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

- Explain the concept of routing.
- Configure temporary and persistent routes.
- Configure a VLAN.
- Discuss the DHCP configuration.
- Configure an NTP server.

Routing commands include **`route`** and **`ip route`**. Dynamic routing protocols include **RIP**, **OSPF**, **BGP** and **IS-IS**.

To create a new run time route use, do:

```
# ip route add 10.1.11.0/24 via 10.30.0.101 dev eth2
```

To create a static rule which will survive a reboot use, do the following:

- On **CentOS**, edit the `/etc/sysconfig/network-scripts/route-<INTERFACE>` file and add a line like this:
  `10.1.11.0/24 via 10.30.0.101 dev eth2`
- **On Ubuntu, edit the `/etc/network/interfaces` file and add a line like this:**
  `up route add -net 10.1.11.0/24 gw 10.30.0.101 dev eth2`
- On **OpenSUSE,** edit the `/etc/sysconfig/network/ifroute-<INTERFACE>` file and add a line like this:
  `10.1.11.0/24 10.30.0.101 - eth2`

The downside to static routes is inflexibility. Dynamic routing protocols are more efficient at detecting and fixing routing problems quickly. Configuring dynamic routing is more complex. Several tools exist to assist in configuring dynamic routing. For example, the **Quagga** and **Zebra** projects seem to be popular.

A **VLAN** (**V**irtual **L**ocal **A**rea **N**etwork) allows multiple network nodes to end up in the same broadcast domain, even if they are not plugged into the same switch or switches. A **VLAN** is also a method for securing two or more **LANs** from each other on the same set of switches.

VLANs can be linked from point to point by doing **VLAN** trunking (**802.1q** is one such protocol).

To create a trunked interface on **CentOS**, create or edit the `/etc/sysconfig/network` file and add the following contents:

```
VLAN=yes VLAN_NAME_TYPE="DEV_PLUS_VID"
```

Create the `/etc/sysconfig/network-scripts/ifcfg-<INTERFACE>.<VLAN>` file with the following contents:

```
DEVICE=<INTERFACE>.<VLAN>
BOOTPROTO=static
ONBOOT=yes
IPADDR=192.168.3.1
NETMASK=255.255.255.0
PHYSDEV="<INTERFACE>"
```

11.4.b VLAN - Virtual Local Area Network (Cont'd)

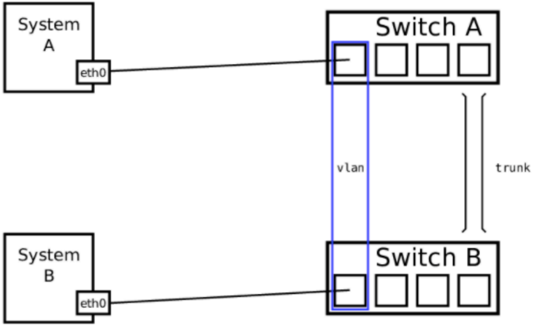

Figure 11.1: Virtual Local Area Network (VLAN)

The **Dynamic Host Configuration Protocol** (**DHCP**) is used to configure the network-layer addressing. The **dhcpd** daemon used to be configured using both a configuration file (`/etc/dhcp/dhcpd.conf`) and a **daemon** options file that was distribution-dependent. Recent versions of **dhcp** have moved the daemon options into **systemd**.

The **daemon** options are configured in a separate file:

- **On CentOS:** `/etc/sysconfig/dhcpd`
- **On Ubuntu:** `/etc/default/isc-dhcp-server`
- **On OpenSUSE:** `/etc/sysconfig/dhcpd`

The **dhcp** server will only serve out addresses on an interface that it finds a subnet block defined in the `/etc/dhcp/dhcpd.conf` file. It is no longer a requirement to explicitly tell **dhcp** which interfaces to use.

Additional or different daemon command line options may be passed to the daemon at start time by the systems' drop-in files. Please see the **COMMAND LINE** section in the **dhcpd man** page for additional details.

Global options are settings which should apply to all the hosts in a network. You can also define options on a per-network basis.

A sample configuration would be:

```
subnet 10.5.5.0 netmask 255.255.255.224 {
  range 10.5.5.26 10.5.5.30;
  option domain-name-servers ns1.internal.example.org;
  option domain-name "internal.example.org";
  option routers 10.5.5.1;
  option broadcast-address 10.5.5.31;
  default-lease-time 600;
  max-lease-time 7200;
  }
```

Many protocols require consistent, if not accurate time to function properly.

The security of many encryption systems is highly dependent on proper time. Industries such as commodities or stock trading require highly accurate time, as a difference of only seconds can mean hundreds if not thousands of dollars lost or earned.

The **Network Time Protocol** (**NTP**) is a method to update and synchronize system time. **NTP** consists of a daemon and a protocol. **NTP** time sources are divided up into "strata".

- A `strata 0` clock is a special purpose time device (Atomic clock, GPS radio, etc).
- A `strata 1` server is any **NTP** server connected directly to a `strata 0` source (over serial or the like).
- A `strata 2` server is any **NTP** server which references a `strata 1` server using **NTP**.
- A `strata 3` server is any **NTP** server which references a `strata 2` server using **NTP**.

NTP may function as a **client**, a **server** ,or a **peer:**

- **Client**: acquires time from a server or a peer.
- **Server**: provides time to a client.

11.7.b. Network Time Protocol (Cont'd)

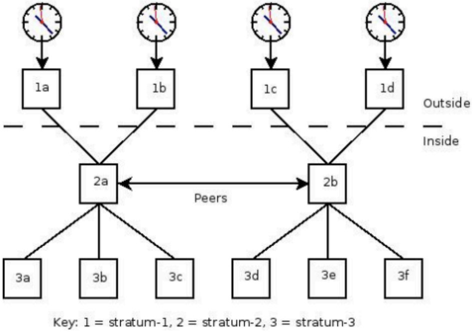

Figure 11.2: Network Time Protocol

web.mit.edu/w1gsl/Public/flyer
http://web.mit.edu/w1gsl/Public/flyer

11.8. NTP Applications

Implementation of **NTP** services varies widely between distributions. The most common elements of the configuration are similar for most popular time synchronization applications. Below, you will see the common locations for the **NTP** application configuration files:

- `/etc/ntp.conf`**ntp**: http://www.ntp.orgThis is the default application ().
- `/etc/chrony.conf`**chrony**: http://chrony.tuxfamily.orgIt is designed to work in a wide range of environments, including intermittent network connections and virtual machines ().
- **systemd-timesyncd:**
  An ntp client is only included in the systemd package (`/etc/systemd/timesyncd.conf`).

You should consult the related **man** pages for configuration details. Since the various **NTP** applications may conflict with each other, you may want to consider having only one **NTP** application active at a time.

To configure the **ntpd** client, choose an **NTP** source as a server or peer, start the **NTP** daemon, and verify that time is synchronizing.

A good **NTP** server is only as good as its time source. The **NTP** pool project (http://www.pool.ntp.org/en/) was created to alleviate the load that was crippling the small number of **NTP** servers. The **pool** directive is aware of round-robin DNS servers and allows a different IP address on each request. If the connection allows **peers**, the **pool** command will establish peers on its own.

The clasic **server** directive may be used if desired.

To configure your **NTP** server to use the **NTP** pool, edit the `/etc/ntp.conf` file and add or edit the following settings:

```
driftfile /var/lib/ntp/ntp.drift
```

```
server 0.pool.ntp.org
server 1.pool.ntp.org
server 2.pool.ntp.org
server 3.pool.ntp.org
```

The **ntpdc -c peers** command can show the time differential between the local system and configured time servers. The **timedatectl** command is in many distributions and may be used to query and control the system time and date.

To configure the **ntpd** server, allow clients to request time, set up access for peers, declare the local machine to be a time reference, and start the **NTP** daemon.

Client access to an **ntpd** server can be restricted:

```
# Default policy prevents queries
restrict default nopeer nomodify notrap noquery
# Allow queries from a particular subnet
restrict 123.123.x.0 mask 255.255.255.0 nopeer nomodify notrap
# Allow queries from a particular host
restrict 131.243.1.42 nopeer nomodify notrap noquery
# Unrestrict localhost restrict 127.0.0.1
```

in the **ntp.conf** as well as defining servers to receive time **peers** may be defined. Two or more **peers** will keep consistent time between themselves in the event of disconnection to the outside time source:

```
peer 128.100.49.12
peer 192.168.0.1
```

It may be desirable to declare the local machine as a time source: