batman-adv - Feature #310

tpmeter: convert any provided address to proper originator address

11/04/2016 03:38 PM - Martin Weinelt

Status:	New	**Start date:**	11/04/2016
Priority:	Low	**Due date:**	
Assignee:	Antonio Quartulli	**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			

Description

```
# batctl -m ffda-bat n
[B.A.T.M.A.N. adv 2016.4, MainIF/MAC: ffda-vpn/56:a3:b3:8b:aa:e4 (ffda-bat/2a:a9:cb:dd:79:4e BATMA
N_IV)]
IF             Neighbor              last-seen
     ffda-vpn      da:ff:61:00:05:03    0.240s
     ffda-vpn      da:ff:61:00:02:03    0.540s
          en1      42:f7:31:6f:6c:c8    0.600s
```

```
# batctl -m ffda-bat tp da:ff:61:00:05:03
Test duration 10110ms.
Sent 0 Bytes.
Throughput: 0 Bytes/s (0 Bps)
# batctl -m ffda-bat tp da:ff:61:00:02:03
Test duration 10110ms.
Sent 0 Bytes.
Throughput: 0 Bytes/s (0 Bps)
```

All hosts involved are running batman-adv 2016.4. The local host from where I'm running the tpmeter has the following setup:

```
# ip netns exec ffda ip link
1: lo: <LOOPBACK> mtu 65536 qdisc noop state DOWN mode DEFAULT group default qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: ffda-bat: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue master ffda-br state UNKNOWN
 mode DEFAULT group default qlen 1000
    link/ether 2a:a9:cb:dd:79:4e brd ff:ff:ff:ff:ff:ff
3: en1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel master ffda-bat state UP mode DE
FAULT group default qlen 1000
    link/ether 00:25:90:0e:66:41 brd ff:ff:ff:ff:ff:ff
4: ffda-br: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP mode DEFAULT group d
efault qlen 1000
    link/ether 2a:a9:cb:dd:79:4e brd ff:ff:ff:ff:ff:ff
11: ffda-vpn: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1280 qdisc fq_codel master ffda-bat state UNKN
OWN mode DEFAULT group default qlen 1000
    link/ether 56:a3:b3:8b:aa:e4 brd ff:ff:ff:ff:ff:ff
```

where

- ffda-bat is the batman-adv if
- ffda-vpn is a fastd tunnel with 1280 MTU
- en1 is a hardlink connecting a local router
- ffda-br is a bridge wrapping the ffda-bat if

There is no firewalling set up:

```
# ip netns exec ffda iptables-save
# Generated by iptables-save v1.6.0 on Fri Nov  4 15:34:37 2016
*filter
:INPUT ACCEPT [17646:4553404]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [1482:94244]
COMMIT
# Completed on Fri Nov  4 15:34:37 2016
```

History

#1 - 11/04/2016 03:56 PM - Sven Eckelmann

Just as info for Antonio: I can see tpmeter packets in the dump (in both directions) when using the filter

```
(batadv.batman.packet_type == 0x43) && (batadv.icmp.msg_type == 15)
```

Martin: "Sent 0 Bytes." is about the acked packets. So packets may have been sent but the sender thinks that no packets were acked.

#2 - 02/27/2018 06:29 PM - Sven Eckelmann

Was da:ff:61:00:05:03 really the primaryif address of the remote?

#3 - 02/27/2018 09:43 PM - Martin Weinelt

Our infrastructure has massively changed so I can't say that for sure.

However the actual issue seems to be that the primaryif address is unnecessarily hard to determine for a remote node that I don't have access to. Consider this:

What are my neighbours?

```
# batctl n
[B.A.T.M.A.N. adv 2017.4, MainIF/MAC: primary0/42:f7:31:6f:6c:cb (bat0/e8:94:f6:69:4d:86 BATMAN_IV)]
IF             Neighbor              last-seen
       br-wan      00:25:90:0e:66:41    0.520s
        mesh0      3e:df:5f:8e:b3:41    1.900s
        mesh0      96:8d:78:23:af:1c    3.620s
        mesh0      7a:fc:d9:66:55:15    0.750s
        mesh0      72:f2:91:1d:c9:f9    4.530s
```

Let's ping … that one!

```
# batctl ping 96:8d:78:23:af:1c
PING 96:8d:78:23:af:1c (96:8d:78:23:af:1c) 20(48) bytes of data
20 bytes from 96:8d:78:23:af:1c icmp_seq=1 ttl=50 time=3.41 ms
20 bytes from 96:8d:78:23:af:1c icmp_seq=2 ttl=50 time=0.47 ms
20 bytes from 96:8d:78:23:af:1c icmp_seq=3 ttl=50 time=0.48 ms
20 bytes from 96:8d:78:23:af:1c icmp_seq=4 ttl=50 time=0.45 ms
^C--- 96:8d:78:23:af:1c ping statistics ---
4 packets transmitted, 4 received, 0% packet loss
rtt min/avg/max/mdev = 0.452/1.203/3.412/1.275 ms
```

And what's the throughput?

```
# batctl tp 96:8d:78:23:af:1c
Test duration 10110ms.
Sent 0 Bytes.
Throughput: 0 Bytes/s (0 Bps)
```

Turn's out … that is not the primary address of the node, but it is enough for ping, why not tp? Feels a bit like unexpected behaviour.

#4 - 02/28/2018 02:07 AM - Antonio Quartulli

Martin Weinelt wrote:

> Turn's out … that is not the primary address of the node, but it is enough for ping, why not tp? Feels a bit like unexpected behaviour.

the address you feed to ping/tr goes through a conversion routine which will extract the proper address to use for you. You can even use an IP and this routine will make the command work anyway.

tp_meter does not use the same routine yet, so you need to specify the correct originator address when performing a test. Some ideas about adapting tp_meter to use this routine have been fluctuating around, but nothing has been implemented yet.

#5 - 02/28/2018 02:15 AM - Martin Weinelt

tp_meter works just fine when using the primary address for the remote node.

#6 - 02/28/2018 02:16 AM - Antonio Quartulli

- Subject changed from tpmeter: nothing sent, zero throughput to tpmeter: convert any provided address to proper originator address

- Tracker changed from Bug to Feature

renaming the ticket and making it a feature wish

#7 - 02/28/2018 10:18 AM - Sven Eckelmann

- File RFC-batctl-tp_meter-Translate-client-mac-and-IPs-to-orig.patch added

Here the IRC discussion about it:

```
<hexa-> # batctl ping 2a:a9:cb:dd:79:4e
<hexa-> PING 2a:a9:cb:dd:79:4e (56:a3:b3:8b:aa:e4) 20(48) bytes of data
<hexa-> 20 bytes from 2a:a9:cb:dd:79:4e icmp_seq=1 ttl=50 time=0.17 ms
<hexa-> # batctl tp 2a:a9:cb:dd:79:4e
<hexa-> Destination unreachable
<hexa-> how does that match up?
<hexa-> the two nodes are using 2017.3 and 2017.4
<neoraider> Hmm, I think tp is more picky about which of a node's addresses you use
<neoraider> Is 2a:a9:cb:dd:79:4e an originator address or a tg entry?
```

```
<hexa-> originator address
<hexa-> it works with the mac in `batctl n`
<hexa-> except sent/throughput is 0 :)
<neoraider> hexa-, hmm, if I interpret this correctly, the originator address should be 56:a3:b3:8b:aa:e4 (wha
t batctl ping shows in the parens)
<neoraider> Have you tried tp'ing that?
<hexa-> 2a:a9:cb:dd:79:4e is the mac of the target batadv interface
<hexa->        br-wan  00:25:90:0e:66:41    0.730s
<hexa-> that's the mac on the slave link
<hexa-> tp'ing the slave link results in 10s of 0 sent/0  throughput
<neoraider> hexa-, so it is a translation table entry and not the orginator address after all
<hexa-> hm, the originator address is derived from what?
<hexa-> the first slave in batadv?
<hexa-> because the vpn has 56:a3:b3:8b:aa:e4
<neoraider> Address of primary member interface
<neoraider> Which is the interface added first
<hexa-> Sent 325156896 Bytes.
<hexa-> Throughput: 30.92 MB/s (259.35 Mbps)
<hexa-> fair
<hexa-> a bit weird, but I'll take it :)
<neoraider> batctl ping resolves through the tg by itself, for tp, you need to pass the originator address dir
ectly
<hexa-> Usage: batctl tp [parameters] <MAC>
<neoraider> So just look it up in the tg, or use what batctl ping shows
<hexa-> yup, good call
<hexa-> # batctl ping 7a:fc:d9:66:55:11
<hexa-> PING 7a:fc:d9:66:55:11 (7a:fc:d9:66:55:11) 20(48) bytes of data
<hexa-> 20 bytes from 7a:fc:d9:66:55:11 icmp_seq=1 ttl=50 time=2.27 ms
<hexa-> # batctl tp 7a:fc:d9:66:55:11
<hexa-> Test duration 10110ms.
<hexa-> Sent 0 Bytes.
<hexa-> Throughput: 0 Bytes/s (0 Bps)
<hexa-> both gluon master nodes with 2017.4
<hexa-> gluon firewalling issue?
<neoraider> Unlikely
<hexa-> the two nodes are 10m apart an have dualband mesh connectivity
<hexa-> neither direction works
<neoraider> 7a:fc:d9:66:55:11 is not the primary addres of a Gluon node, the address of primary0 always ends w
ith 3 or b
<neoraider> Try 7a:fc:d9:66:55:13
<hexa-> Throughput: 7.23 MB/s (60.69 Mbps)
<hexa-> so the ping tg lookup was deceptive
<hexa-> misleading
<neoraider> I'm not sure if there is a good way to determine the primary address of a remote node then :/
<hexa-> hehe
<neoraider> Except on Gluon, where you can follow this schema: http://gluon.readthedocs.io/en/v2017.1.x/dev/ma
c_addresses.html
<hexa-> aye
<hexa-> basically the mac of primary0
<neoraider> Exactly
<neoraider> We basically introduced primary0 to get a stable primary address
```

As you can see, the introduction of the ping translation function in batctl (as shown in the attached patch RFC-batctl-tp_meter-Translate-client-mac-and-IPs-to-orig.patch) will not help here. You can create a test setup based on Emulation_Debug and use following script to prepare the system:

```
#! /bin/sh

set -e
set -v

MAC_PART="$(ip link show enp0s3 | awk '/ether/ {print $2}'| sed -e 's/.*://' -e 's/[\n\ ].*//'|awk '{print ("0
x"$1)*1 }')"
IP_PART="$(echo $MAC_PART|awk '{ print $1+50 }')"

ip link add dummybat type dummy
ip link set up dev dummybat

rmmod batman-adv || true
insmod /host/batman-adv/build/net/batman-adv/batman-adv.ko
batctl if add dummybat
batctl if add enp0s3
```

```
batctl it 100
batctl gw server

ip link set up dev enp0s3
ip link set up dev bat0
ip -4 addr flush dev bat0
ip addr add 192.168.5.${IP_PART}/24 dev bat0

exit 0
```

As you can see, the ping will work from node1 to node2 but not the tp_meter run with the patch RFC-batctl-tp_meter-Translate-client-mac-and-IPs-to-orig.patch:

```
batctl p 192.168.5.52
PING 192.168.5.52 (42:ae:b7:fa:be:1d) 20(48) bytes of data
20 bytes from 192.168.5.52 icmp_seq=1 ttl=50 time=1.39 ms
20 bytes from 192.168.5.52 icmp_seq=2 ttl=50 time=1.39 ms
20 bytes from 192.168.5.52 icmp_seq=3 ttl=50 time=5.74 ms
^C--- 192.168.5.52 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss
rtt min/avg/max/mdev = 1.388/2.838/5.738/2.051 ms

root@bentobox:~# batctl p batctl p 02:ba:de:af:fe:02
PING 02:ba:de:af:fe:02 (02:ba:de:af:fe:02) 20(48) bytes of data
20 bytes from 02:ba:de:af:fe:02 icmp_seq=1 ttl=50 time=1.44 ms
20 bytes from 02:ba:de:af:fe:02 icmp_seq=2 ttl=50 time=1.72 ms
^C--- 02:ba:de:af:fe:02 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss
rtt min/avg/max/mdev = 1.444/1.582/1.720/0.138 ms

root@bentobox:~# batctl tp 192.168.5.52
Test duration 10040ms.
Sent 516737268 Bytes.
Throughput: 49.08 MB/s (411.74 Mbps)

root@bentobox:~# batctl tp 02:ba:de:af:fe:02
Test duration 10308ms.
Sent 0 Bytes.
Throughput: 0 Bytes/s (0 Bps)

root@bentobox:~# batctl t 192.168.5.52
42:ae:b7:fa:be:1d

root@bentobox:~# batctl t 02:ba:de:af:fe:02
02:ba:de:af:fe:02
```

Problem is here how the tp packets are handled in the kernel vs. the rest of the batadv "ICMP" code. A way has to be found which is ok for the end user but also allows to test single links (for B.A.T.M.A.N. V).

Files

tpmeter.pcap	509 KB	11/04/2016	Martin Weinelt
RFC-batctl-tp_meter-Translate-client-mac-and-IPs-to-orig.patch	4.21 KB	02/28/2018	Sven Eckelmann