**Cisco IOS Commands:**

1. When you start the console, will see the prompt of the User EXEC mode (Router>). To see which commands are available in this mode, type “?”:

**Router1>** ?

1. To view and change system parameters of a Cisco router, you must enter the privileged EXEC mode, by typing:

**Router1>** enable
**Router1#**

1. To modify system wide configuration parameters, you must enter the global configuration mode. This mode is entered by typing:

**Router1#** configure terminal

**Router1(config)#**

1. To make changes to a network interface, enter the interface configuration mode, with the command:

**Router1(config)#** interface FastEthernet0/0
**Router1(config-if)#**

The name of the interface is provided as an argument. Here, the network interface that is configured is FastEthernet0/0.

1. To return from the interface configuration to the global configuration mode, or from the global configuration mode to the privileged EXEC mode, use the “exit” command:

**Router1(config-if)#** exit
**Router1(config)#** exit
**Router1#**

Or, to directly return to the privileged EXEC mode from any configuration mode, use the “end” command:

**Router1(config-if)#** end
**Router1#**

1. To return from the privileged EXEC mode to the user EXEC mode, type:

**Router1#** disable
**Router1>**

1. To terminate the console session from the user EXEC mode, type:

**Router1>** logout

Router1 con0 is now available
Press RETURN to get started.

Or type “logout” or “exit” from the privileged EXEC mode:

**Router1#** exit

Router1 con0 is now available
Press RETURN to get started.

In **Previleged Exec Mode** you can check your router configuration using the following commands:

 interfaces

show running-config

**IOS MODE: PREVILEGED EXEC**

**show ip route**

**clear ip route \***

Displays the contents of the routing table and deletes all routing table entries

**show ip cache**

**clear ip cache**

Displays the routing cache and clears the routing cache

**show arp**

**clear arp-cache**

Displays the arp cache and clears the arp cache

**IOS MODE: GLOBAL CONFIGURATION**

**ip route-cache**

Enables route caching. By default, route caching is enabled on a router.

**no ip route-cache**

Disables route caching

show ip ospf

Displays general information about the OSPF configuration

show ip ospf database

Displays the link state database.

show ip ospf border-routers

Displays the Area Border Router (ABR) and Autonomous System Boundary Router (ASBR).

clear ip ospf process-id process

Resets the specified OSPF process.

Adding a static host or network route via a next hop gateway:

**ip route** *IPAddress* *netmask* *gw\_address*

**ip route** *NetAddress* *netmask* *gw\_address*

Adding a static host or network route via a connected interface:

**ip route** *IPAddress* *netmask* *Iface*

**ip route** *NetAddress* *netmask* *Iface*

Adding a default route:

**ip route** 0.0.0.0 0.0.0.0 *gw\_address*

Deleting static route entries:

Command same as above but use "**no ip route**" instead of "**ip route**"

**IOS MODE: INTERFACE Configuration**

Configuring an IP address:

**Router1>** enable
**Router1#** configure terminal

**Router1(config)#** no ip routing
**Router1(config)#** ip routing
**Router1(config)#** interface FastEthernet0/0
**Router1(config-if)#** ip address xxx.yyy.zzz.aaa netmask
**Router1(config-if)#** no shutdown
**Router1(config-if)#** end

Enabling and disabling proxy arp (enabled by default on every interface):

**ip proxy-arp**

**no ip proxy-arp**

**Linux Commands**

ARP Cache Related:

arp -a

Display the content of the ARP cache.

arp –d **IPAddress**

Deletes the entry with the IP address **IPAddress**.

arp –s **IPAddress MACAddress**

Adds a static entry to the ARP cache that is never overwritten by network events. The MAC address is entered as 6 hexadecimal bytes separated by colons.

**Example:** arp –s 10.0.1.12 00:02:2D:0D:68:C1

ip –s –s neigh flush all

command to clear the arp cache

Checking on the network status of the PC:

netstat –i

Displays a table with statistics of the currently configured network interfaces.

netstat –rn

Displays the kernel routing table. The *–n* option forces netstat to print the IP addresses. Without this option, netstat attempts to display the host names.

netstat –an
netstat –tan

netstat -uan

Displays the active network connections. The *–a* option display all active network connections, the *–ta* option displays only information on TCP connections, and the *–tu* option displays only information on UDP traffic. Omitting the *–n* option prints host names, instead of IP addresses.

netstat –s

Displays summary statistics for each protocol that is currently running on the host.

Configuring a PC interface:

ifconfig

Displays the configuration parameters of all active interfaces.

ifconfig **interface**

Displays the configuration parameters of a single interface. For example, ifconfig **eth0** displays information on interface **eth0**.

ifconfig **interface** down

Disables the interface. For example: ifconfig **eth0** down. No traffic is sent or received on a disabled interface.

ifconfig **interface** up

Enables an interface.

ifconfig **interface** **IPAddr/xx**

**e.g.** ifconfig **eth0 10.0.1.8/24**

Assigns interface *eth0* the IP address *10.0.1.8/24* and a broadcast address of *10.0.1.255*

ifconfig **interface** mtu **xxx**

Assigns MTU size **xxx** bytes to **interface**

Enabling and Disabling IP Forwarding on a Linux PC:

sysctl net.ipv4.ip\_forward

 shows current status of ipforwarding

echo 1 > ‘/proc/sys/net/ipv4/ip\_forward’

 enables IPforwarding

echo 0 > ‘/proc/sys/net/ipv4/ip\_forward’

 disables IPforwarding

Additional commands that do the same as the above commands:

sysctl -w net.ipv4.ip\_forward=1

sysctl -w net.ipv4.ip\_forward=0

Setting Path Discovery (MTU probing) on a Linux PC:

PC% echo 2 > `/proc/sys/net/ipv4/tcp\_mtu\_probing'

Sets/enables probing on PC

PC% cat /proc/sys/net/ipv4/tcp\_mtu\_probing'

Checks the current setting for probing. If "0" or "1" then it is not set for default probing. Has to "=2" to enable probing.

Routing commands on a PC:

route add –net **netaddress** netmask **mask** gw **gw\_address**

route add –net **netaddress** netmask **mask** dev **iface**

Adds a routing table entry for the network prefix identified by IP address **netaddress** and netmask **mask**. The next-hop is identified by IP address **gw\_address** or by interface **iface**.

route add –host **hostaddress** gw **gw\_address**

route add –host **hostaddress** dev **iface**

Adds a host route entry for IP address **hostaddress** with the next-hop identified by IP address **gw\_address** or by interface **iface**

route add default gw **gw\_address**

Sets the default route to gateway with IP address **gw\_address**

route del –net **netaddress** netmask **mask** gw **gw\_address**

route del –host **hostaddress** gw **gw\_address**

route del default gw **gw\_address**

Deletes an existing route from the routing table with specific arguments.

route -e

Displays the current routing table with extended fields. The command is identical to the netstat –r command.

ip route flush table main

Deletes all entries in the routing table on a PC. Please note that the local interface route(s) need to be added before adding any other static route entries after a flush table command. To add interface route(s) use the ifconfig **interface** down and ifconfig **interface** up (where e.g. interface = eth0)

ip route flush cache

deletes all entries in the routing cache

ip route get **IPAddress**

ip route flush cache **IPAddress**

Displays the cached route for IPAddress and flushes the cached route entry for IPAddress

Commonly used Linux commands:

ping **IPAddr**

Pings host with IP address **IPAddr.**

ping **IPAddr –c num**

Where **num** is the number of pings you want issued to destination **IPAddr**.

ping **IPAddr –s num**

Where **num** is the number of data bytes in the ICMP request message you want sent to destination **IPAddr**.

traceroute **IPAddr** –m**xxx** -q**yyy**

Command used to trace the route between an origin and a destination IP address **IPAddr**, where –m indicates the max TTL value and –q indicates the number of queries. E.g. m=2, and q=1.