Last updated September, 2000. Click here for updates.

Click here to see additional documents related to this study guide.

Contents

Cramsession™ for Sun Certified Network Administrator for Solaris 7

Abstract:

This Cramsession will help you to prepare for Sun exam 310-042, Sun Certified Network Administrator for Solaris 7. Exam topics include the OSI and TCP/IP models, Ethernet hardware, routing, DHCP, DNS, Sendmail and Troubleshooting hardware and software in TCP/IP LAN environments.

NETWORK MODELS

Describe each layer in the ISO/OSI network model

The OSI model can be described as a division of labor. Each layer accepts information from a lower neighbor, processes it, and passes it along to the next highest layer. From highest to lowest:

Application Layer
Represents the user level. TELNET, FTP, SMTP, NFS

Presentation Layer
Different computers interpret information in different ways. The presentation layer facilitates the encoding and decoding required between platforms. Examples would include ASCII and EBCDIC. XDR (external data representation) resides at this layer

Session Layer
The session layer manages services like authentication, dialog management and encryption between connected clients. It also reestablishes interrupted connections

Transport Layer
Handles transport-specific functions like flow-control and quality between two communicating stations

Network Layer
The network layer addresses, routes, and delivers data traffic on a network. Routing tables are found at this layer

Data Link Layer
This layer addresses the physical medium directly. This is the first location where bits are arranged into a recognizable format. Checksum error detection occurs here. MAC addresses

Physical Layer
Operating at the lowest level, this layer moves unstructured bit streams using electrical signals

Describe each layer in the TCP/IP network model

Sun's implementation of the TCP/IP protocol stack includes 5 layers:

Application Layer
User-accessed application programs and network services

Transport Layer
Connection-oriented TCP and connectionless UDP data transfer

Internet Layer
Data is fragmented, addressed, and routed at the Internet layer

Network Interface Layer
Error detection and packet framing. 802.3, 802.4, 802.5

Hardware Layer
Electrical signals that move raw bits through the ether

Identify the similarities and differences between the ISO/OSI and TCP/IP models

<table>
<tr><th>OSI</th><th>TCP/IP</th><th>Encapsulation</th></tr>
<tr><td>Application</td><td rowspan="3">Application</td><td rowspan="3">Data</td></tr>
<tr><td>Presentation</td></tr>
<tr><td>Session</td></tr>
<tr><td>Transport</td><td>Transport</td><td>Segment</td></tr>
<tr><td>Network</td><td>Internet</td><td>Packet</td></tr>
<tr><td>Data Link</td><td>Network</td><td>Fragment</td></tr>
<tr><td>Physical</td><td>Physical</td><td>Bit(s)</td></tr>
</table>

Describe how applications use TCP/IP to exchange data through Ethernet networks

TCP/IP is protocol based. The sender and receiver both understand predefined methods of communicating, and expect traffic to adhere to fairly strict guidelines. Several tricks are used to add intelligence to the exchange of data, which include special syntax (coding), semantics (control and error handling) and timing (speed and sequence synchronization).

Describe the following protocols: TCP, UDP, IP, and Internet control message protocol (ICMP)

TCP – Connection-oriented. The hallmark of TCP is the acknowledgement between systems that ensures all the sent data was received. Can operate two-way (full-duplex)
UDP – Connectionless. UDP packets traverse the network by themselves when they leave the sender. The receiver and sender do not communicate about the status of

UDP packets. Faster and easier to implement than TCP
IP – Determines a packets path based on the listed destination IP

ICMP – Communicates system status and error messages using IP datagrams

Describe peer-to-peer communications

Communication between network clients is known as peer-to-peer. On each layer, the sender communicates (albeit indirectly) with the corresponding layer on the receiver. Data is sliced up and identified by encapsulating the data within headers and trailers added by each layer.

Identify common TCP/IP protocols by name and function

- **SLIP/PPP** – IP communication over serial links
- **ARP/RARP** – Method used to convert between MAC hardware addresses and IP addresses
- **DNS** – Domain Name Service, a database of hostnames to IP addresses
- **FTP** – File Transfer Protocol, to exchange files between systems
- **DHCP** – Dynamically assign IP addresses to hosts on a network
- **SMTP** – Simple Mail Transport Protocol, manages the exchange of email
- **SNMP** – Simple Network Management Protocol, monitor and control network devices
- **POP3/IMAP4** – relays email from a central server to (roving) clients

INTRODUCTION TO LOCAL AREA NETWORKS

Describe the benefits of a LAN

Resource Sharing – Files, printers, etc
Management – Centralized resources, decentralized clients
Data Access – Workgroup sharing and between remote sites

Identify various LAN topologies

Bus – One coaxial cable with BNC taps cut into it for each machine
Star – A central hub with a segment of cable running between it and each client
Ring – Each node is connected 'serially' to the next node. A 'token' must be passed around to allow each device to communicate

List the components of a LAN

Backbone – the common, shared method of connectivity
Repeater – A simple device that regenerates the signal on a lengthy wire
Hub – A device that forms the center of a twisted-pair network
Bridge – A link-layer device that connects two or more network segments
Switch – A link-layer device that dedicates traffic between two senders
Router – A device that examines addresses and selects optimal paths
Gateway – Interconnects two networks with uncommon protocols
Segment – A length of cable that connects a point-to-point component
Concentrator – Provides multiple functions between segments and nets

Define the following networking terms: topology, backbone, segment, repeater, bridge, router, and gateway

Topology – A description of the various devices and components that make up a network.

ETHERNET INTERFACE

Define the following terms: Ethernet, packet, and maximum transfer unit (MTU)

Ethernet is the most popular LAN technology. It consists of units of data (packets) traversing physical cabling and circuitry regulated by a flow-control protocol called CSMA/CD.

The MTU of a network is a hardware-mandated value that specifies the largest amount of data that can be transmitted within a packet.

Describe Ethernet addresses

An Ethernet address is a 48-bit unique identifier for each network device. The standard is administered by the IEEE and designates the first three octets (in bold) as vendor-specific. An example: **00:10:A4**:EB:AD:87, the 48-bit representation is **00000000:00010000:10100100:**11101011:10101101:10000111

List the components of an Ethernet frame

Preamble – 64 bits – Used for synchronization
Destination Address – 48 bits – Ethernet address of the destination
Source Address – 48 bits – Ethernet address of the source
Type – 16 bits – Describes the data in the frame (IP, ICMP, ARP, RARP)
Data – 1500 bytes – The original data from the sending application
CRC – 32 bits – Value for error checking, calculated from frame contents

An Ethernet frame that is less than 46 bytes in size is known as a **runt**. A frame greater than the MTU is known as a **jabber**.

Define encapsulation

Encapsulation is the nesting of one data structure within another. Each layer prefixes or appends a header or trailer containing control information to the data. The corresponding layer on the receiving end interprets and removes this information.

Describe the purpose of Carrier Sense, Multiple Access/Collision Detection (CSMA/CD)

Hosts on a network transmit whenever they want to. In the event that two hosts choose the exact same interval to transmit, a collision signal causes each host to 'back-off' and wait to retransmit according to some algorithm.

The more hosts on a network, the more transmissions (and collisions) will occur (increasing exponentially until almost every transmit results in a collision). Modern computer hardware does make use of more intelligent componentry to limit the damage done by collisions on a network.

Determine an Ethernet broadcast address

A broadcast address is indicated by an Ethernet frame that has a destination address of all 1s (FF:FF:FF:FF:FF:FF). The message is received by all hosts on the subnet.

Use the commands netstat and snoop

`netstat -i` displays the state of the Ethernet interfaces on a host. It displays information like IP address, MTU size, number of input and output packets, as well as the number collisions and errors.

```
Name  Mtu   Net/Dest  Address    Ipkts  Ierrs  Opkts  Oerrs  Collis  Queue
lo0   8232  loopback  localhost  75036  0      75036  0      0       0
le0   1500  sparty    sparty     20584  145    55808  85     2134    0
```

`snoop` captures network packets, and displays their output to stdin or redirected to a file. It can be run in standard mode, where it displays high-level packet information, or in verbose mode, where it shows everything. Only the superuser (root) can run snoop.

ARP AND RARP

Define address resolution

Address resolution is the translation of 32-bit IP addresses to 48-bit Ethernet addresses (and vise versa). The mapping occurs between the Network layer and the Internet layer.

Describe the process used to map a destination Internet address to a destination Ethernet address

A sending host must complete each field of a frame it transmits. To obtain the destination Ethernet address, the host must send a broadcast alerting other hosts to its request, in the hopes that it will receive a matching reply.

The request contains the sender's IP address and MAC address, and the target's IP address. The host with the matching target IP address then replies directly back to the source. ARP will also cache the response.

If none of the hosts on the local subnet have the matching IP address, a router or gateway can be contacted to assist the requesting host.

Describe the process used to map a destination Ethernet address to a destination Internet address

This process is called RARP (ReverseARP), and is the opposite of the IP to Ethernet address mapping described above. Diskless clients, or new systems utilizing Jumpstart™, benefit the most from this protocol because the IP address is not recorded locally. A special server on the local subnet responds to RARP packets with an unused IP number from a pool of reserved addresses. This server is running the `in.rarpd` daemon.

INTERNET LAYER

Define the terms: IP, datagrams, and fragmentation

IP is the built-in protocol in charge of sending data across a network. The basic unit transferred by IP is called a datagram. Each IP datagram is limited to a certain number of bytes (the MTU) by the transmission medium it crosses. Fragmentation occurs at the router when data must fit into multiple Ethernet frames.

Describe the four IPv4 address classes

An IP address is a 32-bit number laid out in 8 bit fields, called octets. Each IP is unique to a particular host and network. The four default classes are:

Class A – Eight bits in the first octet define the network; 127 usable networks.
Class B – Sixteen bits in the first and second octet define the network; 16,384 usable networks.
Class C – Twenty-one bits in the first three octets define the network; 2,097,152 usable networks
Class D – The first 4 bits are 1110, which results in a first octet value of 224 – 239. The remaining bits define a multicast group.

IPV4 was first described in RFC 791

Define the three standard netmasks

A network mask is used in conjunction with an IP address to compute a network number for hosts on a subnet. The defaults are:
Class A – 255.0.0.0
Class B – 255.255.0.0
Class C – 255.255.255.0

Define the network number

The network number is used to determine if the destination host is on a local subnet. It is computed by using the logical AND operator on the IP address and its associated subnet mask. If the operation does not result in a match, the packet needs to be sent to a remote network, and is forwarded to the gateway for the subnet.

Determine the benefits of Variable Length Subnet Masks (VLSM)

Subnet masks logically segment a network to reduce collision domains (and the associated traffic), confine network protocols to segments, and logically associate departments and administrative functions in particular regions.

The default subnet masks are relatively limited because they define networks that are either too large or too small to be practical. VLSM builds on the idea of subnetting by allowing an organization to nest subnets within a particular class of address space.

For example, subnetting the network **10.0.0.0 / 255.255.0.0** would designate the second octet to be treated as a network, yielding networks 10.1.0.0 – 10.255.0.0.

```
IP Address:     00001010.00000000.00000000.00000000
Subnet Mask:    11111111.11111111.00000000.00000000
```

Further subnetting the network with a **24-bit** mask **255.255.255.0** would mark the 3rd octet as being associated with the network, allowing 10.subnet.subnet (i.e. 10.1.55.0).

```
IP Address:     00001010.00000001.00110111.00000000
Subnet Mask:    11111111.11111111.11111111.00000000
```

For more granularity, one more level of subnetting with a **27-bit** subnet mask 255.255.255.224 would reveal subnets 10.1.55.32 - 10.1.55.224. The remaining 5 bits would be associated with hosts on each of the subnetworks.

```
IP Address:     00001010.00000001.00110111.00000000
Subnet Mask:    11111111.11111111.11111111.11100000
```

VLSM is implemented using the **/etc/netmasks** file in Solaris™.

Configure files for automatic start-up of network interfaces

To configure devices to automatically obtain network information, several files must be consulted:
/etc/hostname.XXX – Where XXX is the name of the interface. This file contains the name of the machine.
/etc/hosts – When the hostname is associated with an IP address in this file, the system reads it and assigns it to the interface specified by `/etc/hostname.XXX`.
/etc/netmasks – Stores netmask information for local subnets.

Upon startup, the script `/etc/rcS.d/S30rootusr.sh` consults the files named above to assign network information. Verification of configured network information can be found by using `ifconfig –a`.

Use the ifconfig command to configure the network interface(s)

Basic form: `ifconfig <interface name>`

Parameters can be sent to the interface using syntax like:
`ifconfig <name> inet <ip adr> netmask <mask> broadcast + up`

WHERE: inet is the IP address, mask is the subnet mask, and broadcast + indicates a value that should be computed by the machine using whatever inet and netmask information is provided.

An interface is enabled using the command: `ifconfig <name> plumb`, and disabled using `ifconfig <name> unplumb`

Verify the network interface

```
#ifconfig –a
lo0: flags=849<UP,LOOPBACK,RUNNING,MULTICAST> mtu 8232
        inet 127.0.0.1 netmask ff000000
le0: flags=863<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST> mtu 1500
        inet 10.1.15.10 netmask ffffff00 broadcast 10.1.15.255
```

Then, ping other hosts on the network.

ROUTING

Describe the routing algorithm

Solaris™ handles routing in the following manner:

The kernel **checks to see if the destination is the local LAN** by computing a network number from the destination IP address and netmask.
If the result of this first computation is not the network number of a locally attached interface, the **routing table is checked for a matching IP address**.
Next, the **routing table is searched for a matching network number**.
Without a network number match, the **default entry of the routing table** is consulted for resolution.
If all else fails, an **ICMP message like "no route to host" is generated**.

Define the following routing terms: table-driven routing, static routing, dynamic routing, and default routing

Table-driven routing – The local host contains a list of devices to which it can forward packets
Static Routing – Routing is defined directly to local hosts or paths to networks that are not local. Static routes are permanent until the next reboot
Dynamic routing – Optimal paths to remote networks change as the environment changes. Routers and clients listen to broadcasts from other hosts to determine the best paths for the traffic they forward
Default routing – The entry in the routing table that contains the best information for sending the packet to its destination. It does not need to have direct contact information about a remote network – it only needs to know which device would have more complete information

Describe the in.routed and in.rdisc processes

`in.routed` implements RIP for Solaris™. A host is configured with the `-q` option, and a router is configured with the `-s` option.

`in.rdisc` implements RDISC in Solaris™. Hosts are configured using the `-s` option, and routers are configured using the `-r` option.

Describe the Routing Information Protocol (RIP) and the Router Discovery (RDISC) protocols

RIP is a distance-vector protocol that associates paths to a destination using the concept of "least cost". It is broadcast based and updates hosts on optimal routes every 30 seconds. When multiple paths to a destination are available, RIP only advertises the path that has the smallest metric (or fewest hops to that destination). Advanced features include hop-count limits (15), hold-downs, split-horizons, and route poisoning.

RDISC maintains default paths to hosts. It is a routing protocol independent network protocol that advertises and listens on the multicast address 224.0.0.1 at 10-minute intervals. It can contain multiple routes to destinations, which increases redundancy.

Describe the /etc/init.d/inetinit routing start-up script

/etc/init.d/inetinit is the script that checks for the existence of certain files in order to make decisions on how to start routing daemons.

If it finds **/etc/defaultrouter**, it creates static routes in the route table and prevents the starting of in.routed or in.rdisc.

If the machine is configured for DHCP, or has the **/etc/notrouter** file, `ip_forwarding` is disabled and `in.routed -q` or `in.rdisc -s` is started.

If there are two IP addresses, or entries in the **/etc/gateways** file, `ip_forwarding` is enabled and the `in.routed -s` (RIP) or `in.rdisc -r` (RDISC) processes start.

Describe the /etc/defaultrouter, /etc/inet/networks, and /etc/gateways files

/etc/defaultrouter – A file that contains hostname or IP addresses of one or more routes on the network
/etc/inet/networks – A file similar to hosts, which assigns names to network numbers
/etc/gateways – An optional file that defines additional passive routes alongside the default routes. Read by `in.routed`

Use the route and netstat commands

route is used to manage routing tables. Some common options:

`route add net <ipaddress> <host> <metric>` - Adds a new route
`route add default <netname>` - Adds a new default route
`route monitor` – Displays routing reports
`route flush` – Clears the routing table

netstat is used to view the current routing table with the `-r` option. Specifying `-rn` will display the table without hostname resolution:

```
Routing Table:
Destination  Gateway      Flags   Ref   Use        Interface
-------------------------------------------------------------
10.1.15.0    10.1.15.10   U       3     1496       le0
224.0.0.0    10.1.15.10   U       3     0          le0
default      10.1.15.1    UG      0     375795
127.0.0.1    127.0.0.1    UH      0     17677      lo0
```

Configure a Sun system as a router

Add a second NIC to the system, and touch the `/reconfigure` file to force Solaris™ to detect new hardware at reboot.

Setup the interface(s) using `ifconfig` and the network configuration files (`/etc/hostname.xxx` and `/etc/hosts`).

Verify that the `in.routed` or `in.rdisc` daemons are running using the `ps –ef` command.

Monitor the routing tables on the machine using `netstat –nr`.

TRANSPORT LAYER

Describe the function of the Transport layer

The transport layer manages end-to-end communication between applications on different hosts. It directs traffic and the sequencing of data by using source and destination port numbers, which are agreed upon by each host before transmission begins. Common features of the protocols it supports are error detection, recovery, and flow-control.

Describe the features of the UDP and TCP

UDP is a connectionless protocol. It trades reliability for speed and, therefore, requires little setup or monitoring by the sender and the receiver. Applications using UDP are responsible for sequencing and message loss. It is functionally similar to the postal service, where letters (packets) are simply addressed to the receiver and left to the network for delivery.

UDP is described in RFC 768

TCP is a connection-oriented protocol. It requires elaborate setup information (known as virtual circuits) before transmission can begin, so that each partner is guaranteed to receive the information it was sent. It uses buffering and full-duplex connection streams. Because of this additional overhead, it is slower and requires more processing power. It is like the phone system, where two callers must be present and agree to talk before communication can begin.

TCP is described in RFC 793

Define the terms: connection-oriented, connectionless, stateful, and stateless

Connection-Oriented transmissions place priority on the quality of data. TCP employs a system of acknowledgements to ensure 100% of the data transmitted is received. Connectionless transmissions are concerned with data quantity. UDP does not track packets as they are sent and received, and can achieve much faster data rates.

Stateful means that the client and server communicate about the information they send and the quality of the data they receive. Stateless means that no such communication occurs, and the client and server operate independent of each other or the current network condition.

CLIENT-SERVER MODEL

Define the terms client, server, and service

The terms client, server, and service reference the relationship between these components over a network, and at a level of interaction found between the highest two layers, application and transport. Clients needing information establish a connection to a known service on a server.

Describe ONC+ technologies

ONC+™ stands for Open Network Computing, and is Sun's brand of technologies, services and tools for building open systems client/server software. It includes XDR, TLI, Sockets, NIS, and NFS.

Define a port and a port number

A port is an abstract representation of an address set aside for a service running on a server. Servers provide certain services over well-known port numbers (Telnet 23, FTP 21, HTTP 80) to requesting clients. A port is defined by software within the kernel designed to accept client connections.

Describe the client-server interaction

A client needing shared resources asks a server using a well-known server port number. It includes in its request the number of an arbitrary port on which it expects to receive the reply. As long as the ports are agreed upon ahead of time, any number can be used.
Servers listen for requests on their well-known port numbers. A daemon listening to the published port number will establish the session with the client using another unused, higher-numbered port. This way additional requests can come in on the well-known port.
The well-known port numbers are configured in the `/etc/services` file.

Describe Internet and RPC services

Internet processes and services are those managed by the `inetd` process. inet starts listening on established port numbers at boot time. When a request is made to a well-known port, the daemon assigned to that port is launched. Internet service daemons are `named in.<daemon>`. Examples include ftp, telnet, rlogin, and talk.

RPC services do not require predefined, unique port numbers to be established at boot time. A process, `rpcbind`, interprets the request and sends it to the appropriate server process. Using RPC, clients are given the actual port number at connection time by `rpcbind` (listening at well-known port 111). RPC services

register themselves with `rpcbind` when they start and are assigned an available port number at that time. RPC services are named `rpc.<daemon>`.

Identify the files used in the client-server model

Internet services on a server use the `/etc/services` file to define well-known ports. The service processes to be started are bound to ports in the `/etc/inet/inetd.conf` file.

RPC services do not require listings in any particular file. The `rpcbind` program returns the actual port number for a requested service at connection time. An error: "`RPC: Program not registered`" is generated if the service cannot be found.

Add and remove Internet services

Internet services are not started during bootup. Each process is started by inetd on behalf of the client. Internet services are managed using the following files:

Inetd is started by the script `/etc/init.d/inetsvc`

It reads from a file `/etc/inet/inetd.conf`, which contains information about what network services are at which well-known port and the location of the process to start.

The `/etc/services` file identifies and registers well-known port numbers, services, and protocols on the machine. The standards are managed by the NIC, but an administrator can add his or her own at any time.

Add and remove RPC services

RPC services are registered with the `rpcbind` process, and do not require special setup in the `/etc/services` file. Some processes are started at boot time, others at client request. `rpcbind` is started at runlevel 2 using `/etc/init.d/rpc`.

The program `/usr/bin/rpcinfo` can be used to monitor the activities of RPC.

Use the commands netstat and rpcinfo to monitor services

`netstat –a` identifies ports on a host and what connections are established

`rpcinfo` displays the program number, version, protocol, port number, service, and owner

`rpcinfo –p <hostname>` shows all RPC services running on a host

`rpcinfo –u <server> <process>` shows if a specific server is running on a host

`rpcinfo –b <process>` broadcasts a request to hosts on a network to see if the specified process is running, and displays their machine name and port

`rpcinfo –d <process>` unregisters a service on a host

DHCP

List the benefits of DHCP

DHCP eases network IP address management by allowing administrators to dynamically configure network information for clients from a centrally administered server. These benefits reduce cost associated with network management, as well as help alleviate the problem of IP address depletion.

DHCP is described in RFC 2131

Define DHCP client functions

DHCP clients are assigned addresses according to a "leasing policy". This means when the client broadcasts its Ethernet address to the network, a DHCP server listening for this request hands out a temporary, but valid, IP address from a pool of pre-allocated addresses. When the machine goes offline, the address is returned. Clients can also move about a network and obtain valid network configuration automatically.

Clients can be setup to use DHCP by using `ifconfig <interface> DHCP` or by 'touching' a file `/etc/dhcp.<interfacename>`

Define DHCP server functions

There are two types of DHCP servers, primary and secondary. DHCP servers listen for network requests using RARP (the daemon `in.rarpd`). Using RARP, the DHCP server maintains a list of the IP address space for networks it is directly attached to. A server can also be configured as a relay agent, which allows DHCP requests to be forwarded between networks. DHCP servers can be configured using the `dhcpconfig` utility.

Choose the appropriate DHCP datastore for your network environment

Two options are available, using files or using NIS+.

The files option has two databases `dhcp_network` and `dhcptab`. These are located in the `/var/dhcp` directory.

The databases can also be managed by NIS+, where they can be coordinated in a multi-server environment.

Customize the DHCP datastore files dhcptab and dhcp_network by using the dhtadm program

`dhcptab` contains the macro table used for DHCP clients. It has three fields, Name, Type, and Value.

Name – user-defined name for the record
Type – (s)ymbol or (m)acro
Value – value pairs that define the symbol (delimited by ',') or macro (delimited by ':')

```
Name     Type  Value
net-30   s     \
              :Timeserv=10.30.86.2:LeaseTim=259000: \
              :DNSdmain=mydomain.com:DNSserv=10.30.86.2 \
              10.30.89.2:LeaseNeg
```

`dhtadm` has several flags for creating entries in the `dhcptab` file.

`dhtadm -C` creates a `dhcptab` configuration file
`dhtadm -A` adds a macro definition to `dhcptab`
`dhtadm -M` modifies an existing macro or symbol in `dhcptab`
`dhtadm -D` deletes a macro definition
`dhtadm –R` removes the `dhcptab` file

`pntadm` manages the `dhcp_network` file.

`pntadm –C` creates a `dhcp_network` file
`pntadm –A` adds a client entry to `dhcp_network`
`pntadm –D` deletes a specified client entry
`pntadm –P` prints the `dhcp_network` file
`pntadm –R` removes the `dhcp_network` file

An entry in dhcp_network might look like the following:

```
Client_ID  Flags  Client_IP     Server_IP    Lease  Macro
10         00     10.30.86.105  10.30.86.15  0      net-30
```

Identify the best address lease policy

Leases can be assigned for a period of time, like 24 hours or three days.

The `LeaseTim` value is an absolute time value that indicates when the lease is up.

`LeaseNeg` indicates whether or not the client can negotiate the expiration time based on usage.

Configure DHCP network services using the dhcpconfig program

`dhcpconfig` is the preferred method of configuring the DHCP tables. It is a ksh script that manipulates `pntadmin` and `dhtadm`.

```
#dhcpconfig

*** DHCP Configuration ***
Would you like to:
```

```
1) Configure DHCP Service
2) Configure BOOTP Relay Agent
3) Unconfigure DHCP or Relay Service
4) Exit
Choice:
```

Use DHCP troubleshooting tools

Monitor the network using `snoop`. Look for `DHCPREQUEST` and `DHCPACK` messages.

```
snoop –o /tmp/dhcpout udp dhcp
```

Make sure the client is configured for DHCP. Look at the `/etc/dhcp.<interface>` file. Stop and restart the DHCP server if the configuration files have changed.

INTRODUCTION TO NETWORK MANAGEMENT TOOLS

Describe network management

Network management includes such concepts as system configuration, fault correction, performance tuning, accounting and security maintenance.

One protocol included in network management is SNMP. It is a UDP standard that uses standard calls – `Get`, `Set`, and `Trap` – to retrieve or place data into object identifiers (**OID**s).

Information is described according to the Structure of Management Information (**SMI**), which describes how objects are stored in a management information base (**MIB**).

SNMP is described in RFC 1157

List some SNMP-based management applications

Some Solaris™ SNMP-based tools are:

Solstice Domain Manager
Site Manager
Enterprise Manager (and corresponding agent software)

DOMAIN NAME SERVICE

Describe the purpose of the Domain Name Service (DNS)

DNS is a solution to the problems inherent to managing computer system hostnames. These hostnames must have an efficient way to resolve to their corresponding numerical addresses and maintain uniqueness on the internet with respect to certain organizations (i.e. `host1.companya.com` and `host1.companyb.com` share similar hostnames but are unique machines on the internet).

DNS is described in RFC 1035

Describe the differences between the DNS namespace, a domain, and a zone of authority

The **DNS namespace** starts with a concept called "nameless root". This is the top-level domain, and contains no name-specific information, only authoritative zones. .com, .edu, .gov, etc are found here.

Domain categories of the DNS namespace can be organizational and geographical. Division based on other political or locale reasons is done in sub-domains.

A **zone of authority** is a delegated portion of the overall DNS namespace to an administrative party. This way, the entire tree can be consistently managed from the "leaf"-level up.

Describe the concept of a nameserver, including the different types of nameservers, such as a primary nameserver, a secondary nameserver, and a caching-only nameserver

A nameserver is the computer that holds the authoritative records for the machine names of a particular domain. The database is managed by the administrator and is located on the primary nameserver machine. The primary server synchronizes and delegates the namespace it manages.

The primaries communicate with secondary servers by way of zone transfers. Secondary servers are used to distribute the load and provide redundancy to primary servers. They localize name-lookups, if placed correctly throughout a domain.

Caching-only servers provide fast, non-authoritative information about a domain. They are used to reduce the overhead of zone transfers and frequently used lookups.

Describe what a resolver is and understand the processes of address resolution and reverse address resolution

A **resolver** is the go-between what a user types as "`easily.rememberedname.com`" and the appropriate numerical IP address of the requested server.

DNS has two types of queries:

Recursive – A recursive request is one that must be satisfied by a nameserver. When a resolver sends a recursive request, the queried nameserver is obliged to return a valid answer. It can't just turn the querier to another name server.

Iterative – An iterative request attempts to locate a server that has the best information. When a resolver sends an iterative request, the queried nameserver returns its best answer, which may be from its non-authoritative cache or the name of a server it believes may have more information.

In Solaris™, address resolution occurs in this order:

- `/etc/nsswitch.conf` is checked for NIS+ domain name information
- `/etc/hosts` is checked to see if the name is defined there
- Query (recursive) request to local DNS server (from `resolv.conf`)
- Local DNS server checks local cache

- Query (iterative) request is sent to root servers
- Local DNS contacts result from root servers
- Remote domain is contact by local DNS, and returns requested hostname
- The result is cached for a short period to satisfy subsequent requests

Describe the syntax of the server-side DNS setup files, including the /etc/named.conf file, the cache file, and zone files

`/etc/named.conf` contains the directory where DNS database files are stored, and the names of the zones for which it is authoritative. Keywords specify primary / secondary roles, as well as default timeout periods.

The cache (or "hints") file is known as `named.root`, and it contains the IP addresses and names of the root servers. This file is used for gathering authoritative name information from remote hosts. It can be obtained directly from the INTERNIC and changes infrequently.

Zone files contain **A records** (hostname to IP) and **PTR records** (IP to hostname). Updates are done on the primary server and pushed out to secondary servers. A primary or secondary server can give out authoritative names for a zone.

Describe the information included in the Start Of Authority (SOA), Name Server (NS), Address (A), and Pointer (PTR) resource records

SOA is the fully qualified domain name

NS records specify other nameservers that contain information about the domain

A records list hostname to IP address information

PTR records list IP to hostname information

Describe the syntax of the client-side DNS setup file /etc/resolv.conf

A client `/etc/resolv.conf` file looks like this:

```
domain subdomain.mydomain.com
search subdomain.mydomain.com mydomain.com
nameserver 10.10.15.31
nameserver 10.1.15.10
nameserver 10.8.2.41
```

`domain` specifies the local domain information (to append to hostnames)
`search` specifies what domains to look in first
`nameserver` specifies the IP of the local DNS server

Solaris™ supports a maximum of three `nameserver` entries.

Describe the various DNS debugging and troubleshooting methods available to the administrator

A built-in tool for testing and troubleshooting DNS is `nslookup`. It can be used interactively to send queries and display replies from any DNS server.

Sun-specific troubleshooting of BIND can be obtained by using the various options provided by pkill.

`pkill -INT in.named` dumps the running daemon from memory to disk.

`pkill -HUP in.named` forces a re-read of config files.

`named -d` can also be started log debug information in a `.run` file.

ELECTRONIC MAIL, MAIL ALIASES, AND MAIL SERVERS

Name and describe the types of machines used for electronic mail (email)

The most popular form of communication on the internet is email. In its most basic form, it is a text message that is stored and forwarded from system to system from the time it leaves the sender until it reaches the receiver.

Electronic mail is routed using information stored in a message "header" tagged with a unique user's name. While mail is passed through the Internet is analogous to that of datagrams, the routing (performed on systems known as **mail relays**) of mail messages occurs at the application layer.

The types of machines and processes for electronic mail include:

Mail User Agents (MUA) – the program that individuals use to create mail messages. These agents are go-betweens for the MUA and the MTA. In Solaris™: `/usr/bin/mail`, `/usr/bin/mailx`

Mail Transport Agent (MTA) – The main component of electronic mail. The *defacto* standard is `sendmail`. `sendmail` accepts messages from the MUA, resolves the destination address, chooses a mailer to deliver the mail, and receives incoming mail. An administrator spends much time configuring this mail element.

Mail Delivery Agent (MDA) – the program that distributes mail to each user's mailbox. In Solaris™: `mail.local`

Describe a mail address

A mail address is a unique identifier for a user on a system. Sendmail examines this text string (which is located in the mail header) to determine who will receive the message.

Mail addresses have several forms:

unqualified – 'username'
qualified – 'username@machine'
fully qualified – 'username@domain.mycompany.com'

The three elements of a mail address:

recipient - (or alias)
delimiter - ('.', '!', '@', etc)
destination - (someplace.somewhere.com)

Name and describe the different alias files

Alias files offer an alternate means of describing names for recipients or groups of recipients. Alias files come in four distinct forms:

`$HOME/.mailrc` – a sender's personal alias file, located in their home directory. It is consulted by the MUA before a message is sent.

`/etc/mail/aliases` – is located on a sending system's local disk. It is referenced by sendmail when a message tagged for local delivery. System Administrators usually modify this file

NIS+ alias table – aliases in the NIS+ database corresponding to the /etc/mail/aliases file

`$HOME/.forward` – is located in a recipient's home directory. It is for redirecting inbound mail to another email address or to a file or executable (such as 'vacation')

Create alias entries in the different alias files

A .mailrc has the format:

```
alias <alias-name> user1@nowhere.com, locuser@mydomain
```

The MUA expands <alias-name> into the headers when a message is addressed to it.

`/etc/mail/aliases` is referenced by any message sent from the system. Local users and inbound messages from remote systems may use entries in aliases. NIS+ can manage alias tables for multiple systems.

Alias names can reference:

```
a username
a /file
a | program
an :include: list
```

Two alias examples:

1.

```
matt: "| /usr/bin/cat | /usr/bin/wc >> mailinfo"
```

This will write word-count information for each message sent to "matt" into a file called mailinfo.

2.
```
ewoks:include:/home/matt/ewoks
/home/matt/ewoks: wicket@endor, teebo@endor, latara@endor,
paploo@endor, kneesaa@endor
```

Messages sent to "ewoks" would be expanded to contain the list of email IDs of those users listed in the `/home/matt/ewoks` file.

Create .forward files

Forward facilitates the redirection of mail to another email address or local program. A `.forward` file exists in the receivers home directory, and must be writable only by the owner in order to be used (for security reasons). `.forward` may look like this:

```
\matt
/home/matt/saved.mail
```

or

```
\matt, "|/usr/bin/vacation matt"
```

Set up a mail server

Some elements to consider when setting up a mail environment:

Make sure clients have a `sendmail.cf` file

Make sure mailhost is specified in the `/etc/hosts` file

Make sure there is sufficient space on `/var/mail` for messages

Designate a server as a local MTA

If using NFS, to mount `/var/mail`, make sure `/etc/vfstab` is correct

Users must exist in the `/etc/passwd` file or in NIS+ tables

Assign MX records to those servers acting as mail hosts in DNS

SENDMAIL

Identify Sendmail features

Sendmail is a reliable, robust mail handler. It can route all types of mail (SMTP, UUCP, local) by examining header information. It is infinitely customizable using special MACRO and RULE declarations.

More information about sendmail can be found at http://www.sendmail.org/

Mail is collected from MUAs and addresses are scanned and parsed. The parsing process determines how to queue, log and create recipients for the message.

A message has three parts:

envelope – the address the message is sent to (used in mail routing)

header – has the From: and To: lines

body – ASCII text of the message

A single blank line separates the header from the body.

In DNS, hosts that are specifically configured to receive mail are tagged with an MX designation. Any mail matching the listed domain is forwarded to the mailhost. A weight is associated with the MX hosts, to provide redundancy.

Example:

```
Mhost       IN A     10.0.0.25
domain.com. IN MX 10 mhost.domain.com.
domain.com. IN MX 20 mhost2.domain.com.
```

Analyze the contents of the /etc/mail/sendmail.cf file

Sendmail.cf is read when the daemon is started. It contains:

MDA listings
Macros
Options
Rule Sets
Rewrite rules

In Solaris™ 7, the sendmail.cf file can be confgured and rebuilt using the m4 preprocessor script collection. Files with the extension .mc are edited by an administrator, and then combined to form sendmail.cf using '`make`'.

LAN PLANNING

Develop a list of considerations when planning a LAN

When designing a LAN, it is important to consider:

Cost
Performance
Flexibility
Reliability
Security

Define LAN management and implementation standards

The idea behind LAN management is tailoring the environment to meet the needs of the business, while allowing for future growth. This includes segmenting a network into manageable subnets that match business-unit functions and administrative tasks. Starting with a good design, using industry standards, and avoiding proprietary products will make development into the future easier.

Discriminate between LAN-cable types and topologies based on cost, performance, flexibility, reliability, and security

There are several choices for LAN cabling types:

10Base5 (Thick Ethernet)	Expensive and outdated	Difficult to install	Difficult to troubleshoot	Vulnerable to EMF and broadcast eavesdropping
10Base2 (Thin Ethernet)	Expensive	Distance restrictions	Difficult to troubleshoot	Vulnerable to EMF and broadcast eavesdropping
10BaseT (Twisted Pair)	Common and affordable	Easy to install in existing structures	Reliable, powerful management capabilities	Vulnerable to EMF and broadcast eavesdropping
10BaseF (Fiber Optic)	Very expensive	Difficult to install	Easy trouble-shooting	Not vulnerable to EMF eavesdropping
100BaseT (Category 5)	Data-grade cable is 40% more expensive than 10BaseT	Easy to install in existing structures and use with 10BaseT	Considered very reliable	Vulnerable to EMF and broadcast eavesdropping

Create a LAN topology that meets case-study business requirements

Understand the business requirements
Evaluate suppliers based on cost and reliability
Select a topology and media type
Determine a LAN management team
Create usage and security policies

Create a LAN blueprint specifying cable runs and the locations of servers, clients, bridges, repeaters, routers, and gateways

An excellent resource for LAN planning is Gerald T. Charles Jr.'s book *LAN Blueprints: Engineering It Right*. (ISBN: 0070117691)

Compile a list of all network cabling and associated hardware required for LAN implementation

Repeaters
Hubs
Switches
Bridges

Gateways
Routers

NETWORK TROUBLESHOOTING

Describe general methods of troubleshooting networking problems

Important concepts to remember when troubleshooting a network include understanding the nature and location of the problem (i.e. hardware or software) and being able to reproduce it.

Try and trace a problem to its source by simplifying the environment. Remove extraneous equipment. Trial and error is a main principle, but it comes back to recreating the problem after changes have been made. Document configuration changes as you work and summarize the corrections for the final solution.

Identify network troubleshooting commands

`ping` – Uses ICMP echo packets to test connectivity between hosts. A sender transmits a ping packet, which the intended receiver echoes back. When used with the `-s` option, sequence and trip-time information is included in the output.

`ifconfig` – Shows the status of configured interfaces. Includes information relating to the MTU, IP address, Netmask, Broadcast address, and MAC address.
In Solaris™, there are actually two `ifconfig` commands. The first, `/sbin/ifconfig`, does not reference and NIS+ `nsswitch.conf` configuration information. The other, `/usr/sbin/ifconfig`, does use name service information in the `nsswitch.conf` file.

`arp` – When used with the `-a` option, it can display the table of cached hardware addresses on the system.

`snoop` – Used to display on-the-fly network information on an interface. Can be an important tool in troubleshooting almost any issue. Used with the `-v` or `-V` options for verbosity. Writes to a file with the `-o` option, and views a file it created using `-i`.

`ndd` – Used to display and set driver configuration parameters. An example would be: `ndd /dev/hme link_speed`, the output of which would be a 0 or 1. A 1 in this case would indicate the link speed was set to 100 Mbps.

`netstat` – Useful for determining the state of the systems interfaces. Displays the routing tables with the `-nr` option. A verbose mode is available with `-v`

`traceroute` – A network troubleshooting tool that reveals the state of the network between the client and the destination. It uses IP time-to-live values to try to max out values of ICMP TIME_EXCEEDED, PORT_UNREACHABLE, and ICMP ECHO_REPLY on routers and destination hosts.

Determine which layer of the TCP/IP layer model is causing the problem

Some hints to try at various levels of the IP model:

Physical – look for LEDs and use verified cabling
Network – snoop will display output on a working network interface
Transport – ICMP error messages, use traceroute and ping
Internet – check protocols, investigate name resolution
Application – use another working system on the same subnet, also look for diagnostic tools built-in to the application

Repair common networking problems

Two frustrating but common network problems:

Faulty cabling
Duplicate IP addresses

Special Thanks to Matthew Kortas for contributing this Cramsession. Make sure to visit his site at: http://www.acm.cse.msu.edu/~kortasma