Advanced

Optimize the shadowsocks server on Linux

First of all, upgrade your Linux kernel to 3.5 or later.

**Step 1, increase the maximum number of open file descriptors**

To handle thousands of concurrent TCP connections, we should increase the limit of file descriptors opened.

Edit the limits.conf

vi /etc/security/limits.conf

Add these two lines

\* soft nofile 51200

\* hard nofile 51200

Then, before you start the shadowsocks server, set the ulimit first

ulimit -n 51200

**Step 2, Tune the kernel parameters**

The priciples of tuning parameters for shadowsocks are

1. Reuse ports and conections as soon as possible.
2. Enlarge the queues and buffers as large as possible.
3. Choose the TCP congestion algorithm for large latency and high throughput.

Here is an example /etc/sysctl.conf of our production servers:

fs.file-max = 51200

net.core.rmem\_max = 67108864

net.core.wmem\_max = 67108864

net.core.netdev\_max\_backlog = 250000

net.core.somaxconn = 4096

net.ipv4.tcp\_syncookies = 1

net.ipv4.tcp\_tw\_reuse = 1

net.ipv4.tcp\_tw\_recycle = 0

net.ipv4.tcp\_fin\_timeout = 30

net.ipv4.tcp\_keepalive\_time = 1200

net.ipv4.ip\_local\_port\_range = 10000 65000

net.ipv4.tcp\_max\_syn\_backlog = 8192

net.ipv4.tcp\_max\_tw\_buckets = 5000

net.ipv4.tcp\_fastopen = 3

net.ipv4.tcp\_mem = 25600 51200 102400

net.ipv4.tcp\_rmem = 4096 87380 67108864

net.ipv4.tcp\_wmem = 4096 65536 67108864

net.ipv4.tcp\_mtu\_probing = 1

net.ipv4.tcp\_congestion\_control = hybla

Of course, remember to execute sysctl -p to reload the config at runtime.

**How to verify your optimizations work**

Use munin or any server monitor tools to generate the graph of your TCP connections. A well tuned server should look like this

