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

<table>
<thead>
<tr>
<th>Source Address</th>
<th>Destination Address</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>IP’s Data Payload</td>
</tr>
</tbody>
</table>

### TCP segment structure

<table>
<thead>
<tr>
<th>Source Port</th>
<th>Destination Port</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TCP’s Data Payload</td>
</tr>
</tbody>
</table>
### TCP/IP packet structure

<table>
<thead>
<tr>
<th>Source Address</th>
<th>Destination Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source Port</td>
<td>Destination Port</td>
</tr>
<tr>
<td>TCP’s Data Payload</td>
<td></td>
</tr>
</tbody>
</table>

IP’s payload is a TCP packet

### 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

<table>
<thead>
<tr>
<th>How netmask is written</th>
<th>Size it indicates</th>
</tr>
</thead>
<tbody>
<tr>
<td>255.255.255.128 or /25</td>
<td>128 addresses</td>
</tr>
<tr>
<td>255.255.255.192 or /26</td>
<td>64</td>
</tr>
<tr>
<td>255.255.255.224 or /27</td>
<td>32</td>
</tr>
<tr>
<td>255.255.255.240 or /28</td>
<td>16</td>
</tr>
<tr>
<td>255.255.255.248 or /29</td>
<td>8</td>
</tr>
<tr>
<td>255.255.255.252 or /30</td>
<td>4</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Destination</th>
<th>Gate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phoenix</td>
<td>33A</td>
</tr>
<tr>
<td>Portland</td>
<td>36B</td>
</tr>
<tr>
<td>international</td>
<td>Terminal 4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Destination</th>
<th>Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>209.233.193.22 /32</td>
<td>ppp0</td>
</tr>
<tr>
<td>192.168.4.0 /24</td>
<td>eth0</td>
</tr>
</tbody>
</table>
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 ~]# ifconfig eth0
eth0 Link encap:Ethernet  HWaddr 00:CO:4F:EF:7F:25
inet addr:192.168.4.98  Bcast:192.168.4.255  Mask:255.255.255.0
inet6 addr: fe80::21c8:fffe:6fa0/64 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:0xc00
```

[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 ~]# ip address add 192.168.4.0 dev eth0
[root@hostz ~]# ip address show dev eth0
2: eth0: <BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast qdisc qdisc qdisc
link/ether 00:60:0e:ff:26:0e brd ff:ff:ff:ff:ff:ff
inet 192.168.4.0/24 scope global eth0
  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
- kernel
- init

Supporting roles
- rc
- rc.sysinit
- mingetty
- login
- shell

Cast of Thousands
- profile
- bash_profile

© David Morgan 2003-2015
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}...

calls
/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

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

```bash
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

```bash
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
DEVICE=eth0
ONBOOT=yes
IPADDR=192.168.3.2
NETMASK=255.255.255.0
```

-or-

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

Documentation: /usr/share/doc/initscripts-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

© David Morgan 2003-2015
Fedora’s system-config-network

Front-end to the config files

(... but personally not particularly recommended)

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)
- http://www.tcpdump.org/