Chapter 12
-
Network Services

INDEX

A little bit of Theory

Knowledge is knowing where to find it! Nevertheless, it is better to have heard of some things.

Protocols and Models

We want to discuss network communication via **TCP/IP**. This acronym stands for a set of standards and protocols, where the *Transmission Control Protocol* (TCP) and the *Internet Protocol* (IP) are the most important. They are organized into:

Addressing:
IP addresses, which uniquely identify every host on the network, deliver data to the correct host.
Routing:
Gateways deliver data to the correct network.
Multiplexing:
Protocol and port numbers deliver data to the correct software module within the host.

The *OSI Reference Model* illustrates the access of applications to the network This is implementation independent and helps a lot in heterogeneous networks. Table 1 shows the standardised terminology, and right column some examples what layer includes.

7	**APPLICATION LAYER** consist of application programs that use the network	
6	**PRESENTATION LAYER** standardizes data presentation to the application	
5	**SESSION LAYER** manages sessions between applications	(i.e.. RPC Protocol)
4	**TRANSPORT LAYER** provides end-to-end error detection and correction	(TCP or UDP Protocol)
3	**NETWORK LAYER** manages connections across the network for the upper layer	(IP-Adresses, Routing)
2	**DATA LINK LAYER** provides reliable data delivery across the physical link	(MAC-Adresses, Duplex-Mode)
1	**PHYSICAL LAYER** defines the physical characteristics of the network media	(Hardware)

Table 1 - OSI Layer

A network packet, meaning the logical form in which the information is transported via a network, contains for each of these layers, as it is being used for the specific communication, its own section. Layers 1 through 4 are organized on behalf of the operating system and/or the TCP/IP protocol. The layers above those correspond to applications.

While layers 1 through 3 are used by all applications, there are big differences in the upper layers. Layer 4 stands for the method to organize a connection. Ordinarily the protocols *UDP* and *TCP* are used here.

UDP stands for *User Datagram Protocol*:

> *The User Datagram Protocol gives application programs direct access to a datagram delivery service, like the delivery service that IP provides. This allows applications to exchange messages over the network with a* ***minimum of protocol overhead****.*
> *UDP is an unreliable,* ***connectionless*** *datagram protocol. As noted previously, "unreliable" merely means that there are no techniques in the protocol for verifying that the data reached the other end of the network correctly. Within your computer, UDP will deliver data correctly. UDP uses 16-bit Source Port and Destination Port numbers in word 1 of the message header, to deliver data to the correct applications process.*

TCP stands for *Transmission Control Protocol*. This protocol goes through significantly more expense to safeguard data integrity:

> *Applications that require the transport protocol to provide reliable data delivery use TCP because it* ***verifies that data is delivered across the network*** *accurately and in the proper sequence. TCP is a reliable,* ***connection-oriented****, byte-stream protocol. Let's look at each of the terms - reliable, connection-oriented, and byte-stream - in more detail.*
> *TCP provides reliability with a mechanism called Positive Acknowledgment with Re-transmission (PAR). Simply stated, a system using PAR sends the data again, unless it hears from the remote system that the data arrived okay. The unit of data exchanged between cooperating TCP modules is called a segment. Each segment contains a checksum that the recipient uses to verify that the data is undamaged. If the data segment is received undamaged, the receiver sends a positive acknowledgment back to the sender. If the data segment is damaged, the receiver discards it. After an appropriate time-out period, the sending TCP module re-transmits any segment for which no positive acknowledgment has been received.*

The correlation between the transmitted data and the applications is accomplished by use of **port numbers**. A lot of applications/server programs bind itself to dedicated ports. If this correlation is documented in a standard RFC we speak from so-called **well-known ports**. HP-UX will handle information this correlation in file **/etc/services** (e.g. telnetd == port 23 ; ftpd == port 21 ; rpcbind == port 111).
An other way to announce which server listens at which port is contained in "**R**emote **P**rocedure **C**alling" protocol technologies. There server process will registrate itself at a central "location brooker" process which listens to a well known port so that clients can ask them to get portinformation. (SUN-RPC: rpcbind == port 111 ; DCE-RPC: rpcd/dced == port 135).

IP-Addresses and Netmasks

By means of the IP-Address we can specify the network in which the system resides and the number of the *Host* in this network. Network information is important for routing, to find way between networks to target. Host information identifies special host in their own network.

How many bits of the 4-byte large IP-address are used is determined via the netmask.

Classically one distinguishes between 3 classes of networks using predetermined netmasks and some reserved areas of address range.

1st byte	type	network part	hosts part	netmask
< 128	class A	byte 1	byte 2,3,4	255.0.0.0
128 - 191	class B	byte 1,2	byte 3,4	255.255.0.0
192 - 223	class C	byte 1,2,3	byte 4	255.255.255.0
224 - 239	multicast	n/a	multicast group	
> 239	reserved	n/a	n/a	n/a

Official network addresses have since become a rare commodity and the philosophy of small, medium and large networks not longer fits the bill of the current field of networking. So we have to speak about subnetting and so-called *classless IP-addresses*. The theory behind this is found in RFC 1122 and 1878, respectively. As effect we see that often to each address the number of bits is attached, which specify the network part.

Using an example we will clarify how netmasks are used:

Network administration (IT) has issued us an official IP-address and netmask. Now we want to determine in which subnet this IP-address is and which broadcast address belongs to it

Example: IP-address: 15.140.10.113 netmask: 255.255.248.0

		binary
IP Address	15.140.10.113	`0000 1111 . 1000 1100 . 0000 1010 . 0111 0001`
Netmask	255.255.248.0	`1111 1111 . 1111 1111 . 1111 1000 . 0000 0000`

Using the logical AND results in the Net portion of the IP address `0000 1111 . 1000 1100 . 0000 1000 . 0000 0000`

in decimal: **15.140.8.0**

setting all hostbits to 1 results in the broadcast address `0000 1111 . 1000 1100 . 0000 1111 . 1111 1111`

in decimal: **15.140.15.255**

Special HP-UX Feature - Usage of RFC 1122

According to RFC 1122 not all possible subnet address combinations are usable. The operating system HP-UX tests these conditions upon activation of the IP-address with the `ifconfig command` (`ioctl` Error message for relevant problems). Since newer RFC's impose less restrictions, one can turn off this test. i.e. in HP-UX 11.X by using:

```
    # ndd -set /dev/ip ip_check_subnet_addr 0
```

Private IP-Addresses

Should one build a network with private IP-addresses, with no connection to the Internet/WAN, it is still recommended that the IP-addresses follow the RFC 1918 standard.

The following address ranges are exclusively used for private nets and never allotted for the Internet.

1	Class A network number:	10.0.0.0	
2	Class B network numbers:	172.16.0.0	...172.31.0.0
256	Class C network numbers:	192.168.0.0	...192.168.255.0

Networking Fundamentals

Installation

Basic statement: Install driver software before installing hardware !

To checkout correctness of driver software and required patches you can use HP-websites for example: http://wtec.cup.hp.com/~netux/lanlinks/link_index.html (HP internal)

Configuration of NIC requires some more informations. You will need at least:

- IP-Address,
- Name for this IP,
- For 100BaseT, information regarding the settings of the Switch/HUB (speed, duplex mode)
- environment information as: netmask, broadcast address, defaultgateway, nameserver

You can configure a new NIC with `sam` or with script `set_parms` .

But it is also possible to add informations via `vi` to configuration files.
System will handle all required informations in following files:

```
/etc/rc.config.d/netconf       ## IP-addr., routing
/etc/rc.config.d/hp*           ## duplexmodi, speed , MAC
/etc/hosts                  ## local names
/etc/resolv.conf               ## nameservers
```

Init process will use this information in startup (`/sbin/rc2.d/S*`) to configure NIC correctly.
It will be done by commands

```
ifconfig                       ## initialize interface
route                          ## add routes
lanadmin                       ## change speed/duplex settings
```

This commands can be used to change network settings at a running system. But be aware that some software dislike changes of networkenvironment at runtime.

Check file `/etc/rc.log` to see if there was any problem in statup.

Sometimes special kernel default values must be changed in order to make special configurations possible. This is accomplished with the following commands:

`ndd`	(HP-UX 11.X)
`nettune`	(HP-UX 10.X)

Commands allow you to change kernel settings without reboot. To make this automatic upon each reboot we utilize at HP-UX 11.X the appropriate entries in the file

```
/etc/rc.config.d/nddconf
```

Routing

When we discuss routing, we have to differentiate between two aspects. On the one hand, to determine the basic functions of the operating system, by which interface the information is passed, and on the other hand, how the information finds its way between the nets. We want to talk here only about the first aspect and functionalities, which serve the second, completely suppressing the latter.
Only one hint from support:
Disable `gated, arpd` and similar programs if you do not know exactly that it is really required and how it is to configure!

You need to be familiar with the following terms, if you like to understand it:

• Loopback Interface	virtual interface to allow locally running processes to communicate with networking methods without usage of any network hardware.
• Host Routes	routes which target is a single host
• Network Routes	routes which target is a network
• Default Route	the way to any other, not explicit listed, system or network

The systems store the corresponding information in the routing table and we can show this using the command:

```
# netstat -rn
```

Kernel will decide which interface a network package has to go in correspondence to this list.

To interpret the output one should consult the, in this case helpful, man-page:

```
# man routing
```

The `route` command is used to inform the system about the routes. So that after a restart the routes are again set, one should enter these accordingly in the file `/etc/rc.config.d/netconf` (using `vi` or `sam`).

Good to know:

- Initialization of interface with `ifconfig` command creates two entries in routing table of system, one host route to new local IP and one network route to subnet to which this IP belongs.
- Source IP of packages of outgoing connections will be determined by routing table (HP-UX 11.X) so we see sometimes funny effects at systems where IP addresses at virtual interfaces are defined.
- When initializing the interface, net and broadcast addresses can be directly modified even normally system defaults will be used. The actual values used by the system can be checked using:

`# ifconfig` <interface>	also shows netmask and broadcast address
`# netstat -in`	also shows the net addresses
`# netstat -rnv`	also shows netadresses and netmasks

- Since HP-UX 11.00 there is a *Dead Gateway Detection* Mechanism, i.e. the active gateways in the routing table are ping-ed regularly to see if they are still active. If there is no answer then the appropriate entry is deactivated. Problems are presented to this technique by rounters which don't answer the ICMP-ECHO-REQUESTS, for example Firewalls and/or Routers, which employ virtual IP-address loadbalancing. In those cases this feature may be disabled using:

  ```
  # ndd -set /dev/ip ip_ire_gw_probe 0
  ```

Name Resolution

A central service, being used by almost every program, is name resolution, i.e. the correlation between names and IP-addresses. By definition problems are thereby a frequent error cause. At the basis are the libc implemented functions gethostbyname() and getnamebyaddr().
By means of these basic functions we can use different sources of information:

- the local file /etc/hosts
- the Network Information System - NIS (previously *Yellow Pages*)
- the Domain Name System - DNS
- NIS+
- LDAP

We will limit ourselves to the first three techniques. Due to the complexity of the subject matter and the limited number of installations we will not discuss NIS+, and a separate LDAP-section will come once this techniques has been more widely adopted.
The default behavior which source will be used is documented in the `nsswitch.conf` man-page.

CAUTION:
It is highly recommended to create the `/etc/nsswitch.conf` file in order to define the desired behavior.

Sample files may be found on the system under `/etc/nsswitch.*` and/or `/usr/newconfig/etc/nsswitch.*` . The default recommendations are however not always meaningful.

Example: (recommendation)

```
# grep host /etc/nsswitch.conf
hosts: files [NOTFOUND=continue] dns [NOTFOUND=continue UNAVAIL=continue]
nis
```

Entry "`files`" corresponds to the most simple method to resolve names at system, this is the `/etc/hosts` file. Independent of other approach there should at least be entries present for the names `localhost`, `loopback` , the eventual hostname an local IP-addresses, e.g.:

```
# grep loopback /etc/hosts
```

```
127.0.0.1       localhost       loopback
# grep <hostname> /etc/hosts
15.140.10.113   grcdg089
```

Will the system be a DNS-Client?

The existence of the file `/etc/resolv.conf` is sufficient. Sensibly this file has the following form (Domain name and IP-addresses naturally correspond to the respective environment):

```
# cat /etc/resolv.conf
domain          grc.hp.com
nameserver      15.137.22.252
```

Shortnames will be expanded to complete DNS-names with the designation specified in the `domain` statement. If information for systems from different subdomains is to be found, one can use a search statement here, e.g.:

```
search       grc.hp.com, bbn.hp.com
```

Thereby one may search all named subdomains for the appropriate name.

There may be up to three nameservers in this file. Thereby one avoids problems should one of the servers be unreachable. Current libc patches allow to customize timeout policy for fallback, see new `retrans` and `retry` options.

It is self-evident that we have to use IP-addresses and not servernames in file `/etc/resolv.conf`.

How does a system become a NIS-Client?

A system becomes an NIS-client when the command `domain name` the NIS-domainname sets and the ypbind process finds an NIS-server.

To enable it in startup process check existence of lines

```
NIS_CLIENT=1
NIS_DOMAIN=<Name of the NIS-Domain>
```

in the `/etc/rc.config.d/namesrvs` file.
The startup script `/sbin/init.d/nis.client` takes over the stopping and starting of this function.

To find a NIS-server `ypbind` process will asks at startup via broadcast request within the network for a server and binds itself to the first server that answers.

If a server from another subnet is to be accessed then options `ypset` or `ypsetme` can be be used (compare the `ypset` and `ypsetme` man-pages; additional entries in `/etc/rc.config.d/namesrvs` are required).

Another possibility to force `ypbind` process to bind itself at specific NIS-servers was introduced by newer patches: PHNE_24034 (UX 11.00) / PHNE_24910 (UX 11.11).
Actually `ypbind` process will check the existence of a file

```
    /var/yp/binding/ypservers
```

at startup nd will try to connect to servers which are named in this file.
Best way to create such a file is to use the command:

```
    # ypinit -c
```

NIS Troubleshooting, first steps:

You can check:

to which NIS-domain client belongs	`# domainname`
to which server did the client bind itself	`# ypwhich`
which maps are made available to NIS	`# ypwhich -m`
content of a specific map	`# ypcat -k` <name of NIS map>

How to check name resolution?

Simplest method for troubleshooting is to check output of following commands for consistency:

```
     # nslookup <suspect name>
     # nslookup <expected IP>
     # ping -o <suspect name> -n 2
```

Please be aware that different programs handles own caching mechanisms for names and resolver policy to improve performance.

Network Services

A standard HP-UX installation is capable of utilizing several *network services*. The configuration takes places by means of various files in the directory `/etc/rc.config.d` and/or the `inetd` configuration file `/etc/inetd.conf` (accessed using `sam` or `vi`).

On which ports a running system offers a service is shown in the output of the command:

```
  # netstat -an | grep LISTEN
```

Which specific program uses a port is best examined with the tool `lsof`:

```
  # lsof -i :<number of port>
```

You will see that a lot of ports will be handled by `inetd` process.

`inetd` is the *internet -super-server*; it takes over listening on the network for all programs entered in `/etc/inetd.conf`.
For syntax of this file consult `man inetd.conf`. Lines which starts with " `#`" are only comments. Work of `inetd` requires information of files `/etc/services` and `/etc/protocols` or corresponding NIS-maps (see also "`man nsswitch.conf`") to determine which service has to listen at which port.

Recommended inetd usage:

Changes to the `/etc/inetd.conf` file are only activated after executing

```
# inetd -c
```

(or after restart of process)

Turn on/of logging of network connections

```
# inetd -l
```

Turn on/of logging of local work

```
# inetd -b
```

Stop inetd process (Do not use `kill -9` !!!)

```
# inetd -k
```

Logging informaion can be found in `/var/adm/syslog/syslog.log`.

Best way to get informations about usage and handling of special services which are started by inetd is to read manpages of server daemons.

Commands and Utilities

Below we will have an overview of some helpfully commands and tools. For proper execution you should consult the corresponding manual pages.

Basic Configuration Commands

`# ioscan -funC lan`
didplay hardware recognition of kernel

`# lanscan`
display hardware state, paths and names how system has recognized NIC

`# ifconfig`
configure and display software state, IP and network parameters of NIC

`# lanadmin`
display and customize driver settings of interface (speed, duplex modi, pmtu, ...)

`# netstat -[ainrv]`
display a lot of statistics and status information about network situation

`# route [add|delete]`
display, add and delete entries of routing table in kernel

`# ndd`
display and customize different kernel settings for networking. See also http://wtec.cup.hp.com/~netux/NDD (HP internal)

Basic Utilities

`# arp -[ansd]`
display and customize arp table in kernel

`# linkloop`
test „layer 2" connectivity, only for interfaces with HP drivers

`# ping [-o]`

test „layer 3" connectivity

```
# traceroute
```
test network connectivity and display route to target

```
# dlpiping
```
This utility is very useful to check connectivity in ServiceGuard environments. Download it from ftp://einstein.grc.hp.com/TOOLS/SG/ (HP internal)

Additional Resources

```
# nettl / netfmt
```
`nettl(1M)` (network tracing an logging) is the HP-UX builtin network log- and tracetool, `netfmt(1M)` is a tool to format binary output of `nettl`. See section below for details.

```
# /opt/nettladm/bin/nettladm
```
graphical front end for nettl and netfmt, available since PHNE_18218

```
# ethereal
```
trace and format tool with nice graphical interface (open source)

```
# tcpdump
```
tracetool with more flexible filter possibilities as nettl (open source)

```
# lsof
```
tool to check usage of local resources, e.g. which port will be used by which process (open source)

```
# tusc
```
tool to trace system calls of a running process (open source)

Tracetools on other operating systems: `snoop`, `netmon`

The open source programs can be found with their source code at http://hpux.aknet.de (non HP). You can find a lot of helpful utilities there.

An extensive collection of HP-UX utilities can be found at:
ftp://einstein.grc.hp.com/TOOLS/ (HP internal)

Network Troubleshooting with nettl

`nettl(1M)` is a useful tool to trace the network. It is located in `/usr/sbin/` directory. It produces a binary output file that can be formatted as desired using `netfmt(1M)`.

1) check if `nettl` is available:

   ```
   # nettl -status
   ```
   should display the current logging configuration.

2) Start tracing:

   ```
   # nettl -tn pduin pduout -tm 10000 -e all -f /tmp/mytrace
   ```

   ATTENTION: The trace file may grow extremly fast depending on net traffic:

   ```
    # ll /tmp/mytrace*
    -rw-------    1 root     sys    1647791 Jun 19 15:31 /tmp/mytrace.TRC000
    # ll /tmp/mytrace*
    -rw-------    1 root     sys    1841065 Jun 19 15:32 /tmp/mytrace.TRC000
   ```

   NOTE: Its recommended to replace the nettl option "`-e all`" with the option "`-e ns_ls_ip`" if trace data from network layer 1 and 2 is not required. The advantage is that the such a trace can be formatted by the graphical tool **ethereal** (http://hpux.asknet.de).

3) Reproduce the problem:

   Start whatever you like to trace.

4) Stop tracing:

   ```
   # nettl -tf -e all
   ```

5) Format the trace file

   Now you need to filter and format the binary output:
   ```
   # netfmt -N -l -f /tmp/mytrace.TRC000 >/tmp/mytrace.txt0
   ```

   and, if available:
   ```
   # netfmt -N -l -f /tmp/my-trace.TRC001 >/tmp/mytrace.txt1
   ```

   You may use filter here. See `netfmt` manual page for details.

6) Analyze the formatted output

The Chapter Network Connectivity contains information about `nettl`, too.

NFS

Basic Functionality

When a client system mounts a filesystem via NFS the following happens:

- The client system generates an RPC request to see if and on which port on the server system a mount daemon - `rpc.mountd` - is reachable (request to portmapper at port 111).
- The port mapper of the server system communicates to the client on which port the `rpc.mountd` can be reached.
- The client systems sends a mount request for the corresponding file system to the server's mount daemon.
- The mount daemon checks to see if the file system is exportable to the client. If so, it then sends the client the file handle of the corresponding file system. Otherwise, it answers with "access denied".
- If the client has the file handle then it generates an RPC request to see if and on which port on the server systems an nfs daemon (`nfsd`) is reachable.
- The port mapper of the server system communicates to the client on which port the `nfsd` can be reached (at HP-UX normally port 2049)..
- The client system sends an NFS request for the corresponding file handle to the servers nfs daemon.
- The nfs daemon searches for the data belonging to the file handle and sends it to the client.
- Thus the client has mounted the file system. For each additional access to the mounted file system only inquiries to the special file handle are sent to the nfs daemon of the server.

Actually all available HP-UX versions supports NFS version 2 (NFSv2) and version 3 (NFSv3) with network protocol UDP but only HP-UX 11.X are able to use NFSv3 with TCP protocol (use actual patches!).

The central configurations file for the NFS functionality is the `/etc/rc.config.d/nfsconf` file. Here are, on one hand, the variables with which the basic functionality is defined and, on the other hand – depending on the version of the operating system and patch level – additional configuration possibilities.

What is configurable on an NFS client?

Actually nothing! The kernel of the system must contain the NFS code, but nowadays that should always be the case. That way we have ensured basic functionalty. For additional functionality one needs more processes:

`rpc.lockd` and `rpc.statd`	for file locking
`biod`	I/O Organization
`automountd`	if you like or need it

Additional functionality requires additional daemons .

`rpc.statd`	for file locking
`rpc.lockd`	for file locking

The startup script ensures that processes are started if you have set in file `/etc/rc.config.d/nfsconf` :

```
NFS_CLIENT=1
```

Thus using

```
# /sbin/init.d/nfs.client start   or
# /sbin/init.d/nfs.client stop
```

the services may be started and stopped and this also ensures that upon reboot everything still functions.

Filesystems that are always to be mounted with NFS are entered in `/etc/fstab`, the temporary ones may be made available by using the automount daemon.

File `/etc/rc.config.d/nfsconf` contains further options to customize system. So You define a value for number of biod (do not change it, if no problems are visible)

```
NUM_NFSIOD=4
```

and options to start automountd: `AUTOMOUNT` and `AUTOFS`

Special effects:

- **WARNING:** In the current script you stop both the client **and** the server if you try to stop client with it!
- With respect to problems in introduction phase of NFSv3 an additional option is usable in `nfsconf` file at HP-UX 10.20.
  `MOUNT_VER=2`
  defines default value for NFS-mount requests, there to version 2.
- Usage of protocol TCP for NFSv3 at HP-UX 11.0 requires a statement
  `NFS_TCP=1`
  in this file . mount command checks existence of such line before it accept option `"proto=tcp"`.

What is to be configured on the NFS-Server?

On the server there is more to do. We need the following processes:

`rpcbind`	the port mapper (`portmap` on UX 10.X)
`rpc.mountd`	
`nfsd`	you need more of these, rule of thumb: four per processor

Additional functionality requires additional daemons .

`rpc.statd`	for file locking
`rpc.lockd`	for file locking
`rpc.pcnfsd`	to handle PCNFSD protocol requests

The line

```
NFS_SERVER=1
```

in the file `/etc/rc.config.d/nfsconf` allows to start and stop daemons processes using

```
# /sbin/init.d/nfs.server start
# /sbin/init.d/nfs.server stop
```

and ensure work also in startup.

Please check in `/etc/rc.config.d/nfsconf` options to customize system behaviour.

Performance discussions shows us importance of adjustment of value of **`NUM_NFSD`**=<number>.
It determines number of nfsd processes which handle UDP-NFS requests - today most of NFS-load. Number should be greater than 4 times number of processors and if system is NFS-client of itself at least greater than 32.

`PCNFS_SERVER=1`
is only required if there are DOS or Windows clients which use special PCNFSD protocol.

With respect to problems in introduction phase of NFSv3 there is an option in `nfsconf` file of HP-UX 10.20:

```
        MOUNTD_VER=2
```

which restrict `rpc.mound` that it allows only NFSv2 mount requests.

You also need to maintain the `/etc/exports` file. This is where you need to enter the file systems that the client may access.
`"exportfs -a"` command trigger rpc.mountd to read this file and make filesystems accessible.
`exportfs` command called without option shows you what mountd currently has exported (see man pages).

`# showmount -e` <servername>
shows you at any client which filesystems of server are exported.

Warning:
Filesystems that are managed by a ServiceGuard package and are available via NFS do

normally not appear in the /etc/exports file because the package startup performs the export via "`exportfs -i`" command.

Involved Scripts and Processes

If one knows exactely where the problem is being caused then it is mostly already solved. Therefore, another overview of the involved processes.

Client	Server
Administration Scripts:	
`/etc/rc.config.d/nfsconf`	`/etc/rc.config.d/nfsconf`
`/sbin/init.d/nfs.client [stop|start]`	`/sbin/init.d/nfs.server[stop|start]`
Worker Processes:	
	`rpcbind`
	`rpc.mountd`
`biod` (a sufficient number)	`nfsd` (min. 4 per CPU)
`automount(d)` (if required)	
File Locking Processes:	
`rpcbind`	`rpcbind`
`rpc.lockd`	`rpc.lockd`
`rpc.statd`	`rpc.statd`

What Information Is Needed When There Is a Problem?

The following information is needed in order to narrow down the problem:

Problem description and actions taken.
A description, as specific as possible, of the network between the systems.

For the Client	For the Server
OS Version & Patchlevel	OS Version & Patchlevel
Name resolution for the server and client (Name after IP and IP after Name)	Name resolution for the server and client (Name after IP and IP after Name)
File system structure (where is what mounted?)	File system structure (where is what mounted?)
Status and structure of the mount points	Status and structure of the mount points
nettl-trace of the problem event	nettl-trace of the problem event
`showmount -e <server>`	`exportfs`
`nfsstat –m`	
`nfsstat –c`	`nfsstat -s`
`netstat -s`	`netstat -s`
`mount -v`	`mount -v`
`ping -o <server> -n 3`	`ping -o <client> -n 3`
`rpcinfo -u <server> nfs`	
`rpcinfo -u <server> mountd`	
`rpcinfo -p <server>`	

For **Problems with NFS-filelocking:**
http://wtec.cup.hp.com/~dolker/Cookbooks/rpc.lockd/index.htm (HP internal)
I have not yet found anything better #;-)!!! The topic is exhaustively covered there.

For messages like **NFS server not responding** see:
http://wtec.cup.hp.com/~netux/ONC/NFS/nfs_server_not_responding.htm (HP internal)

and, last but not least, there is also a nice text on the topic **NFS Performance**
"NFS Performance Tuning for HP-UX 11.0 and 11i Systems" at
http://wtec.cup.hp.com/~netux/ONC/onc_bookmarks_2.htm (HP internal)

Most Common Error Messages at Client

`"nfs server … not responing"` that means that answer of server do not reach client fast enough, so it is to check if request reaches server, if server is able to handle request, to organize datas and to send an answer, if answer reaches client and if client is able to handle it.

" … access denied" that means that rpc.mountd do not allow request, check at client with `"showmount -e <server name>"` what server allows and login to server from this client to check how server recognize client (`# who -R am I`)

Automounter

Basics

You can use the automount daemon to make mounting of NFS file systems more flexible.

Since some days (6/2002) it is also possible to handle ISO 9660 filesystems with Rock Ridge extentions with automountd. It requires all actual "Rock Ridge" related patches and usage of filesystemtype cdfs.

Current HP-UX installations contain two technically different versions whose basic configuration, however, is identical.
The choice is made in the file `/etc/rc.config.d/nfsconf`

`AUTOMOUNT=1`	determines if the nfs client script starts the daemon
`AUTOFS=1`	determines if the technically newer program is started

The central configuration for the daemon is:

```
/etc/auto_master
```

This contains the information or alternatively directs most to other files, *maps*, with the information regarding which file systems are to be mounted where by which server.

Useful hints:

- If the system is an NIS client then, by default, the daemon will use the configuration which is made available via NIS unless the file `/etc/nsswitch.conf` says otherwise.
- Actually the man-pages for the `automount` topic is garbage. Please use it carefully. (as of 6/2002).
- Best help for intelligent configurations can be found at http://docs.hp.com and *Managing NFS/NIS* from Hal Stern.
- The *new fashioned* automount daemon process is easily recognized by the path of the binary and the last letter of its name `/usr/lib/netsvc/fs/autofs/automountd`
- Various problems in software quality had led to the previous recommendation to use the tried and true, older technology. In the meantime, given a current patch level, the newer one is more stable.
- The *new fashioned* automount daemon is mandatory when newer NFS-features are used (NFSv3, Files >2GB).
- The *new fashioned* automount includes some nice new features, e.g.

`# mount -v |grep autofs`	shows you which filesystems
`# automount -v`	triggers automountd to reread configuration

WARNING:
Never use **`kill -9`** on an automount daemon! Since the daemon administers information for the kernel this will corrupt the entire system and you know, there is only one way to cleanup such situation - Reboot!

What Can Be Tweaked on the Automounter?

You can trace a problem by turning on logging of the process. This may be enabled by a start option, or in the `nfsconf` file, or using

```
# kill -SIGUSR2 <process id>    or
# cd /net/=<debug level>
```

(the latter only when /net is in place for the special map `/net -hosts`). The information is found in file `/var/adm/automount.log`.

You can turn off logging via

```
# cd /net/=0                   or with
# kill -SIGUSR2 <process id>
```

If the automount process no longer responds to the SIGUSR2 signal it most often will also not respond to the SIGTERM signal. In this case there is no way to fully restore the functionality of the daemon other than to reboot.

Well Known Problem

The old-fashioned automount daemon **hangs** at startup.

Problem description:
The system no longer boots, it hangs starting the NFS-client subsystem.

Cause:
The daemon works with symbolic links following `/tmp_mnt/...` . Should such a link not get cleaned up after the daemon process terminates (`kill -9` / panic / ...) then it will impede all restarts.

Solution:
Start the system such that no automount process is running. Scan all files for which the daemon is responsible and delete all symbolic links that are found, also

```
  # cd /tmp_mnt
  # rm -rf *
```

Then it should all work.

DNS Server - Domain Name System

DNS is the internet's central name service. The term BIND, used during the implementation, is short for *Berkeley Internet Name Domain*.
Currently, HP provides with installation media BIND versions 4.9.7 and a very special HP-customized version "8.1.2 upgrade version 1.2". Both are special HP adaptations and cannot be compared directly to the OpenSource Releases, see also http://www.software.hp.com/products/DNS_BIND/.
Actually (6/2002) you can find an official HP-version of Bind 9.2 for UX 11.11 at http://www.software.hp.com. A corresponding version for UX 11.00 is announced for end of summer.

If you need newest features or you prefere open source products best way is to compile actual code from http://www.isc.org (non HP) yourself.

WARNING:
Before configuring DNS-servers one might want to look at a book to understand the concepts. Practical are, for example "DNS and BIND" by P.Albitz and C.Liu from O'Reilly & Associates. See http://rogue.boi.hp.com/OReilly-CompleteBookshelf (HP internal).

The Main Principle

The `named` process runs on the DNS server, listens on port 53 for requests and answers according to the existing information. Local files or other DNS servers may serve as information sources.

Configuration

Starting named during bootup:

In the `/etc/rc.config.d/namesvrs` file enter

```
NAMED=1
```

The central configuration files:

`/etc/named.boot`	for BIND Versions 4.X alternatively
`/etc/named.conf`	for newer BIND Versions (8.X, 9.X)

Please be aware that syntax of central configuration files was changed heavily between version 4.X and newer ones. The `directory` statement in this file refers to the directory where all locally administered information is to be found.

In a simple network neighborhood using the script **`hosts_to_named`** you can get all needed configuration files from a well maintained `/etc/hosts` file (see man-page). If you configure the name server using `sam` you also work with this script.
Actually versions of hosts_to_named will check version of `/usr/sbin/named` binary to

decide if it creates a file `/etc/named.boot` only or both. In addition it will create required database files which are version independent.

Notes:

- Never start `named` without a reasonable configuration – otherwise there will be confusion!
- It is possible to use other central configuration files - start named process with `"-f <file>"` option - but it would be helpfully for your colleagues if you do not do it.

Additional Information

If you want to know more about the theory then you should take some relevant training or learn read books. Besides that, there is a wealth of information available on the internet.

General Knowledge:

TCP/IP – Network Administration	C. Hunt	O'Reilly & Associates
Managing NFS and NIS	H. Stern	O'Reilly & Associates
DNS and BIND	Albitz/Liu	O'Reilly & Associates

see http://rogue.boi.hp.com/OReilly-CompleteBookshelf/index.html (HP internal)

Internet Standards - RFC: http://www.rfc-editor.org (non HP)

HP-UX Cookbooks:

http://docs.hp.com
http://wtec.cup.hp.com/~netux/ONC/ (HP internal)
Network Services Knowledge Tree:
at ITRC http://itrc.hp.com or
http://www.grc.hp.com/cgi-bin/IV/SW/bin/iv/InterView.cgi (HP internal)

LAN Interface Driver:

http://wtec.cup.hp.com/~netux/lanlinks/link_index.html (HP internal)

DNS:

http://www.software.hp.com/products/DNS_BIND/index.html
http://www.acmebw.com (non HP)
http://www.isc.org (non HP)

Security:

Germany	http://www.bsi.de (non HP)
International	http://www.cert.org (non HP)
Bugs	http://www.rootshell.com (non HP)