3.2. Learning Objectives

By the end of this session, you should be able to:

- Modify the network configuration interactively.
- · Make persistent changes to the network configuration.
- · Discuss virtual private networks.





3.3.a. Layer 2: Configuration

Changes to the runtime configuration of a network interface will not be reflected in subsequent reboots. Make sure to modify the system configuration files to make these changes permanent.

The following commands are used to change the MTU:

- # ifconfig eth0 mtu 1492
- # ip link set mtu 1492 dev eth0

The following command is used to change the link speed:

```
# ethtool -s eth0 speed 1000 duplex full
```



3.3.b. Layer 2: Configuration (Cont'd)

The following commands are used to check the link status:

- # ethtool eth0
- # mii-tool eth0

Sometimes it becomes necessary to blacklist hardware. This is done in different files on different distributions. The syntax is:

## blacklist <MODULENAME>:

- SUSE: /etc/modprobe.d/blacklist.local
- Ubuntu: /etc/modprobe.d/blacklist.conf
- CentOS: /etc/modprobe.d/blacklist.conf

3.4. Layer 2: Advanced Configuration

The **udev** (user **dev**ice facility) is used to manage network hardware interfaces for the **Linux** kernel. Sometimes changes are needed to rename interfaces or change configurations to match hardware MAC addresses.

The **udev** configuration file for network settings may be something like:

/etc/udev/rules.d/70-persistent-net.rules

Please consult the man udev documentation for the location of the udev configuration files for your distribution.

Another advanced Layer 2 configuration option is the addition of a MAC VLAN. This is an additional Layer 2 MAC address added to an interface

- # ip link add link eth0 dev peth0 type macvlan
- # ip link set up dev peth0

3.5. Layer 3: Configuration

Use one of the following commands to manually set a network (Layer 3) address:

- # ip addr add 10.0.2.25/255.255.255.0 dev eth0
- # ifconfig peth0 10.0.2.22 netmask 255.255.255.0

Use one of the following commands to manually set or change the default route:

- # ip route add default via 10.0.2.2
- # route add default gw 10.0.2.2

To add the address of a DNS server, use:

\$ echo nameserver 4.2.2.1 >> /etc/resolv.conf

To manually request a **DHCP** configuration, use the **dhclient** command:

# dhclient eth0

3.5. Layer 3: Configuration

Use one of the following commands to manually set a network (Layer 3) address:

- # ip addr add 10.0.2.25/255.255.255.0 dev eth0
- # ifconfig peth0 10.0.2.22 netmask 255.255.255.0

Use one of the following commands to manually set or change the default route:

- # ip route add default via 10.0.2.2
- # route add default gw 10.0.2.2

To add the address of a DNS server, use:

- \$ echo nameserver 4.2.2.1 >> /etc/resolv.conf
- To manually request a **DHCP** configuration, use the **dhclient** command:
- # dhclient eth0

₹



3.6. Boot Time Network Configuration

Network settings are stored in configuration files which allow for persistent configuration across reboots. The locations and syntax of these files differ for different distros.

There have also been some efforts at creating tools which manage network settings automatically. One such tool, which is used by **Debian**, **Ubuntu**, **SUSE**, and **CentOS**, is the **Network Manager**.

3.7.a. Network Manager

\_

**Network Manager** is now a part of most of the popular **Linux** distros. The **Network Manager** system remains the same, however, each distro can write their own plugins or interfaces.

## Network Manager provides:

- GUI tool (nm-connection-editor).
- applet (nm-applet).
- · text interface (nmtui).
- CLI interface (nmcli).

The current release of **Network Manager** will automatically configure and start a network interface with **DHCP** if there is no network configuration file for the adapter. If there is a configuration file, **Network Manager** will set the device to an "unmanaged" state and initialize the adapter with the attributes of the configuration file.

3.7.b. Network Manager (Cont'd)

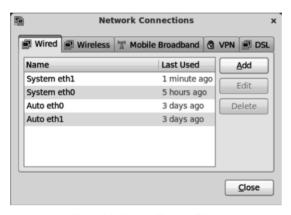


Figure 3.1: Network Manager Editor

## Network Manager is used by:

- · Ubuntu.
- CentOS.
- · SUSE.
- · Debian.

3.8. SUSE Configuration

•

The network configuration files on an **OpenSUSE** system are located in the /etc/sysconfig/network directory and match the ifcfg<interface> pattern.

The hostname is configured on an OpenSUSE system by editing the /etc/HOSTNAME file.

The DNS client settings are managed by either the /etc/sysconfig/network/config file or by manually editing the /etc/resolv.conf file. This is managed by a tool called netconfig.

If you are not using Network Manager, you can modify the network settings using the command:

yast network



3.9. CentOS Configuration

The network configuration files on a **CentOS** system are located in the /etc/sysconfig/network-scripts directory and match the ifcfg-<interface> pattern.

The hostname is configured by changing the entry in the /etc/hostname file.

The DNS client settings are managed by editing the /etc/resolv.conf file.

There are a few different tools specific to CentOS:

- . The setup tool is a text-based user interface (TUI) for system configuration items (network settings included).
- The system-config-network tool runs in either a GUI or TUI mode to configure network settings.
  You can specify which mode you want by running system-config-network-tui or system-config-network-gui.

3.10. Ubuntu Configuration

The network configuration files on a **Ubuntu** system reside in the <code>/etc/network</code> directory with the <code>/etc/network/interfaces</code> interface configuration file.

The hostname configuration file is /etc/hostname.

The DNS configuration for **Ubuntu** is managed by a system called **resolvconf**. The nameserver configuration is part of the per-interface configuration in /etc/network/interfaces.

The network-config command is an Ubuntu-specific tool to manage settings. However, the Network Manager tools are preferred.

•



3.11. DNS Client, the "Resolver"

ı

Historically, the DNS client or the /etc/resolv.conf file nameserver entries were created at configuration time, and were never changed. This static configuration worked well for servers that never changed locations or IP addresses. Network configuration is more dynamic, and the nameserver entries need to be adjusted. The following are some of the utilities that can and will change the behavior of /etc/resolv.conf:

- /etc/resolv.conf
- This is the traditional static file used to configure the resolver.
- Network Interface Configuration files
  In most distros, the nameserver information may be added to the interface configuration file, overwriting or modifying the /etc/resolv.conf when the interface is started.
  - DHCP Client

The **DHCP Server** often provides nameserver information as part of the information sent to the **DHCP Client**, replacing the existing nameserver records.

- · resolvconf service
- Most popular in **Ubuntu**, this service uses additional files like /etc/resolvconf.conf and a background service resolvconf.service to "optimize" the contents of /etc/resolv.conf.
- dnsmasq

Sets up in "mini" caching DNS server and may alter the resolver configuration to look at **dnsmasq** instead of the items listed in /etc/resolv.conf.

systemd.resolved

As of **systemd** version 233, the systemd-resolved is available. It provides a DNS stub listener on IP address **127.0.0.53** on the loopback adapter, and takes input from several files, including: /etc/systemd/resolved.conf,

3.12. What is a VPN?

The Virtual Private Network (VPN) allows for secure, remote access through untrusted networks.

Organizations take advantage of VPNs to make it easy for remote workers to connect back to the internal resources:

- Printers
- File Shares
- Databases.

There are **VPN** types provided by many different protocols:

- SSL
- IPSEC
- · SSH.

3.13. OpenVPN Software

•

One of the more popular VPN tools is OpenVPN, which provides SSL/TLS-based VPN connectivity. OpenVPN is a single binary for both the server and the client, and is a command-line tool.

**OpenVPN** has many examples of configurations for proof of concept and experimentation; one of the simplest is <a href="https://openvpn.net/index.php/open-source/documentation/miscellaneous/static-key-mini-howto.html">https://openvpn.net/index.php/open-source/documentation/miscellaneous/static-key-mini-howto.html</a>.

Besides using Linux to Linux console commands to configure the VPN server and client, you should know that there are many different client software solutions for managing your OpenVPN client.

Some of the characteristics of the OpenVPN software are:

- It can traverse NAT and firewalls.
- It can run over TCP or UDP.
- · It is cross platform.

For additional information on OpenVPN, see: https://openvpn.net/index.php/open-source/overview.html.