TRILL Smart Endnodes and Dumb RBridges

Radia Perlman, Intel
Radia.perlman@intel.com

Motivation

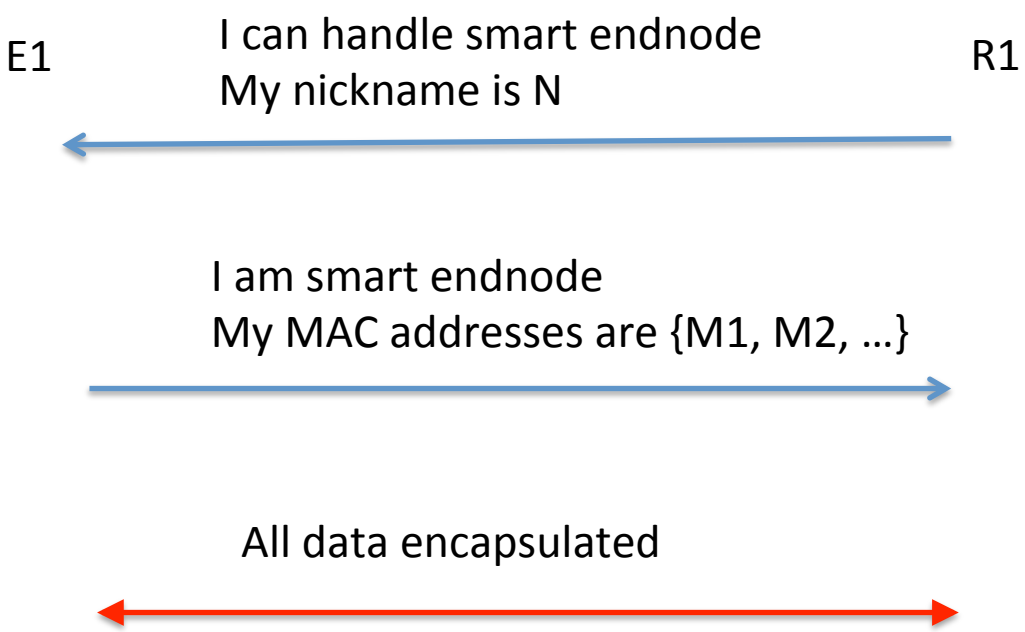
- Save endnode learning table (MAC, RB) space in border RBs
- Smart endnode E need only know info about nodes it is currently talking to
- Endnode more likely to quickly notice if destination has moved (can't be reached)

Dumb RBridge

- Endnode pretends to be Rbridge
- Generate LSP, using “Overload” bit so no paths computed through it
- Obtain nickname
- Ignore LSPs (other than choosing nickname)
- Only downside…but it’s a big downside… consume nicknames
- So instead…smart endnode

Smart Endnode

- Invisible to campus
  - Don't generate LSPs
  - Don't consume nickname
- But do learning of (MAC, RBridge)
- Do encapsulation/decapsulation
- Using attached RBridge nickname


E1
I can handle smart endnode
My nickname is N
R1
I am smart endnode
My MAC addresses are {M1, M2, …}
All data encapsulated

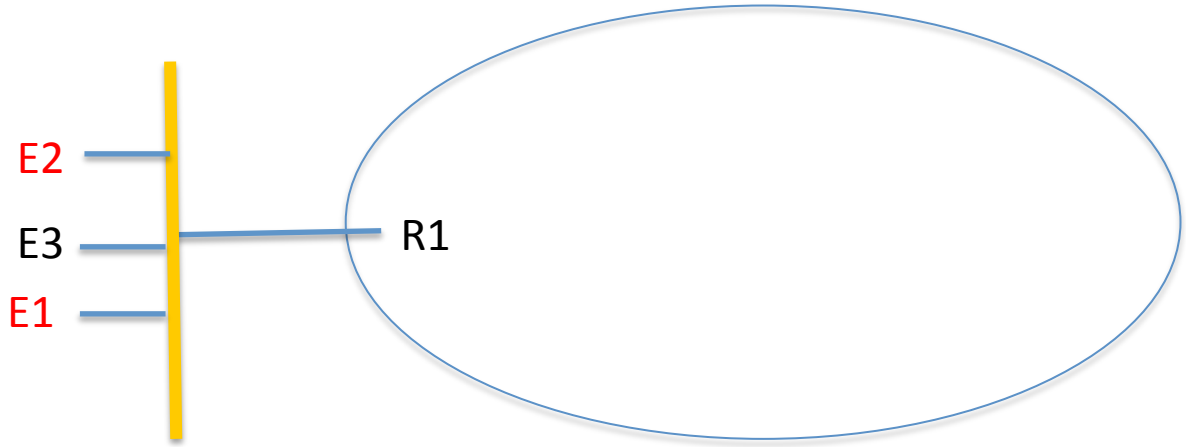
R1

- Attached to access (or universal) link L
- If receive encapsulated packet from L, perhaps check source MAC, VLAN, ingress nickname
  - Otherwise, treat the packet like any encapsulated packet
- If receive encapsulated packet from campus with egress=R1
  - Check MAC. If it belongs to smart endnode on link L, then forward it to L still encapsulated
  - Else, decapsulate

Simplification from draft

- Link has to be completely "smart endnode only" or "no smart endnodes"
- AF announces whether link L is "smart only"
- If there's a dumb endnode on L, AF R will ignore it
  - R will ignore native traffic on L
  - R will not decapsulate traffic onto L

Simplification: Don't allow "hybrid" link



R1 configured that this port is for smart endnodes
R1 announces that
E1 and E2 act like smart endnodes
E3 gets ignored

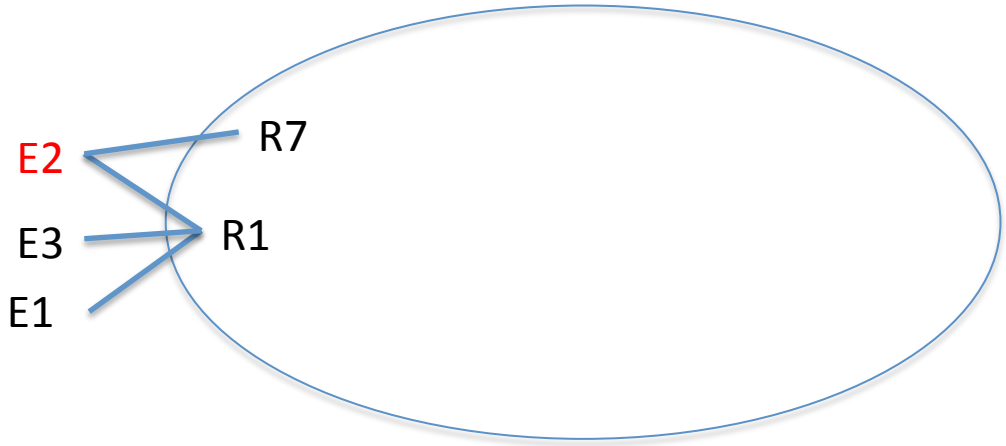
R1 rules

- Assume R1 has some "smart endnode" links, say L_a, L_b, L_c, and some "normal" links, say L_d, L_e
- If R1 receives packet from campus with egress="R1", R1 chooses which port
  - If MACx belongs to smart endnode E, E will have explicitly announced to R1, and R1 knows to send to, say, L_a, and leave encapsulated
  - If MACx is dumb endnode, R1 has to learn based on seeing traffic from MACx
  - If R1 does not know where MACx is…then it only transmits it natively, and only on non-smart links

Another subtlety

- If smart E1 is multihomed to R1 and R2, which nickname should E1 use?
  - Pseudonode
  - R1's
  - R2's
- If choose R1 or R2
  - Should be careful not to switch between…it will confuse endnode learning
  - return traffic will go via that RB
- If choose pseudonode, has to be to EXACT set of attached RBs

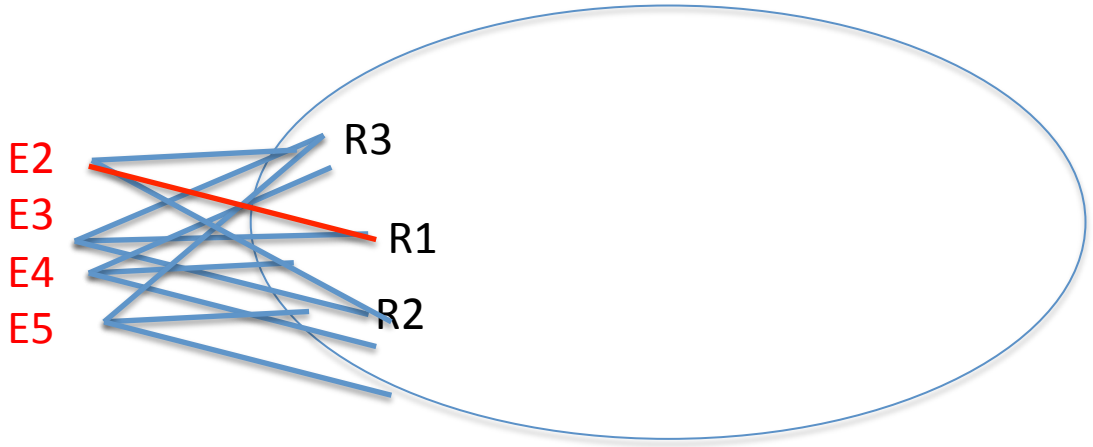
Smart endnode double homed



When encapsulating E2 can use as ingress nickname:

- either R7's nickname or R1's.
- Choose one, unless it's down
- Be told a pseudonode nickname by R1/R7 and use that
- If lots of endnodes dual-homed to {R1, R7}, pseudonode nickname is very useful

Problem if using pseudonode



Suppose E2, E3, E4, E5 told to use pseudonode P

- R3, R1, R2 all report in LSP they can reach P
- If E2's link to R1 dies, it can't use P anymore
- We have the same problem with active/active solution

Possible solution

- E1 use pseudonode P if all its uplinks are working
- If any uplink goes down, E1 chooses one of its attached Rbridges, say R2, and always uses that nickname as ingress
  - Downside; traffic to E1 will always go via R2

Contrast with Linda's "Directory Reliant" smart endnode

- Directory Reliant
  - Does not learn from data; ONLY learns from directory
  - Therefore, does not need to see packets encapsulated
  - Does not need to announce itself to R
  - Can mix smart and dumb endnodes on the same link

If RBridges and smart endnodes ONLY learn from directory

- Makes pseudonode with active/active unnecessary
- If directory advertises that E is attached to {R1, R2, R3}, then you don't need a pseudonode for all the endnodes attached to the same set of RBridges
- And you don't have to worry about RPF check, or endnode bouncing for multicast…just use as "ingress" whichever Rbridge link you chose

But….

- The nice simplicity of the directory approach for active/active assumes EVERYONE will use the directory only