RIFT Multicast

Jeffrey Zhang
Pascal Thubert

IETF108, Virtually Madrid
Previously on RIFT Multicast …

- Borrow PIM-Bidir concept
  - Establish (*, *), (*, G-prefix), (*, G) Bi-directional trees
  - Hash to a north neighbor and send N-TIE as equivalent of PIM joins
  - The joins stop at some sub-TOF nodes, forming sub-trees

- Sub-Trees need to be joined by a virtual RPL
  - A spanning tree among sub-TOF and TOF nodes

- -01 revision with a bit more details
  - Per-neighbor flooding scope
  - THRIFT schema
  - Some spanning Tree details
    - However a new method will be discussed here and documented in -02
Per-neighbor Flooded Multicast TIE

```c
struct TIEHeader {
  ...
  13: optional common.SystemIDType flooding_scope_neighbor;
}

struct IPMulticastTIEElement {
  /** Multicast TIEs are for (*, group-prefix) joins. */
  1: required common.IPPrefixType group_prefix;
  ...
}
```

- The originator sends to specified neighbor only
- The receiver accepts if it is the target, and won’t reflood
RPL Problem: disjoint sub-trees rooted at the Sub ToF

Problem: Build a meta tree (a tree of sub-trees).
Goal: connect the sub-trees

Proposal: Build a loopless a meta-tree (a tree of trees) by joins those trees via the superspine
**Approach:**

**build a spanning tree of ToF and SubToF**

The spanning tree must span all subToF and may span some or optionally all ToF nodes.
Feasibility

ToF (L=X, X>0)

subToF(L=X-1)

Build a Spanning Tree here

ToF + SubToF may be complete bipartite mesh => Easy
ToF + SubToF must be connected if not => Need a protocol
ToF + SubToF cannot be partitioned (disconnected)
=> Partitions kill auto-disaggregation, this is why we have a ring in multiplane
Proposal: Simple DV

Composite metric (Tree Root’s system ID, hop count)

- Many possible parents: need to select one

SubToF nodes advertise the metric to ToFs and reciprocally

=> in Node TIEs?

Result: a trees
Proposal: Simple DV
Optimizations

Use a subToF node as Root, e.g., N1 below

Elect groups of ToF nodes, e.g., (S1, S2), (S3, S4, S5)

e.g., of size $\sim \sqrt{\text{CARD(subToF)}}$,

composite metric (root sysID, ToF group, hops)

SubToF Nodes pick random parent in the same group
Variations / optimizations

Prebuild (\(^*,*\)) for mice flows

- Multiple \(^*,*\) trees selected by hash of mice flow?
- Different root / Group of ToF nodes per tree
- Caution: group -> tree must be same for all nodes else loops

Build \(^*,G\)-prefix) on demand for girafe/elephant flows to avoid flooding

- Forwarding based on longest match
- Root / ToF group selection for each tree?
- Composite Metric (hash (root, g), root, hash(TofGrp, g), TofGrp, hops)
Result a spanning structure with subset of ToF

ToF (L=3)

subToF 2 (L=2)

Mid 1 (L=1)

Leaves (L=0)