

Network stack tuning

```
# Set the max OS send buffer size (wmem) and receive buffer size (rmem)
# to 12 MB for queues on all protocols.
# In other words set the amount of memory that is allocated for each TCP
# socket when it is opened or created while transferring files:

# Warning! The default value of rmem_max and wmem_max is about 128 KB
# in most Linux distributions, which may be enough for a low-latency general
# purpose network environment or for apps such as DNS / Web server. However,
# if the latency is large, the default size might be too small.
net.core.wmem_max = 12582912
net.core.rmem_max = 12582912

# You also need to set minimum size, initial size, and maximum size in
# bytes:
net.ipv4.tcp_rmem = 10240 87380 12582912
net.ipv4.tcp_wmem = 10240 87380 12582912

# Turn on window scaling which can be an option to enlarge the transfer
# window:
net.ipv4.tcp_window_scaling = 1

# Enable timestamps as defined in RFC1323:
net.ipv4.tcp_timestamps = 1

# Enable select acknowledgments:
net.ipv4.tcp_sack = 1

# By default, TCP saves various connection metrics in the route cache when
# the
# connection closes, so that connections established in the near future can
# use these to set initial conditions.
# Usually, this increases overall performance, but may sometimes cause
# performance degradation.
# If set, TCP will not cache metrics on closing connections.
net.ipv4.tcp_no_metrics_save = 1

# Set maximum number of packets, queued on the INPUT side, when the
# interface
# receives packets faster than kernel can process them.
net.core.netdev_max_backlog = 5000
```

From:
<https://www.ricozome.net/dokuwiki/> - Ricozome Wiki

Permanent link:
https://www.ricozome.net/dokuwiki/doku.php?id=pages_it:network

Last update: 2017/07/23 20:27



