

batman-adv - Bug #173

Mesh packets on bat0

07/13/2013 12:31 AM - Oliver Möschen

Status:	Closed	Start date:	07/13/2013
Priority:	High	Due date:	
Assignee:		% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:	2015.0		

Description

Batman 2013.2.0 sends incoming mesh packets to bat0 on Linux 2.6.32

Testsetup

Two identical nodes, running Debian 6.0.7 (x86), eth1 of both nodes connected

```
node1 ~ # uname -a
Linux node1 2.6.32-5-686 #1 SMP Fri May 10 08:33:48 UTC 2013 i686 GNU/Linux
node1 ~ # dpkg -l | grep 2.6.32-5-686
ii  linux-headers-2.6.32-5-686          2.6.32-48squeeze3      Header files for Linux 2.6.32
-5-686
ii  linux-image-2.6.32-5-686           2.6.32-48squeeze3      Linux 2.6.32 for modern PCs

node1 ~ # brctl show
bridge name      bridge id          STP enabled      interfaces
node1 ~ # ip link set eth1 up
node1 ~ # batctl if add eth1
node1 ~ # batctl if
eth1: active
node1 ~ # ip link set bat0 up
node1 ~ # ip add add 10.10.10.1/24 dev bat0
node1 ~ # ip add
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue state UNKNOWN
link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
inet 127.0.0.1/8 scope host lo
inet6 ::1/128 scope host
valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
link/ether 08:00:27:4b:75:b3 brd ff:ff:ff:ff:ff:ff
inet 10.1.1.215/24 brd 10.1.1.255 scope global eth0
inet6 fe80::a00:27ff:fe4b:75b3/64 scope link
valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,SLAVE,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast master bat0 state UP ql
en 1000
link/ether 08:00:27:00:00:01 brd ff:ff:ff:ff:ff:ff
inet6 fe80::a00:27ff:fe00:1/64 scope link
valid_lft forever preferred_lft forever
4: bat0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UNKNOWN
link/ether 9a:fa:b3:73:f6:d8 brd ff:ff:ff:ff:ff:ff
inet 10.10.10.1/24 scope global bat0
inet6 fe80::98fa:b3ff:fe73:f6d8/64 scope link
valid_lft forever preferred_lft forever

node2 ~ # brctl show
bridge name      bridge id          STP enabled      interfaces
node2 ~ # ip link set eth1 up
node2 ~ # batctl if add eth1
node2 ~ # batctl if
eth1: active
node2 ~ # ip link set bat0 up
node2 ~ # ip add add 10.10.10.2/24 dev bat0
node2 ~ # ip add
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue state UNKNOWN
```

```

link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
inet 127.0.0.1/8 scope host lo
inet6 ::1/128 scope host
    valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 08:00:27:d9:50:55 brd ff:ff:ff:ff:ff:ff
    inet 10.1.1.216/24 brd 10.1.1.255 scope global eth0
    inet6 fe80::a00:27ff:fed9:5055/64 scope link
    valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,SLAVE,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast master bat0 state UP ql
en 1000
    link/ether 08:00:27:00:00:02 brd ff:ff:ff:ff:ff:ff
    inet6 fe80::a00:27ff:fe00:2/64 scope link
    valid_lft forever preferred_lft forever
4: bat0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UNKNOWN
    link/ether b2:40:25:42:d3:05 brd ff:ff:ff:ff:ff:ff
    inet 10.10.10.2/24 scope global bat0
    inet6 fe80::b040:25ff:fe42:d305/64 scope link
    valid_lft forever preferred_lft forever

```

```

node1 ~ # batctl o
[B.A.T.M.A.N. adv 2013.2.0, MainIF/MAC: eth1/08:00:27:00:00:01 (bat0)]
Originator      last-seen (#/255)      Nexthop [outgoingIF]:  Potential nexthops ...
08:00:27:00:00:02  0.780s  (255) 08:00:27:00:00:02 [ eth1]: 08:00:27:00:00:02 (255)

```

```

node2 ~ # batctl o
[B.A.T.M.A.N. adv 2013.2.0, MainIF/MAC: eth1/08:00:27:00:00:02 (bat0)]
Originator      last-seen (#/255)      Nexthop [outgoingIF]:  Potential nexthops ...
08:00:27:00:00:01  0.420s  (255) 08:00:27:00:00:01 [ eth1]: 08:00:27:00:00:01 (255)

```

Let's send some traffic to the mesh:

```

node1 ~ # ping 10.10.10.2
PING 10.10.10.2 (10.10.10.2) 56(84) bytes of data.
64 bytes from 10.10.10.2: icmp_req=1 ttl=64 time=1.97 ms
64 bytes from 10.10.10.2: icmp_req=2 ttl=64 time=0.318 ms
64 bytes from 10.10.10.2: icmp_req=3 ttl=64 time=0.314 ms
[...]

```

On eth1 of node2 everything looks fine:

```

node2 ~ # batctl td eth1
23:55:17.975137 BAT 08:00:27:00:00:01 > 08:00:27:00:00:02: UCAST, ttvn 1, ttl 50, IP 10.10.10.1 >
10.10.10.2: ICMP echo request, id 1588, seq 34, length 64
23:55:17.975261 BAT 08:00:27:00:00:02 > 08:00:27:00:00:01: UCAST, ttvn 1, ttl 50, IP 10.10.10.2 >
10.10.10.1: ICMP echo reply, id 1588, seq 34, length 64
23:55:18.024977 BAT 08:00:27:00:00:02: OGM IV via neigh 08:00:27:00:00:02, seq 288776459, tq 255,
ttvn 1, ttcrc 13868, ttl 50, v 14, flags [...F.], length 26
23:55:18.128469 BAT 08:00:27:00:00:02: OGM IV via neigh 08:00:27:00:00:01, seq 288776459, tq 225,
ttvn 1, ttcrc 13868, ttl 49, v 14, flags [.D...], length 46
23:55:18.461849 BAT 08:00:27:00:00:01: OGM IV via neigh 08:00:27:00:00:01, seq 2362693815, tq 255,
ttvn 1, ttcrc 52668, ttl 50, v 14, flags [...F.], length 46
23:55:18.564340 BAT 08:00:27:00:00:01: OGM IV via neigh 08:00:27:00:00:02, seq 2362693815, tq 225,
ttvn 1, ttcrc 52668, ttl 49, v 14, flags [.D...], length 26
23:55:18.981405 BAT 08:00:27:00:00:01 > 08:00:27:00:00:02: UCAST, ttvn 1, ttl 50, IP 10.10.10.1 >
10.10.10.2: ICMP echo request, id 1588, seq 35, length 64
23:55:18.981457 BAT 08:00:27:00:00:02 > 08:00:27:00:00:01: UCAST, ttvn 1, ttl 50, IP 10.10.10.2 >
10.10.10.1: ICMP echo reply, id 1588, seq 35, length 64
23:55:19.036161 BAT 08:00:27:00:00:02: OGM IV via neigh 08:00:27:00:00:02, seq 288776460, tq 255,
ttvn 1, ttcrc 13868, ttl 50, v 14, flags [...F.], length 26

```

```
23:55:19.138587 BAT 08:00:27:00:00:02: OGM IV via neigh 08:00:27:00:00:01, seq 288776460, tq 225,
ttvn 1, ttcrc 13868, ttl 49, v 14, flags [.D...], length 46
23:55:19.465249 BAT 08:00:27:00:00:01: OGM IV via neigh 08:00:27:00:00:01, seq 2362693816, tq 255,
ttvn 1, ttcrc 52668, ttl 50, v 14, flags [...F.], length 46
23:55:19.564327 BAT 08:00:27:00:00:01: OGM IV via neigh 08:00:27:00:00:02, seq 2362693816, tq 225,
ttvn 1, ttcrc 52668, ttl 49, v 14, flags [.D...], length 26
23:55:19.987974 BAT 08:00:27:00:00:01 > 08:00:27:00:00:02: UCAST, ttvn 1, ttl 50, IP 10.10.10.1 >
10.10.10.2: ICMP echo request, id 1588, seq 36, length 64
23:55:19.988016 BAT 08:00:27:00:00:02 > 08:00:27:00:00:01: UCAST, ttvn 1, ttl 50, IP 10.10.10.2 >
10.10.10.1: ICMP echo reply, id 1588, seq 36, length 64
23:55:20.024850 BAT 08:00:27:00:00:02: OGM IV via neigh 08:00:27:00:00:02, seq 288776461, tq 255,
ttvn 1, ttcrc 13868, ttl 50, v 14, flags [...F.], length 26
23:55:20.128600 BAT 08:00:27:00:00:02: OGM IV via neigh 08:00:27:00:00:01, seq 288776461, tq 225,
ttvn 1, ttcrc 13868, ttl 49, v 14, flags [.D...], length 46
^C
```

But there are many packets on bat0 which shouldn't be there:

```
node2 ~ # batctl td bat0
```

```
23:55:35.531057 BAT 08:00:27:00:00:01: OGM IV via neigh 08:00:27:00:00:01, seq 2362693832, tq 255,
ttvn 1, ttcrc 52668, ttl 50, v 14, flags [...F.], length 46
23:55:36.090997 BAT 08:00:27:00:00:01 > 08:00:27:00:00:02: UCAST, ttvn 1, ttl 50, IP 10.10.10.1 >
10.10.10.2: ICMP echo request, id 1588, seq 52, length 64
23:55:36.091051 IP 10.10.10.1 > 10.10.10.2: ICMP echo request, id 1588, seq 52, length 64
23:55:36.091062 IP 10.10.10.2 > 10.10.10.1: ICMP echo reply, id 1588, seq 52, length 64
23:55:36.192880 BAT 08:00:27:00:00:02: OGM IV via neigh 08:00:27:00:00:01, seq 288776477, tq 225,
ttvn 1, ttcrc 13868, ttl 49, v 14, flags [.D...], length 46
23:55:36.529641 BAT 08:00:27:00:00:01: OGM IV via neigh 08:00:27:00:00:01, seq 2362693833, tq 255,
ttvn 1, ttcrc 52668, ttl 50, v 14, flags [...F.], length 46
23:55:37.097030 BAT 08:00:27:00:00:01 > 08:00:27:00:00:02: UCAST, ttvn 1, ttl 50, IP 10.10.10.1 >
10.10.10.2: ICMP echo request, id 1588, seq 53, length 64
23:55:37.097071 IP 10.10.10.1 > 10.10.10.2: ICMP echo request, id 1588, seq 53, length 64
23:55:37.097081 IP 10.10.10.2 > 10.10.10.1: ICMP echo reply, id 1588, seq 53, length 64
23:55:37.158529 BAT 08:00:27:00:00:02: OGM IV via neigh 08:00:27:00:00:01, seq 288776478, tq 225,
ttvn 1, ttcrc 13868, ttl 49, v 14, flags [.D...], length 46
23:55:37.536005 BAT 08:00:27:00:00:01: OGM IV via neigh 08:00:27:00:00:01, seq 2362693834, tq 255,
ttvn 1, ttcrc 52668, ttl 50, v 14, flags [...F.], length 46
23:55:38.103986 BAT 08:00:27:00:00:01 > 08:00:27:00:00:02: UCAST, ttvn 1, ttl 50, IP 10.10.10.1 >
10.10.10.2: ICMP echo request, id 1588, seq 54, length 64
23:55:38.104027 IP 10.10.10.1 > 10.10.10.2: ICMP echo request, id 1588, seq 54, length 64
23:55:38.104037 IP 10.10.10.2 > 10.10.10.1: ICMP echo reply, id 1588, seq 54, length 64
23:55:38.177199 BAT 08:00:27:00:00:02: OGM IV via neigh 08:00:27:00:00:01, seq 288776479, tq 225,
ttvn 1, ttcrc 13868, ttl 49, v 14, flags [.D...], length 46
23:55:38.554654 BAT 08:00:27:00:00:01: OGM IV via neigh 08:00:27:00:00:01, seq 2362693835, tq 255,
ttvn 1, ttcrc 52668, ttl 50, v 14, flags [...F.], length 46
^C
```

Batman 2013.1.0 and Debian kernel 2.6.32-48squeeze3 doesn't show this problem.

Batman 2013.2.0 and Debian kernel 3.2.46-1~bpo60+1 also doesn't show this problem.

These are the attached files

```
node2 ~ # tcpdump -i eth1 -s 1500 -w node2-eth1.pcap
node2 ~ # tcpdump -i bat0 -s 1500 -w node2-bat0.pcap
node2 ~ # batctl ll all
node2 ~ # batctl l > batman-node2.log
```

History

#1 - 07/13/2013 12:43 AM - Antonio Quartulli

Hello Oliver,

after a quick reading of your report I have the feeling that there is some compatibility issue between batman-adv and the kernel you are running.

Oliver Möschen wrote:

...

Batman 2013.1.0 and Debian kernel 2.6.32-48squeeze3 doesn't show this problem.

...

Does this mean that batman-adv 2013.1.0 is running fine in the same environment where batman-adv 2013.2.0 is having troubles?
Or the kernel is not exactly the same?

Cheers,

#2 - 07/13/2013 01:20 AM - Oliver Möschen

Hello Antonio,

in the same environment batman-adv 2013.1.0 is running fine.

Oliver

#3 - 07/13/2013 10:38 AM - Antonio Quartulli

Oliver, do you think you can test 2.6.32 vanilla? In this way we can understand if any of the patches brought by debian is affecting batman-adv somehow.

#4 - 07/13/2013 04:04 PM - Oliver Möschen

OK, I will test 2.6.32 vanilla tonight.

#5 - 07/14/2013 01:40 AM - Oliver Möschen

I've tested several vanilla kernels with batman-adv 2013.2.0 using the setup described above. These are the results:

```
2.6.32      Mesh traffic on bat0
2.6.32.27  Mesh traffic on bat0
2.6.33.7   Mesh traffic on bat0
2.6.34.7   Mesh traffic on bat0
2.6.35.9   Mesh traffic on bat0
2.6.36.4   Mesh traffic on bat0
2.6.37.6   Mesh traffic on bat0
2.6.38.8   Mesh traffic on bat0
2.6.39     OK
```

#6 - 07/14/2013 10:32 AM - Antonio Quartulli

Thanks for you precious information Oliver.

I will try to understand what's wrong with those kernels. We have a compat layer which makes batman-adv compatible with old kernels despite the API changes brought by the new versions and I think something went wrong there.

We will try to provide you a patch that you can test.

#7 - 09/10/2013 10:55 PM - Antonio Quartulli

It looks like the problem is in the new RTNL API implementation we introduced in v2013.2.0, in particular in its "compatibility" code which is used when batman-adv is compiled on older kernels.

We should now try to understand what is different in kernels <2.6.39 for what concerns this API.

#8 - 02/19/2015 11:38 AM - Marek Lindner

- Status changed from New to Closed

I just pushed a patch into the master branch addressing this compat issue. The mailing list has the entire discussion leading up to this patch:

<https://lists.open-mesh.org/mailman3/hyperkitty/list/b.a.t.m.a.n@lists.open-mesh.org/message/WC6WIL6VTC7OVCI5F7U6KN6DOQTRCSZ5/>

Feel free to re-open the ticket if your problem persists.

#9 - 02/11/2017 09:12 AM - Sven Eckelmann

- Target version set to 2015.0

Files

batman-node2.log	12 KB	07/12/2013	Oliver Möschen
node2-bat0.pcap	3.49 KB	07/12/2013	Oliver Möschen
node2-eth1.pcap	4.45 KB	07/12/2013	Oliver Möschen