Sun Solaris 8 Certified Network Administration - Cramsession

Network Model

Identify the purpose of each layer in the TCP / IP five layer Model

- **Application Layer**
  User accessed application programs and network services. Examples at this layer would include FTP, telnet, and SMTP.
- **Transport Layer**
  Connection-oriented TCP and connectionless UDP data transfer occur here. Flow control, sequencing, and acknowledgements are the means by which this occurs.
- **Internet Layer**
  Here data is fragmented, addressed and routed. Also, encapsulation of packet happens at this layer.
- **Network Interface Layer**
  Handles error detection and packet framing. Standards like 802.3 and 802.5 arrange bits into understandable data using fields that describe destination and source addresses, data, and error correction.
- **Hardware Layer**
  Electrical signals that move raw bits through the ether. Today, twisted pair and fiber optic cabling is the most common transmission medium for network data.

Describe the functionality of the following Network Protocols: TCP UDP, IP, and ICMP

- **TCP** – Connection-oriented. The hallmark of TCP is the acknowledgements between systems that ensure all of 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

SCSA Solaris 8 II
Citrix
CIW
IBM
Chauncey
SCO
SANS
SAIR
CNX
Adobe
Nortel
Java
Discussion Boards
Certifications
Training Locators
Find a Tech Job
Study Tips
Bookstore
Links
Reviews
Study Break

Global Navigation
Post/Edit Resume
Skill Assessment
Top Tech Sites
Discounts/Freebies
Newsletters
Tell-A-Friend
Email This Page!
Newsletter Archive
Advertising
Site Map
Help/Info

- **IP** – Determines a packets path based on the listed destination IP.
- **ICMP** – Communicates system status and error messages using IP datagrams

Describe the relationship between the following Network Protocols: TCP, UDP, IP, and ICMP

TCP (transport layer) is the protocol that is used to synchronize hosts on the network ensuring complete data transmission. It provides reliable data streaming so that applications don't need to worry about incomplete information traversing the network.

UDP (transport layer) is the less reliable peer of TCP. It is simply a datagram delivery service. No coordination or error checking occurs with UDP.

IP (internet layer) is the protocol that determines the best route for a packet traversin a network. It chooses the best path, in real-time, for a packet with a correct destinatio address.

ICMP (internet layer) uses datagrams to check for errors and perform management tasks.

Describe peer-to-peer communication

Peer-to-peer is the term used to describe the communication between corresponding network layers on source and destination hosts. At each layer, data is encapsulated and passed on to the next layer. From the source, data is encapsulated in the 'downward' direction. At the receiving end, data is passed 'upward' through the layers as it is un-encapsulated.

Local Area Networks

Identify the role of the following LAN components: Bridge, Repeater, Router, Switch, and Gateway

- **Bridge** – A link layer device that connects two or more network segments.
- **Repeater** – A simple device that regenerates the signal on a lengthy wire.
- **Router** – A device that examines addresses and selects optimal paths.
- **Switch** – A link layer device that dedicates traffic between two senders.
- **Gateway** – Interconnects two networks with uncommon protocols.

Identify the network topologies

- **Bus** – One central device, usually a hub, which connects cabling between othe machines. The signals they send are shared with every node on the network.
- **Star** – A central hub with a segment of cable running between it and each clie and other network hubs. The advantage is that there are never more than 2 hops between any two nodes.
- **Ring** – Each node's output is connected ‘serially’ to the next node's input. A ‘token’ must be passed around to allow each device to communicate. Intelliger systems can increase reliability over that of a bus or star configuration.

Ethernet Interface

State the purpose of the Ethernet address

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.

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

Identify the commands to get and set driver configuration

Getting and setting driver configuration is accomplished with the ndd command.

```
# ndd /dev/hme \?
transceiver_inuse              (read only)
link_status                    (read only)
link_speed           (read only)
<...>
```

shows the status of features of the adaptor hme.

```
# ndd /dev/hme link_status
1
```

queries the parameter link_status and outputs the value 1 (meaning the link is up)

```
# ndd /dev/hme adv_100fdx_cap 1
```

sets the /dev/hme adaptors to 100 megabit full-duplex.

ARP and RARP

Explain the process of address resolution using ARP and RARP

ARP and RARP (ReverseARP) are protocols to match a host's unique MAC address (hard-coded into the network interface card) with an assigned ethernet address (IP). To obtain a destination IP or Ethernet address, a host must send a broadcast alerting other hosts on the network to its request. The broadcast packet contains the MAC address.

ARP maps a 32-bit IP to a 48-bit MAC (or Ethernet) address.

RARP maps a 48-bit MAC address with a 32-bit IP address.

Identify the commands to manage ARP cache

ARP replies received by a host are stored temporarily in cache so that the resolution process does not need to be repeated as frequently.

```
# arp -a
Net to Media Table: IPv4
```

```
Device   IP Address               Mask      Flags   Phys Addr
------ -------------------- --------------- ----- ---------------
hme0   209.55.84.129        255.255.255.255       00:a0:c5:a8:da:b3
hme0   198.93.198.102       255.255.255.255       00:80:5f:89:62:38
hme0   209.55.84.137        255.255.255.255   SP  00:80:3f:ff:47:99
hme0   BASE-ADDRESS.MCAST.NET 240.0.0.0       SM  01:00:5e:00:00:00
```

Identify the configuration files and scripts used to configure a network interface

A host must be properly set up on the network in order for ARP messages to be heard During the system boot, the init process starts scripts located in `/sbin/rcS`. A script in this directory, `/etc/rcS.d/S30network.sh`, looks for files `/etc/hostname.hme0` and `/etc/inet/hosts` to contain the correct information about the host's IP addresses so that it can configure the interface. Ultimately, the `/sbin/ifconfig` command is run to configure the interface.

Internet Layer

Describe the following: IP address, Broadcast address, Netmask, Datagram, and Fragment

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.

The broadcast address is a reserved address that will relay any packets it receives to every host on the network segment. Hosts are grouped on similar subnets using a netmask.

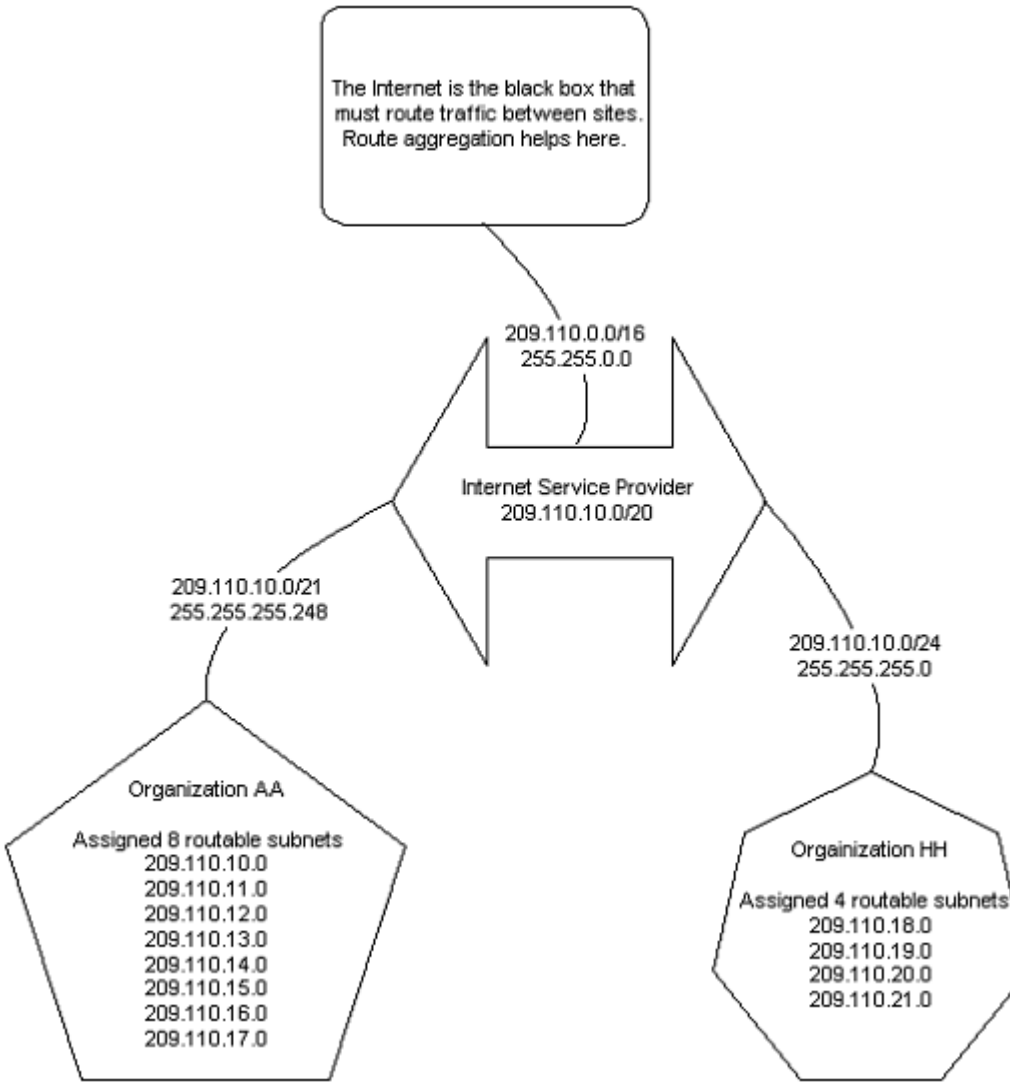
The netmask is used in a logical AND operation with a destination IP to determine if a router must assist with the resolution and routing.

Describe Classless Inter-Domain Routing (CIDR)

Classless Inter-Domain Routing (CIDR) is an answer to the rapid growth of the Internet. The standard IPV4 addresses specified in RFC 791 did not account for the large number of hosts many organizations have. Because the subnetting techniques IPV4 caused inefficient use of the classful addressing schemes and increased the siz of the routing tables needed to direct information, a task force designed and implemented CIDR in RFC-1517, 1518, 1519, and 1520.

CIDR is classless. This means that netmasks are used to create networks of various sizes. Now, Internet addresses are written using the notation X.X.X.X/YY. For example, 10.2.48.130/26 would indicate a host address on network with subnet mask 255.255.255.192.

CIDR works because assigned IP address space can be recursively broken down intc smaller sizes that fit networks of any design. Large ISPs are given blocks of addresse that they then delegate to other smaller ISPs or private organizations. Since they control a large block, they can aggregate routes to their hosts using the technique of supernetting. This reduces the size of the routing tables and increases performance since fewer lookups need to be performed at a high level.



Identify the file used to set netmasks

The `/etc/inet/netmasks` file is used to store netmask information. It has the format:

```
network-number  netmask
128.50.0.0      255.255.255.0
```

Identify the features and benefits of the 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:` **`11111111.11111111.11111111.111`**`00000`

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

Configure a Network Interface

The `/sbin/ifconfig` command is used to configure the network interface in Solaris™

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 the value that should be computed by the machine using whatever information is provided.

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

Routing

Describe IP routing

At its most basic level, routing is the method of forwarding packets from one network to another. For large networks, dedicated equipment is used to communicate cross-network traffic and shield hosts from the task of determining the best route for data to take across a network.

IP routing occurs at the Internet layer. It can be described as either direct or indirect. Direct routing is when the destination host is directly attached to the same network as the sending host. Indirect routing is when the destination host is not on the same network as the sending host.

Routes are stored and referenced by the kernel using routing tables. Several schemes exist for filling these routing tables: static and dynamic.

Solaris™ handles dynamic routing using RIP or RDISC protocols.

Identify the Solaris 8 daemons which implement routing protocols

`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.

Identify the files used to configure routing

`/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_forwardin` is enabled and the `in.routed -s` (RIP) or `in.rdisc -r` (RDISC) processes start.

Identify the purpose of the files used to configure routing

`/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, that assigns names to network numbers
`/etc/gateways` – An optional file that defines additional passive routes alongside the default routes. Read by `in.routed`

Administer the routing table

The command `netstat -rn` shows the routing table with host addresses in numerical format:

```
# netstat -rn
Routing Table: IPv4
  Destination           Gateway           Flags  Ref   Use   Interface
-------------------- -------------------- ----- ----- ------ ---------
220.137.157.128      220.137.157.130       U         1   9230  hme0
224.0.0.0            220.137.157.130       U         1      0  hme0
default              220.137.157.129       UG        1  26156
127.0.0.1            127.0.0.1             UH        3  68199  lo0

Routes can be added to the routing table using the route command.
route [-fn] add|delete [host|net] destination [gateway [metric]]
# route add net 128.50.3.0 128.50.3.1 1
```

Routes can be added to the routing table using the route command.

```
route [-fn] add|delete [host|net] destination [gateway [metric]]
```

```
# route add net 128.50.3.0 128.50.3.1 1
```

Transport Layer

Identify the features of the TCP and UDP

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. Its functionally is 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 t 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's analogous to 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, connection-less, stateful, and stateless

- **Connection-oriented** protocols are implemented using TCP.
- **Connectionless** protocols are implemented using UDP.
- **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 serve operate independent of each other or the current network condition.

Describe the relationships between port numbers, network services and inetd

A port number is a virtual address that the kernel associates with any service it provides. Clients wishing to interact with a network service will contact the server and specify this special port. A daemon running on the server, inetd, listens for incoming requests, and then contacts the program responsible for the service and instructs it to respond.

Client-Server Model

Explain 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.

Administer Internet services and RPC 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 wha network services are at which well-known port, and the location of the process to star

```
#ident  "@(#)inetd.conf 1.33    98/06/02 SMI"   /* SVr4.0 1.5   */
#
#
# Configuration file for inetd(1M).  See inetd.conf(4).
#
# To re-configure the running inetd process, edit this file, then
# send the inetd process a SIGHUP.
#  <service_name> tli <proto> <flags> <user> <server_pathname> <args>
#
# Ftp and telnet are standard Internet services.
#
ftp     stream  tcp     nowait  root    /usr/sbin/in.ftpd       in.ftpd
telnet  stream  tcp     nowait  root    /usr/sbin/in.telnetd    in.telnetd
#
# Tnamed serves the obsolete IEN-116 name server protocol.
#
#name   dgram   udp     wait    root    /usr/sbin/in.tnamed     in.tnamed
#
# Shell, login, exec, comsat and talk are BSD protocols.
#
#shell  stream  tcp     nowait  root    /usr/sbin/in.rshd       in.rshd
#login  stream  tcp     nowait  root    /usr/sbin/in.rlogind    in.rlogind
#exec   stream  tcp     nowait  root    /usr/sbin/in.rexecd     in.rexecd
#comsat dgram   udp     wait    root    /usr/sbin/in.comsat     in.comsat
#talk   dgram   udp     wait    root    /usr/sbin/in.talkd      in.talkd
<...>
```

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.

`/etc/services` has entries that look like:

```
ftp-data 20/tcp
ftp 21/tcp
telnet 23/tcp
smtp 25/tcp
pop3 110/tcp
```

RPC services is a service / port numbering scheme that eliminates the dedicated por requirements of inetd.

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 a client request. `rpcbind` is started at runlevel 2 using `/etc/init.d/rpc`. Clients requesting a service interact with rpcbind on port 111. They are then given a unique port number when the associated service runs.

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

Collect information about services configured on hosts

`netstat` and `rpcinfo` are used to collect information about services on hosts.

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

```
UDP: IPv4
   Local Address         Remote Address     State
-------------------- -------------------- -------
      *.ntp                                Idle
localhost.ntp                              Idle
myhost.nowhere.net.ntp                     Idle
localhost.domain                           Idle
myhost.nowhere.net.domain                  Idle
      *.*                                  Unbound
      *.65249                              Idle
      *.syslog                             Idle
      *.40227                              Idle
      *.*                                  Unbound

TCP: IPv4
   Local Address        Remote Address    Swind Send-Q Rwind Recv-Q  State
-------------------- -------------------- ----- ------ ----- ------ -------
      *.*                  *.*                0      0 24576      0 IDLE
      *.22                 *.*                0      0 24576      0 LISTEN
myhost.nowhere.net.22 209.110.245.243.2792 31856     0 24616      0 ESTABLISHED
myhost.nowhere.net.22 209.110.245.243.2793 31856     0 24616      0 ESTABLISHED
localhost.domain           *.*                0      0 24576      0 LISTEN
myhost.nowhere.net.domain  *.*                0      0 24576      0 LISTEN
      *.smtp               *.*                0      0 24576      0 LISTEN
      *.587                *.*                0      0 24576      0 LISTEN
      *.3306               *.*                0      0 24576      0 LISTEN
localhost.8080             *.*                0      0 24576      0 LISTEN
localhost.8021             *.*                0      0 24576      0 LISTEN
localhost.8099             *.*                0      0 24576      0 LISTEN
      *.443                *.*                0      0 24576      0 LISTEN
      *.*                  *.*                0      0 24576      0 IDLE


Active UNIX domain sockets
Address  Type          Vnode     Conn  Local Addr      Remote Addr
300009a01a8 stream-ord 30000bc0958 00000000 /usr/local/Zope/pcgi.soc
300009a0828 stream-ord 3000080faa8 00000000 /tmp/mysql.sock
```

`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

State 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

Identify DHCP configuration files

`dhcptab` and `dhcp_network` are the two files used to configure DHCP behaviors. DHCP options are stored in the `/etc/dhcp/inittab` file.

```
#
#ident "@(#)inittab 1.3 99/08/16 SMI"
#
# This file provides information about all supported DHCP
options, for
# use by DHCP-related programs. This file should only be
modified to
# add support for SITE options or new STANDARD options; no
existing
# options should be modified. Please note that errors introduced
into
# this file may cause programs to crash.
#
# Please consult dhcp_inittab(4) for further information. Note
that
# this interface is "Unstable" as defined by attributes(5).
#


Subnet          STANDARD,       1,      IP,         1,   1,      sdmi
UTCoffst        STANDARD,       2,      SNUMBER32,  1,   1,      sdmi
Router          STANDARD,       3,      IP,         1,   0,      sdmi
Timeserv        STANDARD,       4,      IP,         1,   0,      sdmi
IEN116n         STANDARD,       5,      IP,         1,   0,      sdmi
DNSserv         STANDARD,       6,      IP,         1,   0,      sdmi
Logserv         STANDARD,       7,      IP,         1,   0,      sdmi
CookieS         STANDARD,       8,      IP,         1,   0,      sdmi

<...>
```

The DHCP configuration file that allows an administrator to tune some RFC parameters is `/etc/default/dhcpagent`

```
#
# This file contains tunable parameters for dhcpagent(1M).
#


# All parameters can be tuned for a specific interface by
prepending
# the interface name to the parameter name. For example, to make
# RELEASE_ON_SIGTERM happen on all interfaces except hme0,
specify:
#
# hme0.RELEASE_ON_SIGTERM=no
# RELEASE_ON_SIGTERM=yes


# By default, when the DHCP agent is sent a SIGTERM, all managed
# interfaces are dropped. By uncommenting the following
# parameter-value pair, all managed interfaces are released
instead.
#
# RELEASE_ON_SIGTERM=yes
<...>
```

Configure the DHCP data store preference (NIS or FILES) using the `/etc/default/dhcp` file.

State the purpose of DHCP configuration files

`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
```

Administer DHCP clients and servers

Solaris™ includes a GUI program called dhcpmgr which is a graphical interface to th older dhcpconfig program. It is an easy, step-by-step wizard to guide an administrato through the server configuration process.

`# /usr/sadm/admin/bin/dhcpmgr`

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

For troubleshooting, DHCP can be started in several debugging modes:

`# /sbin/dhcpagent -d3` (debugging level 3)

Network Management Tools

Identify tools which use the Simple Network Management Protocol (SNMP)

SNMP agents can be set up on many types of network and operating system devices Information sent out by these agents can be read by vendor applications like HP Openview, Sun Management Center™, Solstice Domain Manager™ and Solstice Sit

Manager™. OpenSource tools are also available. In particular, UC-Davis' UCD-SNMP.

Describe the Simple Network Management Protocol (SNMP)

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

SNMP is a UDP standard that uses standard calls – `Get`, `Set`, and `Trap` – to retrieve or place data into object identifiers (**OID**s). SNMP is based on UDP, because it is generally accepted that a poorly performing device will be able to send out messages relating to the problem quickly in the event of a failure.

The functions of SNMP are:

- **Get** - management consoles will poll all of their SNMP-managed devices in the field to obtain status information. The `gets` update a graphic or table.
- **Set** - should a management console need to change a managed device in some way, it can send out a `set`.
- **Trap** - When a managed device has a failure or needs to communicate with the management station, it sends a `trap`.

Information is described according to the Structure of Management Information (**SMI**) which describes how objects are stored in a management information base (**MIB**). Objects in a MIB are defined according to ASN.1 notation. ASN is the Abstract Syntax Notation.

SNMP is described in RFC 1157

Domain Name System

Identify the purpose of DNS

DNS is a solution to the problems inherent in managing computer system hostnames. These hostnames must have an efficient way to resolve their corresponding numerical addresses, and maintain uniqueness on the Internet with respect to individual organizations. (i.e., `host1.companya.com` and `host1.companyb.com` share similar hostnames but are unique machines on the Internet). The most widely used version of DNS in UNIX is bind.

DNS is described in RFC 1035

Describe 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. The process of matching a domain name with an IP address is called resolution.

In Solaris™, address resolution occurs in this order:

- `/etc/nisswitch.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

Address resolution matches hostnames with IP addresses. Machines will also sometimes perform reverse address resolution to verify that a requestor is valid. Too like nslookup can perform both regular and reverse address resolution.

DNS has two types of queries:

1. **Recursive** – a recursive request is one that must be satisfied by a nameserve 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.
2. **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.

Identify the correct Resource Record syntax

Entries in the name server database are known as resource records. A record has the format:

```
[name] [time-to-live] [class] [type] [data]
```

Domain Name is the name of the record.

Time to live is a value that indicates when the record can no longer be trusted.

Class for the Internet is almost always IN.

Record type indicates the kind of information the record provides:

SOA is the fully qualified domain name
NS records specify other nameservers that have domain information
A records list hostname to IP address information
PTR records list IP to hostname information
MX records list configured mail hosts
CNAME records list aliases for a host name

Explain the steps needed to configure DNS

DNS requires some specific information before a host can be configured. Particularly

- Names of root servers that have information about other domains
- Name to Address translations of all hosts in the domain
- Address to Name translations (reverse lookups) of all hosts
- Any servers that contain information about subdomains

Identify the configuration files for DNS

Configuration files for DNS include the:

- `/etc/named.conf`, the main configuration file
- cache file (also called the root hints file)
- zone files, that contain information about hosts in the domain
- `/etc/resolv.conf` file (on the client to specifiy DNS resolvers)

State the purpose of DNS configuration files

`/etc/named.conf` contains the directory where DNS database files are stored, as well as 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 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 does not change often.

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.

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.

Network Time Protocol

Describe the NTP features

Network Time Protocol is a method of keeping system times accurate, regardless of the world's time zones. It uses the reliable time standard Universal Time Coordinated (UTC) to synchronize clocks. UTC is an accepted estimate of the current time based on several institutions' calculations.

NTP uses broadcasts and multicasts to learn about time updates. Accurate time is a necessity for systems because it aids in network management, logging and file time stamps, and even as an element of generating encryption keys.

Identify NTP configuration files

NTP is provided in the packages `SUNWntpr` and `SUNWntpu`. The main configuration file for NTP is `/etc/inet/ntp.conf`. An example configuration file looks like:

```
# @(#)ntp.server     1.5       99/09/21 SMI
```

```
#
# /etc/inet/ntp.server
#
# An example file that could be copied over to /etc/inet/ntp.conf and
# edited; it provides a configuration template for a server that
# listens to an external hardware clock, synchronizes the local clock,
# and announces itself on the NTP multicast net.
#

#
# * All TrueTime receivers are now supported by one driver, type 5.
#   Types 15 and 25 will be retained only for a limited time and may
#   be reassigned in future.
#
# Some of the devices benefit from "fudge" factors.  See the xntpd
# documentation.

# Either a peer or server.  Replace "XType" with a value from the
# table above.
server 127.127.XType.0 prefer
fudge 127.127.XType.0 stratum 0

broadcast 224.0.1.1 ttl 4

enable auth monitor
driftfile /var/ntp/ntp.drift
statsdir /var/ntp/ntpstats/
filegen peerstats file peerstats type day enable
filegen loopstats file loopstats type day enable
filegen clockstats file clockstats type day enable

keys /etc/inet/ntp.keys
trustedkey 0
requestkey 0
controlkey 0
```

A drift file must also exist. Make sure that `/var/ntp/ntp.drift` exists.

NTP logs messages to the `/var/adm/messages` file. Any updates to the system tim are recorded here.

The NTP daemon can be monitored using the `xntpdc` program. It is interactive and can be used to obtain information about configured time servers or other host peers.

State the purpose of NTP files

`/etc/inet/ntp.conf` is the file that lists the preferred timeservers according to strata. Stratum servers, listed highest to lowest in terms of accuracy, determine how precise the time is going to be.

The drift file lists the internal clock's frequency offset and helps keep internal timekeeping more accurate.

NTP clients send NTP broadcast packets that include their local time onto the networ NTP servers respond to the packets by inserting their own time. The clients then use these numbers to determine how long the packet was on the network, and sets intern time according to estimates gained from receiving several responses.

`/etc/init.d/xntpd` is the daemon that starts NTP.

Describe how to configure NTP

It is important that a server use external reference servers. See http://www.eecis.udel.edu/~mills/ntp/clock1.htm for a list of clocks to reference.

Copy the template `/etc/inet/ntp.server` to `/etc/inet/ntp.conf`.

Make entries in the conf file similar to:

```
128.50.10.254 prefer
fudge 127.127.XType.0 stratum 0
```

Create the drift file:

```
# touch /var/ntp/ntp.drift.
```

Start the NTP daemon using `/etc/init.d/xntpd start`

Network Troubleshooting

Identify common network problems

There is no book that can teach network troubleshooting. Only through experience ar experimentation can an administrator become skilled at solving whatever types of problems are thrown her way. Since networked systems depend on interoperability, a important skill to develop is to be able to rule out causes and narrow the scope and focus of the troubleshooting effort.

Try and trace a problem to its source by simplifying the environment by removing 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.

Diagnose network problems

Network problems can occur at all layers of the TCP/IP model. Faulty wiring can occu at the physical layer. Duplicate addressing, both MAC and IP addresses, cause problems in the middle Network and Internet layers. And misconfigured applications can be the source of many headaches at the highest application layer.

Don't assume anything when troubleshooting network problems. You might want to verify systems with known working cabling and network cards. Be sure to follow guidelines when configuring systems (that aren't on a DHCP network) to ensure that addressing doesn't cause issues. And at the application layer, be sure that any and al settings that affect the network are set correctly (duplex mismatches are a common cause of slow networks!).

Resolve network problems

Sometimes the best tools for fixing network problems are built right in. These may include:

- **`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 outpu
- **`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 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 b 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 t try and max out values of ICMP TIME_EXCEEDED, PORT_UNREACHABLE and ICMP ECHO_REPLY on routers and destination hosts.

IPv6

Describe IPv6

IPv6 (RFC 2460) is another solution to the problem of IP address depletion. The Internet Architecture Board invented it in 1991.

IPv6 designates 128 bit unique IP addresses. This provides many more addresses than the IPv4 scheme. These addresses are assigned to clients automatically once a administrator has configured name-to-address tables for their domains. This helps eliminate duplication problems. And, because of a simplified header, packets addressed using IPv6 can be routed much faster than a standard IPv4 packet.

Autoconfiguration of the IPv6 protocol comes in two forms: stateful and stateless. A stateful configuration involves another device (like a DHCP server) that must assign the address to the client. Stateless means that a host can generate its own address based on information it has about the environment.

When a host auto-configures its unique address, it creates a locally-linked address from messages received from neighboring hosts. All neighboring devices and system will receive a broadcast telling them about the new address.

The system MAC address is used in generating an IPv6 global address. Remember, the MAC is made of two parts, a Company ID (24 bits) and a Vendor ID (24 bits).

First, the MAC is converted into binary. Next, the seventh bit from the left, known as the universal bit, is toggled (from 1 to 0 or 0 to 1). Finally, two constant octets -- 0xFF and 0xFE -- are inserted between the Company ID and Vendor ID. The final address then converted to HEX.

Each address begins with what is known as a FP (format prefix), or address indicator It can be:

FE8 (local-link)
FEC (site-local)
FF (multicast)

Local Link addresses cannot be routed. Site-Local addresses can be routed. Multicas are delivered to all interfaces on the network (similar to a broadcast).

Familiar protocols in IPv4 have incompatible IPv6 cousins. ICMPv6, IGMP and Neighbor Discovery Protocol (which replaces ARP/RARP) are some of them. ICMP and NDP work together to find and assign local routers, advertise routes and determine neighboring hosts and their addressing schemes.

Configure an IPv6 network interface

Solaris™ 8 supports IPv6 in *dual-stack* mode. This means that the kernel can understand traffic for both IPv4 and IPv6. The operating system cannot support an IPv6-only network scheme at this time.

A mechanism exists for passing IPv4 traffic over an IPv6 system. For systems that support version 6, a IPv4 address 35.9.12.12 will appear as `0:0:0:0:0:0:35.9.12.12`, where the sixth bit grouping from the left is zeros. For systems that do not support IPv6, the address looks like `0:0:0:0:0:FFFF:35.9.12.12`, where the sixth bit grouping from the left is ones.

To configure a Solaris™ 8 system to use IPv6, create a `/etc/hostname6.hme0` an restart.

The IPv6 equivalent to the `/etc/hosts` file is `/etc/inet/ipnodes`.

Once IPv6 is set up on the system, configuring an interface using ifconfig is as easy as:

```
# ifconfig hme0 inet6
hme0: flags=2000841<UP,RUNNING,MULTICAST,IPv6> mtu 1500 index 2

      inet6 fe80::d01:77ff:fe29:aaeb/10
```

Note that the IP was assigned automatically. The FP `fe80` indicates that the address is locally linked.

Some tools for IPv4 have IPv6 equivalents. These include `netstat`, which can be run as family type ip6:

```
# netstat -f inet6
TCP: IPv6
Local            Remote          Swind Send  Rwind   Recv       State      If
Address          Address          -Q          -Q
------------------------------ ------------------------------------------
localhost.33382  localhost.33319  32768  0  32768  0   ESTABLISHED
localhost.33319  localhost.33382  32768  0  32768  0   ESTABLISHED
```

DNS is configured similarly, except the records are longer. A new record type is the AAAA (quad A).

There is only one routing daemon for IPv6, which is called `in.ripngd`. There are nc configuration parameters for this daemon.

The neighbor discovery protocol is implemented by `in.ndpd`. This daemon is startec only if a configuration file `/etc/init/ndpd.conf` exists.

Special thanks to
Matthew Kortas
for contributing this
Cramsession. Please visit his
site at
http://acm.cse.msu.edu/~kortasma