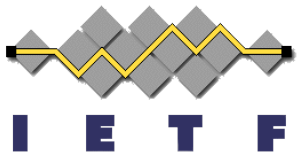
Flood Reduction Architecture
draft-prz-lsr-interop-flood-reduction-architecture
IETF 122

T. Przygienda
S. Hegde
Juniper Networks



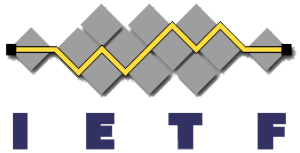
Agenda

- History
- Why new framework needed?
- Distributed Architecture
  - Terminologies
  - Procedures
  - Protocol extensions
- Advantages
- Next steps

History

- Distributed flooding architecture first published as part of draft-ietf-lsr-distoptflood
- Presented in IETF 120 and 121
- WG suggestion to move the architecture to separate document
- WG consensus to work on a distributed architecture for deploying FR algorithms

Recap: Why new framework is needed?



- Ability to introduce Flood-Reduction algorithm in parts of the network
- Ability to upgrade to a new version of the algorithm (or bug fix upgrades) node by node
- Minimum blast radius
- Simplified operations, No config needed
- No centralized control for changing the algorithm throughout the network
- Lower risk of operational mistakes causing network outages

Terminologies

- Pruner – A node running a Flood Reduction (FR) algorithm
- Zero Pruner – Special case of a pruner which floods all LSPs on all interfaces (Basically today's flooding)
- Connected Component – Set of nodes running same Pruner that have atleast one path connecting all nodes in the set
  - Connected component of a Pruner P is denoted by P|
  - Disjoint set of connected component denoted by P|' , P|''
- Connected Dominating Set – A component with paths connecting each pair of node. Denoted by P|*. Must ensure information can reach each node in the component.

Flooding Pruner procedures

- Every Pruner MUST advertise the Pruning algorithm it is currently running
- Every Pruner is allowed to run only one Pruning algorithm at a time
- A node not advertising a Pruning Algorithm is assumed to be running Zero Pruner

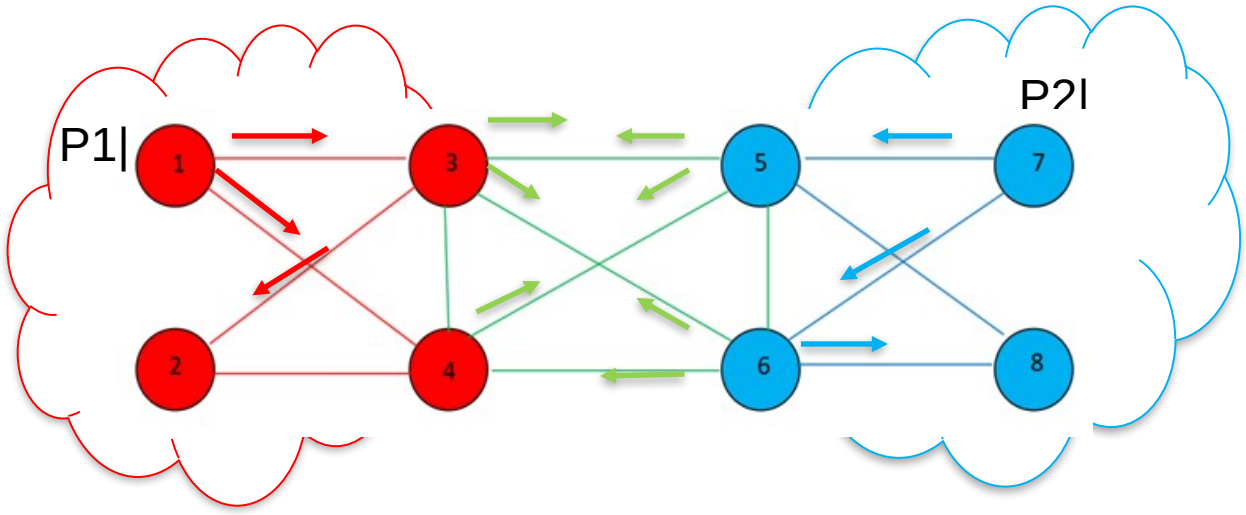
Flooding Pruner procedures

- A Pruner can Prune links for flooding only within its own connected component
- A Pruner MUST flood on all the links outside its connected component

  In other words

  If a Pruner finds a neighbor is running a different Pruner, it MUST flood to that node
- Further optimization possible if neighbor is Zero Pruner. Its left to the discretion of individual Pruner algorithm

Example



- P1| and P2| are connected components for P1 and P2 Pruning algorithm
- Nodes 3,4 ,5,6 are border nodes having adjacency with another Pruner. Hence they MUST flood on all adjacencies connected to another Pruner
- Nodes 1,2 run Pruner algorithm P1, Nodes 7, 8 run Pruner algorithm P2

Protocol Extensions

Sub TLV of TLV-242

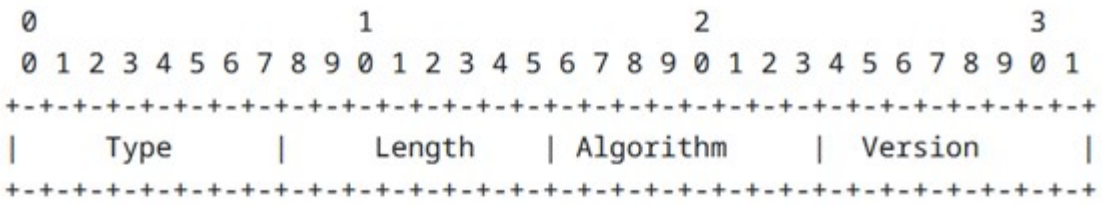


Figure 2

* Type: TBD1

* Length: 2

* Algorithm: 8 bits of a numeric identifier in the range 0 -- 255 that identifies the algorithm used to calculate CDS (flooding topology) of the component.

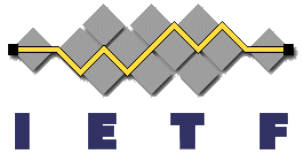
* Version: 8 bits of the algorithm version. Whether versions of the same algorithm can be considered as the same component by the involved algorithm or not is governed by each algorithm independently. All other algorithms MUST consider the versions of an algorithm as different algorithms for the purpose of Section 2.1.4.

Advantages

- Nodes within a connected component are free to use any Pruner
- Pruner itself can be Centralized or distributed
- Node has to advertise the Pruner algorithm its running. No other config required
- When a Node decides to change Pruner algorithm only adjoining nodes are affected.
- Node by Node deployment or migration, minimum blast radius.

Next Steps

- Request for review and comments
- Request WG adoption



Thank you