

batman-adv - Bug #252

TT: size check before local entry add is incorrect (not threadsafe?)

05/15/2016 01:57 PM - Sven Eckelmann

Status:	Feedback	Start date:	05/15/2016
Priority:	Normal	Due date:	
Assignee:	Antonio Quartulli	% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:			
Description			
<p>Just tested TT with the Emulation_Debug environment. Two nodes were enabled and I've just send 3000 packets (different mac addresses) with the attached program to the other node. Then I can see that the remote node sends TT full table requests. But the node which send the 3000 packets never sends the response. The problem seems to be that the tvlv length is 31616 bytes (tvlv_len in batadv_tt_prepare_tvlv_local_data) but this is larger than the maximum packet_size (bat_priv->packet_size_max).</p> <p>If you print the size check in batadv_tt_local_add right before the if (table_size > packet_size_max) { then you can see that the transmission size jumps (each "foobar" is an add to the local table):</p> <pre>foobar 116 22080 foobar 11156 22080 foobar 11156 22080</pre>			
<p>The test was started on my node1 via</p> <pre>insmod /host/batman-adv/net/batman-adv/batman-adv.ko</pre>			

```
/host/batctl/batctl ra BATMAN_IV
/host/batctl/batctl if add eth0
ifconfig eth0 up
ifconfig bat0 up
# sleep 3
/host/batctl/batctl o

/host/rawsend_massive bat0 02:ba:de:af:fe:02
```

History

#1 - 05/15/2016 03:08 PM - Antonio Quartulli

- Assignee changed from Antonio Quartulli to Sven Eckelmann
- Status changed from New to Feedback

Sven,
what are you actually showing in your debug output ? If you are printing table_size and packet_size_max I don't understand when you see the table_size getting larger than packet_size_max.

Another question: are you saying that the "jump" looks suspicious to you ?

#2 - 05/15/2016 03:11 PM - Sven Eckelmann

- Assignee changed from Sven Eckelmann to Antonio Quartulli

Yes, the jump looks suspicious to me.

The elements printed are table_size and packet_size_max

#3 - 05/15/2016 03:46 PM - Antonio Quartulli

In my opinion the "jump" is due to the TT commit. When a bunch of local entries are detected by a node they are marked with the N(ew) flag. Such entries are not yet part of the full table (although they are stored in the hash table) and therefore they are not sent when a TT request comes in. As soon as a TT local commit event is triggered (usually together with the next local OGM) all the N(ew) entries are "committed" and thus the table size is increased (the N flag disappears at this point).

Still, I don't understand why the node would not reply with a TT response since the maximum size seems to always be smaller than the packet_size_max...

#4 - 05/15/2016 03:53 PM - Sven Eckelmann

No, it isn't always smaller. I just forgot to post the relevant log at the end of the run:

```
batadv_tt_local_add:690 31652 22080
batadv_tt_local_add:690 31652 22080
batadv_tt_local_add:690 31652 22080
batadv_tt_local_add:690 31652 22080
batadv_tt_local_add:690 31652 22080
batadv_tt_local_add:690 31652 22080
batadv_tt_local_add:690 31652 22080
batadv_tt_local_add:690 31652 22080
```

The local table will also not pruned (for some reason I haven't checked).

#5 - 05/15/2016 03:57 PM - Antonio Quartulli

oh ok, thanks for adding the missing part. Yes in this case it makes sense.

The timeout for a local entry is defined as

```
main.h:46 #define BATADV_TT_LOCAL_TIMEOUT 600000 /* in milliseconds */
```

Files

rawsend_massive.c	2.75 KB	05/15/2016	Sven Eckelmann
-------------------	---------	------------	----------------