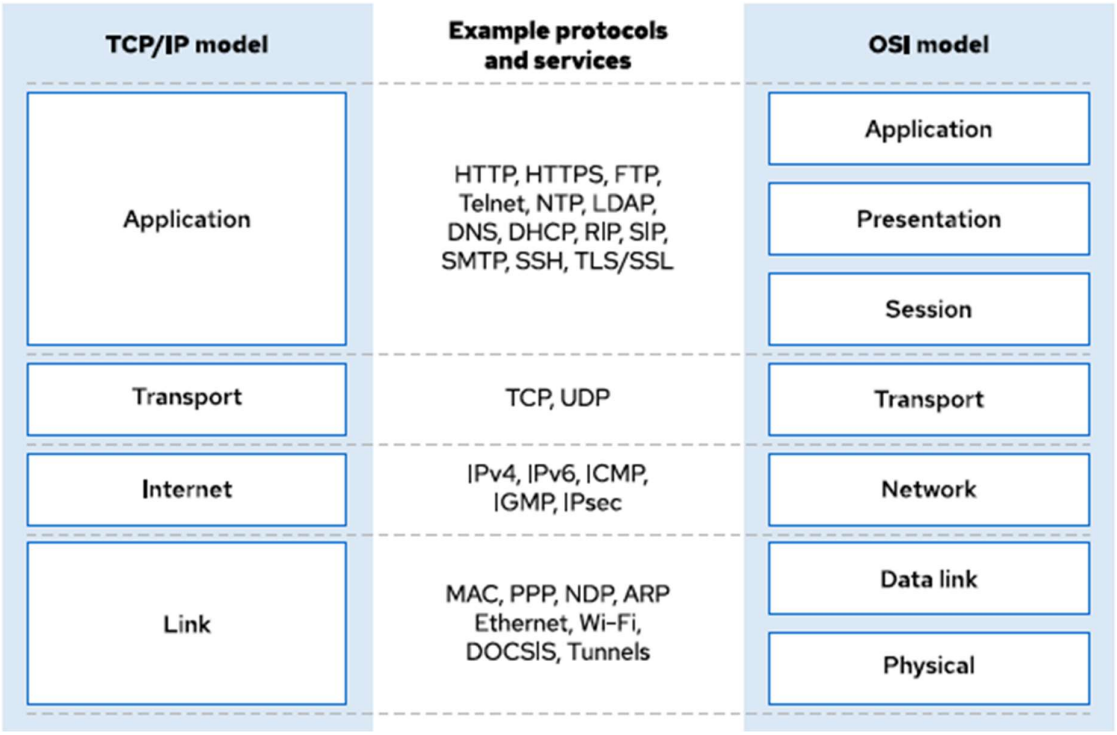
CHAPTER-12 MANAGE NETWORKING

Describe Networking Concepts

TCP/IP Network Model

The TCP/IP network model is a simplified, four-layered set of communication protocols that describes how data communications are packetized, addressed, transmitted, routed, and received between computers over a network.

The protocol is specified by RFC 1122, Requirements for Internet Hosts — Communication Layers.



Describe Network Interface Names

Each network port on a system has a name, which you use to configure and identify it.

Network interface names start with the type of interface:

- Ethernet interfaces begin with en
- WLAN interfaces begin with wl
- WWAN interfaces begin with ww

The rest of the interface name after the type is based on information from the server's firmware or determined by the location of the device in the PCI topology.

- oN indicates an on-board device with unique index N provided by the server's firmware. The eno1 name is on-board Ethernet device 1.
- sN indicates a device in PCI hotplug slot N. For example, ens3 is an Ethernet card in PCI hotplug slot 3.

- pMsN indicates a PCI device on bus M in slot N. A wlp4s0 interface is a WLAN card on PCI bus 4 in slot 0.

IPv4 Networks

An IPv4 address is a 32-bit number, expressed as four 8-bit octets in a decimal format that range in value from 0 to 255, separated by single dots The address is divided into two parts: the network prefix and the host number. The network prefix identifies a unique physical or virtual subnet. The host number identifies a specific host on the subnet. All hosts on the same subnet have the same network prefix and can talk to each other directly without a router. A network gateway connects different networks and a network router commonly operates as the gateway for a subnet.

A network mask (netmask) is a binary mask whose length indicates how many bits belong to the network prefix that identifies the subnet. Because an IPv4 address is always 32 bits long, a subnet with a longer network mask will have less bits available to identify hosts, meaning fewer possible hosts. A subnet with a shorter network mask will have more bits available to identify hosts, meaning more possible hosts and a larger subnet.

IPv4 Subnets and Netmasks

The number of available host addresses in a subnet depends on the network prefix size. For example, a network prefix of /24 leaves 8 bits, or 255 possible host addresses in the subnet. A network prefix of /16 leaves 16 bits, or 65536 possible host addresses in the subnet.

- The network address for a subnet is the lowest possible address on a subnet, where the host number is all binary zeros.
- The broadcast address for a subnet is the highest possible address on a subnet, where the host number is all binary ones, and is a special address for broadcasting packets to all subnet hosts.
- The gateway address for a subnet can be any unique host number in the subnet, but is commonly set to the first available host number, which is a binary number of all zeroes except for a '1' in the last bit. This gateway numbering convention is not mandatory, and subnets that do not need external communication will not set a network gateway.

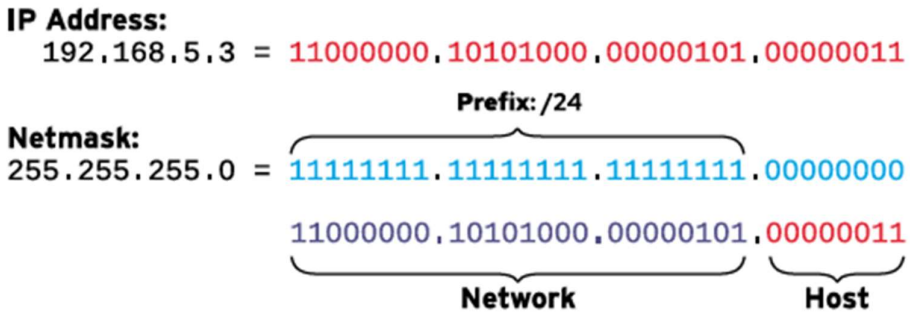
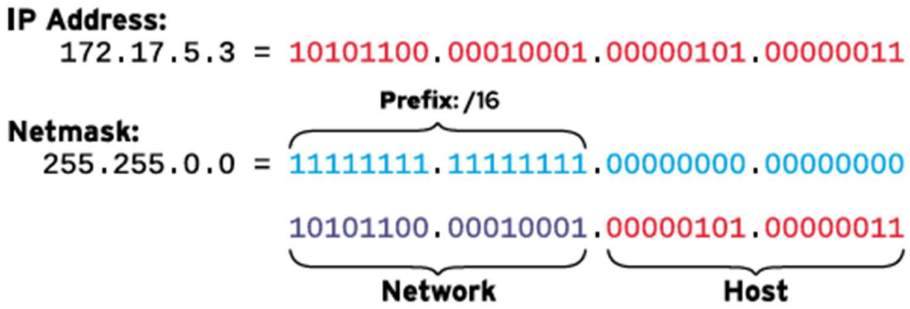


Figure 12.2: IPv4 netmask calculation for a small network



Example Network Calculations

In the following example, identify the netmask first, then perform the binary calculations. A netmask of /24 means that the leading 24 bits of the address define the network address (192.168.1.0). In this scenario, 8 bits, or 254 addresses, are available for host addressing.

IPv4 address of 192.168.1.107/24

Network prefix	/24 or 255.255.255.0	11111111.11111111.11111111.00000000
Host address	192.168.1.107	11000000.10101000.00000001.01101011
Network address	192.168.1.0	11000000.10101000.00000001.00000000
Address range for hosts on subnet	192.168.1.1 - 192.168.1.254	11000000.10101000.00000001.00000001 to 11000000.10101000.00000001.11111110
Broadcast address	192.168.1.255	11000000.10101000.00000001.11111111

In the following example, a /19 netmask is a valid network prefix that uses only a partial octet. Variable length netmasks allow subnets with a different quantity of hosts than the full-octet netmasks. The remaining 13 bits, or 8190 addresses, are available for host addressing.

IPv4 address of 172.16.181.23/19

Network prefix	/19 or 255.255.224.0	11111111.11111111.11100000.00000000
Host address	172.16.181.23	10101100.00010000.10110101.00010111
Network address	172.16.160.0	10101100.00010000.10100000.00000000
Address range for hosts on subnet	172.16.160.1 - 172.16.191.254	10101100.00010000.10100000.00000001 to 10101100.00010000.10111111.11111110
Broadcast address	172.16.191.255	10101100.00010000.10111111.11111111

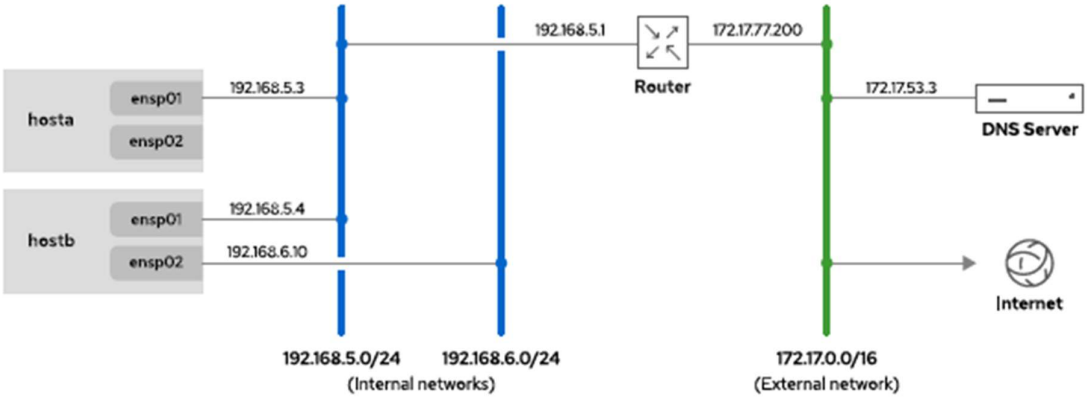
In the following example, the /8 indicates a large network. Only the first octet is used for the network prefix (10.0.0.0). The remaining 24 bits, or 16,777,214 addresses, are available for host addressing. The 10.255.255.255 broadcast address is the last address of the network.

IPv4 address of 10.1.1.18/8

Network prefix	/8 or 255.0.0.0	11111111.00000000.00000000.00000000
Host address	10.1.1.18	00001010.00000001.00000001.00010010
Network address	10.0.0.0	00001010.00000000.00000000.00000000
Address range for hosts on subnet	10.0.0.1 - 10.255.255.254	00001010.00000000.00000000.00000001 to 00001010.11111111.11111111.11111110
Broadcast address	10.255.255.255	00001010.11111111.11111111.11111111

IPv4 Routes

Network packets move from host to host on a subnet and through routers from network to network. Each host has a routing table, which determines which network interface is correct for sending packets to particular networks. A routing table entry lists the destination network, which network interface to use, and the IP address of the router which will forward the packet on its way to the final destination. The routing table entry that matches the network prefix of the destination address is used to route the packet. If multiple entries are valid for the destination address, then the entry with the longer prefix is used.



Example network topology

IPv6 Networks

IPv6 is designed to greatly expand the number of total available device addresses. IPv6 is used in both enterprise networks and for mobile communications. Most if not all Internet Service Providers (ISPs) use IPv6 extensively for assigning to internal equipment and for dynamic assignment for customer devices.

IPv6 can also be used in parallel with IPv4 in a dual-stack mode. A network interface can have both IPv6 and IPv4 addresses. Red Hat Enterprise Linux operates in a dual-stack mode by default.

IPv6 Addresses

An IPv6 address is a 128-bit number, which is normally expressed as eight colon-separated groups of four hexadecimal nibbles (half-bytes). Each nibble represents four bits of the IPv6 address, so each group represents 16 bits of the IPv6 address.

2001:0db8:0000:0010:0000:0000:0000:0001

To make IPv6 addresses easier to write, leading zeros in a colon-separated group are not needed. However, at least one hexadecimal digit must be written in each colon-separated group.

2001:db8:0:10:0:0:0:1

Because addresses with long strings of zeros are common, one or more consecutive groups of zeros only can be combined with exactly one block of two colon :: characters.

2001:db8:0:10::1

The 2001:db8::0010:0:0:0:1 IPv6 address, though a valid representation, is a less convenient way to write the example address. This different representation can confuse administrators who are new to IPv6. The following list shows tips for writing consistently readable addresses:

- Suppress leading zeros in a group.
- Use a two-colon :: block to shorten the address as much as possible.
- If an address contains two consecutive groups of zeros, equal in length, then it is preferred to shorten the leftmost groups of zeros to :: and the rightmost groups to :0: for each group.
- Although it is allowed, do not use :: to shorten a single group of zeros. Use :0: instead, and save :: for consecutive groups of zeros.
- Always use lowercase letters for hexadecimal numbers a through f.

IPv6 Subnets

A normal IPv6 unicast address is divided into two parts: the network prefix and interface ID. The network prefix identifies the subnet. Two network interfaces on the same subnet cannot have the same interface ID; the interface ID identifies a particular interface on the subnet.

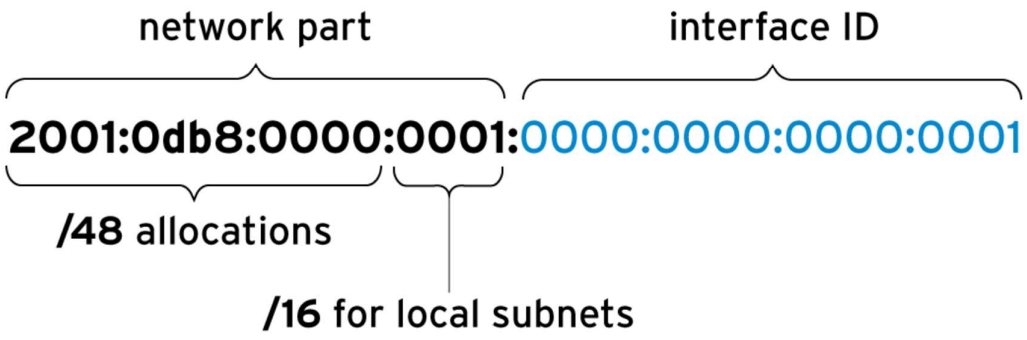
IPv4, IPv6 has a standard subnet mask, /64, which is used for almost all normal addresses. In this case, half of the 128 bit address is the network prefix and the other half is the interface ID. With 64 bits for host addresses, a single subnet could theoretically contain 2^64 hosts.

Typically, the network provider allocates a shorter prefix to an organization, such as a /48. This prefix leaves the rest of the network part for assigning subnets (up to the maximum /64 length) from that allocated prefix. For example, when assigned a /48 allocation prefix, 16 bits are available for up to 65536 subnets.

IPv6 address parts and subnetting

IPv6 address is **2001:db8:0:1::1/64**

Allocation from provider is **2001:db8::/48**



Common IPv6 Addresses and Networks

IPv6 address or network	Purpose	Description
`::1/128`	localhost	The IPv6 equivalent to the `127.0.0.1/8` address, set on the loopback interface.
`::`	The unspecified address	The IPv6 equivalent to `0.0.0.0`. For a network service, it might indicate that it is listening on all configured IP addresses.
`::/0`	The default route (the IPv6 internet)	The IPv6 equivalent to the `0.0.0.0/0` address. The default route in the routing table matches this network; the router for this network is where all traffic is sent in the absence of a better route.
`2000::/3`	Global unicast addresses	"Normal" IPv6 addresses are currently allocated from this space by the *Internet Assigned Numbers Authority (IANA)*. The addresses include all the networks ranging from `2000::/16` through `3fff::/16`.
`fd00::/8`	Unique local addresses (RFC 4193)	IPv6 has no direct equivalent of the RFC 1918 private address space, although this network range is close. A site can use these networks to self-allocate a private routable IP address space inside the organization, but these networks cannot be used on the global internet. The site must *randomly* select a `/48` from this space, but it can subnet the allocation into `/64` networks normally.
`fe80::/10`	Link-local addresses	Every IPv6 interface automatically configures a link-local unicast address that works only on the local link on the `fe80::/64` network. However, the entire `fe80::/10` range is reserved for future use by the local link. This topic is discussed in more detail later.
`ff00::/8`	Multicast	The IPv6 equivalent to the `224.0.0.0/4` address. Multicast is used to transmit to multiple hosts at the same time, and is particularly important in IPv6 because it has no broadcast addresses.

DESCRIPTION	COMMANDS / OPTIONS
Show command to the link-local ip address	**Syntax:** ip [OPTIONS] OBJECT **To view information about network interfaces and their associated IP addresses** **Example**: ip addr show ↲ **To set the IP address of an interface** **Example:** ip addr add 192.168.1.100/24 dev eth0 ↲ **To delete an existing route from the routing table** **Example:** ip route delete 10.0.0.0/24 via 192.168.1.1 dev eth0 ↲ **To change the default gateway for outgoing traffic** **Example:** ip route add default via 192.168.1.254 dev eth0 ↲ **To bring an interface up** **Example:** ip link set eth0 up ↲ **To change the MTU (maximum transmission unit) of a network interface** **Example:** ip link set eth0 mtu 1500 ↲ **To bring an interface up (activate it)** **Example:** ip link set eth0 up ↲ **To monitor real-time network traffic on a specific interface** **Example:** watch -n 1 "ip -s link show eth0 | grep 'RX bytes'" ↲ **Displaying Interface Errors** **Example**: ip -s link show eth0 | grep -E 'errors|dropped' ↲ **To show all IP addresses associated with all network devices** **Example:** ip address ↲

	To display link layer information **Example:** ip link ↲ **To show the statistics of the various network interfaces** **Example:** ip -s link ↲ **To get information about a particular network interface** **Example:** ip -s link show (interface) ↲ **To display the state of devices, addresses and routes continuously** **Example:** ip moniter ↲ **To show routing information** **Example:** ip route ↲ **To view the MAC address of the devices connected to your system** **Example:** ip neighbor ↲
Validate Network Configuration	**Syntax:** ping [options] host_or_IP_address **-c** the number of packets to send to the server/host **-s** send light and heavy packet **-i** change wait time **-q** To only get the summary about the network **-w** to set a timeout for the PING **-f** To flood a network with PING packets for testing network performance **-T** Timestamps record the current time of an event over a network

	tsonly (timestamp only) **tsandaddr** (timestamp and address) **tsprespec** (timestamp pre-specified for multiple hosts)
To display socket statistics	**Syntax**: ss [options] **-n** Show numbers instead of name for interface and port **-t** Show TCP sockets **-u** Show UDP sockets **-l** Show only listening sockets **-a** show all **-p** Show the process that uses the sockets **-A** inet Display active connections

Configure Networking from the Command Line

The NetworkManager service monitors and manages a system's network settings. In the GNOME graphical environment, a Notification Area applet displays network configuration and status information that is received from the NetworkManager daemon. You can interact with the NetworkManager service via the command line or with graphical tools. Service configuration files are stored in the /etc/NetworkManager/system-connections/ directory.

DESCRIPTION	COMMANDS / OPTIONS
nmcli command	**status of all network interfaces.** nmcli dev status **List all connections.** nmcli con show **List the current settings for the connection** nmcli con show name **Add and name a new connection profile**. nmcli con add con-name name **Modify the connection name** nmcli con mod con-name ipv4.method manual ipv4.method auto ipv4.addresses ipv4.gateway ipv4.dns ipv4.dns-search ipv4.ignore-auto-dns connection.autoconnect connection.id ens3 connection.interface-name ens3 **Reload the configuration files, after manual file editing.** nmcli con reload **Activate the connection name** nmcli con up name **Disconnect the interface, which also deactivates the current connection**. nmcli dev dis dev **Delete the specified connection and its configuration file**. nmcli con del name

DESCRIPTION	COMMANDS / OPTIONS
Modify Network Configuration	Depending on the purpose of the connection profile, NetworkManager uses the following directories to store the configuration files: **etc/NetworkManager/system-connections/** Directory stores persistent profiles that the user created and edited. NetworkManager copies them automatically to the /etc/ NetworkManager/system-connections/ directory. **/run/NetworkManager/system-connections** Directory stores temporary profiles, which are automatically removed when you reboot the system. **/usr/lib/NetworkManager/system-connections/** Directory stores predeployed immutable profiles. When you edit such a profile with the NetworkManager API, NetworkManager copies this profile to either the persistent or the temporary storage.
Configure Hostnames	**Syntax:** hostnamectl [OPTIONS...] COMMAND **set-hostname NAME --options** Static: Assigned by system admin and it is used to initialize the kernel hostname during boot time **Dynamic or Transient:** Assigned by mDNS server or DHCP server during run time. **Pretty:** It's a high-level hostname assigned by system admin or end-user set-icon-name NAME set-chassis NAME
Configure Name Resolution	**configuration Files** **/etc/hosts** To resolve the query **/etc/nsswitch.conf** **/etc/resolv.conf** To controls query is performed

Test DNS Name Resolution	**Test DNS Name Resolution** host classroom.example.com **To test DNS server connectivity** dig classroom.example.com **To test the /etc/hosts file** getent hosts classroom.example.com