

# Linux Networking: interfaces and routing

David Morgan

## Configuring networking

- Concepts
- Manual configuration
- Automating it at bootup

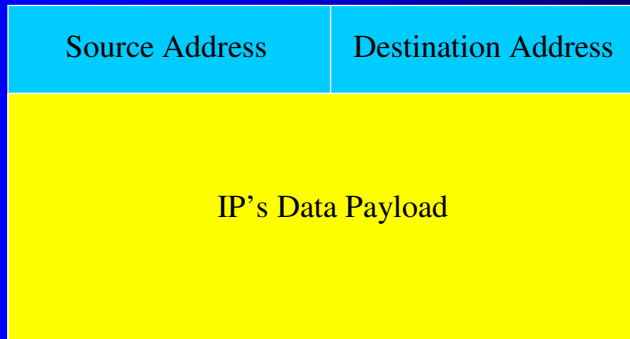
## Concepts

- Packets
- Addresses
- Interfaces
- Routes

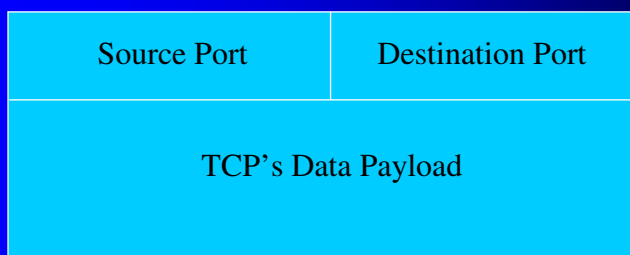
## “Packets,” also known as:

- frames (esp. for ethernet and other datalink layer)
- datagrams (esp. for UDP and other transport layer)
- segments (esp. for TCP)
- packets (esp. for IP and other network layer)
- pdu's (generally, “protocol data units”)

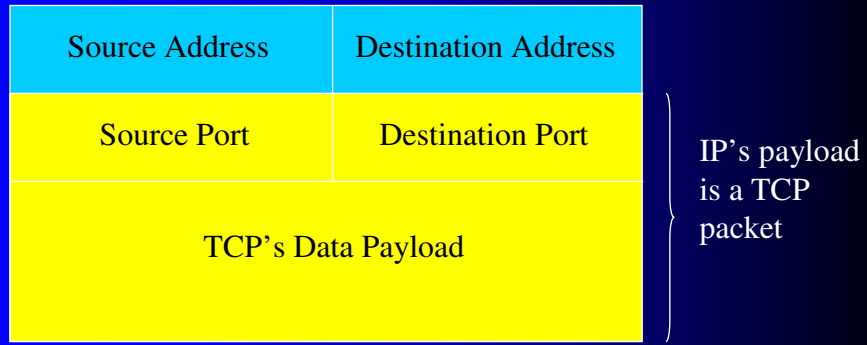
## IP packet structure



## TCP segment structure



## TCP/IP packet structure



## IP addresses

- 32 bit numbers
  - 11000000 10101000 00000100 00000001
- Expressed as “dot quads” or “dotted decimal”
  - 192.168.4.1

## IP addresses - subnet masks

- Go with addresses
- Are also 32-bit numbers
- Operationally, like shoe sizes but for networks
  - they express the *size* of a network
- e.g., Netmask 255.255.255.248 is synonym for “network size is 8 addresses”

## Common netmasks, small LANs

How netmask is written	Size it indicates
255.255.255.128 or /25	128 addresses
255.255.255.192 or /26	64
255.255.255.224 or /27	32
255.255.255.240 or /28	16
255.255.255.248 or /29	8
255.255.255.252 or /30	4

## Interfaces

- Communication outlets to the external world
  - how many doors in your house?
  - how many passenger gates in the airport terminal?
  - how many interfaces in your box?
- Interface devices
  - ethernet cards `/dev/eth0`, `/dev/eth1` ...
  - modems (point-to-point) `/dev/ppp0`, ...
  - exotic `/dev/isdn0`, `/dev/fddi0`

## Routes

- Electronic location of other computers
- By IP address
- Via interfaces

# Routing – IPdest-Iface correlation

Maintained in a “routing table”:

```
[root@EMACH1 /root]# route
Kernel IP routing table

```

Destination	Gateway	Genmask	Iface
209.233.193.22	*	255.255.255.255	ppp0
192.168.4.0	*	255.255.255.0	eth0
default	209.233.193.22	0.0.0.0	ppp0

```
[root@EMACH1 /root]#
```

# Analogy – airport departure board

Departure board

Destination	Gate
Phoenix	33A
Portland	36B
international	Terminal 4

Local, not outside of airport

Local, not outside of computer

Routing table

Destination	Interface
209.233.193.22 /32	ppp0
192.168.4.0 /24	eth0

## Commands to config networks

- Older collection of special-purpose commands
  - ifconfig (for setting up addresses)
  - route (for setting up routes)
  - others (arp, netstat...)
- Newer rewritten umbrella command “ip”
  - “ip address” alternative equivalent to ifconfig
  - “ip route” alternative to route
  - “ip neighbor” alternative to arp
- old commands implemented elsewhere, but “ip” is linux-only

## ifconfig command

- manually configuring interfaces
- View interface status
  - ifconfig -a
- Set interface characteristics
  - ifconfig eth0 192.168.4.1



## ifconfig command

```
root@hostz:~  
File Edit View Terminal Tabs Help  
[root@hostz ~]# ifconfig eth0 192.168.4.98  
[root@hostz ~]#  
[root@hostz ~]# ifconfig eth0  
eth0      Link encap:Ethernet  HWaddr 00:C0:4F:27:FF:2E  
          inet addr:192.168.4.98  Bcast:192.168.4.255  Mask:255.255.255.0  
          inet6 addr: fe80::2c0:4fff:fe27:ff2e/64 Scope:Link  
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1  
          RX packets:32 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:32 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:1000  
          RX bytes:10591 (10.3 KiB)  TX bytes:2830 (2.7 KiB)  
          Interrupt:11 Base address:0xdc00  
  
[root@hostz ~]#
```

## “ip address” command

- manually configuring interfaces

- View interface status
  - ip address show
- Set interface characteristics
  - ip address add 192.168.4.1 dev eth0

## “ip address” command

```
root@hostz:~  
File Edit View Terminal Tabs Help  
[root@hostz ~]# ip address add 192.168.4.99 dev eth0  
[root@hostz ~]#  
[root@hostz ~]# ip address show dev eth0  
2: eth0: <BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast qlen 1000  
link/ether 00:c0:4f:27:ff:2e brd ff:ff:ff:ff:ff:ff  
inet 192.168.4.99/32 scope global eth0  
inet6 fe80::2c0:4fff:fe27:ff2e/64 scope link  
valid_lft forever preferred_lft forever  
[root@hostz ~]#
```

## route command

— manually configuring routes

- host route - to a single machine
  - route add -host 192.168.4.2 eth0
- network route, local - to a group of machines
  - route add -net 192.168.4.0 netmask 255.255.255.0 eth0
- network route, thru gateway - to a group of machines
  - route add -net 192.168.5.0 netmask 255.255.255.0 gw 192.168.4.1
- default route - to “any and all” else
  - route add default gw 192.168.4.1

## or “ip route” command

— manually configuring routes

- host route - to a single machine
  - ip route add 192.168.4.2 dev eth0
- network route, local - to a group of machines
  - ip route add 192.168.4.0/24 dev eth0
- network route, thru gateway - to a group of machines
  - ip route add 192.168.5.0/24 via 192.168.4.1
- default route - to “any and all” else
  - ip route replace default via 192.168.4.1

## Great. But that’s too hard.

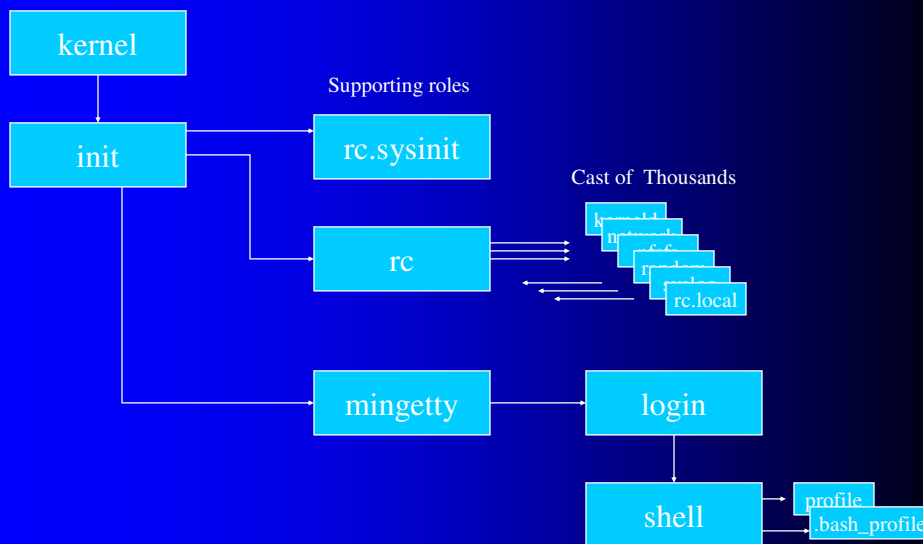
- Can’t somebody else run the commands for me?
- To the rescue: pre-written scripts do it!
- You just feed them the values to use

## General boot process control: /etc/sysconfig

- boot process runs scripts (eg, rc.sysinit)
- scripts pick up parameter values from files...
- ...to incorporate into their commands
- such files centralized in /etc/sysconfig
- edit them to feed desired values to scripts

## Boot process runs scripts

Starring roles



## Scripts pick up values from files...

Initialization scripts: /etc/rc.d/init.d/network  
/etc/sysconfig/network-scripts/ifup

informed by

/etc/sysconfig/network  
/etc/sysconfig/network-scripts/ifcfg-ethX

Initialization script: /etc/profile.d/lang.sh

informed by

/etc/sysconfig/i18n

## ...to incorporate in their commands

Initialization script: /etc/rc.d/init.d/network, calls  
/etc/sysconfig/ifup, calls  
/etc/sysconfig/ifup-eth, contains

```
ip route replace default via ${GATEWAY}...
```

```
etc/sysconfig/network  
NETWORKING=yes  
FORWARD_IPV4=no  
GATEWAY=192.168.3.2
```

becomes gateway

## Boot time automation

- Initialization script: `/etc/rc.d/init.d/network`
- `/etc/sysconfig/network`
  
- `/etc/sysconfig/network-scripts/ifup`
- `/etc/sysconfig/network-scripts/ifcfg-ethX`

## `/etc/rc.d/init.d/network`

Calls “ifup” script for each interface

```
# bring up interfaces configured to come up at boot time
for i in $interfaces; do
    action $"Bringing up interface $i: " ./ifup $i boot
Done
```

Establishes gateway

```
ip route replace default via  ${GATEWAY} ...
```

...from next slide

[ Excerpts, Fedora3's “network” initscript, line 98 ff. ]

## /etc/sysconfig/network

Sets environment variables to values the scripts use for guidance

```
NETWORKING=yes  
FORWARD_IPV4=no  
GATEWAY=192.168.3.1
```

to previous slide...

## /etc/sysconfig/network-scripts/ifup

Reads settings from ifcfg-ethX, configures interface and routes

```
if ! LC_ALL=C ip addr ls ${REALDEVICE} | LC_ALL=C grep -q  
"${IPADDR}/${PREFIX}" ; then  
    if ! ip addr add ${IPADDR}/${PREFIX}...; then  
        echo $"Error adding address ${IPADDR} for ${DEVICE}."  
    fi  
fi  
[ Fedora3's "ifup" script, line 383 ff. ]
```

### PSEUDOCODE:

```
if <the interface doesn't already have an address> ; then  
    if <trying to give it one fails>; then  
        <print error message>  
    endif  
endif
```

## /etc/sysconfig/ network-scripts/ifcfg-eth0

Sets environment variables to values the scripts use for guidance

```
BOOTPROTO=none           -or-    BOOTPROTO=dhcp
DEVICE=eth0               DEVICE=eth0
ONBOOT=yes                ONBOOT=yes
IPADDR=192.168.3.2
NETMASK=255.255.255.0
```

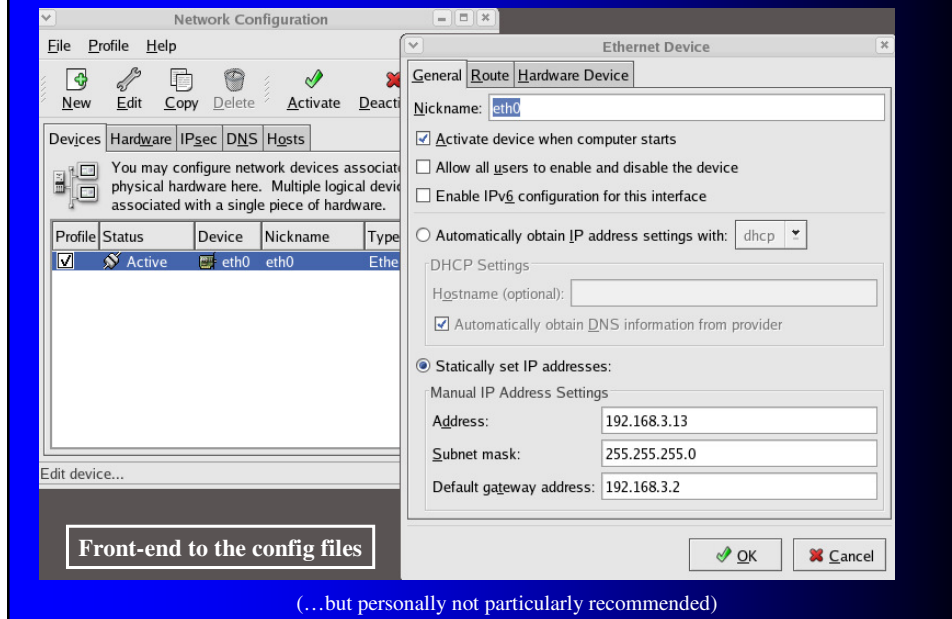
Documentation: [/usr/share/doc/initialscripts-7.93.2/sysconfig.txt](#)

## Network config control at bootup

- Edit the network/ifcfg-ethX files yourself
- Use an admin tool, which does the same thing
  - /usr/sbin/system-config-network (Fedora)
  - webmin



# Fedora's system-config-network



## It's up. What can you do with it?

- test it - ping
- watch it – tcpdump
- interfere with it - iptables
- work with others - services

## Biblio

- “IP Command Reference,” Alexey Kuznetsov (run “gv \$(locate ipcref.ps)” in your linux GUI)
- The Linux Network Administrator’s Guide, Olaf Kirsch (<http://www.tldp.org/LDP/sag/html/index.html>)
- <http://www.tcpdump.org/>