

Neighbor Management Policy for 6LoWPAN

(focusing on the protocol change-free guidance)

<https://tools.ietf.org/html/draft-jadhav-lwig-nbr-mgmt-policy-00>

Rahul, Rabi@ Huawei

Simon @ INRIA

Joakim @ Yanzi Networks

IETF99

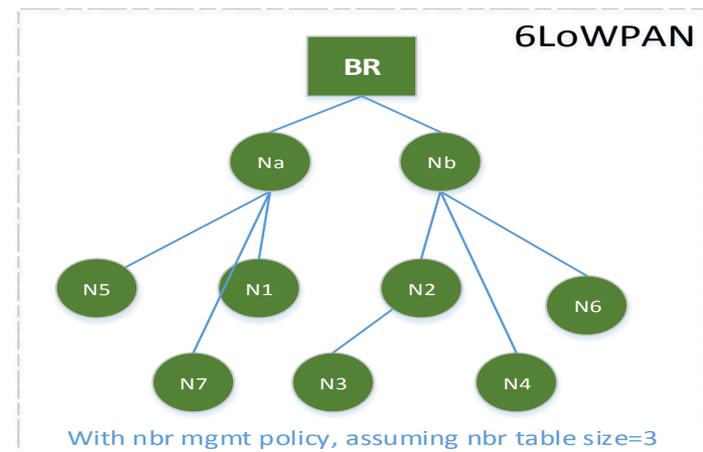
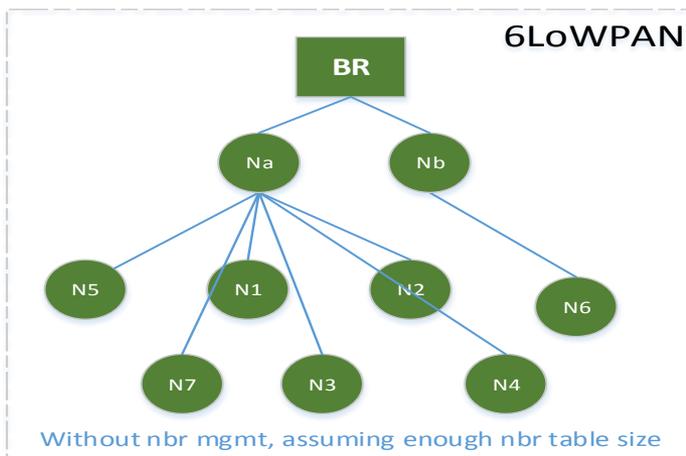
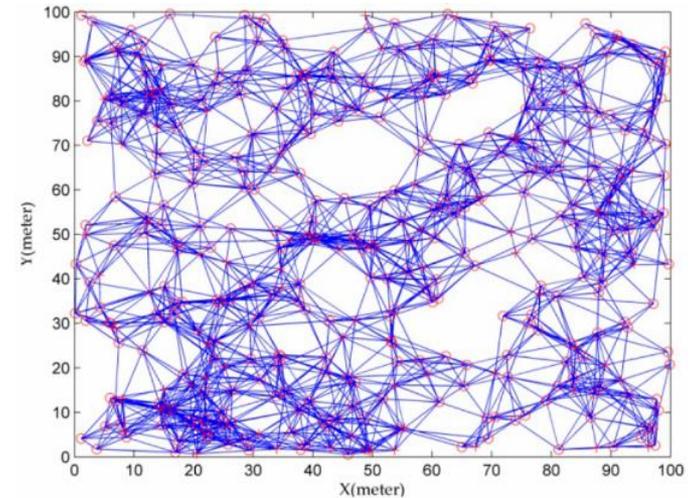
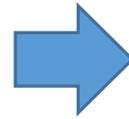
History:

IETF97: Presented the problem statement, without the draft

IETF98: First presentation based on the draft, sample reservation policy described

Why Neighbor Management?

- Implementation dilemma:
 - Node density is higher than nbr cache size
- Quite an implementation issue!!



Unbalanced & unstable topology
without proper nbr mgmt policy

Stable and scalable topology
with a proper policy

The problem of trivial policies

- **Trivial policies**

- LRU: Replace the least recent used entry
- FCFS: First come first serve

- **An example given by Joakim:**

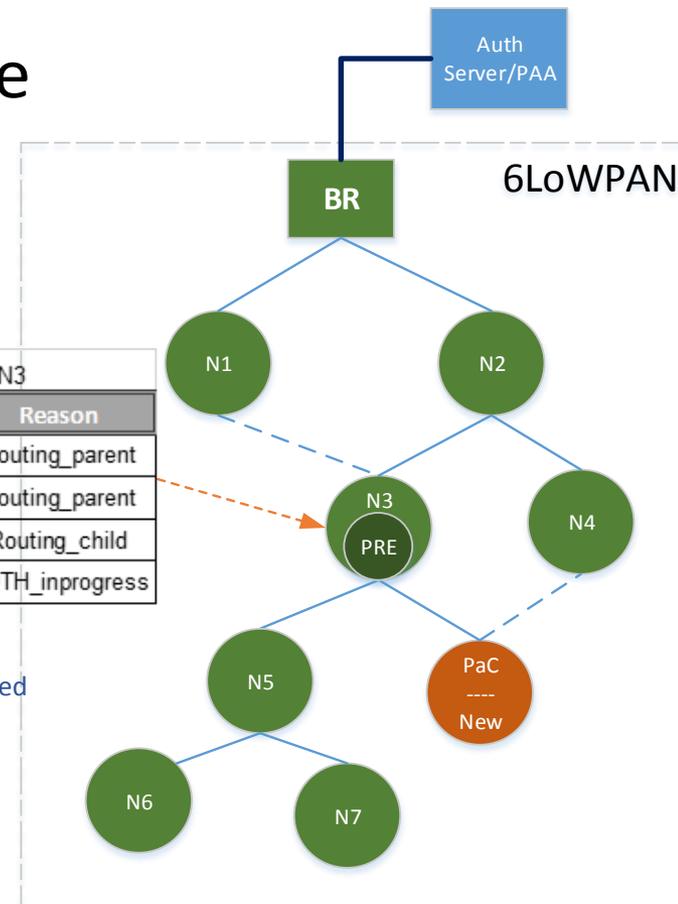
- 50-60 smart plugs on a table
- trying to switch all of them at the same time
- However in this case only 10 entries in the neighbor table
- Result: the network went into a continuous topology churn

Considerations....

- Security-enabled 6LoWPAN/RPL network
 - Key management protocols before RPL network formation
- Cases where neighbor table update happens
 - Relay based signaling during authentication
 - RPL's parent selection using DIO messaging
 - RPL's routing child node

Neighbor cache @ N3		
MAC addr	LL IP addr	Reason
N1_mac	N1_llip	Routing_parent
N2_mac	N2_llip	Routing_parent
N5_mac	N5_llip	Routing_child
PaC_mac	Pac_llip	AUTH_inprogress

PRE = PANA Relay Element aka Join Proxy
 PaC = PANA Client aka Pledge
 PAA = Auth Server aka Join Registrar/Coordinator



Neighbor Management Operations

• Insertion

- Problem with simple logic (If table space is available: insert)
 - RPL's DIO storm in dense network overwhelms neighbor cache
 - DAO storm to the same parent

• Eviction

- Problems currently...
 - An routing child eviction may have ripple effect on all grand-children
- Good news: Evicting non-preferred parent NCE is possible without much implications!!

• Reinforcement

- Reinforcement allows the link quality estimation to be updated, eventually helping in eviction/retention decision

Clearing unused NCEs

- For storing MOP, route invalidation is important since routing entries are mapped to NCEs
- For Non-storing MOP,
 - since there is no route invalidation procedure, the child node needs to deregister using NS(lifetime=0) ...
- PRE neighbors
 - After authentication is successful, the PRE auth entries can be removed
 - However there is no way of explicit identification of auth finish
 - Usually reachability timeout will remove such entries. For neighbors added for authentication, the reachability timer can be reduced to a lower value.

Proposed reservation based policy

- Basic principles

- “Reservation”

- Parent NCEs can be inserted at will and can occupy reserved entries

- Because they could be evicted if necessary, unlike routing direct childs and relay element entries

- Insertion reason (RPL_parent, RPL_child, Other) is attached with every NCE

- Makes decision making easier when it comes to eviction



Issues with reactive policy

- Whatever I presented as of now is reactive ...
- Limitations of reactive policy
 - Scenario: A parent whose nbr cache is full sends a DIO (a 6LR(router node) cannot avoid sending DIOs) ...
 - A child node may still select this parent node since DIO does not signal NCE metric
 - Thus there would be an additional signaling to reject this parent node
 - Worse, in the future, the child node may again select the same parent based on new DIO from the parent node.

Proactive way:

- A “Parental Advisory” to signal availability
- “Currently there is no standard way of signaling such neighbor cache space availability information. RPL's DIO messages carry metric information and can be augmented with neighbor cache space as an additional metric”
- Out of scope for LWIG, could bring to Roll/6lo for discussion

Discussions

- WG Adoption in LWIG
 - As a general protocol agnostic guidance for nbr mgmt.
- Next steps:
 - Contiki implementation already began prior to this draft.
Contiki will keep aligning with this draft...

Thank You