

How many sockets can STCP really support and how do you determine the number currently in use?

by Noah Davids

There is a fair amount of confusion regarding how many TCP sockets STCP can really support and how to count them. This article addresses that confusion. This is for VOS users running some version of VOS 14.7, 15.1 or 15.2.

The number of TCP sockets that can be created at any one time is based on several different limits and on which of these limits you hit first.

The first limit is the size of the `tc_entry` array. There are 16130 entries in this array and once they are all in use no more TCP sockets can be created. Until STCP changes the way that it searches its internal data structures it is not feasible to increase the size of this array, so 16130 is the absolute maximum number of TCP sockets that you can have.

The second limit is the `stp.mN` clone limit. This is the `stp` device that is created in the `devices.tin` file. In release 14.6.1 this limit was raised from 8120 to 16000, if you upgraded from a pre-14.6.1 version of VOS you may still be using a limit of 8120 or even 5120 which was the clone limit STCP originally (VOS release 14.1) shipped with. Almost every time a TCP socket is created, a clone of this device is also created. Once you have reached the clone limit no more TCP sockets can be created – even if there are free entries in the `tc_entry` array.

Note that in the previous paragraph I said “almost every time”. When the `telnetd` process accepts a connection it first creates a clone but then after some processing it frees the clone. The TCP socket still takes an entry in the `tcb_entry` array but does not use a clone. The same is true of the sockets used in incoming and outgoing OSL connections. But NOT the OSL listening sockets. What this means is that typically the number of used `tc_entries` will be greater than the number of `stp.mN` device clones. If you have more than 130 OSL and `telnetd` connections you will not be able to create all 16000 `stp.mN` device clones.

If you have a clone limit below 16000 you can increase it by

1. edit the `devices.tin` file to change the clone limit
2. create a new `devices.table` file
3. move the `devices.table` file to `>system`
4. reboot

Actually, all you have to do is stop the STCP stack, unload the STCP driver and do a “`configure_devices -flush`”. Then load the driver back up and restart the STCP stack. In my opinion it is easier to reboot, especially on a VOS release 15-based system.

The next limit is the total number of devices that the system is configured to support. You can see this by displaying the value of `max_local_devices$` in `analyze_system`. The value is a short and the default is 4000x or 16384. The current number of devices is stored in the variable `configured_local_devices$`. Remember that when the system boots every

entry in the devices table for the module will count toward the maximum. If, when a socket is created an stcp.mN clone is also created, then configured_local_devices\$ is incremented by 1. When the clone is freed the value is decremented by 1. If the number of devices defined when the system is booted is X you will not be able to create more than 16384 -X sockets. Note that this holds true if the sockets are used by OSL or telnetd since an stcp.mN clone is initially created for those sockets. If the clone cannot be created you cannot create the socket.

You can raise the maximum number of devices just by patching the max_local_devices\$ with the analyze_system request set_word. You can do this at any point after the system is booted (see figure 1).

```
analyze_system -request_line 'set_word max_local_devices$ 4e20' -quit
VOS Release 14.7.2am, analyze_system Release 14.7.2am
Current process is 4055, ptep C34C84C0, Noah_Davids.CAC
addr      from  to
C094250C  4000  4E20
```

Figure 1 – increasing maximum number of devices supported on a module

The final limitation is the amount of memory available to the STREAMS subsystem. When a socket is created some amount of memory is needed. The amount used is added to the amount used by the Streams subsystem. If that amount exceeds certain thresholds you will not be able to create the socket. There are three thresholds LO, MED, HI. Socket creation checks both the HI and MED thresholds so if your Streams memory usage exceeds the MED threshold you will not be able to create a socket.

You can increase the amount of memory available to the STREAMS subsystem by changing the sys_denominator parameters with the analyze_system request set_streams_param. The default value is 8 which corresponds to 1/8 of system memory. Changing the value to 4 will double the available amount of streams memory. Like max_local_devices\$, you can make this change any time after the system is booted.

```
analyze_system -request_line 'set_streams_param sys_denominator 4' -
quit
VOS Release 14.7.2am, analyze_system Release 14.7.2am
Current process is 4055, ptep C34C84C0, Noah_Davids.CAC

Changing sys max heap denomin (sys_denominator)
from 8 to 4
```

Figure 2 – increases the amount of memory available to the STREAMS subsystem

The following macro, tcp_socket_count displays the currently used number of entries in the tc_entry array, the number of stcp.mN clone devices and just calls netstat and counts the number of TCP sockets displayed. It also displays the current streams memory usage as a percentage of the LO, MED and HI thresholds. The LO threshold because if you have exceeded the LO threshold you will have problems moving data through STREAMS even if you can still create sockets.

```
tcp_socket_count
VOS Release 14.7.2am, analyze_system Release 14.7.2am
Current process is 4055, ptep C34C84C0, Noah_Davids.CAC
as: as: as: as: as: as:
```

```
The number of tc_entries at 06-11-09 13:04:27 mst is 72 out of 5130
The stcp.m14 clone count at 06-11-09 13:04:29 mst is 53 out of 8120
TCP sockets from netstat at 06-11-09 13:04:29 mst is 70
The number of devices at 06-11-09 13:04:30 mst is 151 out of 16384
0.46% of LO Streams memory threshold in use as of 06-11-09 13:04:30 mst
0.42% of MED Streams memory threshold in use as of 06-11-09 13:04:30 mst
0.402% of HI Streams memory threshold in use as of 06-11-09 13:04:30 mst
```

Figure 3 – output from the tcp_socket_count command macro

```
& socket_count begins here
&
& socket_count.cm
& version 2.0 06-11-01
& Noah Davids Stratus CAC noah.davids@stratus.com
&
&attach_input
&set START (calc (length (system_name)) + 3)
&set_string STCP_DEVICE stcp.(substr (current_module) &START&)
&
analyze_system
..attach_default_output (process_dir)>tc_entries
match tcb; dump_tcentry
..detach_default_output
..attach_default_output (process_dir)>ntcpcon
d ntcpcon
..detach_default_output
..attach_default_output (process_dir)>clones
match clone_count; dump_dvt -name &STCP_DEVICE&
match clone_limit; dump_dvt -name &STCP_DEVICE&
..detach_default_output
..attach_default_output (process_dir)>devices
d configured_local_devices$
d max_local_devices$
..detach_default_output
..attach_default_output (process_dir)>pmm
match 'pmm_limit' ; dump_stream -stm_msg
match 'hq_pool[0]->pmm_total_size' ; dump_stream -stm_msg
match 'hq_pool[0]->pmm_allocated_size' ; dump_stream -stm_msg
..detach_default_output
quit
attach_default_output (process_dir)>netstat
>system>stcp>command_library>netstat -numeric -all_sockets -protocol tcp
detach_default_output
&
attach_default_output (process_dir)>tc_entries2
display_file_status (process_dir)>tc_entries
detach_default_output
&set records (substr (contents (process_dir)>tc_entries2 23) 27)
&
```

```

display_line
display_line
&set NTCPCON (calc (substr (contents (process_dir)>ntcpcon 1) 19 8)x)
display_line The number of tc_entries at &+
(file_info (process_dir)>tc_entries date_used) is (calc &records& - 2) &+
out of &NTCPCON&
&
&set CLONE_COUNT (substr (contents (process_dir)>clones 1) 25)
&set CLONE_LIMIT (substr (contents (process_dir)>clones 2) 25)
display_line The &STCP_DEVICE& clone count at &+
(file_info (process_dir)>clones date_used) is &CLONE_COUNT& &+
out of &CLONE_LIMIT&
&
attach_default_output (process_dir)>netstat2
display_file_status (process_dir)>netstat
detach_default_output
&set records (substr (contents (process_dir)>netstat2 23) 27)
&if &records& = 0
&then display_line No info from netstat, probably no sockets available
&else display_line TCP sockets from netstat at &+
(file_info (process_dir)>netstat date_used) is (calc &records& - 2)
&
&set CONFIG_DEVICES (calc (substr (contents (process_dir)>devices 1) 19 4)x)
&set MAX_DEVICES (calc (substr (contents (process_dir)>devices 2) 19 4)x)
display_line The number of devices at ' ' &+
(file_info (process_dir)>devices date_used) is &CONFIG_DEVICES& &+
out of &MAX_DEVICES&
&
&set ALLOC_MEM (substr (contents (process_dir)>pmm 6) 46)
&set TOTAL_MEM (substr (contents (process_dir)>pmm 5) 46)
&set LO_MEM (substr (contents (process_dir)>pmm 2) 41)
&set MED_MEM (substr (contents (process_dir)>pmm 3) 41)
&set HI_MEM (substr (contents (process_dir)>pmm 4) 41)
&set LO_PER (substr (calc &ALLOC_MEM& / &LO_MEM& * 100) 1 5)
&set MED_PER (substr (calc &ALLOC_MEM& / &MED_MEM& * 100) 1 5)
&set HI_PER (substr (calc &ALLOC_MEM& / &HI_MEM& * 100) 1 5)
display_line &LO_PER&% of LO Streams memory &+
threshold in use as of (file_info (process_dir)>pmm date_used)
display_line &MED_PER&% of MED Streams memory &+
threshold in use as of (file_info (process_dir)>pmm date_used)
display_line &HI_PER&% of HI Streams memory &+
threshold in use as of (file_info (process_dir)>pmm date_used)
&
& socket_count ends here

```