

batman-adv - Bug #409

DAT: received packet on bat0.20/eth0 with own address as source address

04/22/2020 12:48 PM - Matteo Fortini

Status:	In Progress	Start date:	04/22/2020
Priority:	Normal	Due date:	
Assignee:	Matteo Fortini	% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:			

Description

I have a batman-adv network with four (openwrt 19.07.2) nodes on an 802.11s mesh, two of which are connected by ethernet, too:

- batman is active on the mesh interface for all nodes, has two VLANs defined (bat0.20 and bat0.107).
 - bat0.20 is the "private" VLAN and is bridged to the ethernet network and a wifi SSID*
- bat0.107 is bridged to a secondary wifi SSID
- All the bridges have STP off, while batman has bl active
 - batman-adv is correctly finding the backbone and the two wired nodes see each other as neighbors in the bbt.
- I changed the MAC address of all wifi interfaces and of the wired ones so that I have no duplicate MAC addresses on the network. [See mac address...](#)[See mac address...](#)

```
bat0      Link encap:Ethernet HWaddr BA:03:29:67:EF:22
bat0.107  Link encap:Ethernet HWaddr BA:03:29:67:EF:22
bat0.20   Link encap:Ethernet HWaddr BA:03:29:67:EF:22
br-IOT    Link encap:Ethernet HWaddr 92:83:C4:00:C3:A4
br-pvtlan Link encap:Ethernet HWaddr 96:83:C4:00:C3:98
eth0      Link encap:Ethernet HWaddr 94:83:C4:00:C3:97
eth0.1    Link encap:Ethernet HWaddr 96:83:C4:00:C3:9A
eth0.2    Link encap:Ethernet HWaddr 96:83:C4:00:C3:AA
ifb4pppoe-wan Link encap:Ethernet HWaddr 3E:AB:72:AC:68:E9
mesh0     Link encap:Ethernet HWaddr 92:83:C4:00:C3:A2
wlan0-1   Link encap:Ethernet HWaddr 92:83:C4:00:C3:A0
wlan0-2   Link encap:Ethernet HWaddr 92:83:C4:00:C3:A4
```

In the logs I have every 30s or so the "received packet on bat0.20 with own address..." message.

I can reproduce the problem with DAT enabled, if I disable DAT just on the offending nodes, the problem goes away

Moreover, sometimes the message is repeated much more frequently, as you can see here (the MAC address is unique in all the network):

[See log...](#)[See log...](#)

```
Wed Apr 22 10:41:46 2020 [1587552106.093] kern.warn kernel: [59860.042409] br-pvtlan: received packet on bat0.20 with own address as source address (addr:96:83:c4:00:c3:98, vlan:0)
Wed Apr 22 10:41:46 2020 [1587552106.094] kern.warn kernel: [59860.053363] br-pvtlan: received packet on bat0.20 with own address as source address (addr:96:83:c4:00:c3:98, vlan:0)
Wed Apr 22 10:42:48 2020 [1587552168.686] kern.warn kernel: [59922.632640] br-pvtlan: received packet on bat0.20 with own address as source address (addr:96:83:c4:00:c3:98, vlan:0)
Wed Apr 22 10:42:48 2020 [1587552168.686] kern.warn kernel: [59922.645463] br-pvtlan: received packet on bat0.20 with own address as source address (addr:96:83:c4:00:c3:98, vlan:0)
Wed Apr 22 10:42:48 2020 [1587552168.699] kern.warn kernel: [59922.659110] br-pvtlan: received packet on bat0.20 with own address as source address (addr:96:83:c4:00:c3:98, vlan:0)
Wed Apr 22 10:42:48 2020 [1587552168.714] kern.warn kernel: [59922.671797] br-pvtlan: received packet on bat0.20 with own address as source address (addr:96:83:c4:00:c3:98, vlan:0)
Wed Apr 22 10:42:48 2020 [1587552168.727] kern.warn kernel: [59922.685158] br-pvtlan: received packet on bat0.20 with own address as source address (addr:96:83:c4:00:c3:98, vlan:0)
Wed Apr 22 10:42:48 2020 [1587552168.739] kern.warn kernel: [59922.698085] br-pvtlan: received packet on bat0.20 with own address as source address (addr:96:83:c4:00:c3:98, vlan:0)
```



```

Wed Apr 22 10:44:40 2020 [1587552280.127] kern.warn kernel: [60034.073480] br-pvttlan: received pac
ket on bat0.20 with own address as source address (addr:96:83:c4:00:c3:98, vlan:0)
Wed Apr 22 10:44:40 2020 [1587552280.127] kern.warn kernel: [60034.086205] br-pvttlan: received pac
ket on bat0.20 with own address as source address (addr:96:83:c4:00:c3:98, vlan:0)
Wed Apr 22 10:44:40 2020 [1587552280.141] kern.warn kernel: [60034.100187] br-pvttlan: received pac
ket on bat0.20 with own address as source address (addr:96:83:c4:00:c3:98, vlan:0)
Wed Apr 22 10:44:40 2020 [1587552280.155] kern.warn kernel: [60034.113136] br-pvttlan: received pac
ket on bat0.20 with own address as source address (addr:96:83:c4:00:c3:98, vlan:0)
Wed Apr 22 10:44:40 2020 [1587552280.167] kern.warn kernel: [60034.126583] br-pvttlan: received pac
ket on bat0.20 with own address as source address (addr:96:83:c4:00:c3:98, vlan:0)
Wed Apr 22 10:44:40 2020 [1587552280.183] kern.warn kernel: [60034.140927] br-pvttlan: received pac
ket on bat0.20 with own address as source address (addr:96:83:c4:00:c3:98, vlan:0)
Wed Apr 22 10:44:40 2020 [1587552280.197] kern.warn kernel: [60034.156150] br-pvttlan: received pac
ket on bat0.20 with own address as source address (addr:96:83:c4:00:c3:98, vlan:0)
Wed Apr 22 10:44:40 2020 [1587552280.220] kern.warn kernel: [60034.177078] br-pvttlan: received pac
ket on bat0.20 with own address as source address (addr:96:83:c4:00:c3:98, vlan:0)

```

Related issues:

Related to batman-adv - Bug #170: br-lan: received packet on bat0 with own ad...

Closed

05/06/2013

History

#1 - 04/22/2020 01:00 PM - Sven Eckelmann

- Assignee changed from batman-adv developers to Simon Wunderlich

#2 - 04/22/2020 01:02 PM - Sven Eckelmann

What packet is reported here? Did you check for uevent BATADV_UEV_LOOPDETECT? Is the mac address of the br-pvttlan the same as bat0?

#3 - 04/22/2020 01:33 PM - Sven Eckelmann

- Assignee changed from Simon Wunderlich to Matteo Fortini

- Status changed from New to Feedback

#4 - 04/22/2020 01:46 PM - Matteo Fortini

I add that the two nodes connected to the wired network are the only ones seeing the issue, the wireless ones are not saying anything.

#5 - 04/22/2020 01:49 PM - Matteo Fortini

Sven Eckelmann wrote:

What packet is reported here? Did you check for uevent BATADV_UEV_LOOPDETECT? Is the mac address of the br-pvttlan the same as bat0?

The mac address of the bridge used to be the same, then I changed it. Problem is still there. I haven't checked the uevent, I will look into it

```

bat0      Link encap:Ethernet  HWaddr BA:03:29:67:EF:22
bat0.107  Link encap:Ethernet  HWaddr BA:03:29:67:EF:22
bat0.20   Link encap:Ethernet  HWaddr BA:03:29:67:EF:22
br-IOT    Link encap:Ethernet  HWaddr 92:83:C4:00:C3:A4
br-pvttlan Link encap:Ethernet  HWaddr 96:83:C4:00:C3:98
eth0      Link encap:Ethernet  HWaddr 94:83:C4:00:C3:97
eth0.1    Link encap:Ethernet  HWaddr 96:83:C4:00:C3:9A
eth0.2    Link encap:Ethernet  HWaddr 96:83:C4:00:C3:AA
ifb4pppoe-wan Link encap:Ethernet HWaddr 3E:AB:72:AC:68:E9
mesh0     Link encap:Ethernet  HWaddr 92:83:C4:00:C3:A2
wlan0-1   Link encap:Ethernet  HWaddr 92:83:C4:00:C3:A0
wlan0-2   Link encap:Ethernet  HWaddr 92:83:C4:00:C3:A4

```

#6 - 04/22/2020 06:21 PM - Matteo Fortini

Sven Eckelmann wrote:

What packet is reported here? Did you check for uevent BATADV_UEV_LOOPDETECT? Is the mac address of the br-pvntlan the same as bat0?

Sorry, how do I dump the uevents in openwrt? I tried procd but it seems I haven't found the right command line yet. Thank you

#7 - 04/22/2020 07:42 PM - Matteo Fortini

The packets seem to be ARP requests.

(I have DAT enabled)

[logread...logread...](#)

```
Wed Apr 22 17:39:47 2020 kern.warn kernel: [ 6674.547387] br-pvntlan: received packet on eth0.2 with own address as source address (addr:96:83:c4:00:c3:98, vlan:0)
Wed Apr 22 17:39:47 2020 kern.warn kernel: [ 6674.558329] br-pvntlan: received packet on eth0.2 with own address as source address (addr:96:83:c4:00:c3:98, vlan:0)
Wed Apr 22 17:39:47 2020 kern.warn kernel: [ 6674.570208] br-pvntlan: received packet on bat0.20 with own address as source address (addr:96:83:c4:00:c3:98, vlan:0)
Wed Apr 22 17:39:47 2020 kern.warn kernel: [ 6674.581221] br-pvntlan: received packet on bat0.20 with own address as source address (addr:96:83:c4:00:c3:98, vlan:0)
Wed Apr 22 17:40:06 2020 kern.warn kernel: [ 6694.292656] br-pvntlan: received packet on eth0.2 with own address as source address (addr:96:83:c4:00:c3:98, vlan:0)
Wed Apr 22 17:40:06 2020 kern.warn kernel: [ 6694.303584] br-pvntlan: received packet on eth0.2 with own address as source address (addr:96:83:c4:00:c3:98, vlan:0)
Wed Apr 22 17:40:07 2020 kern.warn kernel: [ 6694.315632] br-pvntlan: received packet on bat0.20 with own address as source address (addr:96:83:c4:00:c3:98, vlan:0)
Wed Apr 22 17:40:07 2020 kern.warn kernel: [ 6694.326606] br-pvntlan: received packet on bat0.20 with own address as source address (addr:96:83:c4:00:c3:98, vlan:0)
Wed Apr 22 17:40:09 2020 kern.warn kernel: [ 6696.397047] br-pvntlan: received packet on bat0.20 with own address as source address (addr:96:83:c4:00:c3:98, vlan:0)
Wed Apr 22 17:40:09 2020 kern.warn kernel: [ 6696.408020] br-pvntlan: received packet on bat0.20 with own address as source address (addr:96:83:c4:00:c3:98, vlan:0)
Wed Apr 22 17:40:09 2020 kern.warn kernel: [ 6696.420887] br-pvntlan: received packet on eth0.2 with own address as source address (addr:96:83:c4:00:c3:98, vlan:0)
Wed Apr 22 17:40:09 2020 kern.warn kernel: [ 6696.431805] br-pvntlan: received packet on eth0.2 with own address as source address (addr:96:83:c4:00:c3:98, vlan:0)
Wed Apr 22 17:40:09 2020 kern.warn kernel: [ 6696.796379] br-pvntlan: received packet on bat0.20 with own address as source address (addr:96:83:c4:00:c3:98, vlan:0)
Wed Apr 22 17:40:09 2020 kern.warn kernel: [ 6696.807339] br-pvntlan: received packet on bat0.20 with own address as source address (addr:96:83:c4:00:c3:98, vlan:0)
Wed Apr 22 17:40:22 2020 kern.warn kernel: [ 6709.356428] net_ratelimit: 603 callbacks suppressed
Wed Apr 22 17:40:22 2020 kern.warn kernel: [ 6709.356444] br-pvntlan: received packet on bat0.20 with own address as source address (addr:96:83:c4:00:c3:98, vlan:0)
```

[tcpdump...tcpdump...](#)

```
tcpdump -vvvn -i bat0.20 ether src '96:83:c4:00:c3:98'
tcpdump: listening on bat0.20, link-type EN10MB (Ethernet), capture size 262144 bytes
17:38:10.754333 ARP, Ethernet (len 6), IPv4 (len 4), Reply 0.0.0.0 is-at 92:83:c4:00:c3:a2, length 28
17:38:10.754257 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 192.168.99.172 tell 192.168.99.1, length 46
17:38:30.273039 IP (tos 0xc0, ttl 1, id 0, offset 0, flags [DF], proto IGMP (2), length 32, options (RA))
    0.0.0.0 > 224.0.0.1: igmp query v2
17:39:47.221090 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 192.168.99.235 tell 192.168.99.1, length 28
17:39:47.236136 ARP, Ethernet (len 6), IPv4 (len 4), Reply 0.0.0.0 is-at 82:16:f9:9b:cd:44, length 46
17:39:47.247056 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 192.168.99.235 tell 192.168.99.1, length 46
17:39:47.247551 ARP, Ethernet (len 6), IPv4 (len 4), Reply 0.0.0.0 is-at 92:83:c4:00:c3:a2, length 28
```


17:40:10.675995 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 192.168.99.156 tell 192.168.99.1, length 46
17:40:10.676962 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 192.168.99.156 tell 192.168.99.1, length 46
17:40:10.677451 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 192.168.99.156 tell 192.168.99.1, length 46
17:40:10.679242 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 192.168.99.156 tell 192.168.99.1, length 46
17:40:10.679323 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 192.168.99.156 tell 192.168.99.1, length 46
17:40:10.681309 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 192.168.99.156 tell 192.168.99.1, length 46
17:40:10.681389 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 192.168.99.156 tell 192.168.99.1, length 46
17:40:10.682609 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 192.168.99.156 tell 192.168.99.1, length 46
17:40:10.683111 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 192.168.99.156 tell 192.168.99.1, length 46
17:40:10.684850 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 192.168.99.156 tell 192.168.99.1, length 46
17:40:10.685311 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 192.168.99.156 tell 192.168.99.1, length 46
17:40:10.686881 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 192.168.99.156 tell 192.168.99.1, length 46
17:40:10.686962 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 192.168.99.156 tell 192.168.99.1, length 46
17:40:22.034404 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 192.168.99.123 tell 192.168.99.1, length 46

#8 - 04/22/2020 07:45 PM - Matteo Fortini

I can reproduce the problem with DAT enabled, if I disable DAT just on the offending nodes, the problem goes away

#9 - 04/23/2020 09:06 AM - Sven Eckelmann

- Related to Bug #170: br-lan: received packet on bat0 with own address as source address added

#10 - 04/23/2020 09:20 AM - Sven Eckelmann

- Assignee changed from Matteo Fortini to Antonio Quartulli

- Status changed from Feedback to In Progress

- Description updated

- Subject changed from received packet on bat0.20/eth0 with own address as source address storming the network to DAT: received packet on bat0.20/eth0 with own address as source address

I can reproduce the problem with DAT enabled, if I disable DAT just on the offending nodes, the problem goes away

Ok, so it sounds like a bug for the DAT author.

Regarding loopdetect. I think you can manually install udevadmin (in a really manual way) to get the monitor command. Maybe Linus can add some note about this (in case it is easy enough to explain).

The other way would be to install a hotplug script which logs the things. Just create a file called /etc/hotplug.d/net/99-batman-adv-ev. Inside this file you can react to various events from batman-adv. Just from my memory, it should work somewhat like this:

```
#!/bin/sh

batadv_uev_gw()
{
    case "$BATACTION" in
    add)
        echo "uev gw add $BATDATA"|logger -t batadv
        ;;
    change)
        echo "uev gw change $BATDATA"|logger -t batadv
        ;;
    del)
        echo "uev gw del"|logger -t batadv
        ;;
    esac
}

batadv_uev_bla()
{
    case "$BATACTION" in
    loopdetect)
        echo "uev loopdetect on vid $BATDATA"|logger -t batadv
        ;;
    esac
}

case "$BATTTYPE" in
gw)
    batadv_uev_gw
    ;;
bla)
    batadv_uev_bla
    ;;
esac
```

#11 - 04/23/2020 09:23 AM - Sven Eckelmann

- *Description updated*

#12 - 05/01/2020 07:42 PM - Marek Lindner

- *Assignee changed from Antonio Quartulli to Matteo Fortini*

Hi Matteo,

I am one of the thinkers behind DAT - it sounds a bit odd that DAT would cause loops. DAT only reacts to ARP packets created by non-mesh clients.

To better understand, can you take a packet capture with tcpdump of the looping packets and attach the pcap to this ticket ? This will allow us to understand what packet is looping.

Also, can you explain the simplest possible setup to reproduce this loop ? Meaning: which interfaces can you eliminate and still get the loop ? What type of non-mesh client needs to be connected where ? Or does the loop even begin without any client ?

Thanks,
Marek