Identifying Defunct DAGs in RPL draft-goyal-roll-defunct-dags-00

Mukul Goyal
University of Wisconsin Milwaukee

Main Idea

- Need a mechanism that allows a node to identify defunct DAGs and delete the state it maintains for such DAGs.
- Specify a “no inconsistency” flag inside a DIS.
- With “no inconsistency “ flag set, a multicast DIS would not lead to the reset of trickle timers in the receiving RPL routers.
- Same mechanism specified in draft-dejean-roll-selective-dis-00 (“Selective DIS for RPL”)
- Plan to merge the two documents

When does a node remove a neighbor from the parent set for a DAG?

- When the DAG's life time is over.
- When the neighbor is no longer reachable
- When the neighbor advertises in its DIO an infinite rank in the DAG
- When keeping the neighbor as a parent would required the node to increase its rank beyond L +DAGMaxRankIncrease
  - L is the minimum rank the node had in the DAG
- When the neighbor advertises in its DIO membership in a different DAG within the same RPL Instance, where a different DAG is recognized by a different DODAGID or a different DODAGVersionNumber.

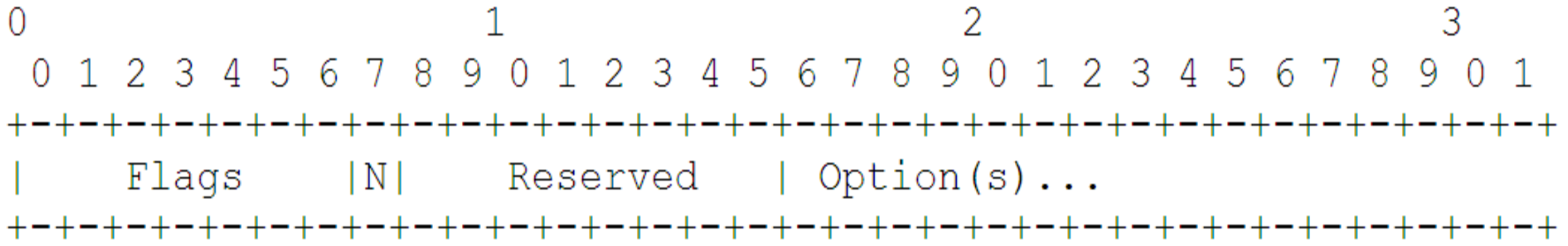
A node may fail to remove a neighbor from parent set

- The node fails to receive the neighbor's DIOs advertising an increased rank or the neighbor's membership in a different DAG
- The node may not check the neighbor's unreachability for a long time.
  - RPL is proactive, hence the lack of data traffic using a DAG can not be considered a reliable indication of the DAG's defunction.
  - Trickle timer based control of DIO transmissions means the possibility of an indefinite delay in the receipt of a new DIO from a functional DAG parent.
- In such cases, a node would continue to consider itself attached to a DAG even if all its parents in the DAG are unreachable or have moved to different DAGs.

Need a mechanism to identify defunct DAGs

- Using unicast DIS to query individual parents may be expensive.
- Hence, the mechanism specified in this document is based on the use of a multicast DIS message to solicit DIOs about a DAG suspected of defunction.
- As per RPL specs, receipt of a multicast DIS is treated as an inconsistency and causes the reset of Trickle timer.

No Inconsistency Flag in DIS



- On receiving a unicast/multicast DIS with N flag set,
  - an RPL node MUST NOT reset the trickle timers for the DAGs that match the DIS predicates.
  - For each DAG matching the predicates of a multicast DIS, an RPL node MUST schedule a DIO transmission after a time duration between Imin/2 and Imin, where Imin is the minimum Trickle interval size
  - For each DAG matching the predicates of a unicast DIS received with N flag set, an RPL node MUST immediately generate a DIO.

draft-dejean-roll-selective-dis

- “No Inconsistency” flag is same in operation as “Leaf Node” flag proposed in draft-dejean-roll-selective-dis
- Nodes responding to a DIS with “Leaf Node” flag set MUST send a unicast DIO
  - Unicast DIO MUST include a Configuration option.
- draft-dejean-roll-selective-dis defines a “Response Spreading” option that specifies the time interval over which the DIO responses must be spread.

Identifying A Defunct DAG

- When an RPL node has not received a DIO from any of its parents in a DAG for more than MaxSilence*Imax seconds
  - The node MUST generate a multicast DIS message that carries a Solicited Information option and has N flag set.
  - The Solicited Information option MUST have the I and D flags set and the RPLInstanceID/DODAGID fields MUST be set to values identifying the DAG.
  - The V flag inside the Solicited Information option SHOULD NOT be set so as to allow neighbors to send DIOs advertising the latest version of the DAG.
  - After sending the DIS, the node MUST wait for Imin duration to receive the DIOs generates by its neighbors.

Identifying A Defunct DAG

- At the conclusion of the wait period:
  - If the node has received one or more DIOs advertising newer version(s) of the DAG, it MUST join the latest version of the DAG, select a new parent set among the neighbors advertising the latest DAG version and mark the DAG status as functional.
  - Otherwise, if the node has not received a DIO advertising the current version of the DAG from a neighbor in the parent set, it MUST remove that neighbor from the parent set.
  - As a result, if the node has no parent left in the DAG, it MUST
    - mark the DAG as defunct
    - schedule the deletion of the state it has maintained for the DAG after DAGHoldTime duration.
- An RPL node SHOULD check the functional status of a DAG it belongs to at least once every CheckDAGStatusTime interval