Linux Networking

Unit 18 - Hands-On Networking - 2018

Prof. Dr.-Ing. Thorsten Herfet, Andreas Schmidt, Pablo Gil Pereira

Telecommunications Lab, Saarland Informatics Campus, 26th Feb. 2018

Iproute2

HON

iproute2

- Manipulates network subsystem of the Linux kernel
- Replaces many traditional commands, e.g.:
  - ifconfig
  - route
  - arp
  - ...
- Unlike the tools above gives you nearly full control over linux network settings
- Functionality split up into sub-commands
- A little more complicated, but very well worth the learning curve

```
# Get help
$ ip help
...


$ ip <subcommand> help
...
```

"Link" Layer

The `link` subcommand can be used to manipulate **interface / link settings**.

```
# show all interfaces / links of the current host
$ ip link show

1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00

2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP mode DEFAULT \
         group default qlen 1000
    link/ether 70:54:d2:7b:7a:db brd ff:ff:ff:ff:ff:ff

3: eth1: <BROADCAST,MULTICAST> mtu 1500 qdisc pfifo_fast state DOWN mode DEFAULT group default \
         qlen 1000
    link/ether 70:54:d2:7b:7a:dc brd ff:ff:ff:ff:ff:ff
```

Flag	Description
`UP`	The interface is enabled
`LOWER_UP`	The lower layer is ready (e.g. "a cable is plugged in")
`NO-CARRIER`	The physical layer is not ready.
`BROADCAST`	Interface supports broadcasts
`MULTICAST`	Interface supports multicast

"Link" Layer

Manipulating Link settings

```
# Enable / Disable an interface
$ ip link set up dev eth0
$ ip link set down dev eth0

# Changing the MTU
$ ip link set mtu 1492 dev eth0

# Enable / Disable promiscuous mode
$ ip link set promisc on dev eth0
$ ip link set promisc off dev eth0

# Set a new MAC address
$ ip link set dev eth0 address 02:ab:00:1f:2b:34

# Add a more human-friendly description to an interface
$ ip link set dev eth0 alias "Local Network"

$ ip link show dev eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP mode DEFAULT \
   group default qlen 1000
    link/ether 70:54:d2:7b:7a:db brd ff:ff:ff:ff:ff:ff
    alias Local Network
```

Linux and its interfaces

Nearly all network functions in Linux are represented by interfaces

Physical Interfaces

- Represent a physical point of attachment to a network.
  i.e. a network card
- Data sent to the interface is put on the wire and the other way round.

Virtual Interfaces

- Logical abstraction of a specific function.
- Data sent to a virtual interface may not go to the physical network.
- Data sent to a virtual interface may be processed in some way before going to the network.

Virtual Interfaces | Example

VLANs

A networked host can belong to **more than one VLAN** requiring it to **tag outgoing data frames**. In Linux different VLANs are represented by virtual interfaces. All data sent to this interface will be **tagged and sent out the underlying link**.

```
$ ip link
...
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP mode DEFAULT \
   group default qlen 1000
    link/ether 70:54:d2:7b:7a:db brd ff:ff:ff:ff:ff:ff

# Add VLAN 10 to interface eth0
$ ip link add name vlan10 link eth0 type vlan id 10

$ ip link
...
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP mode DEFAULT \
   group default qlen 1000
    link/ether 70:54:d2:7b:7a:db brd ff:ff:ff:ff:ff:ff
3: vlan10@eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP mode DEFAULT
    group default
    link/ether 70:54:d2:7b:7a:db brd ff:ff:ff:ff:ff:ff

# Remove the VLAN
$ ip link del dev vlan10
```

Network Layer

The `address` subcommand can modify network layer settings, i.e. addresses.

```
# Show current IP configuration
$ ip address show

1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP \
   group default qlen 1000
    link/ether 70:54:d2:7b:7a:db brd ff:ff:ff:ff:ff:ff
    inet 134.96.86.110/25 brd 134.96.86.127 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::7254:d2ff:fe7b:7adb/64 scope link
       valid_lft forever preferred_lft forever


# Show IPv6 Addresses only
$ ip -6 address show

1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qlen 1000
    inet6 fe80::7254:d2ff:fe7b:7adb/64 scope link
       valid_lft forever preferred_lft forever
```

Network Layer

Manipulating IP Addresses

```
# Set an IP address
$ ip address add 10.0.0.1/24 dev eth1
$ ip address add 192.0.2.1/24 dev eth1
$ ip address add fd1e:1186:9d39::1/64 dev eth1

$ ip address
...
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 70:54:d2:7b:7a:dc brd ff:ff:ff:ff:ff:ff
    inet 10.0.0.1/24 brd 10.0.0.255 scope global eth1
       valid_lft forever preferred_lft forever
    inet 192.0.2.1/24 brd 192.0.2.255 scope global eth1
       valid_lft forever preferred_lft forever
    inet6 fd1e:1186:9d39::1/64 scope link
       valid_lft forever preferred_lft forever
    inet6 fe80::7254:d2ff:fe7b:7adc/64 scope link
       valid_lft forever preferred_lft forever


# Remove IP address from interface
$ ip address del 192.0.2.1/24 dev eth1


# Remove all addresses from interface
$ ip address flush dev eth1
```

Network Layer

Route Management

```
# Show the current routing table
$ ip route show

default via 192.0.2.1 dev eth0  proto static
192.0.2.0/25 dev eth0  proto kernel  scope link  src 192.0.2.11  metric 1

# Add a route via a gateway (i.e. router)
$ ip route add 192.0.2.128/25 via 192.0.2.1

# Change an existing route
$ ip route change 192.0.2.128/25 via 192.0.2.2

# Delete a route
$ ip route delete 192.0.2.128/25 via 192.0.2.2

# Add a default route
$ ip route add default via 192.0.2.1

# Discard traffic to unwanted destinations, e.g. known malicious hosts
$ ip route add blackhole 203.0.113.0/24        # drop silently
$ ip route add unreachable 203.0.113.0/24      # ICMP Host Unreachable
$ ip route add prohibit 203.0.113.0/24         # ICMP Administratively Prohibited
$ ip route add throw 203.0.113.0/24            # ICMP Network Unreachable
```

Network Layer

Think about the neighbors ...

```
# Show the ARP / NDP neightbor cache
$ ip neighbor show

134.96.86.96 dev eth0 lladdr 30:05:5c:53:d3:86 STALE
134.96.86.1 dev eth0 lladdr 00:23:5d:ff:e4:00 REACHABLE
fe80::468a:5bff:fe27:880b dev eth0 lladdr 44:8a:5b:27:88:0b REACHABLE


# Clear the Cache
$ ip neighbor flush


# Manually add an entry to the Cache
$ ip neighbor add 192.0.2.14 lladdr 02:4a:bb:68:9c:d0
```

Making it permanent

Iproute2 only changes the kernel's runtime settings.

```
$ nano /etc/network/interfaces
```

```
# Start interface automatically
auto eth0

# Use DHCP ...
iface eth0 inet dhcp

# ... or use a static IP config
iface eth0 inet static
  address 192.0.2.33
  netmask 255.255.255.0
  gateway 192.0.2.1             # default gateway

# VLAN 10 on interface eth0
iface vlan10 inet static
  address 10.0.0.1
  netmask 255.255.255.0
  vlan-raw-device eth0
```

```
# Configure IPv6 ...


# ... using SLAAC
# No configuration necessary.
# Linux uses SLAAC by default


# ... using DHCPv6
iface eth0 inet6 dhcp


# ... statically
iface eth0 inet6 static
  address fd1e:1186:9d39::33
  netmask 64
  gateway fd1e:1186:9d39::1
```

```
# Enable / Disable interface listed in interfaces file
$ ifup eth0
$ ifdown eth0
```

Linux Kernel Internals

Sysctl

- Sysctl is a **config interface** to the **kernel's runtime settings**.
- Allows you to **tune** many aspects of the **running kernel**.
  - CPU scheduling
  - Memory management
  - **Network settings**
  - ...
- Individual settings are represented as **files under `/proc/sys`**.
- **`sysctl`-command** can be used to change settings.

Sysctl | Examples

```
# Check value of a setting
$ sysctl net.ipv4.ip_forward
net.ipv4.ip_forward = 0

# Linux does not forward packets by default
# Enable routing for IPv4
$ sysctl -w net.ipv4.ip_forward=1

# Enable routing for IPv6
$ sysctl -w net.ipv6.conf.all.forwarding=1

# Disable responses to ICMP echo requests
$ sysctl -w net.ipv4.icmp_echo_ignore_all=1

# Ignore IPv6 router advertisements received on eth0
$ sysctl -w net.ipv6.conf.eth0.ignore_ra=0
```

To change settings permanently you can add them to `/etc/sysctl.conf` or a file below `/etc/sysctl.d/` so they survive a system reboot.

Be careful when changing sysctl parameters. Some settings may break your network connection or make your system unstable.

Make sure you know what you are doing.

Linux and Names

Linux Hostname

```
# Display the hostname
$ hostname
honlinux

# Display the fully qualified hostname
$ hostname -f
honlinux.my.domain

# Set the hostname to "gateway"
# (until the next reboot)
$ hostname gateway
```

To set the hostname permanently, edit `/etc/hostname`.

```
echo "gateway" > /etc/hostname
```

Local Resolver

- Part of the system **libc**
- Syscall **interface for name resolution**
- **Resolves names** to network addresses
- Specific features vary by implementation

/etc/resolv.conf

Config file for the local resolver

- List nameservers
- List default search domains
- Resolver options

```
## /etc/resolv.conf

## List of nameservers
nameserver 134.96.7.5
nameserver 134.96.7.100

## Search suffixes
search nt.uni-saarland.de uni-saarland.de

## Options
# rotate: try a different server for every query
# timeout: number of seconds to wait for replies
options rotate timeout:5
```

/etc/hosts

- Define hostname/IP pairs for the resolver
- Can be used to override names in the public DNS on a per-host basis
- One entry per line
- Each entry can contain **exactly one IP address** and **one or more hostnames**

```
## /etc/hosts

# Entries for the local host
127.0.0.1   honlinux.my.domain localhost
::1         honlinux.my.domain localhost

# A machine not in public DNS with an
# address we'd rather not have to remember
fd2f:5f10:b0a0:9277:aab:26a:0:18ba    dev-system  dev  d
```

Dnsmasq

- Network infrastructure for **small networks**
  - DNS
  - DHCP / DHCPv6
  - IPv6 Router advertisements
  - Network boot
- Lightweight
- Easy to configure (`/etc/dnsmasq.conf`)

Network Resolver (using Dnsmasq)

Just start Dnsmasq. It will use the local resolver to forward requests by default.

```
$ service dnsmasq start
```

Checking the Resolver

- host ( )
- dig (,)
- nslookup (, ,)

```
# Check the DNS resolver running on the local host
$ host www.uni-saarland.de 127.0.0.1
...

$ dig @127.0.0.1 www.uni-saarland.de
...

$ nslookup
> server 127.0.0.1
Default server: 127.0.0.1
Address: 127.0.0.1#53
> www.uni-saarland.de
...
```

DNS Server (using Dnsmasq)

Dnsmasq answers DNS requests matching hostnames in the local /etc/hosts.

```
## /etc/hosts
...
192.0.2.10    server.my.domain
...
```

```
$ dig @127.0.0.1 server.my.domain
...
;; ANSWER SECTION:
server.my.domain.             170        IN         A          192.0.2.10
...
```

Dynamic Addressing (using Dnsmasq)

```
## /etc/dnsmasq.conf

# Define an IPv4 address pool (192.0.2.10-20, 12h lease time)
dhcp-range=192.0.2.10, 192.0.2.20, 12h

# DHCPv6 (201 addresses with /64 prefix, 12h lease time)
dhcp-range=fd1e:1186:9d39::100, fd1e:1186:9d39::200, 64, 12h

# SLAAC (enable router advertisements, no DHCP)
# (Note for the tutorials: SLAAC breaks VPCS!)
dhcp-range=fd1e:1186:9d39::, ra-stateless

# Fixed IP address for a specific host
dhcp-host=11:22:33:44:55:66, 192.0.2.44

# Dnsmasq announces itself as the default gateway, DNS and NTP server by default
```

```
# Don't forget to restart Dnsmasq for changes to take effect
$ service dnsmasq restart
```

```
# Make sure Dnsmasq is started on system boot
$ rc-update add dnsmasq default
```

iptables | Examples

```
# DNAT (Change destination address)
iptables -t nat -A PREROUTING -s 192.0.2.12 -p tcp --dport 80 -j DNAT --to-destination 203.0.113.44

# SNAT (Change Source address)
iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to-source 198.51.100.1

# masquerading NAT (Change source address to that of outgoing interface)
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

```
# To filter IPv6 traffic, Linux uses a different set of tables.
# You can use ip6tables to manipulate them.
$ ip6tables -s fd1e:1186:9d39::1aa4 -j DROP
```

```
# Once you checked your current ruleset, you can persist it ...
$ service iptables save
$ service ip6tables save

# ... and automatically load it on system boot
$ rc-update add iptables default
$ rc-update add ip6tables default
```

Wrap-Up

Further Reading

- Iproute2 Cheat Sheet
- `man interfaces`
- Sysctl Documentation
- Sysctl Network Settings Documentation
- `man resolv.conf`
- Dnsmasq Website
- `man dnsmasq`