Quick Reference Guide to basic Linux networking commands

Connectivity

ping <host> —- sends an ICMP echo message (one packet) to a host. This may go continually until you hit Control-C. Ping means a packet was sent from your machine via and echoed at the IP level. ping tells you if the other Host is Up.

telnet host <port> —- talk to “hosts” at the given port number. By default, the telnet port is port 23. Few other famous ports are:
7 – echo port,
25 – SMTP, use to send mail
79 – Finger, provides information on other users of the network

Use control-] to get out of telnet.

ARP

Arp is used to IP addresses into Ethernet addresses. Root can add and delete arp entries. Deleting them can be useful if an entry is malformed or just wrong. Arp entries explicitly added by root are permanent — they can also be by proxy. The arp table is stored in the kernel and manipulated dynamically. Arp entries are cached and will time out and are deleted normally in 20 minutes.

arp –a : Prints the arp table
arp –s <ip_address> <mac_address> [pub] to add an entry in the table
arp –a –d to delete all the entries in the ARP table

Routing

netstat –r —- Print routing tables. The routing tables are stored in the kernel and used by to route packets to non-local networks.
route add —- The route command is used for setting a static (non-dynamic by hand route) route path in the route tables. All the traffic from this PC to that

IP/SubNet will go through the given Gateway IP. It can also be used for setting a default route; i.e., send all packets to a particular gateway, by using 0.0.0.0 in the pace of IP/SubNet.
routed —– The BSD daemon that does dynamic routing. Started at boot. This runs the RIP routing protocol. ROOT ONLY. You won't be able to run this without root access.
gated —– Gated is an alternative routing daemon to RIP. It uses the OSPF, EGP, and RIP protocols in one place. ROOT ONLY.
traceroute —- Useful for tracing the route of IP packets. The packet causes messages to be sent back from all gateways in between the source and destination by increasing the number of by 1 each time.
netstat –rnf : it displays the routing tables of IPv4
net.inet.ip.forwarding=1 : to enable packets forwarding (to turn a host into a router)
route add|delete [-net|-host] <destination> <gateway> (ex. route add 192.168.20.0/24 192.168.30.4) to add a route
route flush : it removes all the routes
route add -net 0.0.0.0 192.168.10.2 : to add a default route
routed -Pripv2 –Pno_rdisc –d [-s|-q] to execute routed daemon with RIPv2 protocol, without ICMP auto-discovery, in , in supply or in quiet mode
route add 224.0.0.0/4 127.0.0.1 : it defines the route used from RIPv2
–n : to query the RIP daemon on a specific host (manually update the routing table)

Important Files

/etc/hosts —- names to addresses
/etc/networks —- network names to addresses
/etc/protocols —– protocol names to protocol numbers
/etc/services —- / service names to port numbers

Tools and Network Performance Analysis

ifconfig <interface> <address> [up] : start the interface
ifconfig <interface> [down|delete] : stop the interface
ethereal & : it allows you open ethereal not foreground

–i <interface> - : tool to capture and analyze packets
netstat –w [seconds] –I [interface] : display network settings and statistics
–p [port] –s [bytes] target_host : it creates UDP traffic
–p [port] : it's able to receive UDP traffic
–p [port] –s [bytes] target_host : it creates TCP traffic
–p [port] it's able to receive TCP traffic
ifconfig <interface> <address> netmask <mask> [up] : it allows to subnet the sub-networks

Switching

ifconfig sl0 srcIP dstIP : configure a serial interface (do " –l /dev/ttyd0" before, and " net.inet.ip.forwarding=1" after)
telnet 192.168.0.254 : to access the switch from a host in its subnetwork
sh or **show running-configuration** : to see the current configurations
configure terminal : to enter in configuration mode
exit : in order to go to the lower configuration mode

VLAN

n : it creates a VLAN with ID n
no N : it deletes the VLAN with ID N
untagged Y : it adds the port Y to the VLAN N
ifconfig vlan0 create : it creates vlan0 interface
ifconfig vlan0 ID em0 : it associates vlan0 interface on top of em0, and set the tags to ID
ifconfig vlan0 <address> [up] : to turn on the virtual interface
tagged Y : it adds to the port Y the support of tagged frames for the current VLAN

UDP-TCP

socklab udp - it executes socklab with udp protocol
sock - it creates a udp socket, it's equivalent to type sock udp and bind
sendto <Socket ID> <hostname> <port #> - emission of data packets
recvfrom <Socket ID> <byte #> - it receives data from socket

socklab tcp - it executes socklab with tcp protocol
passive - it creates a socket in passive mode, it's equivalent to socklab, sock tcp, bind, listen
accept - it accepts an incoming connection (it can be done before or after creating the incoming connection)
connect <hostname> <port #> - these two commands are equivalent to socklab, sock tcp, bind, connect
close - it closes the connection
read <byte #> - to read bytes on the socket
write (ex. write ciao, ex. write #10) to write "ciao" or to write 10 bytes on the socket

NAT Firewall

rm /etc/resolv.conf - it prevent address resolution and make sure your filtering and firewall rules works properly
ipnat –f file_name - it writes filtering rules into file_name
ipnat –l - it gives the list of active rules
ipnat –C –F - it re-initialize the rules table
map em0 192.168.1.0/24 -> 195.221.227.57/32 em0 : mapping IP addresses to the interface
map em0 192.168.1.0/24 -> 195.221.227.57/32 portmap tcp/udp 20000:50000 : mapping with port
ipf –f file_name : it writes filtering rules into file_name
ipf –F –a : it resets the rule table
ipfstat –I : it grants access to a few information on filtered packets, as well as active filtering rules

---

This quick reference guide has been created by It's FOSS, your ultimate source of Open Source and Linux learning.