Linux TCP Dump in Debug Shell to Troubleshoot Diameter Connection Issues

Contents

[Introduction](#)
[Background Information](#)
[Peering Issues](#)
[Packet Capture Collection](#)
[PCAP Analysis](#)

Introduction

This document describes how to use the TCP Dump in the StarOS debug shell to troubleshoot Diameter connection issues. Often cases are raised requesting assistance in troubleshooting why a Diameter connection will not come up or has gone down, even though (supposedly) no configuration or network changes have occurred. The diameter connection can fail to establish at the initial TCP/IP negotiation level, or at the Capabilities Exchange Request (CER) / Capabilities Exchange Answer (CEA) level.

Background Information

While there is no typical diameter peering issue, however, they do fall into a few categories:

- All the peers for a specific endpoint and/or protocol are down.
  <==== this example
- Peers for a specific port number are down.
- Peers tied to a specific PSC, DPC, or SF card are down.

Typically TCP port 3868 (default) is used for the Diameter server-side, though other ports can be specified as well, and confirmed to be different than 3868 in the configuration if the peer config lines have a port # specified at the end of the line.

Peering Issues

In the example here, the peers for endpoint 3gpp-aaa-s6b were reported down by **show diameter peer full all** and have no port # specified in the peer lines and so by default use port 3868, while the peers for Gy use a combination of 3868, 3869, and 3870 for the various peers.

show diameter peers all reports all the configured peers for all diameter endpoints. Here we see 6 peers configured and the associated configuration lines for 3gpp-aaa-s6b (broken) as well as for Gy (working), noting that Gy has some custom port #s:

```
diameter endpoint 3gpp-aaa-s6b

origin realm epc.mnc260.mcc310.3gppnetwork.org
```

```
use-proxy
origin host s6b.IEPCF201.epc.mnc260.mcc310.3gppnetwork.org address 10.168.86.144
max-outstanding 64
route-failure threshold 100
route-failure deadtime 600
route-failure recovery-threshold percent 50
dscp af31
peer mp2.daldra01.dra.epc.mnc260.mcc310.3gppnetwork.org realm epc.mnc260.mcc310.3gppnetwork.org
address 10.160.113.136
peer mp2.elgdra01.dra.epc.mnc260.mcc310.3gppnetwork.org realm epc.mnc260.mcc310.3gppnetwork.org
address 10.160.114.136
peer mp2.nvldra01.dra.epc.mnc260.mcc310.3gppnetwork.org realm epc.mnc260.mcc310.3gppnetwork.org
address 10.160.115.136
peer tsa06.draaro01.dra.epc.mnc260.mcc310.3gppnetwork.org realm
epc.mnc260.mcc310.3gppnetwork.org address 10.162.6.73
peer tsa06.drasyo01.dra.epc.mnc260.mcc310.3gppnetwork.org realm
epc.mnc260.mcc310.3gppnetwork.org address 10.164.57.41
peer tsa06.drawsc01.dra.epc.mnc260.mcc310.3gppnetwork.org realm
epc.mnc260.mcc310.3gppnetwork.org address 10.177.70.201
route-entry peer mp2.daldra01.dra.epc.mnc260.mcc310.3gppnetwork.org
route-entry peer mp2.elgdra01.dra.epc.mnc260.mcc310.3gppnetwork.org
route-entry peer mp2.nvldra01.dra.epc.mnc260.mcc310.3gppnetwork.org
route-entry peer tsa06.draaro01.dra.epc.mnc260.mcc310.3gppnetwork.org
route-entry peer tsa06.drasyo01.dra.epc.mnc260.mcc310.3gppnetwork.org
route-entry peer tsa06.drawsc01.dra.epc.mnc260.mcc310.3gppnetwork.org

#exit



[local]IEPCF201# show diameter peers all

Friday December 11 20:27:43 UTC 2020

Diameter Peer details

======================

-------------------------------------------------------------------------------

Context: billing              Endpoint: 3gpp-aaa-s6b

-------------------------------------------------------------------------------

Peer: mp2.daldra01.dra.epc.mnc260.mc Addr:Port 10.160.113.136:3868

Peer: mp2.elgdra01.dra.epc.mnc260.mc Addr:Port 10.160.114.136:3868

Peer: mp2.nvldra01.dra.epc.mnc260.mc Addr:Port 10.160.115.136:3868

Peer: tsa06.draaro01.dra.epc.mnc260. Addr:Port 10.162.6.73:3868

Peer: tsa06.drasyo01.dra.epc.mnc260. Addr:Port 10.164.57.41:3868

Peer: tsa06.drawsc01.dra.epc.mnc260. Addr:Port 10.177.70.201:3868

-------------------------------------------------------------------------------
```

```
diameter endpoint credit-control

origin realm starent.gy.com
use-proxy
origin host iepcf201.gy address 10.168.86.151
destination-host-avp always
route-failure threshold 100
route-failure deadtime 600
route-failure recovery-threshold percent 50
peer ln24.daldra01.dra.epc3.mnc260.mcc310.3gppnetwork.org realm nsn-gy address 10.160.113.136
port 3869
peer ln24.drawsc01.dra.epc3.mnc260.mcc310.3gppnetwork.org realm nsn-gy address 10.177.70.201
port 3870
peer tsa05.drachr01.dra.epc3.mnc260.mcc310.3gppnetwork.org realm nsn-gy address 10.164.144.88
peer tsa05.draphx01.dra.epc3.mnc260.mcc310.3gppnetwork.org realm nsn-gy address 10.198.93.88
peer tsa05.drapol01.dra.epc3.mnc260.mcc310.3gppnetwork.org realm nsn-gy address 10.182.16.88
peer tsa06.drachr01.dra.epc3.mnc260.mcc310.3gppnetwork.org realm nsn-gy address 10.164.144.89
peer tsa06.draphx01.dra.epc3.mnc260.mcc310.3gppnetwork.org realm nsn-gy address 10.198.93.89
peer tsa06.drapol01.dra.epc3.mnc260.mcc310.3gppnetwork.org realm nsn-gy address 10.182.16.89
route-entry peer ln24.drawsc01.dra.epc3.mnc260.mcc310.3gppnetwork.org weight 20
route-entry peer ln24.daldra01.dra.epc3.mnc260.mcc310.3gppnetwork.org
route-entry peer tsa05.drapol01.dra.epc3.mnc260.mcc310.3gppnetwork.org
route-entry peer tsa06.drapol01.dra.epc3.mnc260.mcc310.3gppnetwork.org
route-entry peer tsa05.drachr01.dra.epc3.mnc260.mcc310.3gppnetwork.org weight 5
route-entry peer tsa05.draphx01.dra.epc3.mnc260.mcc310.3gppnetwork.org weight 5
route-entry peer tsa06.drachr01.dra.epc3.mnc260.mcc310.3gppnetwork.org weight 5
route-entry peer tsa06.draphx01.dra.epc3.mnc260.mcc310.3gppnetwork.org weight 5

#exit
```

Also worth noting, for most setups, the use-proxy configurable is specified to setup peering on the ASR side to use the diamproxy process running on all of the active cards, for example, this is a vPC-DI where the cards are called Service Function Cards.

```
[local]IEPCF201# show task resources facility diamproxy all
Friday December 11 20:34:37 UTC 2020
                    task   cputime        memory     files      sessions
cpu facility      inst  used  allc   used  alloc used allc  used  allc S status
---------------------- ----------- ------------- --------- ------------- ------
3/0 diamproxy        5 0.12%   90% 41.62M 250.0M   38 2500    --    -- -   good
5/0 diamproxy        2 0.11%   90% 41.63M 250.0M   51 2500    --    -- -   good
6/0 diamproxy        6 0.13%   90% 41.62M 250.0M   35 2500    --    -- -   good
7/0 diamproxy        3 0.12%   90% 41.64M 250.0M   34 2500    --    -- -   good
8/0 diamproxy        4 0.13%   90% 41.65M 250.0M   34 2500    --    -- -   good
10/0 diamproxy       1 0.10%   90% 41.64M 250.0M   49 2500    --    -- -   good
Total                6  0.71%     249.8M            241           0
[local]IEPCF201#
```

Here **show diameter peers full all** is taken from the show support details captures the fact that the Diameter peers for the 3gpp-aaa-s6b endpoint are all down. Note that this is a special debug version of the **show diameter peers full** command taken from the show support details (SSD) and so it also shows all peer connections to the **aaamgr** processes (not showing the output here) and so the final count of connections is much higher than if this was run normally, but shown at the bottom is the summary output as if it were run normally with the fewer number of connections (144). The FULL output is attached to this article, and so just the connections for one peer (but with all 6 diamproxies) are shown for brevity.

Also shown is an example of one open working connection for the Gy endpoints, where you can see an extra field called **Local Address** that captures the connection being up on the ASR side, whereas on the broken 3gpp-aaa-s6b peers that field does not exist. (Later shown is the output after the issue was fixed by the customer for the 3gpp-aaa-s6b peer where that **Local Address** is

included.)

```
******** show diameter peers full *******
Sunday December 13 15:19:00 UTC 2020
--------------------------------------------------------------------------------
Context: billing Endpoint: 3gpp-aaa-s6b
--------------------------------------------------------------------------------

Peer Hostname: mp2.daldra01.dra.epc.mnc260.mcc310.3gppnetwork.org
Local Hostname: 0001-diamproxy.s6b.IEPCF201.epc.mnc260.mcc310.3gppnetwork.org
Peer Realm: epc.mnc260.mcc310.3gppnetwork.org
Local Realm: epc.mnc260.mcc310.3gppnetwork.org
Peer Address: 10.160.113.136:3868
State: IDLE [TCP]
CPU: 10/0 Task: diamproxy-1
Messages Out/Queued: 0/0
Supported Vendor IDs: None
Admin Status: Enable
DPR Disconnect: N/A
Peer Backoff Timer running:N/A



Peer Hostname: mp2.daldra01.dra.epc.mnc260.mcc310.3gppnetwork.org
Local Hostname: 0002-diamproxy.s6b.IEPCF201.epc.mnc260.mcc310.3gppnetwork.org
Peer Realm: epc.mnc260.mcc310.3gppnetwork.org
Local Realm: epc.mnc260.mcc310.3gppnetwork.org
Peer Address: 10.160.113.136:3868
State: IDLE [TCP]
CPU: 5/0 Task: diamproxy-2
Messages Out/Queued: 0/0
Supported Vendor IDs: None
Admin Status: Enable
DPR Disconnect: N/A
Peer Backoff Timer running:N/A



Peer Hostname: mp2.daldra01.dra.epc.mnc260.mcc310.3gppnetwork.org
Local Hostname: 0003-diamproxy.s6b.IEPCF201.epc.mnc260.mcc310.3gppnetwork.org
Peer Realm: epc.mnc260.mcc310.3gppnetwork.org
Local Realm: epc.mnc260.mcc310.3gppnetwork.org
Peer Address: 10.160.113.136:3868
State: IDLE [TCP]
CPU: 7/0 Task: diamproxy-3
Messages Out/Queued: 0/0
Supported Vendor IDs: None
Admin Status: Enable
DPR Disconnect: N/A
Peer Backoff Timer running:N/A



Peer Hostname: mp2.daldra01.dra.epc.mnc260.mcc310.3gppnetwork.org
Local Hostname: 0004-diamproxy.s6b.IEPCF201.epc.mnc260.mcc310.3gppnetwork.org
Peer Realm: epc.mnc260.mcc310.3gppnetwork.org
Local Realm: epc.mnc260.mcc310.3gppnetwork.org
Peer Address: 10.160.113.136:3868
State: IDLE [TCP]
CPU: 8/0 Task: diamproxy-4
Messages Out/Queued: 0/0
Supported Vendor IDs: None
```

```
Admin Status: Enable
DPR Disconnect: N/A
Peer Backoff Timer running:N/A



Peer Hostname: mp2.daldra01.dra.epc.mnc260.mcc310.3gppnetwork.org
Local Hostname: 0005-diamproxy.s6b.IEPCF201.epc.mnc260.mcc310.3gppnetwork.org
Peer Realm: epc.mnc260.mcc310.3gppnetwork.org
Local Realm: epc.mnc260.mcc310.3gppnetwork.org
Peer Address: 10.160.113.136:3868
State: IDLE [TCP]
CPU: 3/0 Task: diamproxy-5
Messages Out/Queued: 0/0
Supported Vendor IDs: None
Admin Status: Enable
DPR Disconnect: N/A
Peer Backoff Timer running:N/A



Peer Hostname: mp2.daldra01.dra.epc.mnc260.mcc310.3gppnetwork.org
Local Hostname: 0006-diamproxy.s6b.IEPCF201.epc.mnc260.mcc310.3gppnetwork.org
Peer Realm: epc.mnc260.mcc310.3gppnetwork.org
Local Realm: epc.mnc260.mcc310.3gppnetwork.org
Peer Address: 10.160.113.136:3868
State: IDLE [TCP]
CPU: 6/0 Task: diamproxy-6
Messages Out/Queued: 0/0
Supported Vendor IDs: None
Admin Status: Enable
DPR Disconnect: N/A
Peer Backoff Timer running:N/A

...

--------------------------------------------------------------------------------
Context: billing Endpoint: credit-control
--------------------------------------------------------------------------------

...

Peer Hostname: ln24.daldra01.dra.epc3.mnc260.mcc310.3gppnetwork.org
Local Hostname: 0001-diamproxy.iepcf201.gy
Peer Realm: nsn-gy
Local Realm: starent.gy.com
Peer Address: 10.160.113.136:3869
Local Address: 10.168.86.151:55584
State: OPEN [TCP]
CPU: 10/0 Task: diamproxy-1
Messages Out/Queued: 0/0
Supported Vendor IDs: 10415
Admin Status: Enable
DPR Disconnect: N/A
Peer Backoff Timer running:N/A

Peers Summary:
Peers in OPEN state: 1404
Peers in CLOSED state: 468
Peers in intermediate state: 0
Total peers matching specified criteria: 1872
```

For reference, here is the normal output from this command showing the connection count without the aaamgrs:

```
Peers Summary:
Peers in OPEN state: 107
Peers in CLOSED state: 36
Peers in intermediate state: 1
Total peers matching specified criteria: 144
```

Packet Capture Collection

As discussed, this scenario shows that ALL the diameter peers are down for the s6b endpoint, the issue is NOT for a specific diamproxy/card, which means that PCAP collection for any of the cards should suitably represent the issue for troubleshooting purposes. IF the issue was only seen on a specific diamproxy, then it would be more important to capture a PCAP for that process. This is important because the process of collection requires specifying a specific card - it can't be run across all cards with a single capture - and though in this scenario the issue is indeed seen across all cards, shown below are captures taken on two cards to help make some points on how to analyze the resulting data.

The first thing to do is look at the card table and pick out a couple of ACTIVE cards (3 and 5) on which to run the capture, as well as noting which is the Demux card which should not be specified.

```
[local]IEPCF201# show card table
Friday December 11 17:15:28 UTC 2020
Slot         Card Type                                Oper State     SPOF  Attach
-----------  ---------------------------------------  -------------  ----  ------
 1: CFC      Control Function Virtual Card            Active         No
 2: CFC      Control Function Virtual Card            Standby        -
 3: FC       4-Port Service Function Virtual Card     Active         No       <=====
 4: FC       4-Port Service Function Virtual Card     Standby        -
 5: FC       4-Port Service Function Virtual Card     Active         No       <=====
 6: FC       4-Port Service Function Virtual Card     Active         No
 7: FC       4-Port Service Function Virtual Card     Active         No
 8: FC       4-Port Service Function Virtual Card     Active         No
 9: FC       4-Port Service Function Virtual Card     Active         No
10: FC       4-Port Service Function Virtual Card     Active         No
[local]IEPCF201#

[local]IEPCF201# show session recovery status verbose
Saturday December 12 21:43:11 UTC 2020
Session Recovery Status:
  Overall Status        : Ready For Recovery
  Last Status Update    : 4 seconds ago

              ----sessmgr---  ----aaamgr----  demux
 cpu state    active standby  active standby  active  status
---- -------  ------ -------  ------ -------  ------  -------------------------
 3/0 Active   12     1        12     1        0       Good
 4/0 Standby  0      12       0      12       0       Good
 5/0 Active   12     1        12     1        0       Good
 6/0 Active   12     1        12     1        0       Good
 7/0 Active   12     1        12     1        0       Good
 8/0 Active   12     1        12     1        0       Good
 9/0 Active   0      0        0      0        8       Good (Demux)
10/0 Active   12     1        12     1        0       Good
[local]IEPCF201#
```

Also, the context # where the diameter peers are defined needs to be retrieved, in this case, the billing context is #2.

```
 ******** show context *******
Sunday December 13 15:14:24 UTC 2020
Context Name    ContextID    State     Description
--------------- --------- ----------   ----------------------
local           1            Active
billing         2            Active    <==========
calea           3            Active
gi              4            Active
sgw             5            Active
```

Next is to log into the Linux debug shell for the cards where the PCAP is to be collected, in this case, cards 3 and 5, in their own CLI session:

> **Note**: Access to the debug shell is not something that most operators will likely have access to unless they have been told the password which is specific to the chassis/customer depending on how it was setup. Take caution when logging into the debug shell, as it is logging into the underlying operating system of the card (PSC or DPC of ASR 5000 or ASR 5500) or virtual machine (Service Function (SF) of vPC-DI).

```
[local]IEPCF201# cli test password <password>
Saturday December 12 21:43:54 UTC 2020
Warning: Test commands enables internal testing and debugging commands
         USE OF THIS MODE MAY CAUSE SIGNIFICANT SERVICE INTERRUPTION
[local]IEPCF201#
[local]IEPCF201# debug shell card 3 cpu 0
Saturday December 12 21:44:02 UTC 2020
Last login: Fri Dec 11 19:26:34 +0000 2020 on pts/1 from card1-cpu0.
qvpc-di:card3-cpu0#
```

Now run a special Linux command **setvr** (set virtual router) only available in this customized StarOS version of Linux, specifying the context # retrieved earlier. Note that the prompt changes:

```
qvpc-di:card3-cpu0# setvr 2 bash
bash-2.05b#
```

At this point, the TCP dump can be run using the parameters as follows. Note that if the port number is different as in the example shown earlier for gy, then that port number should be used. Also, a host IP address can be specified with **host <host ip address>** if there is a specific peer address for which to capture packets. Run the command for a couple of minutes, and stop the capture with Control-C. IF packets are captured, the number of packets are displayed.

```
bash-2.05b# tcpdump -i any -s 0 -w /tmp/diameter_SF3.pcap "port 3868"
tcpdump: listening on any
^C
1458 packets received by filter
0 packets dropped by kernel
bash-2.05b#
```

Next, exit the virtual router space with the exit command, and then copy the file to the active management card's flash, which for ASR 5500 would be MIO 5 or 6, or in the case here for vPC-DI, 1 or 2.

```
bash-2.05b# exit
exit
qvpc-di:card3-cpu0# scp /tmp/diameter_SF3.pcap card1:/flash/sftp/diameter_SF3.pcap
diameter_SF3.pcap            100%  110KB 110.4KB/s   00:00
qvpc-di:card3-cpu0# exit
[local]IEPCF201#
```

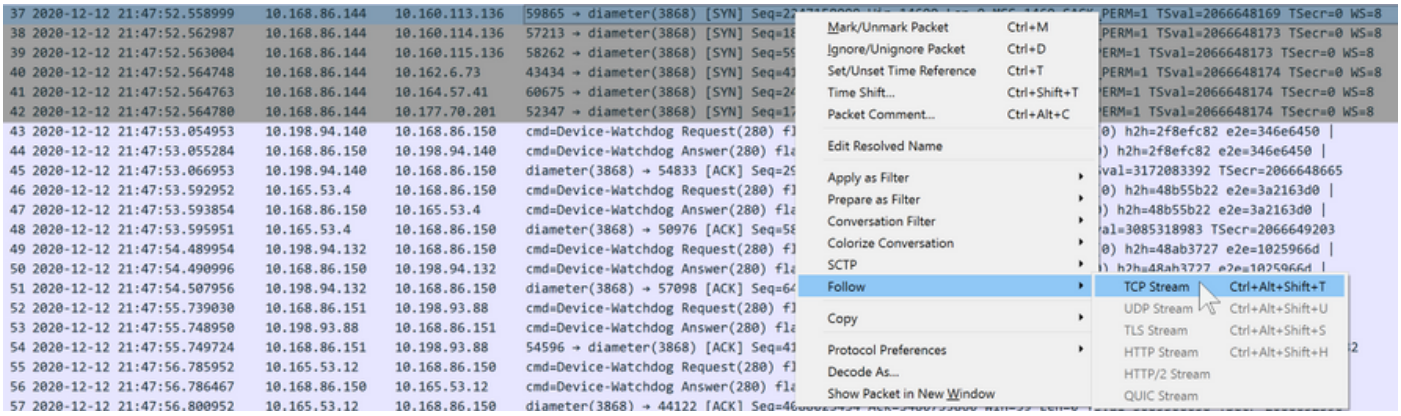
At that point, the file can be retrieved with sftp using whatever means exist within the network to reach the /flash directory.

Here are the commands for SF 5 as well, which is a repeat of what was just shown for SF 3. Ideally, run both sessions at the same time in order to have simultaneous captures for analysis (though this may not be necessary).

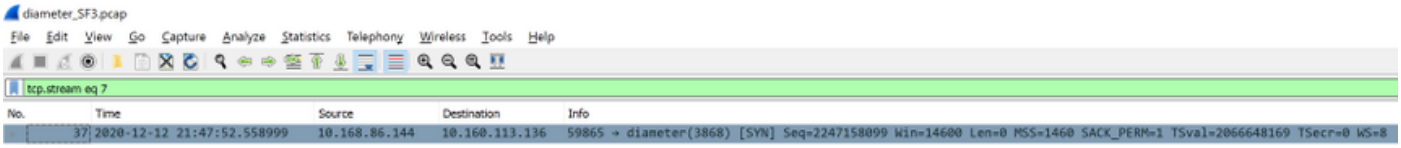
```
[local]IEPCF201# cli test password <password>
Saturday December 12 21:43:28 UTC 2020
Warning: Test commands enables internal testing and debugging commands
         USE OF THIS MODE MAY CAUSE SIGNIFICANT SERVICE INTERRUPTION
[local]IEPCF201# debug shell card 5 cpu 0
Saturday December 12 21:44:13 UTC 2020
qvpc-di:card5-cpu0#
qvpc-di:card5-cpu0# setvr 2 bash
bash-2.05b#  tcpdump -i any -s 0 -w /tmp/diameter_SF5.pcap "port 3868"
tcpdump: listening on any
^C
1488 packets received by filter
0 packets dropped by kernel
bash-2.05b# exit
exit
qvpc-di:card5-cpu0# scp /tmp/diameter_SF5.pcap card1:/flash/sftp/diameter_SF5.pcap
diameter_SF5.pcap           100%  113KB 112.7KB/s   00:00
qvpc-di:card5-cpu0# exit
[local]IEPCF201#
```

PCAP Analysis

The goal here is to determine where the breakdown is in the diameter connection establishment process. As mentioned earlier, it could be in the TCP/IP connection or it could be at the ensuing CER/CEA step. For TCP/IP, look to see if a TCP SYN is being sent, and if a TCP SYN ACK is being received, followed by an ACK sent from ASR. Packets can be filtered with any number of filters to help with analysis and in this case the filter tcp.flags.syn == 1 shows that the SYN is being sent for all 6 peers for this particular card. Looking at an unfiltered view, right-click a SYN packet and take advantage of the TCP stream feature in Wireshark which aggregates all TCP packets that are using the same TCP port #, by choosing Follow ... TCP Stream to see if there is a corresponding exchange of TCP packets that establish the connection.

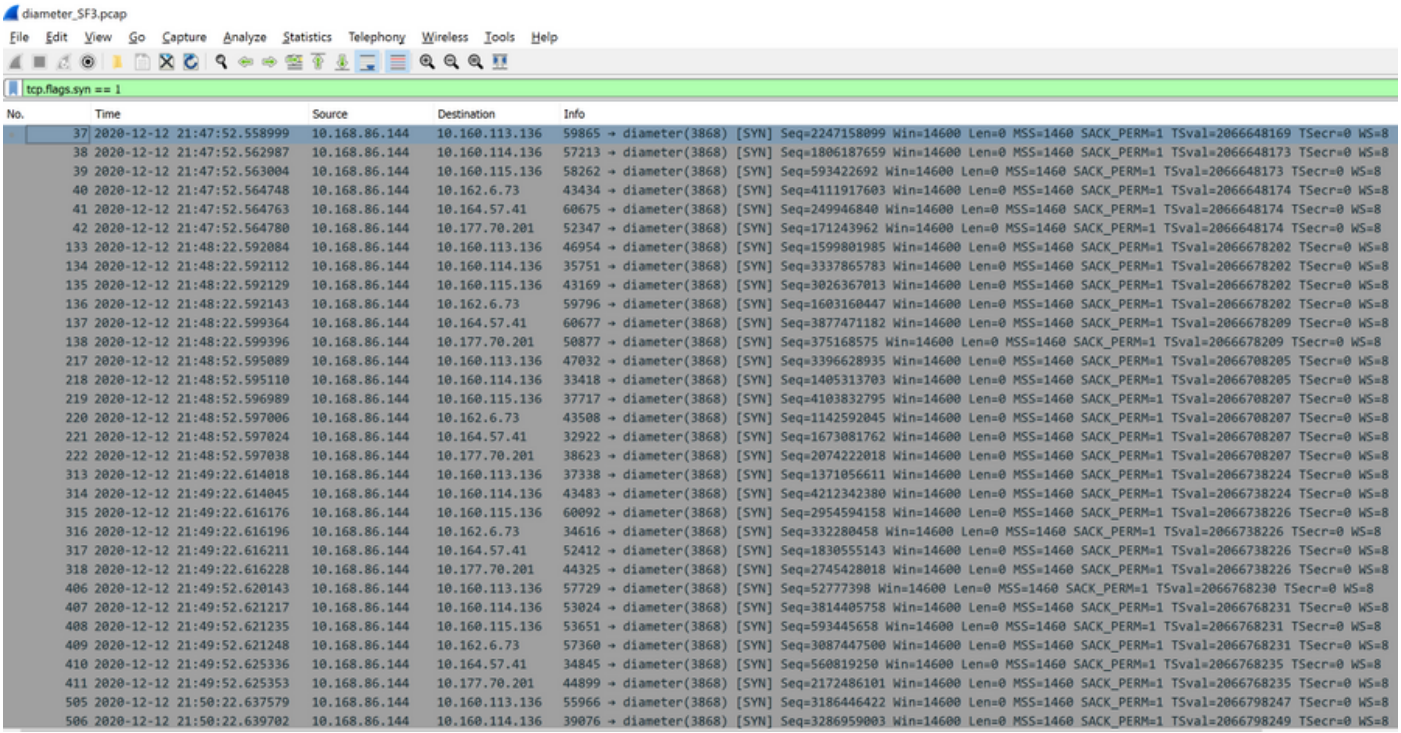


In this scenario, note that there are NO further packets beyond the SYN, and this confirms that the ASR is likely sending a SYN but not getting back any response, which would eliminate the ASR from being the cause of the failure to setup the connection (though this is not guaranteed to be the case, possibly the packet is not being sent, or that the response is being dropped, in which case an external PCAP would be helpful in further narrowing down the issue).



Also worth noting is that the pattern is being repeated every 30 seconds, which matches the default configuration for the diameter endpoint of 30 seconds to retry the connection - the ASR is not giving up but rather will retry forever until successful. The PCAP for SF 5 shows the exact same behavior.

```
context billing
    diameter endpoint 3gpp-aaa-s6b
      connection timeout 30
      connection retry-timeout 30
```



Tying things together, the diameter base statistics show that the number of failed connections is

incrementing at a rate commensurate with the number of SF/diamproxies and the retry timeout. The math is the following: 6 peers * 6 diamproxies = 36 attempts every 30 seconds. So, over a minute that would be 72 attempts, and this can be seen by running **show diameter statistics proxy** and looking at **Connection Timeouts** incrementing from 60984 to 61056 = 72 over a minute period as shown by the CLI timestamps.

```
[local]IEPCF201# show diameter statistics proxy | grep "Connection Timeouts"
Friday December 11 20:39:10 UTC 2020
  Connection Timeouts:                                          60984
[local]IEPCF201# show diameter statistics proxy | grep "Connection Timeouts"
Friday December 11 20:39:12 UTC 2020
  Connection Timeouts:                                          60984
[local]IEPCF201# show diameter statistics proxy | grep "Connection Timeouts"
Friday December 11 20:39:14 UTC 2020
  Connection Timeouts:                                          60984
[local]IEPCF201# show diameter statistics proxy | grep "Connection Timeouts"
Friday December 11 20:39:17 UTC 2020
  Connection Timeouts:                                          60990
[local]IEPCF201# show diameter statistics proxy | grep "Connection Timeouts"
Friday December 11 20:39:19 UTC 2020
  Connection Timeouts:                                          60990
[local]IEPCF201# show diameter statistics proxy | grep "Connection Timeouts"
Friday December 11 20:39:21 UTC 2020
  Connection Timeouts:                                          60996
[local]IEPCF201# show diameter statistics proxy | grep "Connection Timeouts"
Friday December 11 20:39:25 UTC 2020
  Connection Timeouts:                                          61002
[local]IEPCF201# show diameter statistics proxy | grep "Connection Timeouts"
Friday December 11 20:39:27 UTC 2020
  Connection Timeouts:                                          61002
[local]IEPCF201# show diameter statistics proxy | grep "Connection Timeouts"
Friday December 11 20:39:29 UTC 2020
  Connection Timeouts:                                          61008
[local]IEPCF201# show diameter statistics proxy | grep "Connection Timeouts"
Friday December 11 20:39:32 UTC 2020
   Connection Timeouts:                                          61014
[local]IEPCF201# show diameter statistics proxy | grep "Connection Timeouts"
Friday December 11 20:39:35 UTC 2020
  Connection Timeouts:                                          61014
[local]IEPCF201# show diameter statistics proxy | grep "Connection Timeouts"
Friday December 11 20:39:37 UTC 2020
  Connection Timeouts:                                          61020
[local]IEPCF201# show diameter statistics proxy | grep "Connection Timeouts"
Friday December 11 20:39:40 UTC 2020
  Connection Timeouts:                                          61020
[local]IEPCF201# show diameter statistics proxy | grep "Connection Timeouts"
Friday December 11 20:39:43 UTC 2020
  Connection Timeouts:                                          61020
[local]IEPCF201# show diameter statistics proxy | grep "Connection Timeouts"
Friday December 11 20:39:45 UTC 2020
  Connection Timeouts:                                          61026
[local]IEPCF201# show diameter statistics proxy | grep "Connection Timeouts"
Friday December 11 20:39:47 UTC 2020
  Connection Timeouts:                                          61026
[local]IEPCF201# show diameter statistics proxy | grep "Connection Timeouts"
Friday December 11 20:39:50 UTC 2020
   Connection Timeouts:                                          61038
[local]IEPCF201# show diameter statistics proxy | grep "Connection Timeouts"
Friday December 11 20:39:56 UTC 2020
  Connection Timeouts:                                          61038
[local]IEPCF201# show diameter statistics proxy | grep "Connection Timeouts"
```

```
Friday December 11 20:39:58 UTC 2020
  Connection Timeouts:                                          61044
[local]IEPCF201# show diameter statistics proxy | grep "Connection Timeouts"
Friday December 11 20:40:01 UTC 2020
  Connection Timeouts:                                          61044
[local]IEPCF201# show diameter statistics proxy | grep "Connection Timeouts"
Friday December 11 20:40:03 UTC 2020
  Connection Timeouts:                                          61050
[local]IEPCF201# show diameter statistics proxy | grep "Connection Timeouts"
Friday December 11 20:40:05 UTC 2020
  Connection Timeouts:                                          61056
[local]IEPCF201# show diameter statistics proxy | grep "Connection Timeouts"
Friday December 11 20:40:07 UTC 2020
  Connection Timeouts:                                          61056
[local]IEPCF201# show diameter statistics proxy | grep "Connection Timeouts"
Friday December 11 20:40:09 UTC 2020
  Connection Timeouts:                                          61056
[local]IEPCF201# show diameter statistics proxy | grep "Connection Timeouts"
Friday December 11 20:40:12 UTC 2020
  Connection Timeouts:                                          61056
[local]IEPCF201# show diameter statistics proxy | grep "Connection Timeouts"
Friday December 11 20:40:14 UTC 2020
  Connection Timeouts:                                          61056
[local]IEPCF201# show diameter statistics proxy | grep "Connection Timeouts"
Friday December 11 20:40:16 UTC 2020
  Connection Timeouts:                                          61062
[local]IEPCF201# show diameter statistics proxy | grep "Connection Timeouts"
Friday December 11 20:40:18 UTC 2020
  Connection Timeouts:                                          61062
[local]IEPCF201# show diameter statistics proxy | grep "Connection Timeouts"
Friday December 11 20:40:20 UTC 2020
  Connection Timeouts:                                          61068
[local]IEPCF201# show diameter statistics proxy | grep "Connection Timeouts"
Friday December 11 20:40:22 UTC 2020
  Connection Timeouts:                                          61074
[local]IEPCF201# show diameter statistics proxy | grep "Connection Timeouts"
Friday December 11 20:40:25 UTC 2020
  Connection Timeouts:                                          61074
[local]IEPCF201# show diameter statistics proxy | grep "Connection Timeouts"
Friday December 11 20:40:27 UTC 2020
  Connection Timeouts:                                          61074
[local]IEPCF201#
```

Also note that the number of CER/CEAs (across all diameter peers) is trivial, which proves that it never gets to the point of trying to exchange these packets, which means it's a TCP/IP setup problem.

```
[local]IEPCF201# show diameter statistics proxy
Friday December 11 20:57:09 UTC 2020
...
Capabilities Exchange Requests and Answers statistics:
  Connection CER sent:                                            109
  Connection CER send errors:                                       0
  CERs received:                                                    0
  Connection CER create failures:                                   0
  CEAs received:                                                  108
  CEA AVPs unknown:                                                 0
  CEA Application ID mismatch:                                      0
  Read CEA Messages:                                              108
  Read CEA Messages Unexpected:                                     0
  Read CEA Missing:                                                 0
  Read CEA Negotiation Failure:                                     0
  Read CER Messages:                                                0
```

```
  Read CER Messages Unexpected:                                    0
  Read CER Missing:                                                0
  Tw Expire Waiting for CEA:                                       0
```

Finally note that after the issue was resolved by the customer, the Peers in CLOSED State goes back to 0 and the **Local Address** field shows up in **show diameter peers full all output**.

```
 Peer Hostname: mp1.daldra01.dra.epc.mnc260.mcc310.3gppnetwork.org
  Local Hostname: 0001-diamproxy.s6b.IEPCF201.epc.mnc260.mcc310.3gppnetwork.org
  Peer Realm: epc.mnc260.mcc310.3gppnetwork.org
  Local Realm: epc.mnc260.mcc310.3gppnetwork.org
  Peer Address: 10.160.113.133:3868
  Local Address: 10.168.86.144:32852
  State: OPEN [TCP]
  CPU: 10/0                               Task: diamproxy-1
  Messages Out/Queued: 0/0
  Supported Vendor IDs: None
  Admin Status: Enable
  DPR Disconnect: N/A
  Peer Backoff Timer running:N/A


 Peers Summary:
  Peers in OPEN state: 144
  Peers in CLOSED state: 0
  Peers in intermediate state: 0
  Total peers matching specified criteria: 144
[local]IEPCF101#
```