# IP Multicast Fast Reroute

**follow-up on draft-dimitri-rtgwg-mfrr-framework-00**

**RTG Working Group**

**IETF 75 meeting**

**Stockholm (Sweden)**

**July 2009**

# Status

- Draft initially presented in Dublin (IETF 72)
- Work on multicast routing recovery specifics
- Work on transient/temporary loops during reconvergence period

# Goal

- Investigate solution space for improving multicast distribution trees (MDT) recovery time

- Topological failures (e.g. links and nodes)

-> Analyze new proposals as well as existing solutions aimed at reducing impact of the scaling factors of PIM convergence

# Convergence & recovery time analysis

- Recovery time ($T_R$): upon MDT failure

  Time after which all receivers have restored connectivity to MDT (so, receive again multicast traffic streams)

- Convergence time ($T_C$): upon MDT failure

  Time after which all MFIB updates have been performed by all the routers

- $T_R$ and $T_C$ dependence
  - PIM variant
  - Network topology size and shape
  - Number of mcast groups affected

# Mcast FRR solution space

- Track 1: re-use/extend existing unicast FRR to protect/recover MDT
  - FRR scheme extended to incorporate a certain level of "multicast-awareness"
    - Decrease time for PIM message exchange

      -> Tuning unicast routing re-convergence to decrease RIB-related operations time

    - Decrease time required to propagate fail-over information by retro-fit into unicast FRR scheme

      -> Tuning failure notification time

# Mcast FRR solution space

- Track 2: PIM built-in extensions to improve convergence time
  - Existing solutions: Anycast RP, Dual multicast topologies
    - Tackle specific failure cases and rely on abstracting reachability and/or topology
    - Drawbacks of tweaking Hello timers
  - Example: upon mcast state change, trigger J/P message *conditionally* to prevent transients loops
    - Transient loops may be induced from the use of multiple MFIBs entries for same mcast group (resulting from PIM Join exchanges prior and after failure)

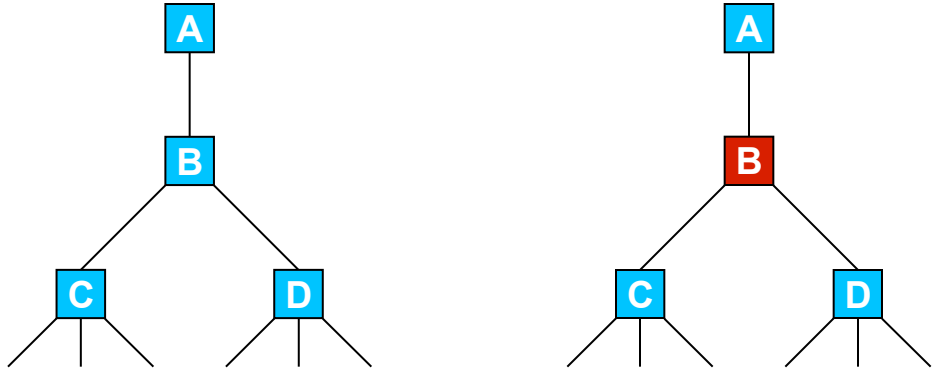
# Track 2: Problem Space

- **Tweaking Hello Timers**
  - May lead to faster failure detection but also increases processing overhead and results in PIM neighbor being declared down due to missed Hellos (if Hello packets are not prioritized)
  - Other drawback: dependence created between *Hello* exchanges for maintaining interface liveness and learn about neighboring PIM routers/ capability negotiation/etc.
- **Alternative**
  - Extend PIM mechanisms (potentially by using another fast failure detection) to improve the convergence time

    Multicast routing-specific components that can benefit from such improvement: time needed for sending a Join/Prune message as a result of multicast state change
  - **Must be accompanied by set of conditions to prevent transients loops** that may be induced from the use of multiple MFIBs entries for mcast group (resulting from PIM-JOIN exchanges prior and after failure)

# Case1: Multicast Routing Failure

- **Condition**: PIM routing is down and *Join*, *Prune* or *Hello* messages cannot be sent or treated anymore: MRIB entries have consistence problem and disrupt node's RPF-neighbor




- **Consequences**
  - *Join/Prune*, *Hello* messages cannot be exchanged anymore: PIM neighbor adjacencies between nodes B-C and A-B will be lost at Holdtime elapsing (3.5 x Hello Period)
  - *Join/Prune* messages periodically exchanged every 60s (by default) between Join/Prune Messages: Holdtime specified in a Join/Prune message should be set to 210s (3.5 x J/P period)

  => MFIB cannot be updated if members arrive or leave

  - However, multicast traffic can still be forwarded according to MFIB as entries are valid for 180 seconds (delay of storage before clearing entries in MFIB)

# Approach

- MFIB entries are valid for only 180 seconds
  - *Join* refresh messages are not sent anymore
  - Need to maintain these entries after this delay expiration to ensure multicast forwarding
- **Idea**: freeze MFIB entries on nodes all along the path (where the failed router is present)

  Failed node does not forward *Join* anymore all along the path (as stored in MRIB)

  -> MFIB entries of nodes (that do not received *Join* messages anymore) need to be freezed and self-refreshed

  To freeze MFIB entries for (pre-determined) period
  - Prior to failure, negotiation between PIM neighbors of "recovery" period
  - Upon failure, timer activation at nodes contiguous to failure (PIM routing and MRIB recovery to be triggered)
  - Contiguous nodes behave "as is" wrt own downstream neighbors

# Algorithms for Recovery

**Algorithm 1: RPF-Check at B is OK**

- Multicast forwarding can be assured in this case by self-refreshing entries during the period of recovery (at downstream neighbors)
- Definition of delay for this period of recovery
- Neighbors of B have to be aware of the period during which that in order to continue sending Join messages to maintain the entries in the MFIB
- If this period is too long: compute new backup tree (excluding) node B
  - Find a backup structure in the multicast routing topology where the node that has failed has been completely removed
  - The alternate paths should be computed without the failed node and all the nodes that compute alternate paths have to be aware of this failure

# Algorithms for Recovery

**Algorithm 2: RPF-Check at B is not OK**

1. Downstream neighbors of B need to find an alternate path

   Node B cannot initiate itself the demand as it cannot forward *Join*/*Prune*
2. Each downstream neighbor computes shortest path towards A or towards nearest node that is crossed by MDT (in the topology where B has been removed)
3. As B has failed and as downstream nodes are not aware of event: some information to be inserted in *Join* message (or before sending Join, specific *Notification* message) so that nodes along the path can avoid node B when computing the new path

# Case2: Multicast Routing Failure

- **Conditions**: PIM routing is ok, RPF Check is down
  - MRIB consistency problem due to some topological changes (due to metric update or a link up or down)
  - Entries do not match good RPF-neighbor
  - However, node can still send Join/Prune or Hello messages

- As some topological changes occurred, there should be a switchover of the current MDT to a new MDT (accounting of new topology)
- Some specific rules for switchover to be enforced as some transient loops may occur

- Cause: No synchronization in propagation of the Join/ Prune messages
  - A part of the old MDT may exist together with a part of the new MDT

-> Recovery algorithm should avoid these loops

# Conditions for transient loops

Loops occur:

- when one node has to send both a *Join* and a *Prune* for the same MDT in different directions
- if topological change implies that downstream node of a failed node will become an upstream node

  ≡ if a path from the failed node toward the source/RP on new topology traverses a prior downstream node
- when distance from downstream node (i.e. node B) to its parent node traversed by the MDT in direction of source/RP (i.e. node A) in the old topology is higher than the same distance in the new topology:

  $d(A,B)_{new} < d(A,B)_{old}$

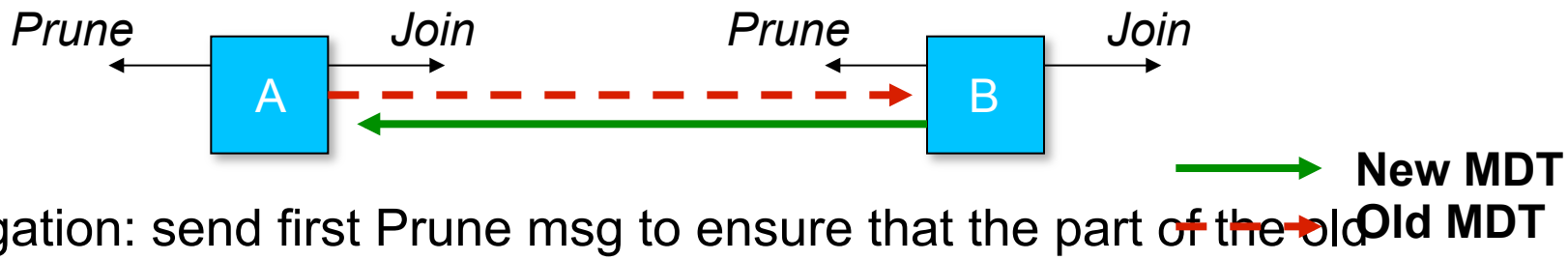
# Check Procedures

- List the nodes that will be traversed by the Join message in the new topology from the failed node (node B)
- Check cycles of size n, n-1, ..., 1 (leads to cycle detection scheme)

# Example

- Cycle of size 2 if
  - node A sends Join to node B
  - node B treats the Join message and stored corr. entry in its MRIB/MFIB
  - node B has not sent Prune message yet or if node A has not yet treated Prune message sent by node B

  Then, the two MDT (old and new one) coexist for a period




- Mitigation: send first Prune msg to ensure that the part of the old MDT will be removed first (BBM)