21 - Network card configuration:

Network Devices Configuration:

- Network card installation:
  - Using Yast2
  - Automatic Detection of Network Card Using `modprobe`
    Automatic try of all network cards until one fits:
    `modprobe -t net \*`
    Starts trying all the net cards until the appropriate one is found.
    It might hang the PC. Then reboot and try them one by one.

  - Individual module can be loaded one by one with the command:
    `modprobe <modulename>`(without `.o`)

  - To check which card got successfull if any, do the command
    `dmesg | grep eth0`

  - In the file `/etc/modules.conf` or
    `/etc/conf.modules` (RedHat) the following line should be:
    `alias eth0 <modulename>`(without `.o`)
    `options <modulename> io=0x280 irq=10`
    This last line exists only if it is not a PnP Network card.

  Note: The file `cd /lib/modules/$(uname -r)/modules.dep` has the list of dependencies for all loadble modules. `insmod` can also be used but it doesn't check for dependencies...!!! `modprobe` uses `insmod` but also checks the presence of dependencies and load them as well.

  - To display the already loaded kernel modules:
    `lsmod`
    - The word `(autoclean)` means that the kerneld will load it as needed and unload it when not used any more.
    - The module name in last column in square brackets [ ] , indicates that this module is a dependency of a main module loaded as well.

- **Network Real Cards IP addressing (`eth0, eth1..`)**
  - Using YaST2
  - Using `ifconfig` but not recommended for beginners since it is not permanent.

- **Virtual cards IP addressing (`eth0:0....eth0:9`)**
  - Using YaST2 (recommended)
  - `ifconfig eth0:doz IP_Addr`
    (also good inside `/etc/init.d/boot.local`)

  Standard configurations (permanent)
  `/etc/sysconfig/network/ifcfg-eth0`
  Standard Configuration file of `eth0`
  `/etc/sysconfig/network/ifcfg-eth0:elop`
  Standard Configuration file of `eth0:elop`

- **Loopback device (`lo`)**
  Principle of internal Packet Routing through the Loopback

- Network config files involved:

`/etc/HOSTNAME`
Local host name

`/etc/hosts`
Hosts names with IP addresses for name resolution
Format:
`<Host IP Addr.> <hostname> <alias1> <alias2>..`

`/etc/networks`
Networks names with IP addresses for name resolution
Format:
`<netname> <Net IP Addr.>`

`/etc/host.conf`
Name resolution sequence file.
Syntax:

`order hosts bind yp`	Order of name resolution.
`multi on`	Allows to have multiple names for the same IPaddress.(`/etc/hosts`)

`/etc/resolv.conf`
List of DNS IP Addr. that will be searched for name resolution.
The sequence is as read in the file.
`search <domain.name1> <domain.name2>` etc.
is the list of domain names that will be appended to any non-fully qualified host name (FQDN).
eg. `search michel.home elop.de`

`/etc/manuf`
List of manufacturer and MAC Addresses for ethernet cards.

- **Start and Stop of Networking Interfaces** (only with SuSE):

`/etc/init.d/network {start|stop|restart|reload|status|probe}`
(SuSE) `rcnetwork {start|stop|restart|reload|status|probe}`

`start`	- Loads the Network card module and starts it
`stop`	- Unloads the network card module.
`restart`	- Does a `stop` and then `start`
`reload`	- Same as `restart`
`status`	- Status of network card(s) excluding loopback (lo) same result as eg. `ifconfig eth0` command.
`probe`	- Checks if the network card is UP and display the word restart if not UP.

- **Network commands:**

`hostname -i`	Host IP Address
`hostname -d`	Host Domainname
`hostname -f`	Host **F**ully **Q**ualified **D**omain **N**ame
`ifconfig`	To get status of all installed network interfaces
`netstat -i`	Display installed interfaces and parameters
`lsmod`	Display all the loaded kernel modules
`ip addr`	Displays the IP addresses of local interfaces

ARP Protocol

`arp -a`	Displays the ARP table
`arp -d` *IP*	Erases the entry of the IP in the ARP table
`arping -D` *IP*	Checks for duplicate IP addresses on a network

To add a `dummy0` network device to the system:

`ifconfig dummy0 ``hostname -i```

Sets the `dummy0` device to the host addr

To add any number of extra IP addresses to an ethernet card eg. `eth0`

`ifconfig eth0:`*`xxxx`* `192.168.10.101 netmask 255.255.255.0 up`

This will add the address `192.168.10.101` to the alsready existing one for `eth0` as `eth0:`*`xxxx`* where '*xxxx*' can be any a-z,A-Z,0-9

eg. `ifconfig eth0:elop 192.168.70.160`

ifconfig

```
eth0   Link encap:Ethernet  HWaddr 00:40:05:5E:B5:2E
          inet addr:192.168.10.200  Bcast:192.168.10.255
Mask:255.255.255.0
          UP BROADCAST RUNNING PROMISC MULTICAST  MTU:1500  Metric:1
          RX packets:2467 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2176 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          Interrupt:11 Base address:0xd400

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:3924  Metric:1
          RX packets:40 errors:0 dropped:0 overruns:0 frame:0
          TX packets:40 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
```

Ethernet `Hwaddr`: Ethernet unique hardware address set by manufacturer

`MTU`: Maximum length of packets allowed for this interface

NETSTAT

netstat List open connections (sockets)

```
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags       Type       State         I-Node Path
unix  1      [ ]         STREAM     CONNECTED     294    @00000019
unix  1      [ N ]       STREAM     CONNECTED     347    @00000027
unix  1      [ ]         STREAM     CONNECTED     290    @00000017
unix  1      [ ]         STREAM     CONNECTED     292    @00000018
unix  1      [ ]         STREAM     CONNECTED     267    @0000000f
unix  1      [ ]         STREAM     CONNECTED     296    @0000001a
unix  1      [ ]         STREAM     CONNECTED     1014   @00000075
unix  9      [ ]         DGRAM                    88     /dev/log
unix  1      [ ]         STREAM     CONNECTED     288    @00000016
```

netstat -i List Network Interfaces

```
Iface   MTU Met    RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR Flg
eth0   1500   0     2531      0      0      0     2240      0      0      0 BMRU
lo     3924   0       40      0      0      0       40      0      0      0 LRU
```

netstat -r Display network Routes

```
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
192.168.10.0    *               255.255.255.0   U         0 0          0 eth0
loopback        *               255.0.0.0       U         0 0          0 lo
default         apacheX.michel. 0.0.0.0         UG        0 0          0 eth0
```

netstat -s Display Network statistics

```
Ip:
    2314 total packets received
    0 forwarded
    0 incoming packets discarded
    529 incoming packets delivered
    2023 requests sent out
Icmp:
    11 ICMP messages received
    0 input ICMP message failed.
    ICMP input histogram:
        echo replies: 11
    0 ICMP messages sent
    0 ICMP messages failed
    ICMP output histogram:
Tcp:
    85 active connections openings
    0 passive connection openings
    0 failed connection attempts
    0 connection resets received
    0 connections established
    1785 segments received
    1494 segments send out
    7 segments retransmited
    0 bad segments received.
    5 resets sent
Udp:
    518 packets received
    0 packets to unknown port received.
    0 packet receive errors
    518 packets sent
TcpExt:
```

Advanced command(s):

`netstat -pan | less`	Shows Id, name, Net Sockets etc.
`watch -n1 "netstat | grep tcp"`	Shows dynamically(each second) tcp connections
`watch -n1 "netstat -n --inet"`	"" "" TCP and UDP ""
`watch -n1 "netstat -n --ip"`	"" "" TCP and UDP ""

`netstat -ct`	Shows tcp connections each seconds (cumulative)
`netstat -ltu`	Shown all the listening TCP and UDP ports

more `netstat` options

`-v`	Verbose
`-c`	Continuouly refreshed on display
`-n`	Display IP addresses, don't resolve names
`-l`	Listening ports
`-t`	TCP ports
`-u`	UDP Ports

IFCONFIG

ifconfig (**i**nter**f**ace **config**uration)

NAME

ifconfig - configure a network interface

SYNOPSIS

ifconfig [interface]
ifconfig interface [aftype] options | address ...

DESCRIPTION

Ifconfig is used to configure the kernel-resident network interfaces. It is used at boot time to set up interfaces as necessary. After that, it is usually only needed when debugging or when system tuning is needed.

If **no arguments** are given, ifconfig displays the status of the currently active interfaces. If a single interface argument is given, it displays the status of the given interface only.
If a single **-a** argument is given, it displays the status of all interfaces, even those that are down. Otherwise, it configures an interface.

Address Families

If the first argument after the interface name is recognized as the name of a supported address family, that address family is used for decoding and displaying all protocol addresses. Currently supported address families include inet (TCP/IP, default), inet6 (IPv6), ax25 (AMPR Packet Radio), ddp (Appletalk Phase 2), ipx (Novell IPX) and netrom (AMPR Packet radio).

OPTIONS

interface
The name of the interface. This is usually a driver name followed by a unit number, for example `eth0` for the first Ethernet interface.

up This flag causes the interface to be activated. It is implicitly specified if an address is assigned to the interface.

down This flag causes the driver for this interface to be shut down.

[-]arp Enable or disable the use of the ARP protocol on this interface.

[-]promisc
Enable or disable the promiscuous mode of the interface.
If selected, all packets on the network will be received by the interface.

[-]allmulti
Enable or disable all-multicast mode.
If selected, all multicast packets on the network will be received by the interface.

metric <N>
This parameter sets the interface metric.

mtu <N>
This parameter sets the Maximum Transfer Unit (MTU) (packet size)of an interface.

dstaddr <addr>
Set the remote IP address for a point-to-point link (such as PPP). This keyword is now obsolete; use the pointopoint keyword instead.

netmask <netmask>
Set the IP network mask for this interface. This value defaults to the usual class A, B or C network mask (as derived from the interface IP address),but it can be set to any value.

add <addr/prefixlen>
Add an IPv6 address to an interface.

del <addr/prefixlen>
Remove an IPv6 address from an interface.

tunnel <aa.bb.cc.dd>
Create a new SIT (IPv6-in-IPv4) device, tunnelling to the given destination.

irq <irq>
Set the interrupt line used by this device. Not all devices can dynamically change their IRQ setting.

io_addr <addr>
Set the start address in I/O space for this device.

mem_start <addr>
Set the start address for shared memory used by this device. Only a few devices need this.

media <type>
Set the physical port or medium type to be used by the device. Not all devices can change this setting, and those that can vary in what values they support. Typical values for type are 10base2 (thin Ethernet), 10baseT(twisted-pair 10Mbps Ethernet), AUI (external transceiver) and so on. The special medium type of auto can be used to tell the driver to auto-sense the media. Again, not all drivers can do this.

[-]broadcast [<addr>]
If the address argument is given, set the protocol broadcast address for this interface. Otherwise, set (or clear) the IFF_BROADCAST flag for the interface.

[-]pointopoint [<addr>]
This keyword enables the point-to-point mode of an interface, meaning that it is a

direct link between two machines with nobody else listening on it. If the address argument is also given, set the protocol address of the other side of the link, just like the obsolete dstaddr keyword does.
Otherwise, set or clear the IFF_POINTOPOINT flag for the interface.

hw <class> <address>
Set the hardware address of this interface, if the device driver supports this operation. The keyword must be followed by the name of the hardware class and the printable ASCII equivalent of the hardware address. Hardware classes currently supported include ether (Ethernet), ax25 (AMPR AX.25), ARCnet and netrom (AMPR NET/ROM).

multicast
Set the multicast flag on the interface.
This should not normally be needed as the drivers set the flag correctly themselves.

address
The IP address to be assigned to this interface.

txqueuelen <length**>**
Set the length of the transmit queue of the device. It is useful to set this to small values for slower devices with a high latency (modem links, ISDN) to prevent fast bulk transfers from disturbing interactive traffic like telnet too much.

-i Undocumented !!!

[-]dynamic Undocumented !!!

outfill <NN> Undocumented !!!

keepalive <NN> Undocumented !!!

NOTES

Since kernel release 2.2 there are no explicit interface statistics for alias interfaces anymore. The statistics printed for the original address are shared with all alias addresses on the same device. If you want per-address statistics you should add explicit accounting rules for the address using the **ipchains(8)** command.

FILES

```
/proc/net/socket
/proc/net/dev
/proc/net/if_inet6
```

BUGS

While appletalk DDP and IPX addresses will be displayed they cannot be altered by this command.

SEE ALSO

route(8), netstat(8), arp(8), rarp(8), ipchains(8)