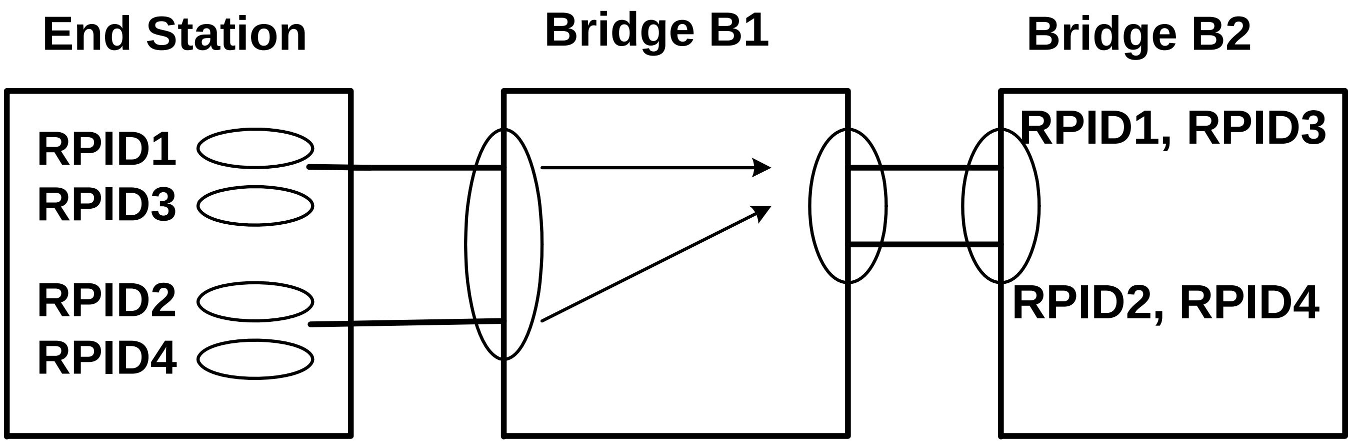
- Need to standardize a method of hashing based on RPID so that switches and NICs agree on which RPIDs are used on a member
  - May diminish the value of the fate sharing capability if the number of RPIDs is different than the number of members in a LAG
- Need to modify end station LAG implementations to deal with flow to RPID assignments
  - May not be easy depending on OS
- Need to worry about stripping these tags off at the edges of CNDs
- **Breaks the parsing functions of every bridge ASIC out there today**

Problems introduced by RPIDs (1)



- If B1 ends up forwarding both RPIDs on the same member link towards B2, having the RPID doesn't help
- Bridges and end stations need to:
  - Use the RPID as the only input to the hash
  - Agree on the hashing algorithm

Problems introduced by RPIDs (2)



- Assume the end stations and the bridges agree on hashing
- Assume end station allocates RPIDs as flows arise - RPID1, RPID2, …
  - What happens when the flows going through RPID1 and RPID3 are the only ones active?
  - Fate sharing even with RPID
- How does the end station pick the RPID to ensure there will not be fate sharing?

Do we absolutely need an RPID?

- Some amount of fate sharing among flows is inevitable
  - RPIDs don’t address every possible situation
- LAGs on multiple NICs is not very common
  - NIC teaming is more common for high availability
- Simulations with flows sharing fate have shown acceptable performance
  - We are doing much, much better than the fate sharing of PFC anyway
  - Some of these problems can be addressed by getting end stations and switches to agree on the hashing algorithm

Deployment considerations

- Introducing a new tag will slow the standardization, development and deployment of CN
  - Data center bridges are starting to be deployed
- Dealing with a new frame format is non trivial
  - New sniffers, debuggers, …

Recommendations for the WG

- Avoid requiring RPIDs in the first version of the spec
  - We have made many compromises with respect to performance of the algorithm arguing for simplicity
  - The goal is achieving “acceptable performance”, not optimization of all possible cases
  - We should not burden all implementations to fix some corner cases like LAG across NICs which is one of the problems solved by RPID
    - LAG across multiple NICs is not common
    - LAG across stateful NICs is not possible
- We can always discuss RPIDs in future revision to the spec
  - It is fairly easy to get RPID/non-RPID implementations to interoperate so that incremental deployment is possible

BROCADE

TM

THANK YOU