

batman-adv - Bug #421

Misconfig or bug: received packet on bat0 with own address as source address

10/28/2020 04:06 PM - Adrian Schmutzler

Status:	New	Start date:	10/28/2020
Priority:	Normal	Due date:	
Assignee:	batman-adv developers	% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:			

Description

General setup:

Freifunk Franken firmware fork, where Batman is used on a distributed Layer-2 network connected to gateways via fastd tunnels.

Each node offers client and mesh via ethernet (e.g. via vlans, eth0.1 for client and eth0.3 for mesh) and via WiFi (e.g. w2ap for 2.4 GHz AP und w2mesh for 2.4 GHz mesh (802.11s), w5ap for 5 GHz AP etc.)

We make sure that all of the "mesh" interfaces (e.g. eth0.3, w2mesh, w5mesh, i.e. what you see with batctl if) have distinct MAC addresses.

Same for all "client" interfaces, i.e. members of the same bridge br-mesh alongside bat0 (e.g. eth0.1, w2ap, w5ap)

MAC addresses are allowed to overlap *between* those groups, though, e.g. eth0.3 (= "mesh") could have the same address as w2ap (= "client/ap").

Test setup:

Isolated device configured as above and connected to Freifunk network via layer-3 (WAN), i.e. no batman neighbors ("batctl o" and "batctl n" are empty).

Device is acting as batman server (gw_mode server), but similar behavior can be produced with batman client nodes. BLA is active (=default).

TP-Link TL-WDR4900 v1

OpenWrt 19.07 (Tested with .3 on the device, the problem itself is present across all subversions including .4 observed on different devices)

Batman-adv openwrt-2019.2-7 (openwrt-routing 19.07 branch; I also tested with the recent 2019.2-10 including a recent BLA patch on a different device)

Problem:

dmesg (and logread) show the following every 10 seconds:

```
[ 179.939430] br-mesh: received packet on bat0 with own address as source address (addr:fa:1a:67:xx:xx:fb, vlan:0)
```

Discussion:

I can remove the warning via one of two measures:

1. Remove the MAC address collision of eth0.3 ("mesh") and w5ap ("client") by giving an arbitrary unique MAC address to eth0.3
2. Disable BLA via uci set network.bat0.bridge_loop_avoidance='0'

Actual question:

From my conceptual understanding, I do not see a reason why an overlap between "client" and "mesh" MAC addresses should be forbidden.

Actually, it's quite strange that particularly the overlap of eth0.3 ("mesh") and w5ap ("client") causes the warning, while the still existing overlap between eth0.1 ("client") and w5mesh ("mesh") is *not* harmful.

Therefore, my actual question is: is this intended behavior, i.e. is this MAC overlap actually forbidden? Or this is a bug (possibly caused/created by BLA)?

Keep in mind that this happens on an isolated device.

As a consequence, since disabling BLA removes the warning, would disabling BLA "solve" the problem then for the moment, since the packets sent by BLA are the root cause, or would disabling BLA just remove a detection tool for the misconfiguration that still exists?

Further info:

MAC addresses:

```
bat0: random
br-mesh: f8:...:fb
eth0:    f8:...:fb (same as eth0.1)
eth0.2:  f8:...:fc
eth0.3:  fa:...:fb
w2ap:    fa:...:fa
w2mesh:  f8:...:fa
w5ap:    fa:...:fb
w5mesh:  f8:...:fb
```

(There are additional AP networks configured, but those have separate addresses and also are completely separate from batman)

OpenWrt network config:

```
config interface 'loopback'
    option ifname 'lo'
    option proto 'static'
    option ipaddr '127.0.0.1'
    option netmask '255.0.0.0'

config globals 'globals'
    option ula_prefix 'fdff:0::/64'

config interface 'wan'
    option ifname 'eth0.2'
    option proto 'dhcp'

config device 'wan_eth0_2_dev'
    option name 'eth0.2'
    option macaddr 'f8:1a:67:xx:xx:fc'

config switch
    option name 'switch0'
    option reset '1'
    option enable_vlan '1'

config switch_vlan 'vlan1'
    option device 'switch0'
    option vlan '1'
    option ports '0t 1t 4 5'

config switch_vlan 'vlan2'
    option device 'switch0'
    option vlan '2'
    option ports '0t 1t'

config interface 'eth0_3'
    option proto 'batadv_hardif'
    option master 'bat0'
    option ifname 'eth0.3'
```

```
config interface 'mesh'
    option type 'bridge'
    option auto '1'
    option ifname 'bat0 eth0.1'
    option macaddr 'f8:1a:67:xx:xx:fb'
    list ip6addr 'fdff:0::0:f81a:67xx:xxfb/64'
...
    option proto 'static'
    list ipaddr '10.xx.xx.1/24'
    option ip4table 'fff'
    option ip6table 'fff'

config switch_vlan 'vlan3'
    option device 'switch0'
    option vlan '3'
    option ports '0t 1t 2 3'

config device 'ethmesh_dev'
    option name 'eth0.3'
    option macaddr 'fa:1a:67:xx:xx:fb'

config interface 'w5mesh'
    option mtu '1560'
    option proto 'batadv_hardif'
    option master 'bat0'

config interface 'configap5'
    option proto 'static'
    option ip6addr 'fe80::1/64'

config interface 'w2mesh'
    option mtu '1560'
    option proto 'batadv_hardif'
    option master 'bat0'

config interface 'configap2'
    option proto 'static'
    option ip6addr 'fe80::1/64'

config interface 'bat0'
    option proto 'batadv'
    option gw_mode 'server'
    option gw_sel_class '200000'
    option network_coding '0'
    option network_coding '0'
    option aggregated_ogms '1'
    option ap_isolation '0'
    option bonding '0'
    option fragmentation '1'
    option orig_interval '1000'
    option distributed_arp_table '1'
    option hop_penalty '30'

# followed by various rules and wireguard interfaces
```

History

#1 - 10/28/2020 04:37 PM - Sven Eckelmann

- Description updated

Looks like this is most likely caused by the regular bridge-loop-avoidance announcements. See also:

- https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/tree/net/batman-adv/bridge_loop_avoidance.c#n1480
- https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/tree/net/bridge/br_fdb.c#n589
- here is the reason why a fdb entry is marked as LOCAL:

https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/tree/net/bridge/br_fdb.c#n915

The bridge loop announcements are send with the source address of the underlying **primary** netdevice. And this address is conflicting with the eth0.3 one in the same bridge - thus the bridge complains about it. If you don't want to see this message (and want to use bla) then you have to make sure that the primary slave/underlying device of your bat0 is not sharing the mac address with any other device in this bridge.

~~There were some discussion in the past to drop the primary_if and use the mac address of the softif. I am currently unable to assess the (negative) effects of such a change but at least Marek was against it.~~

Simon can answer questions about bridge loop avoidance if you have them.

#2 - 10/28/2020 06:54 PM - Adrian Schmutzler

The bridge loop announcements are send with the source address of the underlying **primary** netdevice.

So, what exactly is the underlying primary netdevice in this case?

#3 - 10/28/2020 10:13 PM - Adrian Schmutzler

As an intermediary solution, I have now created a bridge br-ethmesh that contains the eth0.3. Instead of eth0.3, the bridge br-ethmesh is then added to batman as mesh device.

eth0.3 keeps the MAC address, but br-ethmesh gets assigned a randomly generated address.

This way it's still possible to use the eth0.3 MAC address for link-local, but the address used for BLA is random ...

#4 - 10/29/2020 08:24 AM - Sven Eckelmann

You can find the primary/main if in the header of various batctl debug tables:

```
$ sudo batctl o
[B.A.T.M.A.N. adv 2020.4-3-g1810de05, MainIF/MAC: enp68s0/2c:f0:5d:04:70:3a (bat0/7a:8b:21:b7:13:b8 BATMAN_IV)
]
  Originator      last-seen (#/255) Nexthop          [outgoingIF]
, - (sven@ripper:pts/13:~/projekte/siwu/2014-12-16_OBX-Interface/web)
$ sudo batctl n
[B.A.T.M.A.N. adv 2020.4-3-g1810de05, MainIF/MAC: enp68s0/2c:f0:5d:04:70:3a (bat0/7a:8b:21:b7:13:b8 BATMAN_IV)
]
IF                Neighbor          last-seen
```