Network Card (NIC)

A network interface controller (NIC, also known as a network interface card, network adapter, LAN adapter or physical network interface, and by similar terms) is a computer hardware component that connects a computer to a computer network.

- Set RX/TX on boot before network.online
- Fix high SoftIRQ on 4.13 kernel

Set RX/TX on boot before network.online

Check current RX TX values

```
ethtool -g enp0s31f6
```

The output

The output of this command should look like this:

```
Ring parameters for enp0s31f6:
Pre-set maximums:
RX:             4096
RX Mini:        0
RX Jumbo:       0
TX:             4096
Current hardware settings:
RX:             256
RX Mini:        0
RX Jumbo:       0TX:             256
```

As you can se we shold set much higher values.

Create ethtool script

```
nano /usr/bin/network_rx_tx.sh
```

Add these lines to `network_rx_tx.sh`

```
#!/bin/bash
ethtool -G enp0s31f6 rx 4096 tx 4096echo "RX TX has been set to 4096"
```

Create the startup script

You can use ifup.pre or other ways to run the script, but with `systemd-networkd` you should follow the next steps.

```
nano /lib/systemd/system/network-rx-tx.service
```

Add these lines to `network-rx-tx.service`

```
[Unit]
Description=NetworkRXTX Setter
#Before=network-pre.target
#Wants=network-pre.target
DefaultDependencies=no
Requires=local-fs.target
After=local-fs.target
[Service]
#Type=oneshot
ExecStart=/usr/bin/network_rx_tx.sh
```

```
RemainAfterExit=yes
```

```
[Install]WantedBy=network.target
```

Enable the service

```
systemctl enable network-rx-tx.service
```

Fix high SoftIRQ on 4.13 kernel

Create `/etc/sysctl.d/99-network-tuning.conf` file

```
touch /etc/sysctl.d/99-network-tuning.conf
```

Insert the following text into `/etc/sysctl.d/99-network-tuning.conf`

The code below also contains various buffer updates.

```
# http://www.nateware.com/linux-network-tuning-for-2013.html# Increase Linux autotuning TCP buffer limits
# Set max to 16MB for 1GE and 32M (33554432) or 54M (56623104) for 10GE# Don't set tcp_mem itself! Let the kernel scale it based on RAM.
net.core.rmem_max = 16777216
net.core.wmem_max = 16777216
```

```
net.core.rmem_default = 16777216
net.core.wmem_default = 16777216
net.core.optmem_max = 40960
# cloudflare uses this for balancing latency and throughput# https://blog.cloudflare.com/the-story-of-one-latency-spike/
net.ipv4.tcp_rmem = 4096 1048576 2097152
net.ipv4.tcp_wmem = 4096 65536 16777216
# Also increase the max packet backlog
net.core.netdev_max_backlog = 100000
net.core.netdev_budget = 50000
# Make room for more TIME_WAIT sockets due to more clients,# and allow them to be reused if we run out of sockets
net.ipv4.tcp_max_syn_backlog = 30000
net.ipv4.tcp_max_tw_buckets = 2000000
net.ipv4.tcp_tw_reuse = 1
net.ipv4.tcp_fin_timeout = 10
# Disable TCP slow start on idle connections
net.ipv4.tcp_slow_start_after_idle = 0
# If your servers talk UDP, also up these limitsnet.ipv4.udp_rmem_min = 8192
net.ipv4.udp_wmem_min = 8192
# Actual fix for high softirq in 4.13net.core.netdev_budget_usecs = 5000
```

Reboot the server

```
/sbin/reboot
```