12.1 Protocols in the IP Protocol Suite

Protocol	Description
Internet Protocol **(IP)**	IP is the main protocol used on the internet. IP is a connectionless protocol that makes routing path decisions. It also handles logical addressing issues through the use of IP addresses.
Transmission Control Protocol **(TCP)**	TCP provides services that ensure accurate and timely delivery of network communications between two hosts. TCP is a connection-oriented protocol. It provides the following services to ensure message delivery: • Sequencing of data packets • Flow control • Error checking
User Datagram Protocol **(UDP)**	UDP is a connectionless protocol. It is a host-to-host protocol like TCP. However, UDP does not include mechanisms for ensuring timely and accurate delivery. Because it has less overhead, it offers fast communications, but at the expense of possible errors or data loss.
Internet Control Message Protocol **(ICMP)**	ICMP works closely with IP to provide error and control information. This protocol allows hosts to exchange packet status information, which helps move the packets through the internetwork. Two common management utilities, ping and traceroute, use ICMP messages to check network connectivity. ICMP also works with IP to send notices when destinations are unreachable and when devices' buffers overflow. It works with IP to send notices of whether devices can communicate across the network as well. Additionally, ICMP relays information about the route and hops that packets take through the network.
Internet Group Membership Protocol **(IGMP)**	IGMP is a protocol for defining host groups. All group members can receive broadcast messages intended for the group (called multicasts). Multicast groups can be composed of devices within the same network or across networks (connected with a router).
HyperText Transfer Protocol **(HTTP)**	HTTP is used by web browsers and web servers to exchange files (such as web pages) through the world wide web and intranets. HTTP can be described as an information requesting and responding protocol. It is typically used to request and send web documents but is also used as the protocol for communication between agents using different IP protocols.
HTTP over SSL **(HTTPS)**	HTTPS is a secure form of HTTP that uses SSL to encrypt data before it is transmitted.
Secure Sockets Layer **(SSL)**	SSL secures messages being transmitted on the internet. It uses RSA for authentication and encryption. Web browsers use SSL (Secure Sockets Layer) to ensure safe web transactions. URLs that begin with https:// trigger your web browser to use SSL.
Transport Layer Security **(TLS)**	TLS is an improved version of SSL. It ensures that messages being transmitted on the internet are private and tamper-proof. TLS is implemented through two protocols. • TLS Record provides connection security with encryption (e.g., with DES). • TLS Handshake provides mutual authentication and choice of encryption method.
File Transfer Protocol **(FTP)**	FTP provides a generic method of transferring files. It can include file security through usernames and passwords, and it allows file transfer between dissimilar computer systems. FTP can transfer both binary and text files, including HTML, to another host. FTP URLs are preceded by ftp://. To log in to an FTP server, use:ftp://username@servername
Trivial File Transfer Protocol **(TFTP)**	TFTP is similar to FTP. It lets you transfer files between a host and an FTP server. However, this protocol provides no user authentication and no error detection. Because it doesn't perform error detection, TFTP is faster than FTP but might be subject to transmission errors.
Secure File Transfer Protocol **(SFTP)**	SFTP is a secure version of FTP that uses Secure Shell (SSH) to encrypt data transfers. SSH ensures that SFTP transmissions use encrypted commands and data, which prevent data from being transmitted over the network in clear text.
Secure Copy Protocol **(SCP)**	SCP allows you to copy files between systems. Like SFTP, SCP relies on SSH to ensure that data and passwords are not transmitted over the network in cleartext.
Simple Mail Transfer Protocol **(SMTP)**	SMTP is used to route electronic mail through the internetwork. SMTP is used in the following ways: • Between mail servers for sending and relaying mail • By all email clients to send mail
Internet Message Access Protocol **(IMAP)**	IMAP is an email retrieval protocol designed to enable users to access their email from various locations without the need to transfer messages or files back and forth between computers. Messages remain on the remote mail server and are not automatically downloaded to a client system. An email client that uses IMAP for receiving mail uses SMTP for sending mail.
Post Office Protocol 3 **(POP3)**	POP3 retrieves email from a remote server to a local client over a TCP/IP connection. With POP3, email messages are downloaded to the client. An email client that uses POP3 for receiving mail uses SMTP for sending mail.
Dynamic Host Configuration Protocol **(DHCP)**	DHCP is a protocol that automatically assigns addresses and other configuration parameters to network hosts. Using a DHCP server, hosts receive configuration information at startup. This reduces the amount of manual configuration required on each host.

Domain Name System **(DNS)**	DNS is a system that is distributed throughout the internetwork to provide address and name resolution. For example, the name www.mydomain.com would be identified with a specific IP address.
Network Time Protocol **(NTP)**	NTP communicates time synchronization information between systems on a network.
Lightweight Directory Access Protocol **(LDAP)**	LDAP allows you to search and update a directory service. The LDAP directory service follows a client-server model. One or more LDAP servers contain the directory data. The LDAP client connects to an LDAP server to make a directory service request.
Simple Network Management Protocol **(SNMP)**	SNMP is a protocol designed for managing complex networks. SNMP lets network hosts exchange configuration and status information. This information can be gathered by management software and used to monitor and manage the network.
Remote Terminal Emulation **(Telnet)**	Telnet allows an attached computer to act as a dumb terminal with data processing taking place on the TCP/IP host computer. Telnet uses unsecured data transmissions and should be avoided. SSH provides the same functionality but does so securely using encryption.
Secure Shell **(SSH)**	SSH allows secure interactive control of remote systems. SSH is a secure and acceptable alternative to Telnet. SSH uses public-key cryptography for both connection and authentication.

12.1 IPv4 Rules and Concepts

IPv4 Address Rules
• Each host must have a unique IPv4 address. • Each host on the same logical network must have the same network address. • Hosts can only communicate directly with other hosts on the same logical network.

Concept	**Description**
Host	A host (also known as a network host) is a computer or device (such as a router) on a network.
IP address	The IP address is a number assigned to identify hosts and other devices on a network.
Network address	The network address (also referred to as the network ID) is the portion of the IP address that identifies a specific network.
Host address	The host address (also referred to as a host ID) is the remaining portion of the IP address that identifies the specific host or other device on the network.
Subnet mask	A subnet mask identifies the portion of the IP address that defines the network address and the portion of the IP address that defines the specific host.
Address class	IPv4 addresses are divided into classes. The address class identifies the range of IPv4 addresses and a default subnet mask used for the range.
Default subnet mask	A default subnet mask is assigned to Classes A - C as follows: • **255.0.0.0** is the default subnet mask for **Class A networks**. • **255.255.0.0** is the default subnet mask for **Class B networks**. • **255.255.255.0** is the default subnet mask for **Class C networks**.
Broadcast address	The broadcast address is the last address in the IP address range and is used to send messages to all hosts on the network.
Default gateway	The default gateway is a device that performs routing and enables a host to communicate with hosts on other networks through the routing process. • A default gateway address must be configured on each host to allow internetwork communication. Without the default gateway, hosts can only communicate with devices within the same subnet. • The default gateway address must be on the same subnet as the host computer. ◦ Routers have multiple network interface cards attached to multiple networks. ◦ When configuring the default gateway, choose the address on the local subnet.

12.1 Classless Inter-Domain Routing (CIDR)

Class	Address Range	First Octet Range	Default Subnet Mask	CIDR Notation
A	1.0.0.0 to 126.255.255.255	**1-126** (**00000001**--**01111110** binary)	**255**.0.0.0	**/8**
B	128.0.0.0 to 191.255.255.255	**128-191** (**10000000**--**10111111** binary)	**255.255**.0.0	**/16**
C	192.0.0.0 to 223.255.255.255	**192-223** (**11000000**--**11011111** binary)	**255.255.255**.0	**/24**
D	224.0.0.0 to 239.255.255.255	**224-239** (**11100000**--**11101111** binary)	n/a	n/a
E	240.0.0.0 to 255.255.255.255	**240-255** (**11110000**--**11111111** binary)	n/a	n/a

12.1 Address Assignment

Method	Uses
Dynamic Host Configuration Protocol (DHCP)	DHCP is a method used to automatically assign IPv4 addresses and other TCP/IPv4 configuration parameters to hosts. Client computers contact a DHCP server to receive TCP/IPv4 configuration information. Use DHCP: • For small, medium, or large networks. • For automatic host configuration. • To automatically deliver additional configuration parameters such as default gateway and DNS servers. *By default, all Windows computers try to use DHCP for TCP/IPv4 configuration information.*
Automatic Private IPv4 Addressing (APIPA)	APIPA is an automatic configuration method where hosts automatically select their own IPv4 address within a specific range. When using APIPA: • Windows computers will use APIPA if a DHCP server cannot be contacted. • Hosts select an IPv4 address in the 169.254.0.1 to 169.254.255.255 range with a mask of 255.255.0.0. After choosing the address, the host verifies that no other host on the network is using the selected address. • APIPA sets only the IPv4 address and mask. Because it does not assign a default gateway, APIPA can be used on a single subnet, but cannot be used if communication with other subnets is required. Use APIPA for small single-subnet networks that do not use DNS servers or do not have internet or connectivity outside of the local subnet.
Static (manual) assignment	Static/manual IPv4 address assignment means that you manually enter in the required IPv4 address and associated IP information for a host. • When you configure a static IPv4 address, you must also configure the subnet mask and default gateway. • When you configure a static IPv4 address, you disable DHCP and APIPA. • If you use DHCP, you can also assign DNS server addresses manually. Use static addressing: • For small networks that do not often change or grow. • If your network does not have a DHCP server or if you want to eliminate DHCP traffic from your network. • For specific hosts that must have the same address each time (such as servers). You can use DHCP on the rest of the network and use static addressing for only a few hosts. However, before you use static addressing, explore the possibility of using a DHCP server to assign the same IPv4 address to specific hosts each time an address is requested. • For non-DHCP hosts (hosts that cannot accept an IPv4 address from DHCP). Ensure that duplicate addresses are not assigned to hosts on the same network.
Alternate IPv4 configuration	When an alternate IPv4 configuration is enabled, the host attempts to use DHCP for TCP/IPv4 configuration information. If a DHCP server cannot be contacted, the alternate IPv4 values are used. Use an alternate configuration: • For computers (such as a laptop) that connect to two networks: one with a DHCP server and another without a DHCP server. • To provide values to properly configure the computer in the event that the DHCP server is unavailable. When you configure an alternate IPv4 address, APIPA will never be used.

12.1 Well-Known Internet Services Ports

Port(s)	Service
20 TCP **21 TCP**	File Transfer Protocol (**FTP**)
22 TCP and **UDP**	Secure Shell (**SSH**)
23 TCP	**Telnet**
25 TCP	Simple Mail Transfer Protocol (**SMTP**)
53 TCP and **UDP**	Domain Name Server (**DNS**)
67 UDP **68 UDP**	Dynamic Host Configuration Protocol (**DHCP**)
69 UDP	Trivial File Transfer Protocol (**TFTP**)
80 TCP	HyperText Transfer Protocol (**HTTP**)
110 TCP	Post Office Protocol (**POP3**)
119 TCP	Network News Transport Protocol (**NNTP**)
123 UDP	Network Time Protocol (**NTP**)
137 UDP **138 UDP** **139 TCP**	**NetBIOS**
143 TCP and **UDP**	Internet Message Access Protocol (**IMAP4**)
161 TCP and **UDP** **162 TCP** and **UDP**	Simple Network Management Protocol (**SNMP**)
389 TCP and **UDP**	Lightweight Directory Access Protocol (**LDAP**)
443 TCP and **UDP**	HTTP with Secure Sockets Layer (**SSL**)
465 TCP	Simple Mail Transfer Protocol over TLS/SSL (**SMTPS**)
514 UDP	**Syslog** (used for remote system logging)
636 TCP and **UDP**	Lightweight Directory Access Protocol over TLS/SSL (**LDAPS**)
993 TCP	Internet Message Access Protocol over TLS/SSL (**IMAPS**)
995 TCP	Post Office Protocol 3 over TLS/SSL (**POP3S**)

12.2 Network Configuration Files

File or Directory	Description
/etc/sysconfig/network-scripts	This directory contains network configuration files. For example, each network interface in the system is configured using a configuration file in this directory named ifcfg-device_name (for example, ifcfg-ens192). Edit the appropriate device file in this directory to modify the following settings: • Boot protocol (static, DHCP, or BootP) • Autoconfiguration information • IP address, mask, and default router (for static configurations)
/var/lib/dhcpcd/dhcpclientn **/var/lib/dhcpcd/dhclient.leases** **/var/lib/dhclient/dhclient.leases**	These files contain DHCP lease information. The specific file used will vary between distributions.
/etc/netplan	This directory contains the *.yaml file that defines the network configuration parameters, such as: • On/off toggle for DHCP4 and DHCP6 • Static IP address • Gateway IP address • IP addresses of nameservers You can test the configuration using sudo netplan try. Once you confirm the configuration is accurate, use sudo netplan apply to apply the configuration.
/etc/dhcp/dhclient.conf	This configuration file defines dhclient parameters, such as: • Protocol timing • Information requested from the server • Information required from the server • Preinitialized addresses for networks without DHCP servers

12.2 Network Configuration Commands

Command	Function
/etc/init.d/network **/etc/rc.d/init.d/network** **service network**	Starts, restarts, or stops networking services on init-based distributions.
systemctl *command* **network**	Manages network services on systemd-based distributions.
ifconfig **ifconfig***interface*	Views network interface information. Use the **-a** option to display the status of all interfaces.
ifconfig *interface parameters*	Creates a static IP configuration for the specified interface. Common **ifconfig** parameters include the following: • **address** sets the IP address • **netmask** sets the subnet mask • **broadcast** sets the broadcast address • **up** activates the interface • **down** deactivates the interface
ifup interface	Starts a network interface.
ifdown interface	Stops a network interface.
ip addr show *interface*	Displays the current networking information for a network interface. Omitting the interface from the command displays networking information for all interfaces in the system. Common ip addr show parameters include the following: • **inet** shows the IPv4 address with the subnet mask in CIDR notation. • **brd** shows the broadcast address. • **up** or **down** shows the interface status. • **inet6** shows the IPv6 IP address.
ip addr add *ip_address* **dev** *interface*	Adds an additional IP address to a network interface.
ip addr del *ip_address* **dev** *interface*	Removes an IP address from a network interface.
ip link set *interface* **down** **ip link set** *interface* **up**	Stops and starts the specified interface.
ethtool	Lists and changes Ethernet card properties such as supported modes, auto-negotiation, speed, wake on, duplex, link detection status, and port. You can also use it to list and change driver settings.
iwconfig	Displays and changes the parameters of the wireless network interfaces.
brctl	Sets up, maintains, and inspects the Ethernet bridge configuration in the Linux kernel.

12.2 Network Interface Bonding (table parameters)

configuration file: **etc/sysconfig/network-scripts/ifcfg-bond*n***

(where *n* is a number that identifies the interface)

Commands: **insmod** and **modprobe**

Parameters	Description
DEVICE	Specifies a number to identify the bond.
TYPE	Specifies driver type.
STARTMODE	Specifies when the driver will be started. The options are onboot and manual. • onboot activates the bond when the system is started. • manual requires the boot to be started.
USERCTL	Allows or prohibist system users from making changes to the bond.
NM_CONTROLLED	Indicates whether the bond will be controlled by NetworkManager.
MTU	Specifies the maximum transmission unit (MTU).
BOOTPROTO	Determines the protocol type to initialize the device. There are three options: • **dhcp** causes the system to search for the DHCP server. • **bootp** indicates the boot protocol looks for a DHCP server. • **static** or **none** indicates DHCP will not be used.
IPADDR	Specifies the IP address for the bond.
SLAVE	Identifies the slave interfaces for the bonding.
BONDING_OPTS	Bonding options include the following: • **arp_interval** specifies the ARP link monitoring frequency in milliseconds. • **downdelay** specifies the time, in milliseconds, to wait before disabling a slave after a link failure has been detected. This option is valid only for the miimon link monitor. The downdelay value should be a multiple of the miimon value or it will be rounded down to the nearest multiple. The default value is 0. • **lacp_rate** specifies the rate to transmit LACPDU (LACP data unit) packets in 802.3ad mode. Slow, or 0, transmits LACPDUs every 30 seconds. Fast, or 1, transmits LACPDUs every second. Slow is the default setting. • **maxbonds** specifies the number of bonding devices to create for this instance of the bonding driver. • **miimon** specifies the MII link monitoring frequency in milliseconds. This determines how often the link state of each slave is inspected for link failures. A value of zero disables MII link monitoring. • **mode** specifies if the links function as either hot standby or load balancing services. The behavior of the single logical bonded interface is specified by the bonding driver mode. The default parameter is balance-rr. Modes can be specified by numbers as follows: ◦ **mode=0** Balance Round Robin (balance-rr): this mode transmits network packets in sequential order from the first available network interface (NIC) slave through the last. This mode provides load balancing and fault tolerance. ◦ **mode=1** Active backup: only one NIC slave in the bond is active. A different slave becomes active if the active slave fails. This mode provides fault tolerance. ◦ **mode=2** Balance XOR: this mode transmits network packets based on a hash of the packet's source and destination. The same NIC slave is used for each destination MAC address, IP address, or IP address and port combination. This mode provides load balancing and fault tolerance. ◦ **mode=3** Broadcast: this mode transmits network packets on all slave network interfaces. This mode provides fault tolerance. ◦ **mode=4** 802.3ad Dynamic link aggregation: each aggregation group shares the same speed and duplex settings. This mode is similar to the XOR mode and supports the same balancing policies. The link is set up dynamically between two LACP-supporting peers. ◦ **mode=5** Adaptive transmit load balancing (balance-tlb): This mode does not require any special network switch support. The outgoing network packet traffic is distributed according to the current load on each network interface slave. ◦ **mode=6** Adaptive load balancing (balance-alb): this mode includes balance-tlb plus receive load balancing (rlb) for IPV4 traffic. It does not require any special network switch support. The receive load balancing is achieved by ARP negotiation. • **primary** specifies which slave is the primary device. The primary device will always be the active slave while it is available. Only when the primary is offline will another device be used. • **updelay** specifies the time, in milliseconds, to wait before enabling a slave after a link recovery has been detected. The updelay value should be a multiple of the miimon value.

12.2 Slave Configuration

Parameter	Description
DEVICE	Specifies a name to identify the interface.
USERCTL	Allows or prohibit system users from making changes to the interface.
ONBOOT	Specifies whether the interface will be started at boot.
MASTER	Identifies the master associated with the slave interface.
SLAVE	Specifies that the interface is a slave.
BOOTPROTO	Determines the protocol type to initialize the device. The protocol must be the same of the protocol specified for the master. There are three options: • **dhcp** causes the system to search for the DHCP server. • **bootp** indicates the boot protocol looks for a DHCP server. • **static** or **none** indicates DHCP will not be used.

12.3 The nmcli Utility

Commands	Description
nmcli -t -f RUNNING general	Checks whether NetworkManager is running
nmcli general	Shows the state of connections
nmcli status	Lists devices and their statuses
nmcli con show	Lists available connections
nmcli hostname	Displays the hostname
nmcli general hostname <*newhostname*>	Updates the hostname
nmcli con add con-name eth2 type ethernet ifname eth2 ipv4.method auto	Creates a new Ethernet connection that uses DHCP
nmcli con mod eth2 ipv4.method manual ipv4.address 192.168.82.5/24 ipv4.gateway 192.168.82.1	Reconfigures a connection from DHCP to a static IP address

12.4 IPv6 Address Types

Address Type	Description
Link-local	Link-local addresses are addresses that are valid on only the current subnet. • Link-local addresses have a FE80::/10 prefix. This includes any address beginning with FE8, FE9, FEA, or FEB. • All nodes must have at least one link-local address, although each interface can have multiple addresses. • Routers never forward packets destined for link-local addresses to other subnets. • Link-local addresses are used for automatic address configuration, neighbor discovery, or for subnets that have no routers.
Unique local	Unique local addresses are private addresses used for communication within a site or between a limited number of sites. In other words, unique local addressing is commonly used for network communications within an organization that do not cross a public network. They are the equivalent of private addressing in IPv4. • Unique local addresses have a FC00::/7 prefix. Currently, however, the 8th bit is always set to 1 to indicate that the address is local (and not global). Thus, addresses beginning with FC or FD are unique local addresses. • Following the prefix, the next 40 bits are used for the global ID. The global ID is generated randomly so that there is a high probability of uniqueness on the entire internet. • Following the global ID, the remaining 16 bits in the prefix are used for subnet information. • Unique local addresses are likely to be globally unique, but are not globally routable. Unique local addresses might be routed between sites by a local ISP. • Earlier IPv6 specifications defined a site-local address that was not globally unique and had a FEC0::/10 prefix. The site-local address has been replaced with the unique local address. • Because unique local addresses are not registered with IANA, they cannot be used on a public network (such as the internet) without address translation.
Global unicast	The process for designing a network addressing scheme when using unique local addresses is similar to that used for global unicast addresses. The key difference is how the prefix is defined. Because the address range is not registered, a global routing prefix does not have to be requested from an ISP. Instead, each organization defines the prefix to be used for their organization. However, there are several requirements that need to be observed when doing so. As with global unicast addressing, using this addressing scheme allows organizations to define a large number (216) of IPv6 subnets. Global unicast addresses are addresses that are assigned to individual interfaces that are globally unique (unique throughout the entire internet). Global unicast addresses are any addresses that are not link-local, unique local, or multicast addresses. Originally, ISPs assigned global unicast addresses with a 2000::/3 prefix (this includes any address beginning with a 2 or a 3). However, this was later amended so that all IPv6 addresses that haven't been specifically reserved for other purposes are defined as global unicast addresses. The global routing prefix assigned to an organization by an ISP is typically 48 bits long (/48). However, it could be as short as /32 or as long as /56, depending upon the ISP. Using this addressing scheme allows organizations to define a large number (216) of IPv6 subnets.
Loopback	The local loopback address for the local host is 0:0:0:0:0:0:0:1 (also identified as ::1 or ::1/128). The local loopback address is not assigned to an interface. It can be used to verify that the TCP/IP protocol stack has been properly installed on the host.

12.4 IPv6 Configuration Methods

Method	Description
Static full assignment	Static full assignment is where the entire 128-bit IPv6 address and all other configuration information is statically assigned to the host.
Static partial assignment	Static partial assignment is where the prefix is statically assigned and the interface ID uses the modified EUI-64 format derived from the MAC address.
Stateless autoconfiguration	Stateless autoconfiguration is where clients automatically generate the interface ID and learn the subnet prefix and default gateway through Neighbor Discovery Protocol (NDP). NDP uses the following messages for autoconfiguration: • Router solicitation (RS) is a message sent by the client to request that the routers respond. • Router advertisement (RA) is a message sent by the router periodically and in response to RS messages to inform clients of the IPv6 subnet prefix and the default gateway address. NDP is also used by hosts to discover the address of other interfaces on the network, replacing the need for Address Resolution Protocol (ARP). Even though NDP provides enough information for the addressing of the client and for clients to learn the addresses of other clients on the network, NDP does not provide the client with DNS server information or other IP configuration information besides the IP address and the default gateway.
DHCPv6	IPv6 uses an updated version of DHCP (called DHCPv6) that operates in one of two different modes: • Stateful DHCPv6 is used when the DHCP server provides each client with the IP address, default gateway, and other IP configuration information (such as the DNS server IP address). The DHCP server tracks the status (or state) of the client. • Stateless DHCPv6 does not provide the client with an IP address and does not track the status of each client, but rather is used to supply the client with the DNS server IP address. Stateless DHCPv6 is most useful when used in conjunction with stateless autoconfiguration.

12.5 Commands to Configure Routing

Command	Function
route add	Adds a static route in the routing table. Use the following options: • **default gw** creates a route for the default router • **-net** specifies a network address • **-host** specifies a single host on the network • **reject** installs a blocking route
route del	Deletes a static route in the routing table.
route	Views the routing table, including the default gateway address.
ip route show	Views routes in the routing table.
ip route add	Adds a route to the routing table. Use the following options: • ***network*** specifies the address of the remote network. Be sure to include the prefix of the network using CIDR notation • **via *router_IP*** specifies the router to which packets addressed to the remote route should be sent • **dev *interfac e*** specifies the network interface to which the new route will be applied
ip route del *network*	Removes a route from the routing table. Replace network with the network address of the route to be removed. Be sure to include the prefix of the network using CIDR notation.

12.6 Name Resolution Settings Configuration

Command	Function
/etc/hosts	When a host needs to resolve a hostname into an IP address, it reads this file first by default. If it finds a mapping for the hostname, it uses it and the name resolution process ends. If it does not find a mapping, it sends the name resolution request to a DNS server.
/etc/resolv.conf	Provides the system with the address of a DNS server that can be used for name resolution. Up to three servers can be listed, and the servers are accessed in the order specified. The file can also specify a fully qualified domain name that will be appended to hostnames that are missing a domain name. *In recent versions of many Linux distributions, the resolv.conf file is generated from other files and is modified there, not directly in resolv.conf.*
/etc/nsswitch.conf	Specifies whether the computer's hosts file or the DNS server takes precedence.
/etc/hostname	Defines the host and domain names.
host	Finds the IP address for a domain name.
hostname	Displays or sets the name of the local host for the current session.
dig	Domain Information Groper (dig) is a command line tool that lets you query Domain Name System (DNS) name servers and displays the answers that are returned from the name server(s) that are queried. dig is useful for verifying and troubleshooting DNS problems. It can also be used to perform DNS lookups to display the answers that are returned from the name server that were queried.
nslookup	This is a program to query internet domain name servers. The nslookup command has two modes, called interactive and non-interactive. Interactive mode allows the user to query name servers for information about various hosts and domains or to print a list of hosts in a domain. Non-interactive mode allows the user to print just the name and requested information for a host or domain. You use non-interactive mode when the name or internet address of the host you want to look up is the first argument. To use nslookup: **1.** Enter **nslookup** at the shell prompt. **2.** Enter the ***hostname*** or ***IP address*** , such as 192.168.1.1 **3.** The DNS server should respond with the requested mapping. **4.** Type **exit** when finished.

12.8 Network Communications Troubleshooting Tools

Tool	Function
ping	Verifies connectivity between hosts within the network. • Ping a host using its IP address. If there is no response, try to ping another host. ◦ If your computer cannot communicate with any other computer, check the network cable, the network interface card, or the IP address configuration on your computer. ◦ If your computer can communicate with computers on the local network, but can't communicate with remote computers (such as the internet), verify the default gateway configuration on your computer. ◦ If all computers on the local network cannot communicate with any remote computer, troubleshoot the router's connection to the remote network. • Ping a host using its DNS name. If a ping by IP address works, but a ping by DNS name fails, there is probably a name resolution problem. • Use the **-c** option to specify how many ICMP echo requests to send to the destination. IPv6 communications can be tested using ping, but the **ping6** command must be used.
netstat	Displays a list of network connections (e.g., sockets), the routing table, and information about the network interface. A socket is an endpoint of a bidirectional communication flow across a computer network. Use the following options for additional information: **-a** lists both listening and non-listening ports. **-i** displays a table of all network interfaces. -l lists listening sockets. **-s** displays statistics for each protocol. **-r** displays the routing table, which includes the IP address of the default gateway. The netstat command is being replaced by **ss**, **ip route** (for **netstat -r**), **ip -s link** (for **netstat -i**), and **ip maddr** (for **netstat -g**)
nc **ncat**	Tests communications between network hosts. The netcat (nc or ncat) command establishes a TCP or UDP connection between two computers. The procedure for using nc is to: • Open a listening TCP or UDP socket on one host. The syntax is **nc -l *port_number*** . The **-l** option tells netcat to wait and listen for incoming connections. If no protocol is specified, TCP is used by default. To use UDP instead of TCP, include the **-u** option in the command. • Connect to the listening socket on the first host from another host. The syntax is **nc *ip_address port_number*** . After the connection is established, text entered at the prompt of the second computer should appear on the screen of the first computer. You must open the appropriate ports in the host firewalls of both systems.
traceroute **tracepath**	Tests connectivity between devices and shows the path between them. The traceroute command: • Can help track down which router (known as a hop) in the route is not working correctly. • Displays the round trip time (RTT) for each hop. The RTT is the time difference between when the probe was sent from traceroute and the time the response arrived for each packet. **tracepath** is similar to **traceroute**, but does not require super user privileges. To test IPv6 routing, use the **traceroute6** or **tracepath6** commands instead of traceroute.
nslookup	Sends a name resolution request. To use nslookup: **1.** Enter **nslookup** at the shell prompt. **2.** Enter the ***hostname*** or ***IP address*** (such as 192.168.1.1). **3.** The DNS server should respond with the requested mapping. **4.** Enter **exit** when finished. *The **nslookup** command is being replaced by the **host** and **dig** commands.*
dig	Sends a name resolution request and receives extensive information about the hostname or IP address. Consider the following options: • **a** resolves a record information. • **ptr** resolves a PTR record. • **cname** resolves CNAME record information. • **p** queries a specific port on the host. • **in** resolves internet record information. • **mx** resolves MX record information. • **soa** resolves start of authority information.

ss	Dumps socket statistics and provides detailed information about communication with other hosts, networks, services, network connections, networking protocol statistics, and Linux socket connections. Consider the following options: **-a** displays all sockets. **-t** displays only TCP sockets. **-u** displays only UDP sockets. **-l** displays listening sockets. **-m** shows socket memory usage. **-p** shows process using socket. **ss > *ss_output*** sends the output to a file.
nmcli	Controls NetworkManager and get its status from the command line. Use nmcli as a complementary utility to nm-applet or other similar clients. Its main usage is on servers, headless machines, or for power users. Consider the following options: **-t** displays terse output. The output is suitable for scripts. **-p** displays pretty output that is easily readable by humans. **-m** specifies mode (tabular or multiline). **-f** specifies column names.
nmtui	Provides a text-base interface for controlling NetworkManager. Consider the following options: **-edit** displays a connection editor that supports adding, modifying, viewing, and deleting connections. **-connect** displays a list of available connections with options to activate or deactivate them. **-hostname** sets the system hostname.
iftop	Listens to network traffic on a named interface. If no interface is named, it listens on the first interface that looks like an external interface. Be aware that ifhop: • Displays a table of current bandwidth usage by pairs of hosts. • Must be run with sufficient permissions to monitor all network traffic on the interface. • Looks up the hostnames associated with addresses it finds in packets. This can cause substantial traffic. You can suppress the display of DNS traffic by using filter codes or switch it off with the -n option. • Includes some of the following options: **-r** suppresses display of DNS traffic when the program is running. **-h** prints a summary of usage. **-N** suppresses resolving port number to service names. **-p** runs in promiscuous mode. It counts traffic that does not pass directly through the specified interface. **-f** allows you to specify filters.
iperf	Performs network throughput measurements. To perform an iperf test, the user must establish both a server (to discard traffic) and a client (to generate traffic). With iperf: **-f** specifies report format: [kmKM] Kbits, Mbits, KBytes, MBytes. **-i** pauses n seconds between periodic bandwidth reports. **-l** sets length read/write buffer (default 8 KB). **-o** specifies output filename for the report or error message. **-p** sets server port to listen on/connect to (default 5001). **-u** uses UDP rather than TCP.
tcpdump	Dumps traffic on a network. Options include: **-A** prints each packet without the link level header in ASCII. **-B** sets the operating system capture buffer size. **-c** exits after receiving count packets. **-d** dumps the compiled packet-matching code in a human readable form to standard output and stops. **-dd** dumps packet-matching code as a C program fragment. **-D** prints the list of the network interfaces available on which tcpdump can capture packets.
ipset	Sets up, maintains, and inspects IP sets in the Linux kernel. Consider the following options: • **-n** creates a set identified with setname and specified type. • **add** adds a given entry to the set. • **del** deletes the specified entry from a set. • **test** tests whether an entry is set or not. • **-x** destroys specified set or all sets if no set is specified. **-t** lists the set names and header; suppresses listing set members..

mtr	Combines the functionality of the traceroute and ping commands in a single network diagnostic tool. Consider the following options: **-r** puts mtr into report mode, causing it to run for the number of cycles specified by the -c option, print statistics, and exit. **-t** forces mtr to use the curses based terminal interface if it is available. **-n** forces mtr to display numeric IP numbers and not try to resolve the host names. **-u** uses UDP datagrams instead of ICMP ECHO. **-4** uses IPv4 only. **-6** uses IPv6 only.
arp	Displays and modifies the internet-to-Ethernet address translation tables used by the Address Resolution Protocol, or ARP. Consider the following options: **-a** displays all of the current ARP entries. **-d** deletes the entry for the specified hostname. When combined with -a, this option deletes all entries and automatically disables hostname lookups. **-f** processes entries in the specified file to be set in the ARP tables. **-F** overwrites entries for a given host when used with -f. **-s** creates an ARP entry for the specified host and Ethernet address. This option is used with the -f option. *The arp command is being replaced by ip n.*
whois	Looks up records in the databases maintained by several network information centers (NICs) Consider the following options: **-a** uses the American Registry for Internet Numbers (ARIN) database. **-d** uses the US Department of Defense database. **-g** uses the US non-military federal government database, which contains points of contact for subdomains of .GOV. **-h** uses the specified host instead of the default NIC. Either a hostname or an IP address may be specified.
Wireshark **tshark**	Tracks, intercepts, and logs network traffic. It can also generate a customized report from captured data. Use a CLI version of the Wireshark packet analyzer. Consider the following options: **-i** specifies the name or index number of the interface. **-s** specifies the packet snapshot length. **-y** specifies the link type. **-c** stops capture after a specified number of packets. **-i** specifies a file to read from. **-i** specifies a file to output to. **-2** performs two-pass analysis.