Update
Draft-ietf-lsr-distoptflood-09
IETF 123

Tony P, HPE former known as Juniper ;-)

07/23

Juniper Business Use Only

What’s going on

- Lots of implementation/test/measurement data
- Discussions in context of **draft-ietf-lsr-isis-flood-reduction-arch-00**
- Treatment of broadcast and other minor clarifications
- Hash function improvements

Juniper Business Use Only

Technical Changes

- Broadcast medium computation has to “jump” over pseudonodes
  - Metric from each “real” node to another is 1 hop
  - No DIS behavior change, i.e. in case of reduction “misses” DIS still generates CSNPs and it responds to PSNPs on the ether
- Answering “misses” MUST NOT use reduction
  - CSNP and PSNP processing MUST NOT try to optimize the resulting flooding, otherwise bugs in reduction could permanently partition
  - This means normal CSNP may “interfere” with flood reduction but none of that affects correctness, at most has some small price in terms of efficiency

Juniper Business Use Only

Hash Changes

- Old hashing did not distribute particularly well and was buggy ;-) since
  - Fragment ID was shifted far to the left in the result
  - Last byte of system ID basically determined what the lowest bits of the hash were exclusively determined by this byte
  - And worst of all he hash “forgot” to shift after last rotation back and 7 * 4 = 28 < u64 size so last 4 bits were always 0
- Hence mod on even numbers did not balance at all for practical purposes
- New hash
  - Adds byte position to its value to help with very regular numberings
  - Last system ID byte is not rotated left anymore
  - Upper part of hash is folded onto lower part at the end to make sure all bytes contribute entropy for power of two balancing cases

Juniper Business Use Only

Minor

- Pseudo node ID is not taken into hash value
- Whoever read -09 realizes that the text mismatches the reference implementation appendix right now (oversight)

Juniper Business Use Only

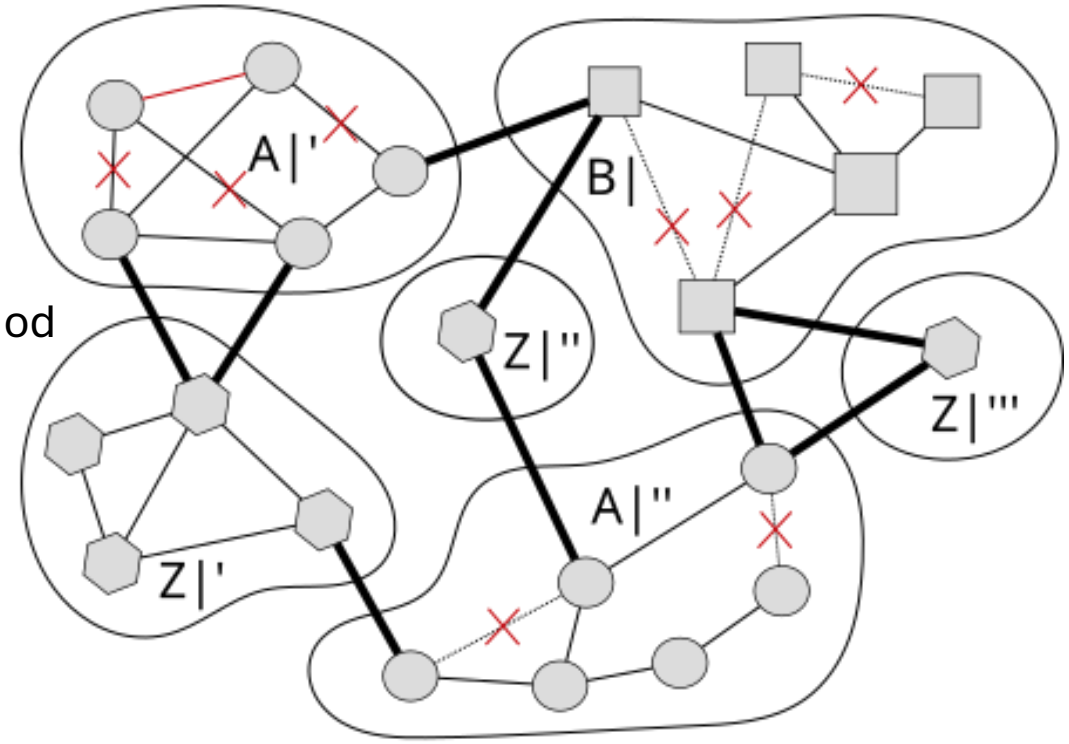
Signaling

- Draft presents now an algorithm only
  - Allocation of value requested in existing “IGP Algorithm Type for Computing Flooding Topology”
- Deployable with **draft-ietf-lsr-isis-flood-reduction-arch-00** allowing for migration and mixing of distributed algorithms

Juniper Business Use Only

Signaling in draft-lsr-interop-flood-reduction-architecture-00

- Each node must signal via router capability TLV (or anything else that is understood) **at most one** running flood reduction algorithm
  - No signaling of running algorithm signifies no reduction (an algo can play tricks with that)
- Set of connected nodes running same algorithm can optimize flooding (removed red crossed links) but **only within their component**
- Links connecting components **MUST** be flooded, i.e. interfaces with 2 different algos on both sides **MUST** flood
  - Will lead de facto to e'one flooding on broadcast interfaces unless e'one agrees on algo



Juniper Business Use Only