batman-adv - Bug #418

BLA: claiming race condition with multicast from mesh

09/14/2020 10:54 AM - Linus Lüssing

Status:	New	**Start date:**	09/14/2020
Priority:	Normal	**Due date:**	
Assignee:	batman-adv developers	**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			

Description

Scenario:

- Two BLA backbone gateways sharing the same LAN, receiving a multicast packet from the mesh.

Issue:

- Both BLA backbone gateways will race for a claim. And can end up with both gateways thinking the other one claimed the client. Resulting in packetloss for traffic from the mesh into the BLA backbone.

This actually seems to be acknowledged in the code:

```
1857         if (!claim) {
1858                 /* possible optimization: race for a claim */
1859                 /* No claim exists yet, claim it for us!
1860                  */
```

https://git.open-mesh.org/batman-adv.git/blob/f2a2e0310dc1c570bdd1439553e897649b000292:/net/batman-adv/bridge_loop_avoidance.c#l1858

Typically this seems to resolve when the claim times out a bit earlier on one of the BLA backbone gateways. However it unfortunately seems quite persistent when the two nodes were set up on the same host at the same time via a script, for instance. And is probably also persistent when physically similar devices / nodes are booted at the same time, for instance after a power outage.

This should be reproducable with the attached script. It creates a fully meshed topology, with nodes 1 and 2 bridged via LAN, like the following:

```
 --[LAN/br0]--
 |            |
(1)          (2)
 |            |
 ---[mesh]----
   /      \
 (3) ... (8)
```

To reproduce, run:

```
$ ./test-mcast-bla.sh setup
$ ping6 ff12::123%br8
```

Then compare *"batctl meshif bat1 cl"* and *"batctl meshif bat2 cl"*. You should see that both nodes assume that the other one claimed the MAC from node 8. Furthermore, a *"tcpdump -i br0 'icmp6 and dst ff12::123'"* should stay silent, showing that the multicasted ICMPv6 Echo Requests are wrongly dropped into the BLA backbone by both node 1 and 2.

You can teardown the test mesh via "$ ./test-mcast-bla.sh teardown" (or restart it via "reload").

Tested in a x86 Debian VM running Linux 5.8.7.

Further notes:

- Probably more likely to trigger via multicast-to-unicasts than via classic flooding. As the latter adds some jitter on forwarding, making a race less likely.
- More likely to trigger if the two BLA backbone gateways are direct neighbors so that they receive the multicast packets at the same time.

Files

test-mcast-bla.sh	1.87 KB	09/14/2020	Linus Lüssing