

Update
draft-ietf-lsr-distoptflood-07

IETF 121

R. White

Akamai

S. Hegde

T. Przygienda

Juniper Networks

L. Jalil

Verizon

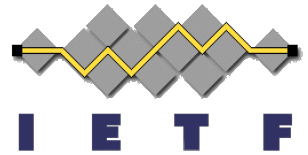
D. Voyer

Bell Canada



What's New

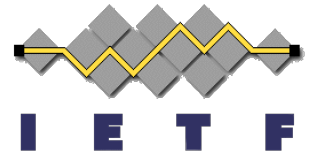
- New version represents the consensus WG Call and authors' consensus incl. input from several large operators
- Whole framework allowing multiple algorithms on the same network using then any type of "Signaling" separated into
 - <https://www.ietf.org/id/draft-lsr-prz-interop-flood-reduction-architecture-00.html>
- LSP-ID has $\gg 4$ on fragment now to keep 16 subsequent LSP-IDs on the same CDS
 - We will move to $\& 0xF7$ based on implementation experience (8 subsequent only and easier operationally to debug)
- Hash reference implementation in Rust added
- Acee caught excellent corner case inefficiency for stale LSP fragments
 - It worked with the PSNP "belts-and-suspenders"



Operational Section

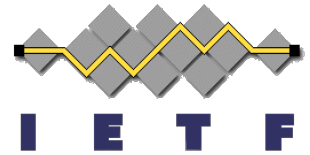
- Based on several large operators' input and simulation of behavior on real large topologies and synthetic SP-like setups
 - Deployment on few key nodes can deliver significant gains
 - Leaderless, configuration-free, “minimal blast radius under any conditions” operation uniformly preferred
 - Optional indication that the algorithm is running on a node strongly preferred with “version” running
 - No significant practical interest in “multiple algorithms” at same time on the network or migrations

Pretty Classical Distribution on Large ISIS Backbones



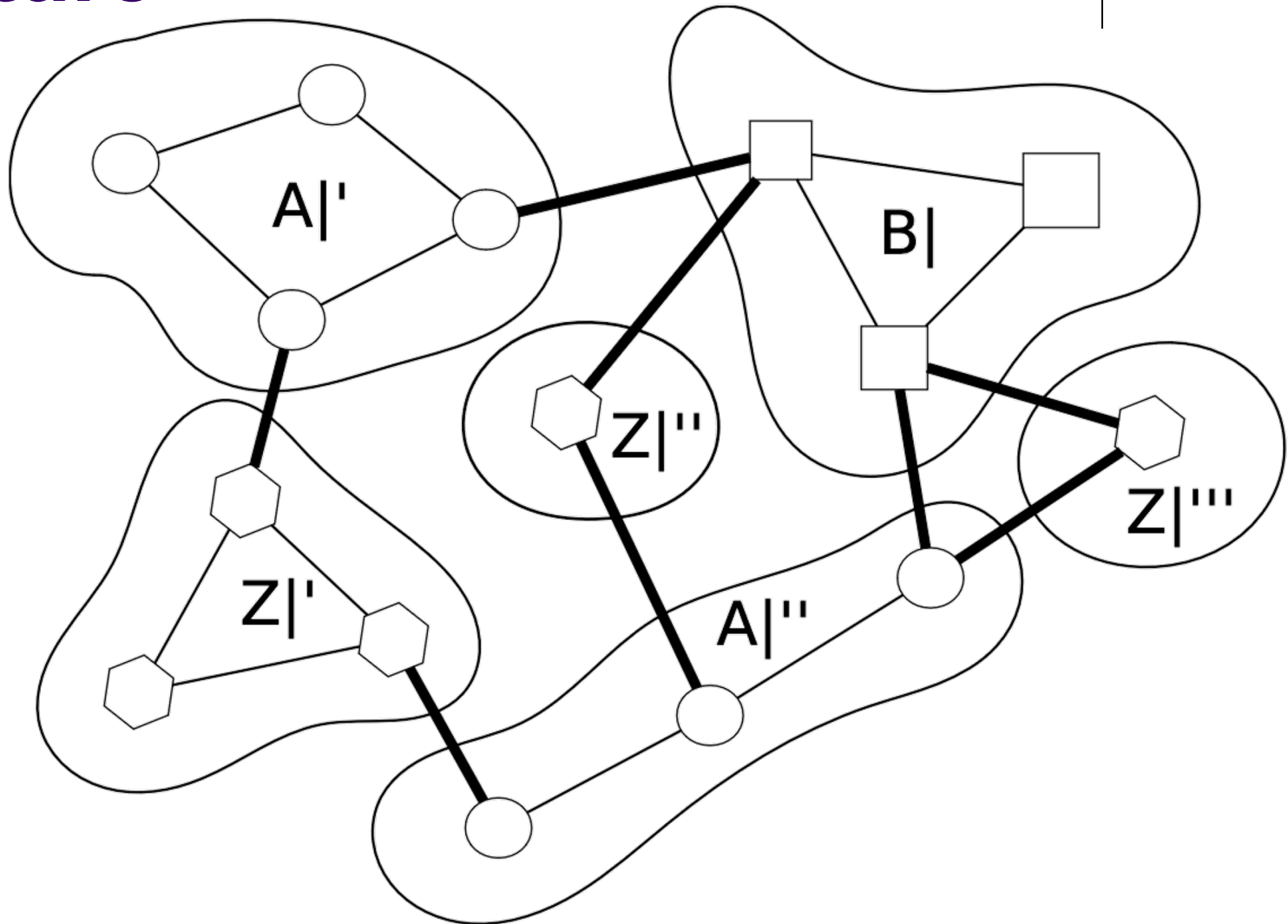
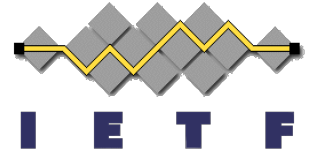
- 1/3 of nodes about 90% reduction in flooding volume
- 1/3 of nodes about 80% reduction
- Small fat tails of 10% of nodes with ~10% reduction
- Some very well built topologies can massively reduce with very, very few nodes

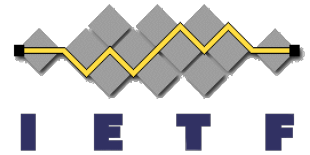
Framework for (Mixed) Leaderless Operation



- In case of just disttopo on a network nothing strictly necessary
 - Hence the “SHOULD”
- In case of possible multiple algorithms some “running algorithm” signaling necessary
 - Would allow arbitrary mix of leader-based and leaderless algorithms at same time
 - Only practical consideration is seamless migration on algorithm improvement or defects
- Leader based can be also in the mix

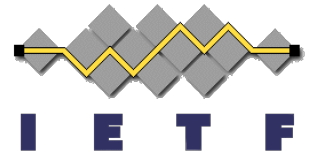
A Picture, a pony for a picture





Now

- First Consensus Call on “Should WG work on leaderless signaling for distributed flood reduction to allow for no configuration, minimal blast radius and no flag day changes?”
- If so, consensus Call on “Is Generic Leaderless & Leader Based Signaling Needed for Mixed Mode”
 - Only needed if multiple algorithms with mix of leader and leaderless at same time on network necessary
- If so, where to put it?



Rest is Almost Trivial

- Simply a TLV indicating algorithm number that is configured **or** running on a node
 - RFC9776 pretty much has that “configured” already for leader-based
 - Leaderless would need a “running algorithm” TLV on top
 - Necessary for any approach avoiding a “flag day” change
- Nodes can obviously change algorithm at will
 - Biggest impact possible is recomputation in 2 components
- Any algorithm **MUST** flood on all adjacencies to another component, i.e. if the other side cannot be assumed to run same algorithm
 - Hence the “running or configured algo” signalling