



[Home](#) | [UNC](#) | [CS Dept.](#) | [Rose-Hulman](#) | [Bandar](#) | [Contact](#)

[Research](#)

[Courses](#)

[Schedule](#)

[Resume](#)

[Biography](#)

[T-Files](#)

How To: Network / TCP / UDP Tuning

This is a very basic step by step description of how to improve the performance networking (TCP & UDP) on Linux 2.4+ for high-bandwidth applications. These settings are especially important for GigE links. Jump to [Quick Step](#) or [All The Steps](#).

Assumptions

This howto assumes that the machine being tuned is involved in supporting high-bandwidth applications. Making these modifications on a machine that supports multiple users and/or multiple connections is not recommended - it may cause the machine to deny connections because of a lack of memory allocation.

The Steps

1. Make sure that you have root privileges.
2. Type: `sysctl -p | grep mem`
This will display your current buffer settings. Save These! You may want to roll-back these changes
3. Type: `sysctl -w net.core.rmem_max=8388608`
This sets the max OS receive buffer size for all types of connections.
4. Type: `sysctl -w net.core.wmem_max=8388608`
This sets the max OS send buffer size for all types of connections.
5. Type: `sysctl -w net.core.rmem_default=65536`
This sets the default OS receive buffer size for all types of connections.
6. Type: `sysctl -w net.core.wmem_default=65536`
This sets the default OS send buffer size for all types of connections.
7. Type: `sysctl -w net.ipv4.tcp_mem='8388608 8388608 8388608'`
TCP Autotuning setting. "The tcp_mem variable defines how the TCP stack should behave when it comes to memory usage. ... The first value specified in the tcp_mem variable tells the kernel

the low threshold. Below this point, the TCP stack do not bother at all about putting any pressure on the memory usage by different TCP sockets. ... The second value tells the kernel at which point to start pressuring memory usage down. ... The final value tells the kernel how many memory pages it may use maximally. If this value is reached, TCP streams and packets start getting dropped until we reach a lower memory usage again. This value includes all TCP sockets currently in use."

8. Type: `sysctl -w net.ipv4.tcp_rmem='4096 87380 8388608'`

TCP Autotuning setting. "The first value tells the kernel the minimum receive buffer for each TCP connection, and this buffer is always allocated to a TCP socket, even under high pressure on the system. ... The second value specified tells the kernel the default receive buffer allocated for each TCP socket. This value overrides the `/proc/sys/net/core/rmem_default` value used by other protocols. ... The third and last value specified in this variable specifies the maximum receive buffer that can be allocated for a TCP socket."

9. Type: `sysctl -w net.ipv4.tcp_wmem='4096 65536 8388608'`

TCP Autotuning setting. "This variable takes 3 different values which holds information on how much TCP sendbuffer memory space each TCP socket has to use. Every TCP socket has this much buffer space to use before the buffer is filled up. Each of the three values are used under different conditions. ... The first value in this variable tells the minimum TCP send buffer space available for a single TCP socket. ... The second value in the variable tells us the default buffer space allowed for a single TCP socket to use. ... The third value tells the kernel the maximum TCP send buffer space."

10. Type: `sysctl -w net.ipv4.route.flush=1`
This will ensure that immediately subsequent connections use these values.

Quick Step

Cut and paste the following into a linux shell with root privileges:

```
sysctl -w net.core.rmem_max=8388608
sysctl -w net.core.wmem_max=8388608
sysctl -w net.core.rmem_default=65536
sysctl -w net.core.wmem_default=65536
```

```
sysctl -w net.ipv4.tcp_rmem='4096 87380 8388608'  
sysctl -w net.ipv4.tcp_wmem='4096 65536 8388608'  
sysctl -w net.ipv4.tcp_mem='8388608 8388608  
8388608'  
sysctl -w net.ipv4.route.flush=1
```

References

All of this information comes directly from these very reliable sources:

- [Pittsburgh Super Computing Center TCP Tuning Guide](#)
- [Summary of Buffer Sizes for Various Os's](#)
- [Ipsysctl tutorial 1.0.3](#)

Feedback

Please send [me](#) some [feedback](#) on how this worked for you. I'd be happy to help you figure it out on yours. I've used these or similar settings for a number of high-bandwidth applications with great results.

Last Modified December 31st, 1969

Copyright c 2003 - [Travis Sparks](#) - All Rights Reserved

DISCLAIMER: All usage of content on this site is subject to the UNC [Honor Code](#). All content on this site is posted with the express purpose of collaboration and general information.