

batman-adv - Feature #291

Reduce DAT Cache misses

07/11/2016 10:35 AM - Linus Lüssing

Status:	New	Start date:	07/11/2016
Priority:	Normal	Due date:	
Assignee:	Linus Lüssing	% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:			
Description			
<p>While the overall ARP overhead is greatly reduced, we generally still have many ARP Requests from gateway nodes / routers. In a 1000 node setup this is about 30kbit/s.</p> <p>In a minimal setup with just two hosts (Linux 4.6-rc6, no batman-adv involved), one being a DHCP server, the other one a DHCP client, as well as one persistent TCP connection between them, I noticed that ARP packets are sent rarely. This seems to break the initial assumption, that at least one ARP exchange would take place during the 5min. DAT cache timeout.</p> <p>In the test setup, during a ~37000 seconds (10h) interval, these were the only ARP packets showing up:</p> <pre>5 106.241867 02:04:64:a4:39:d3 -> ff:ff:ff:ff:ff:ff ARP 60 Who has 192.168.123.1? Tell 192.168.123.50 6 106.241958 02:04:64:a4:39:f2 -> 02:04:64:a4:39:d3 ARP 42 192.168.123.1 is at 02:04:64:a4:39:f2 14 111.246595 02:04:64:a4:39:f2 -> 02:04:64:a4:39:d3 ARP 42 Who has 192.168.123.50? Tell 192.168.123.1 15 111.247439 02:04:64:a4:39:d3 -> 02:04:64:a4:39:f2 ARP 60 192.168.123.50 is at 02:04:64:a4:39:d3 2092 5217.550877 02:04:64:a4:39:d3 -> 02:04:64:a4:39:f2 ARP 60 Who has 192.168.123.1? Tell 192.168.123.50 2093 5217.550911 02:04:64:a4:39:f2 -> 02:04:64:a4:39:d3 ARP 42 192.168.123.1 is at 02:04:64:a4:39:f2</pre> <p>Which would of course be insufficient to keep the DAT Cache fully up to date during the time a client is connected.</p>			

History

#1 - 07/25/2016 08:23 AM - Sven Eckelmann

- Assignee set to Linus Lüssing
- Subject changed from DAT Cache misses to Reduce DAT Cache misses
- Tracker changed from Bug to Feature

Looks like this ticket is just as reminder for Linus and not actually a problem description for any other developer. So just assigning it to Linus.

This is what he wrote on the mailing list:

On Dienstag, 19. Juli 2016 08:29:07 CEST Linus Lüssing wrote in

<https://lists.open-mesh.org/mailman3/hyperkitty/list/b.a.t.m.a.n@lists.open-mesh.org/message/FZ3FVN46IRICF55BPGLET6A2WJ7HFISO/>:

After the observations in ticket [#291](#) [0], Antonio and I had been discussing this topic a little further on IRC.

The Linux kernel seems to refresh its ARP table by snooping the IP traffic, too. But IP traffic does not create any ARP table entries.

What do you think about doing the same in batman-adv? That is, only updating, but not creating DAT cache entries from IP traffic. That should get rid of the main two concerns, accidentally snooping routed IP traffic and performance implications, shouldn't it?

On Dienstag, 19. Juli 2016 18:15:37 CEST Linus Lüssing wrote in

<https://lists.open-mesh.org/mailman3/hyperkitty/list/b.a.t.m.a.n@lists.open-mesh.org/message/5LC5XZJZU434UVPA46XKGYHS7ANKHCUY/>:

On Tue, Jul 19, 2016 at 09:06:38AM +0200, Ruben Wisniewski wrote:

What exactly is the target of it? Just refresh the timer for arp entries? Why don't just ask the kernel arp table if it knows something about the mac-address?

Yes, refreshing DAT cache entry timers more frequently was the idea.

There usually is no entry in a host's ARP table if bridges are involved, though. Which is the typical use-case with batman-adv.

#2 - 04/04/2017 05:34 AM - Linus Lüssing

Forwarding an interesting draft I just found out about. It might point to the right direction for our ARP issues:

<https://tools.ietf.org/html/draft-perkins-intarea-multicast-ieee802-02>

It seems to acknowledge the issue of many ARP requests from routers triggered from the WAN side. And that a missing ARP reply from the LAN is usually not cached, therefore every TCP retry for instance will probably cause another unanswered ARP request. The draft lists NAT as a mitigation strategy - which is used here but does not seem to fully solve the issue for us (probably NAT only helps against ARP issues triggered by port scans from the WAN to public IPs in the LAN).

My current guess is, that those are "zombie" NAT entries, entries for which hosts have already vanished from the mesh. Probably, the default NAT timeout of 5(!) days is not helpful either (/proc/sys/net/netfilter/nf_conntrack_tcp_timeout_established).

#3 - 05/22/2018 10:12 PM - Linus Lüssing

Further observations, explanations and potential patches:

Verified to have the desired effect, but NAK'd:

- [Noflood mark approach](#)

Explanation & analysis:

- https://www.open-mesh.org/projects/batman-adv/wiki/DAT_DHCP_Snooping

WIP attempt for a more elegant, more transparent solution:

- [batman-adv: Increase DHCP snooped DAT entry purge timeout in DHT](#)