Linux Networking Commands

Dr. Vimal Baghel, Assistant Professor
SCSET, BU

Outline

- Understanding Network Devices
- Configuring NIC IP address
- Configuring Networking with Command-line Utilities
- Important Files
- Tools and Network Performance Analysis
- Commands for Connectivity, ARP, Routing, Switching, VLAN, NAT Firewall
- Q & A

Understanding Network Devices in Linux

- Linux networking devices
  - Not shown in /dev directory
  - Do not “exist” on system until appropriate device driver installed in kernel
- Networking device
  - Named channel over which network traffic can pass
- Device drivers for networking are kernel modules

Understanding Network Devices in Linux (continued)

- Kernel modules can be loaded or unloaded while Linux is running
- /dev/eth0
  - First Ethernet card installed on system
- Media Access Control (MAC) address
  - Unique address assigned by Ethernet card manufacturer

66.01:A1:4

48

Understanding Network Devices in Linux (continued)

- To obtain MAC address
  - Host (switch) broadcasts message to entire network segment using Address Resolution Protocol (ARP)
  - Host with IP address responds directly to computer that sent ARP request with MAC address
  - Source host stores MAC address and IP address

Understanding Network Devices in Linux (continued)

- arp command
    - Display ARP cache
        - Mapping of IP addresses to hardware addresses
    - Used mainly for troubleshooting network connectivity
    - Refreshed frequently

Configuration NIC IP address

- NIC: Network Interface Card
- Use “ipconfig” command to determine IP address, interface devices, and change NIC configuration
- Any device use symbol to determine
    - eth0: Ethernet device number 0
    - eth1: ethernet device number 1
    - lo : local loopback device
    - Wlan0 : Wireless lan 0

Determining NIC IP Address

```
[root@tmp]# ifconfig -a

eth0 Link encap:Ethernet HWaddr 00:08:C7:10:74:A8
BROADCAST MULTICAST MTU:1500 Metric:1
RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:100
RX bytes:0 (0.0 b) TX bytes:0 (0.0 b)
Interrupt:11 Base address:0x1820

lo Link encap:Local Loopback
inet addr:127.0.0.1 Mask:255.0.0.0
UP LOOPBACK RUNNING MTU:16436 Metric:1
RX packets:787 errors:0 dropped:0 overruns:0 frame:0
TX packets:787 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:82644 (80.7 Kb) TX bytes:82644 (80.7 Kb)
```

32

6.1.0.0

f

6

0

09

0

f

A

Changing IP Address

- We could give this eth0 interface an IP address using the **ifconfig** command.

```
[root@tmp]# ifconfig eth0 10.0.0.1 netmask 255.255.255.0 up
```

- **The "up" at the end of the command activates the interface.**

- To make this permanent at each boot up time, **add this command in /etc/rc.local file which is run at the end of every reboot.**

Permanent IP configuration

- Fedora Linux also makes life a little easier with interface configuration files located in the /etc/sysconfig/network-scripts directory.
- **Interface eth0 has a file called ifcfg-eth0, eth1 uses ifcfg-eth1, and so on.**
- Admin can place your IP address information in these files

File formats for network-scripts

```
root@network-scripts]# less ifcfg-eth0

DEVICE=eth0
IPADDR=192.168.1.100
NETMASK=255.255.255.0

BOOTPROTO=static
ONBOOT=yes
#
# The following settings are optional
#
BROADCAST=192.168.1.255
NETWORK=192.168.1.0
[root@network-scripts]#
```

Getting the IP Address Using DHCP

```
[root@tmp]# cd /etc/sysconfig/network-scripts

[root@network-scripts]# less ifcfg-eth0

DEVICE=eth0
BOOTPROTO=dhcp
ONBOOT=yes

[root@network-scripts]#
```

Activate config change

- After change, the values in the configuration files for the NIC you must deactivate and activate it for the modifications to take effect.
- The ifdown and ifup commands can be used to do this:

```
[root@network-scripts]# ifdown eth0
[root@network-scripts]# ifup eth0
```

Multiple IP Addresses on a Single NIC(1)

```
[root@tmp]# ifconfig –a

wlan0 Link encap:Ethernet HWaddr 00:06:25:09:6A:B5
inet addr:192.168.1.100 Bcast:192.168.1.255 Mask:255.255.255.0
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:47379 errors:0 dropped:0 overruns:0 frame:0
TX packets:107900 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:100
RX bytes:4676853 (4.4 Mb) TX bytes:43209032 (41.2 Mb)
Interrupt:11 Memory:c887a000-c887b000

wlan0:0 Link encap:Ethernet HWaddr 00:06:25:09:6A:B5
inet addr:192.168.1.99 Bcast:192.168.1.255 Mask:255.255.255.0
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
Interrupt:11 Memory:c887a000-c887b000
```

Multiple IP Addresses on a Single NIC(2)

- In the previous slide, there were two wireless interfaces: wlan0 and wlan0:0.
- Interface wlan0:0 is a child interface of wlan0, a virtual subinterface (an IP alias.)
- IP aliasing is one of the most common ways of creating multiple IP addresses associated with a single NIC.
- Aliases have the name format parent-interface-name:X, where X is the sub-interface number of your choice.

The process for creating an IP alias

- First ensure the parent real interface exists
- Verify that no other IP aliases with the same name exists with the name you plan to use. In this we want to create interface wlan0:0.
- Create the virtual interface with the ifconfig command

```
[root@tmp]# ifconfig wlan0:0 192.168.1.99 netmask 255.255.255.0 up
```

- Shutting down the main interface also shuts down all its aliases too. Aliases can be shutdown independently of other interfaces

The process for creating an IP alias

- Admin should also create a /etc/sysconfig/network-scripts/ifcfg-wlan0:0 file
- so that the aliases will all be managed automatically with the ifup and ifdown commands

```
DEVICE=wlan0:0
ONBOOT=yes
BOOTPROTO=static
IPADDR=192.168.1.99
NETMASK=255.255.255.0
```

- The commands to activate and deactivate the alias interface would therefore be:

```
[root@tmp]# ifup wlan0:0
[root@tmp]# ifdown wlan0:0
```

How to View Current Routing Table

- The netstat -nr command will provide the contents of the touting table.
- Networks with a gateway of 0.0.0.0 are usually directly connected to the interface.
- No gateway is needed to reach your own directly connected interface, so a gateway address of 0.0.0.0 seems appropriate.
- The route with a destination address of 0.0.0.0 is your default gateway

#natstat –nr command

```
[root@tmp]# netstat -nr

Kernel IP routing table
Destination     Gateway         Genmask         Flags MSS Window irtt Iface
172.16.68.64    172.16.69.193   255.255.255.224 UG    40  0      0    eth1
172.16.11.96    172.16.69.193   255.255.255.224 UG    40  0      0    eth1
172.16.68.32    172.16.69.193   255.255.255.224 UG    40  0      0    eth1
172.16.67.0     172.16.67.135   255.255.255.224 UG    40             0     0    eth0
172.16.69.192   0.0.0.0         255.255.255.192 U     40  0      0    eth1
172.16.67.128   0.0.0.0         255.255.255.128 U     40  0      0    eth0
172.160.0       172.16.67.135   255.255.0.0     UG    40  0      0    eth0
172.16.0.0      172.16.67.131   255.240.0.0     UG    40  0      0    eth0
127.0.0.0       0.0.0.0         255.0.0.0       U     40  0      0    lo
0.0.0.0         172.16.69.193 0.0.0.0     UG    40 0     0    eth1
[root@tmp]#
```

How to Change Default Gateway

```
[root@tmp]# route add default gw 192.168.1.1 wlan0
```

- In this case, make sure that the router/firewall with IP address 192.168.1.1 is connected to the same network as interface wlan0
- Once done, we'll need to update “/etc/sysconfig/network” file to reflect the change. This file is used to configure your default gateway each time Linux boots.

```
NETWORKING=yes
HOSTNAME=bigboy
GATEWAY=192.168.1.1
```

How to Delete a Route

```
[root@tmp]# route del -net 10.0.0.0 netmask 255.0.0.0 gw 192.168.1.254 wlan0
```

Linux router

- Router/firewall appliances that provide basic Internet connectivity for a small office or home network are becoming more affordable every day
- when budgets are tight you might want to consider modifying an existing Linux server to be a router

Configuring IP Forwarding

- For your Linux server to become a router, you have to enable packet forwarding.
- In simple terms packet forwarding enables packets to flow through the Linux server from one network to another.
- The Linux kernel configuration parameter to activate this is named net.ipv4.ip_forward and can be found in the file /etc/sysctl.conf.
- Remove the "#" from the line related to packet forwarding.

/etc/sysctl.conf changing

```
Before: # Disables packet forwarding

net.ipv4.ip_forward=0

After: # Enables packet forwarding

net.ipv4.ip_forward=1
```

- To activate the feature immediately you must force Linux to read the /etc/sysctl.conf file with the sysctl command using the -p switch

```
[root@tmp]# sysctl -p
```

Configuring /etc/hosts File

- The **/etc/hosts** file is just a list of IP addresses and their corresponding server names.
- Your server will typically check this file before referencing DNS. If the name is found with a corresponding IP address, then DNS won't be queried at all.
- Unfortunately, if the IP address for that host changes, you also must also update the file. This may not be much of a concern for a single server but can become laborious if it must be done companywide.
- Use a centralized DNS server to handle most of the rest.
- Sometimes we might not be the one managing the DNS server, and in such cases, it may be easier to add a quick **/etc/hosts** file entry till the centralized change can be made.

/etc/hosts

```
192.168.1.101  smallfry
```

- You can also add aliases to the end of the line which enable you to refer to the server using other names.
- Here we have set it up so that smallfry can also be accessed using the names tiny and littleguy.

```
192.168.1.101  smallfry  tiny  littleguy
```

/etc/hosts

- You should never have an IP address more than once in this file because Linux will use only the values in the first entry it finds.

```
192.168.1.101  smallfry    # (Wrong)
192.168.1.101  tiny        # (Wrong)
192.168.1.101  littleguy   # (Wrong)
```

Using ping to Test Network Connectivity

- The Linux ping command will send continuous pings, once a second, until stopped with a Ctrl-C.
- Here is an example of a successful ping to the server bigboy at 192.168.1.100

```
[root@smallfry tmp]# ping 192.168.1.101
PING 192.168.1.101 (192.168.1.101) from 192.168.1.100 : 56(84) bytes of data.
64 bytes from 192.168.1.101: icmp_seq=1 ttl=128 time=3.95 ms
64 bytes from 192.168.1.101: icmp_seq=2 ttl=128 time=7.07 ms
64 bytes from 192.168.1.101: icmp_seq=3 ttl=128 time=4.46 ms
64 bytes from 192.168.1.101: icmp_seq=4 ttl=128 time=4.31 ms

--- 192.168.1.101 ping statistics ---
4 packets transmitted, 4 received, 0% loss, time 3026ms
rtt min/avg/max/mdev = 3.950/4.948/7.072/1.242 ms

[root@smallfry tmp]#
```

Using ping to Test Network Connectivity

- Most servers will respond to a ping query it becomes a very handy tool.
- A lack of response could be due to:
  - A server with that IP address doesn't exist ✓
  - The server has been configured not to respond to pings ✓
  - A firewall or router along the network path is blocking ICMP traffic
  - You have incorrect routing. Check the routes and subnet masks on both the local and remote servers and all routers in between.
  - Either the source or destination device having an incorrect IP address or subnet mask.

Configuring Networking with Command-line Utilities

- ifconfig command
  - Set up network configuration in Linux kernel
  - Parameters include:
    - Network interface
    - IP address assigned to interface
    - Network mask
  - Syntax
    - *ifconfig device ip_address netmask address broadcast address*
    - $ ifconfig eth0

Configuring Networking with Command-line Utilities (continued)

- Packet: Unit of data that network card transmits
- Broadcast address sends packet to all computers on same part of network ✓
- Maximum transmission unit (MTU)
  - Maximum size of packet interface supports

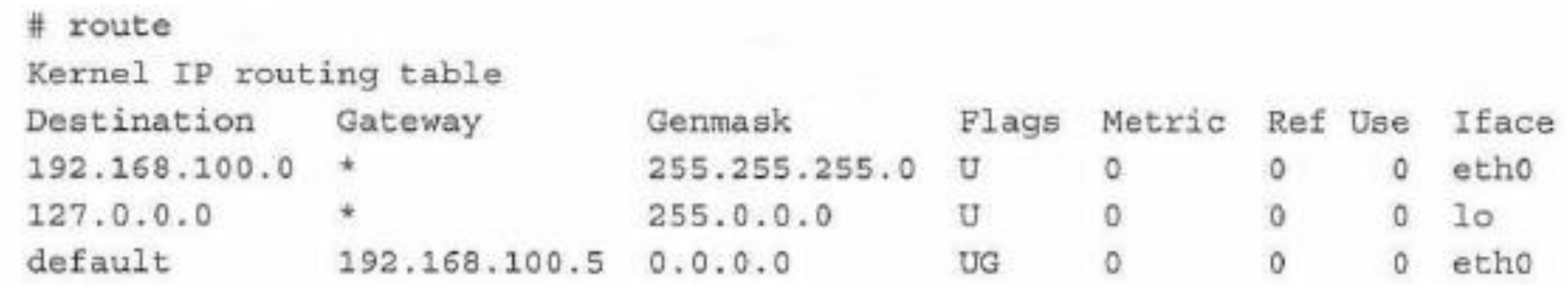
10 x 5 = 50

Configuring Networking with Command-line Utilities (continued)

- View status of interface: ifconfig eth0
- Stop Ethernet interface: ifconfig eth0 down
- Start Ethernet interface: ifconfig eth0 up
- Routing table tells networking software where to send packets that are not part of local network
- A real example of configuring an Ethernet card at the command line might look like this:
- # ifconfig eth0 192.168 . 100.1 netmask 255.255.255.0 broadcast 192. 168.100.255

Configuring Networking with Command-line Utilities (continued)

- route command
  - View or configure routing table within kernel
  - Executed at boot time when networking initialized
  - Output information for addresses
    - 192.168.100.0 (eth0 IP address)
    - 127.0.0.0
    - Other



```
# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref  Use Iface
192.168.100.0   *               255.255.255.0   U     0      0      0 eth0
127.0.0.0       *               255.0.0.0       U     0      0      0 lo
default         192.168.100.5   0.0.0.0         UG    0      0      0 eth0
```

Configuring Networking with Command-line Utilities (continued)

- Route command output
  - Destination
  - Gateway
  - Genmask
  - Flags
    - – Ref
    - – Use
    - – Iface
- Add route example:
  - route add -net 192.168.100.0 netmask 255.255.255.0 dev eth0
- This command adds a default gateway route,
- # route add default gw 192.168.100.5

Configuring Networking with Command-line Utilities (continued)

- service command
  - Start or stop networking
  - Relies on script   /etc/rc.d/init.d/network
- /etc/sysconfig/networking/devices configuration directory
  - Contains file for each network device
  - ifcfg-eth0 file
    - Used by /etc/rc.d/init.d/network script
    - As it executes ifconfig and route commands

Changing IP Address/Other Parameters

- Change the information in /etc/sysconfig/network-scripts/ifcfg-eth0
- Execute this command:
  - **# service network restart**

Configuring Networking with Command-line Utilities (continued)

- **ifup** and ifdown scripts manage single interface, rather than all network interfaces
  - Example:
    - # ./ifup eth0
    - # ./ifdown eth0
- Some systems have two or more physical network devices

Configuring Networking with Command-line Utilities (continued)

- IP forwarding
  - Allows packets to be passed between network interfaces
  - Required for any router
  - To enable:
    - `# echo 1 > /proc/sys/net/ipv4/ip_forward`

Thanks

Q & A