TRILL issue: Using Pseudonode Nicknames for Ingress RBridge

Radia Perlman

radiaperlman@gmail.com

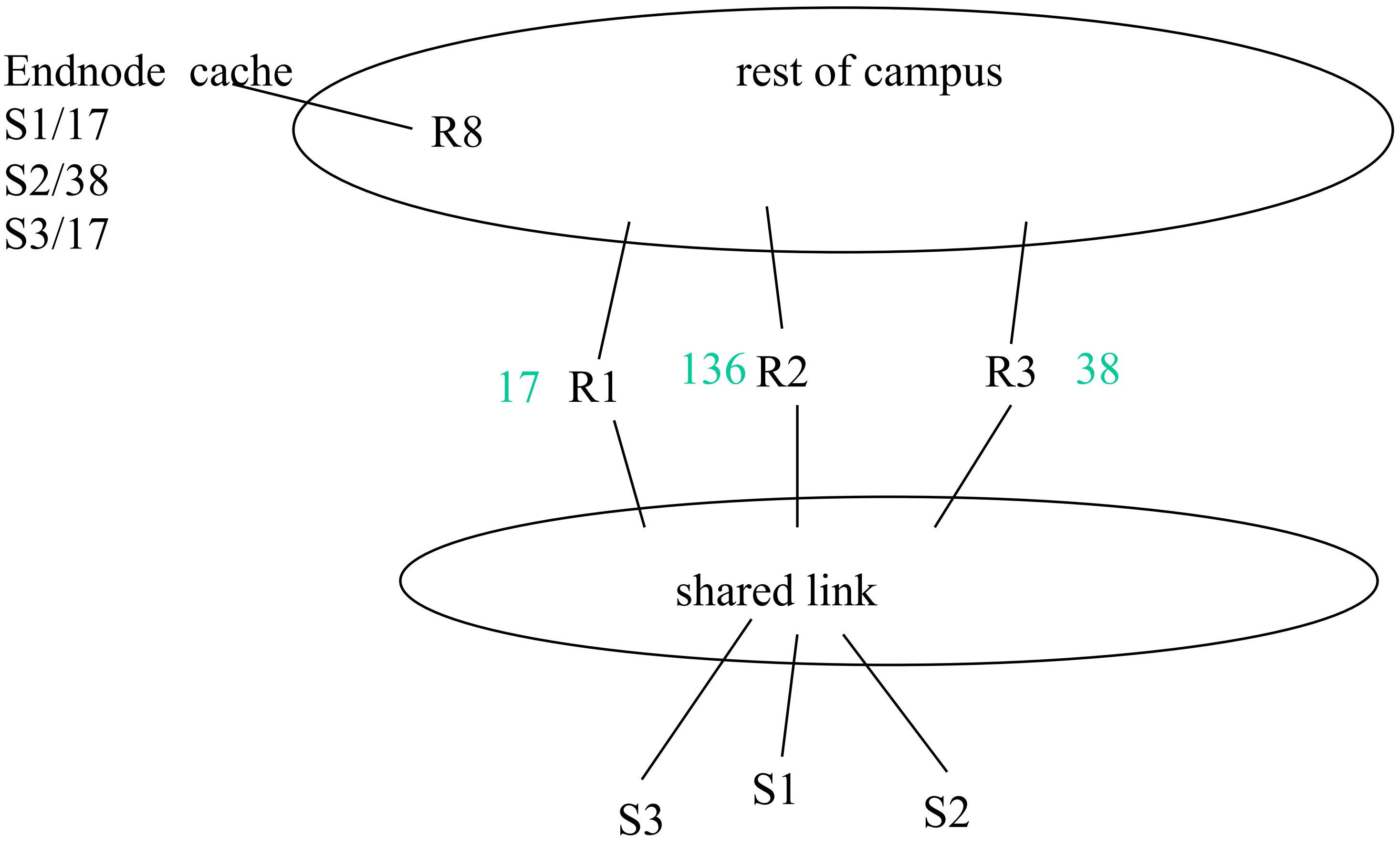
Hongjun Zhai

Fangwei Hu

Issue

- If the Appointed forwarder on a link changes from R1 to R2, remote RBridge endnode caches will be incorrect

Endnode cache wrong if AF changes



Endnode cache
S1/17
S2/38
S3/17

rest of campus

R8

17 R1

136 R2

R3 38

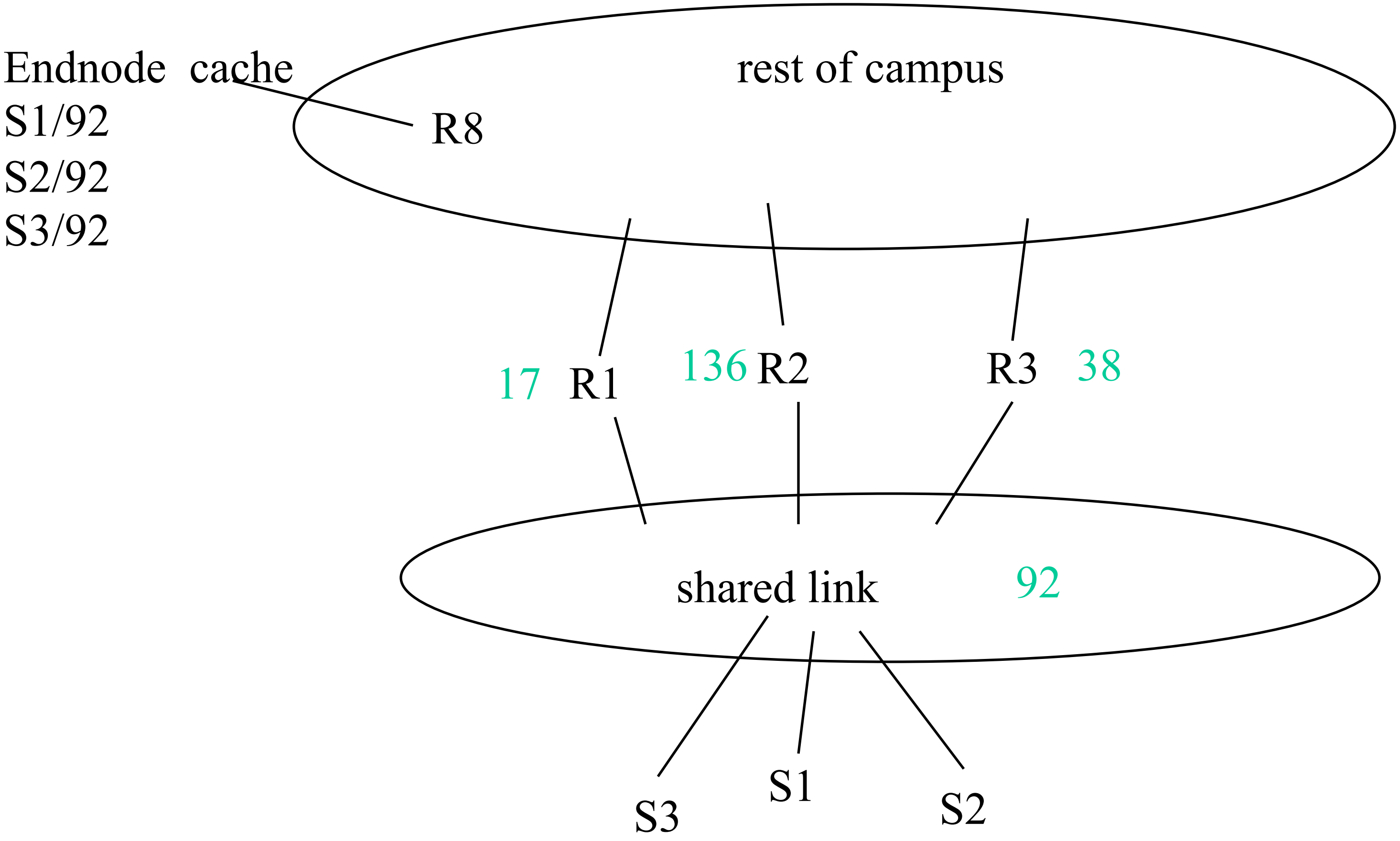
shared link

S3

S1

S2

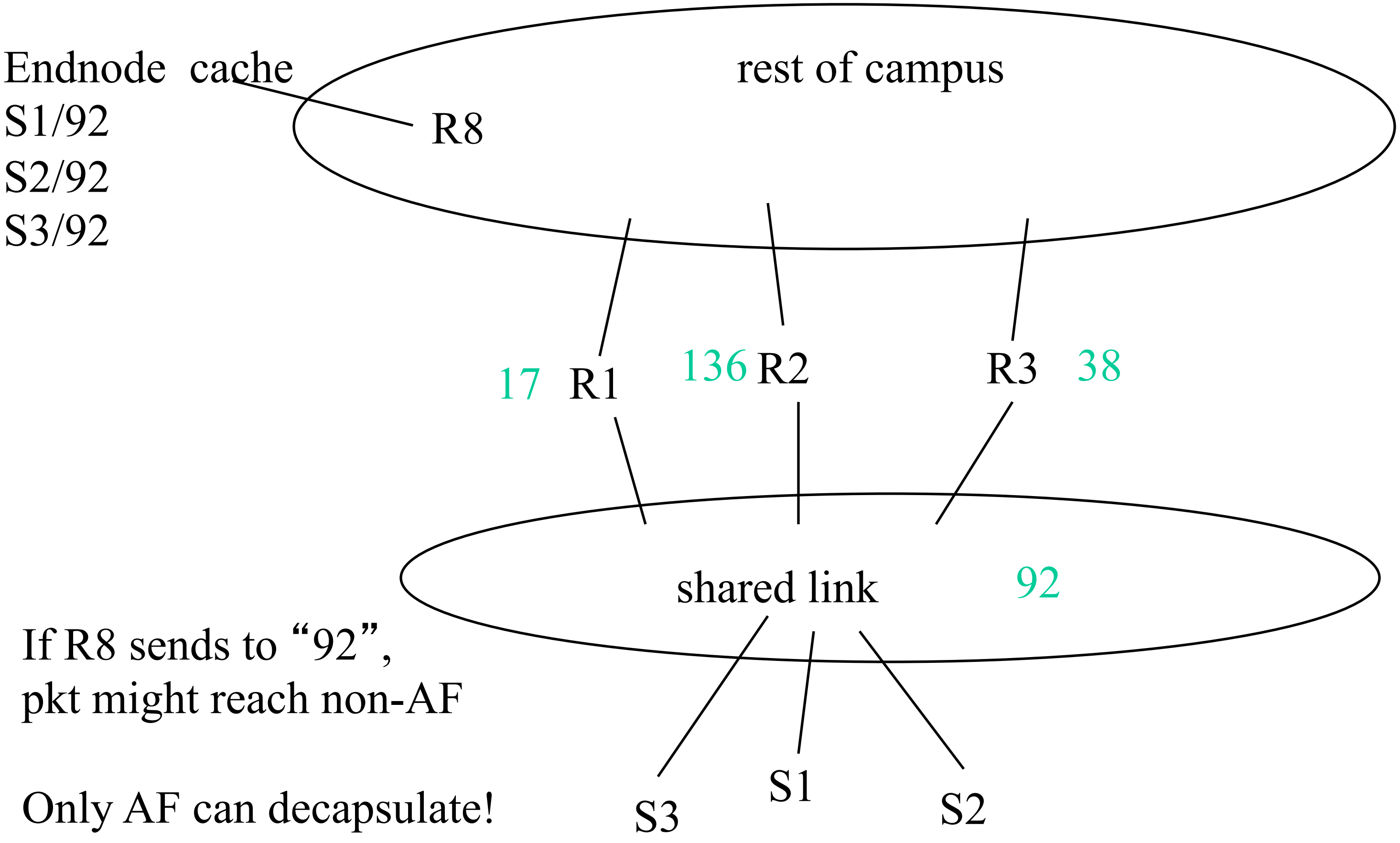
Solution: Use pseudonode nickname for ingress



Some subtleties

- Interaction with access links (links that are supposed to only be leaves…no inter-RB traffic… no inter-RB links advertised)
  - Can be done by not using a pseudonode (and having all RBs on the link claim they are using nickname “92”)
  - Or a pseudonode with nickname 92, and “overload” bit set, so paths through 92 not formed

Access link: need to forward rcv’d pkt addressed to “92” to AF



Endnode cache
S1/92
S2/92
S3/92

If R8 sends to “92”,
pkt might reach non-AF

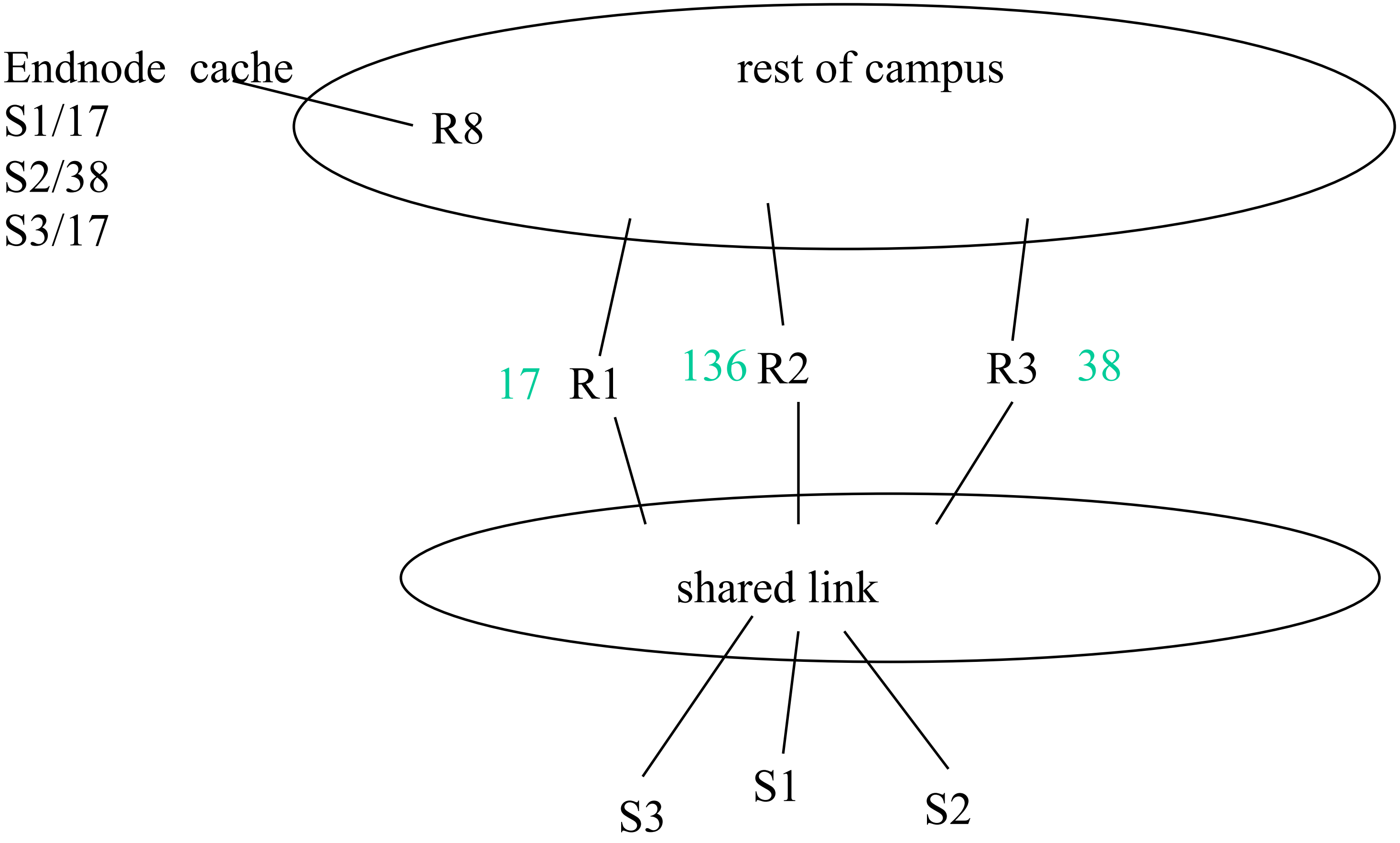
Only AF can decapsulate!

Special case: might have "link aggregation port group"

- There's a feature where a bridge B has two "up-links" to the RBs, only forwarding on one up-link (chosen at random), and never forwarding between the up-links
- But there wouldn't be any AF's in that case, and the RBs wouldn't see each other's Hellos

But in general case, need to forward on last hop to AF

Or not use pseudonode nickname on access links



Endnode cache
S1/17
S2/38
S3/17

rest of campus

R8

17 R1

136 R2

R3 38

shared link

S3

S1

S2

Another subtlety: Reusing nickname when DRB changes

Reuse nickname if DRB changes

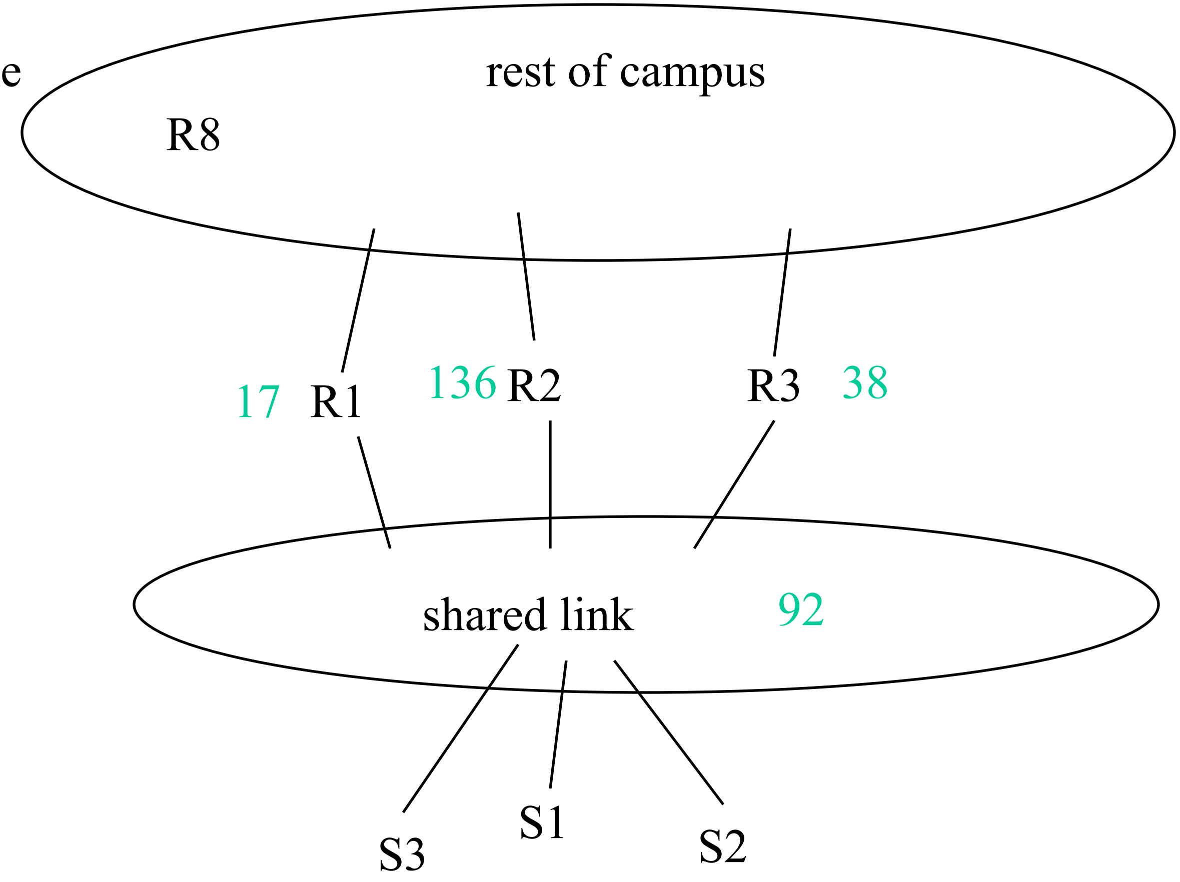
- DRB needs to tell other RBs what the pseudonode nickname is (in Hellos)
- If new DRB comes up, perhaps old RBs that remember the pseudonode nickname should tell the new DRB (in Hellos) what the pseudonode nickname was

But what if the link partitions into two links?

- Can the new DRB even tell the difference between a link partitioning and the DRB dying?

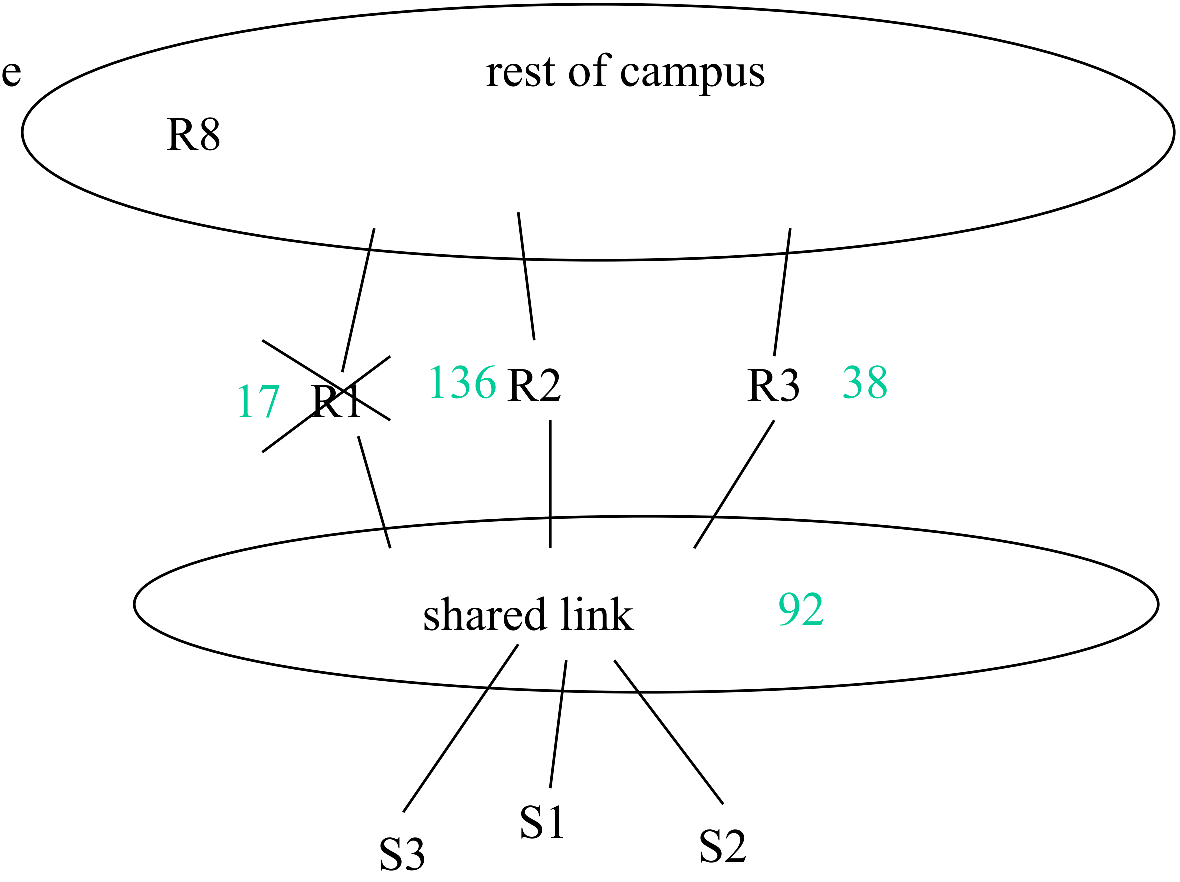
Issue: LAN partition vs DRB dies

Endnode cache
S1/92
S2/92
S3/92



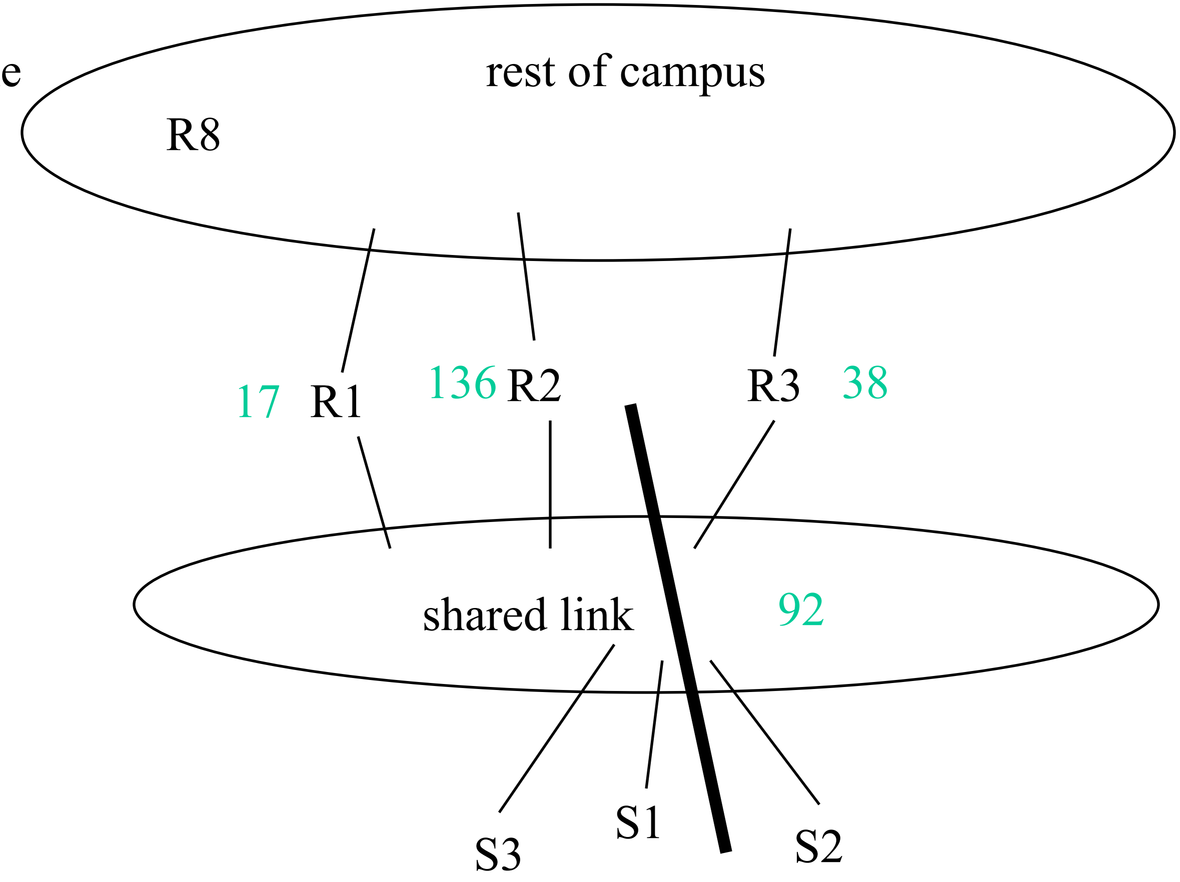
Issue: DRB dies: Reuse "92"

Endnode cache
S1/92
S2/92
S3/92



Issue: LAN partition: Can R3 reuse "92"? Both R1 and R3 will want 92

Endnode cache
S1/92
S2/92
S3/92



Recommendation

- Be optimistic and reuse the nickname
- If it’s really a partition, LSPs will resolve it
- Whoever has higher priority gets to keep it
- No reason why it’s better for old DRB to keep it rather than new one
  - in either case, some endnodes will have incorrect entries in distant RBridges

Another issue

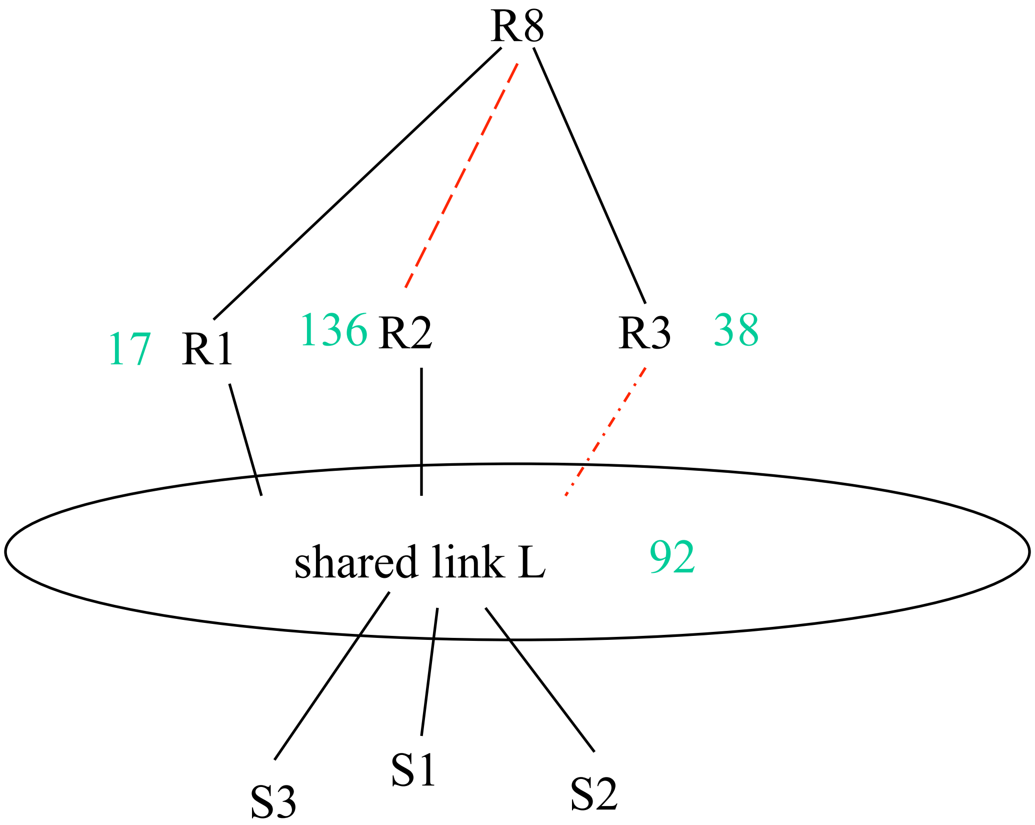
that their endnode cache is now wrong

– Either tell them to delete entries associated with nickname “92”, or tell them “entries that were 92 should now be 51”

Subtle issue: RPF check

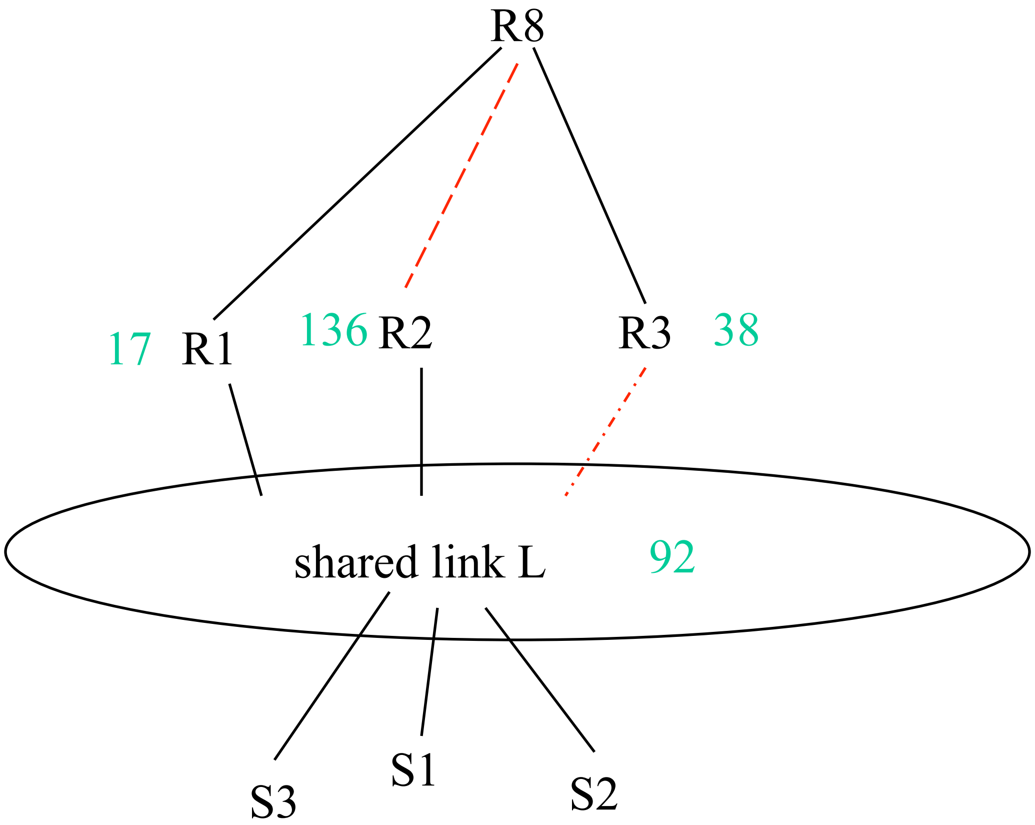
nickname, and the RPF check

Assume R3 is AF
Chooses tree T4:



rcv packet via R1

Assume R3 is AF
Chooses tree T4:



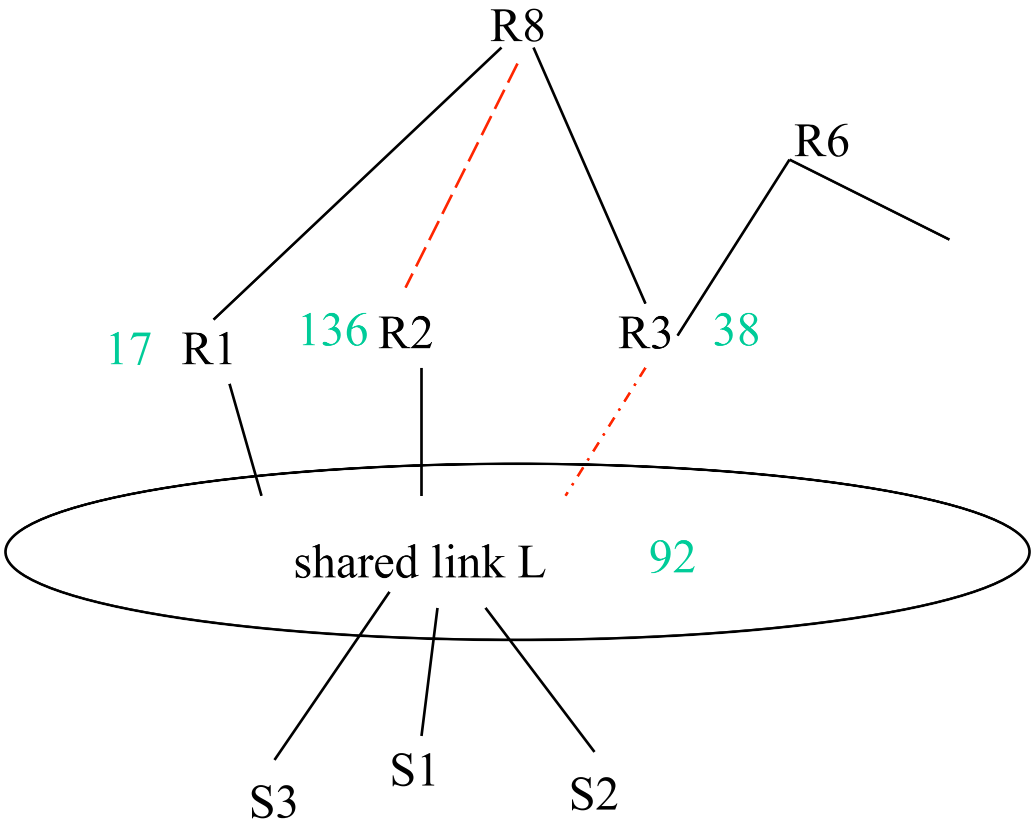
the frame

And send it back onto the link

- But that's not the same as "receiving the packet on the tree"
- So assume R3 is AF, and look at previous slide…
- R3 should encapsulate the frame, send it onto the link, but not forward it further until it receives the frame on a port in the tree

rcv packet via R1

Assume R3 is AF
Chooses tree T4:



In that case, RPF check just works

– And only forwards encapsulated pkt on tree if pkt received on port in the tree

comes from the pseudonode

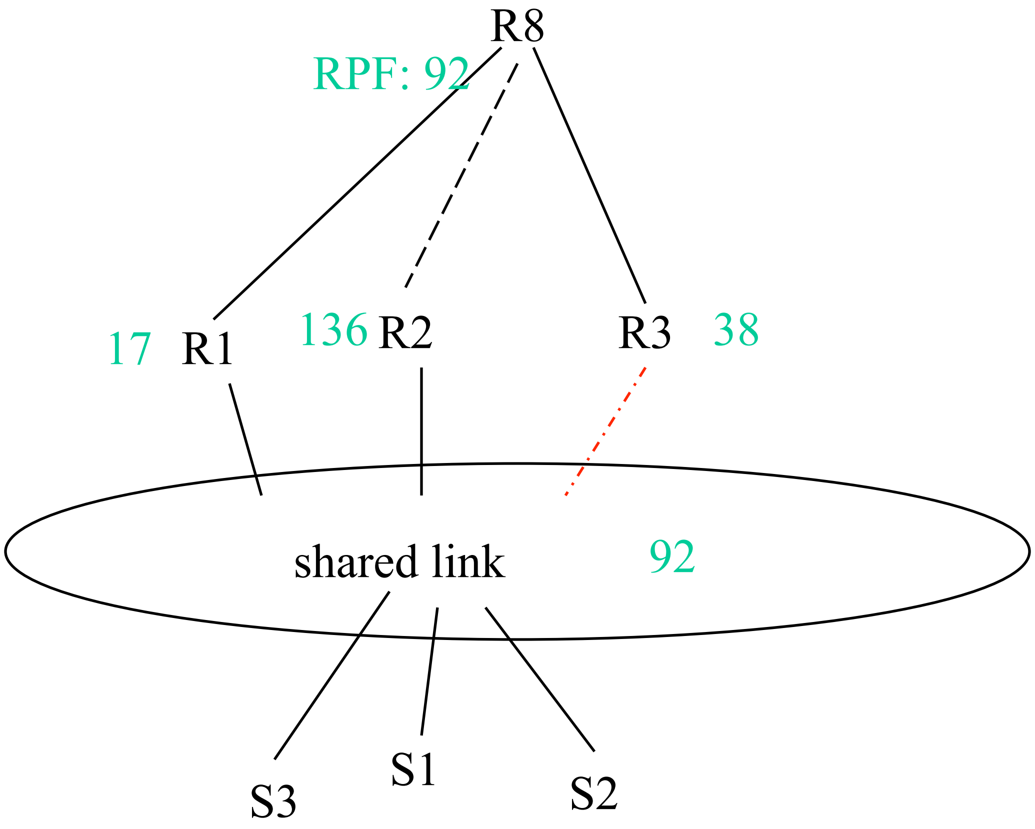
comes from the pseudonode

- And will be received via only one path

• And will be received via only one path

So the RPF check will always be OK

via R1
via R1



traffic on L

- Twice as much multicast traffic on L
  - native, and encapsulated
  - in both directions (first hop and last hop)

nickname

nickname’

- And can’t be avoided

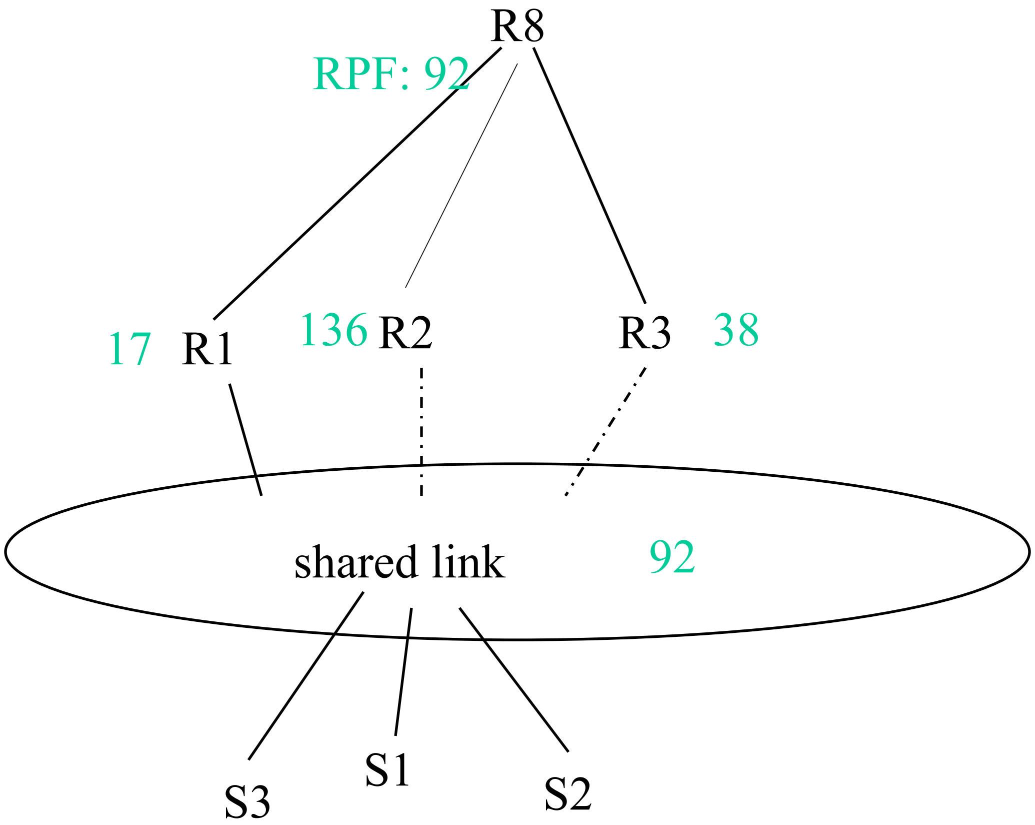
- And can’t be avoided

Access links

is not in the tree, R3 must encapsulate and transmit onto L

even though spec says not to ever send encapsulated traffic on an access link

even though spec says not to ever send encapsulated traffic on an access link



Potential solution

- R3 should not volunteer to be an AF on L if R3's port to L is not in ***any***

R3's port to L is not in

R3 should only ingress on behalf of L ***any*** tree

- Else (R3's port to L is in at least one tree)