Efficient route invalidation for RPL

History:

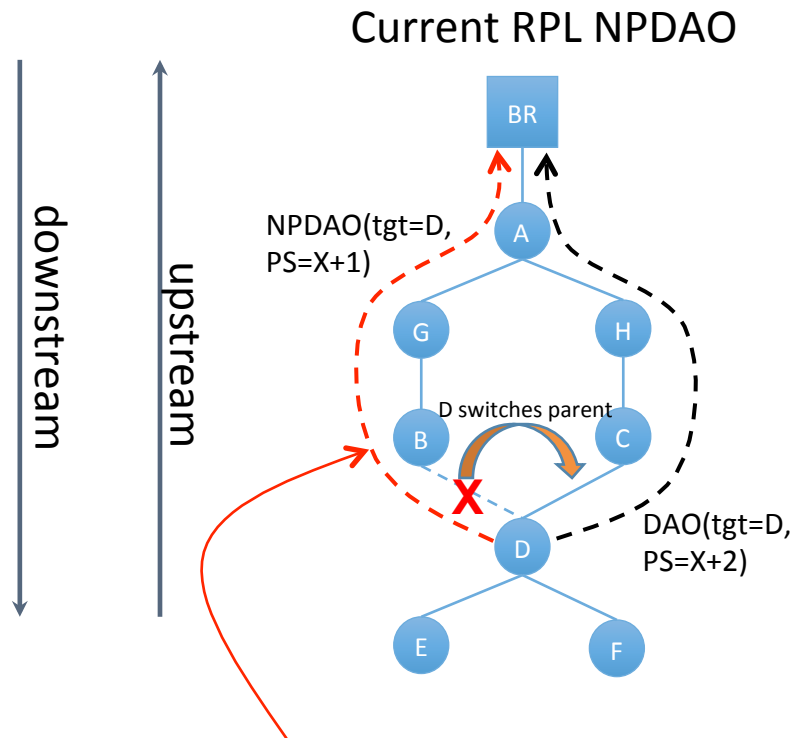IETF95 - Presented the problem statement

IETF96 - Presented existing solutions based on comments rcvd and why those fall short

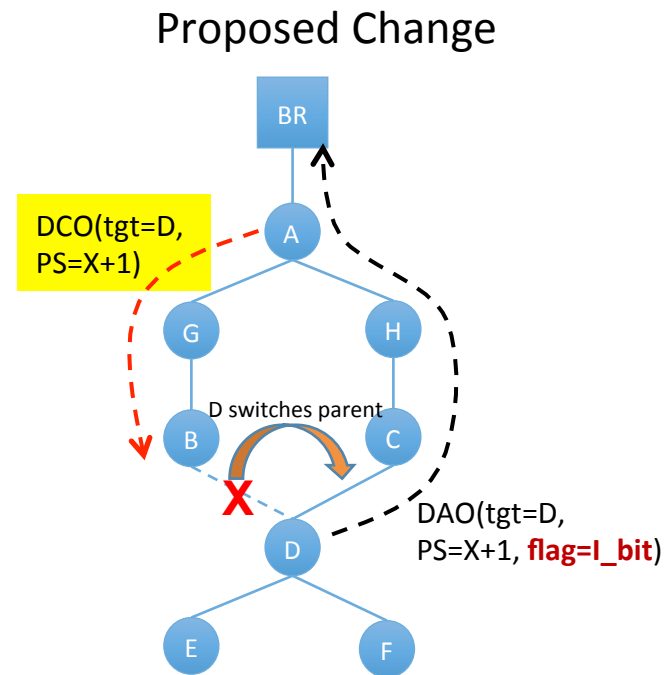IETF98 – Presented new solution for improving route invalidation

IETF99 – adopted as WG document , thank you for the review

IETF100 – Changes to message codes

Recap: the problem and the solution

Current RPL NPDAO

downstream

upstream

BR

NPDAO(tgt=D, PS=X+1)

A

G   H

D switches parent

B   C

X

D

DAO(tgt=D, PS=X+2)

E   F

NP-DAO via broken links will cause many problems such as reachability and efficiency

Proposed Change

BR

DCO(tgt=D, PS=X+1)

A

G   H

D switches parent

B   C

X

D

DAO(tgt=D, PS=X+1, **flag=I_bit**)

E   F

- Send the DAO via the new parent;
- Common parent to trigger the DCO to invalidate the previous path

PS = PathSequence
Tgt = Target

Primary update: new RPL message code

Per RFC 6550, DAO (including NPDAO) traverses upstream only.

Previous version,

NPDAO had a flag which said, invalidate and "send downstream".

New version,

A new message DCO (Dest Cleanup Object) is added for proactive route cleanup.

Primary update: new RPL message code

Reasons for using new code:

- Existing implementations do not need to change, thus incremental code update possible

- RPL message codes are available in plentiful as compared to DAO flags.

Side-Effects – Along comes the ACK

DAO optionally has a DAO-ACK.

DCO would also need one, thus DCO-ACK.

```
+------+--------------------------------------------+---------------+
| Code |                 Description                 |   Reference   |
+------+--------------------------------------------+---------------+
| 0x85 |          Destination Cleanup Object        | This document |
| 0x86 | Destination Cleanup Object Acknowledgement | This document |
+------+--------------------------------------------+---------------+
```

Next Step

- Change the implementation (trivial)

- Welcome any feedback from the Open-source community (while we believe the technique description is stable enough)

Thank you