D Y PATIL INSTITUTE OF ENGINEERING & TECHNOLOGY AMBI,PUNE	**D Y PATIL INSTITUTE OF ENGINEERING & TECHNOLOGY, AMBI**	LABORATORY MANUAL
	EXPERIMENT TITLE: Explore and Study of TCP/IP utilities and Network Commands on Linux.	
DEPARTMENT OF INFORMATION TECHNOLOGY		
EXPERIMENT NO. : DYPIET/IT/TE/SL-IV	**SEMESTER** : VI(TE)	**PAGE:1-**

AIM: Explore and Study of TCP/IP utilities and Network Commands on Linux.

OBJECTIVES: To study,

TCP/IP utilities

Network commands

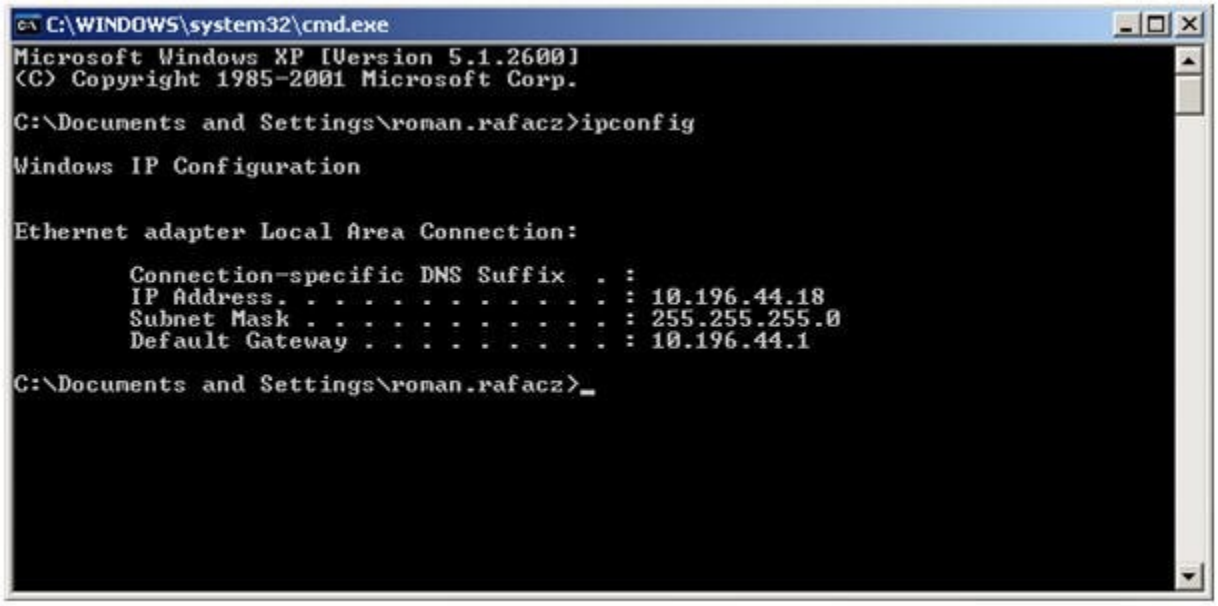
THEORY:

Ifconfig : ifconfig utility is used to configure network interface parameters. Mostly we use this command to check the IP address assigned to the system.

```
[root@localhost ~]# ifconfig -a
eno16777736: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      ether 00:0c:29:c5:a5:61 txqueuelen 1000 (Ethernet)
      RX packets 0 bytes 0 (0.0 B)
      RX errors 0 dropped 0 overruns 0 frame 0
      TX packets 0 bytes 0 (0.0 B)
      TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
      inet 127.0.0.1 netmask 255.0.0.0
      inet6 ::1 prefixlen 128 scopeid 0x10<host>
      loop txqueuelen 0 (Local Loopback)
      RX packets 2 bytes 140 (140.0 B)
      RX errors 0 dropped 0 overruns 0 frame 0
      TX packets 2 bytes 140 (140.0 B)
      TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
[root@localhost ~]#
```

IPConfig : Not part of the TCP/IP utilities but it is useful to show current TCP/IP settings.

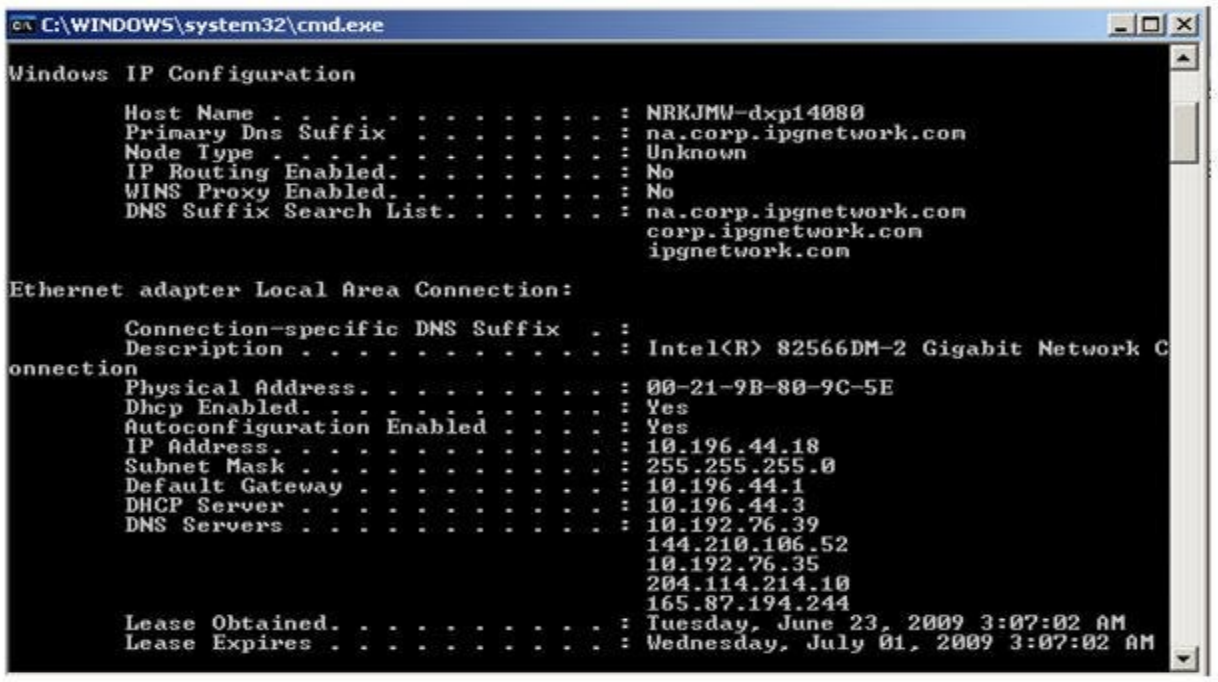
The IPConfig command line utility will show detailed information about the network you are connected to. It also helps with reconfiguration of your IP address through release and renew.



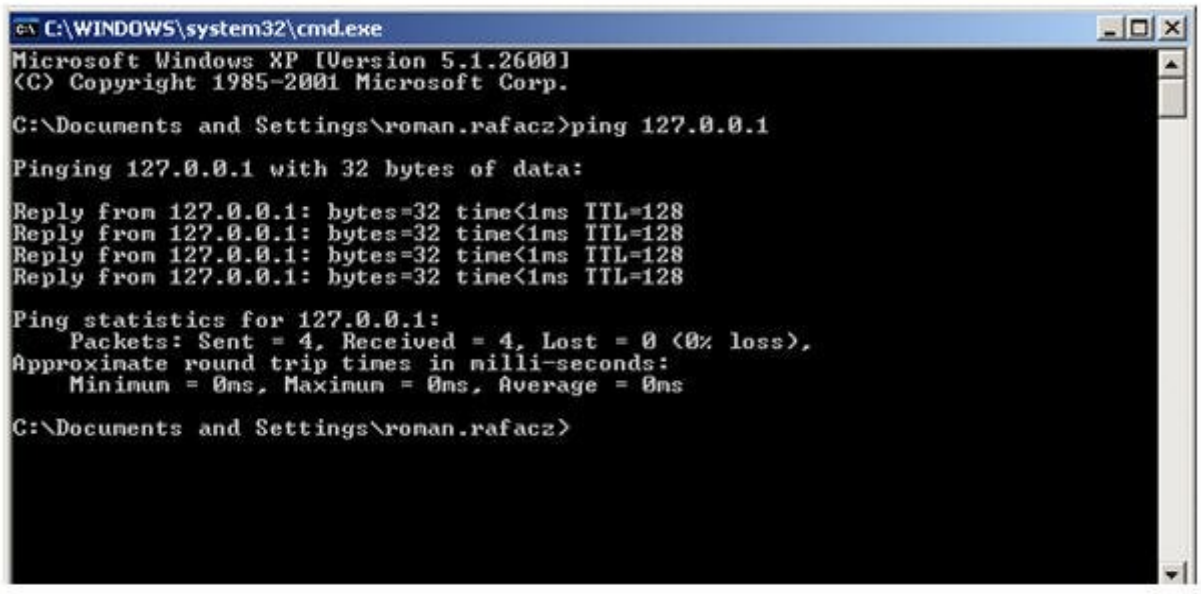
ipconfig will give a quick view of you IP address, your subnet mask and default gateway.

ipconfig /all will give you more detailed information. Through **ipconfig /all** we can find DNS severs, if we have DHCP enabled, MAC Address, along with other helpful information. All good things to know if we have trouble getting connected to the internet.

Other IPConfig tools that are helpful include **ipconfig /release** and **ipconfig /renew**. But before I get into this let's discuss how we actually get an IP Address.There are two ways to obtain an IP address. One way is to have a static IP address which we manually assign. The second one is to have a dynamic IP address obtained through a DHCP server.

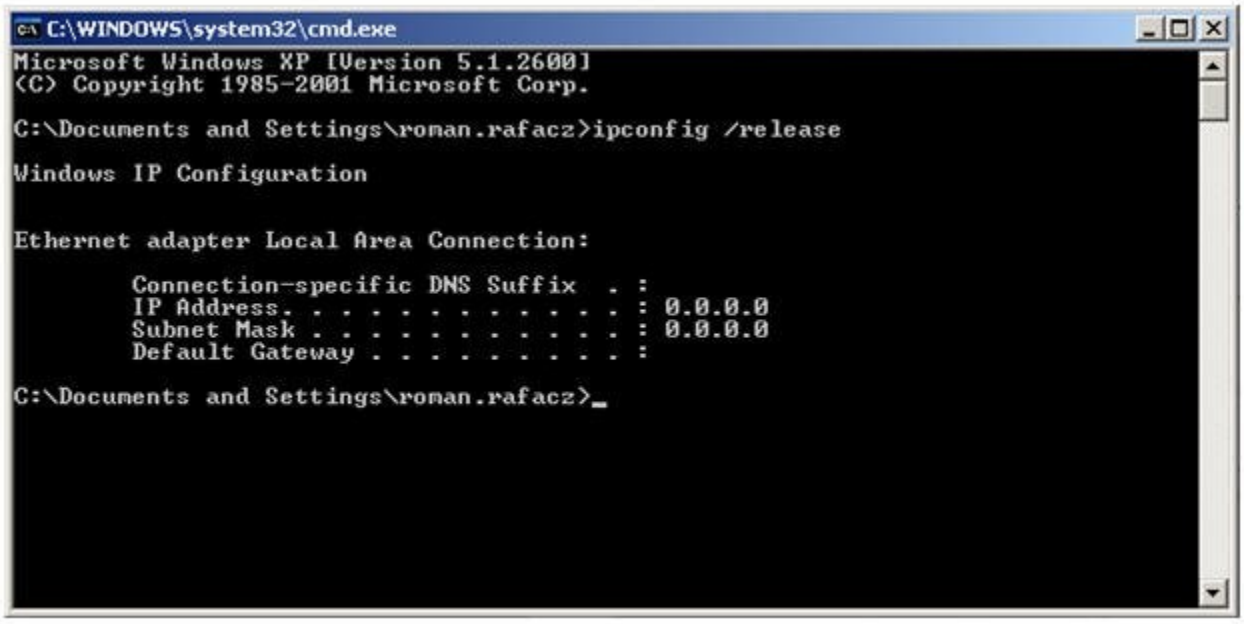


If you were to right click on Network Connects, go to Properties, right click on Local Area Connection, scroll down to Internet Protocol (TCP/IP), and select Properties -- you'll see two options: Obtain an IP address automatically and Use the following IP address

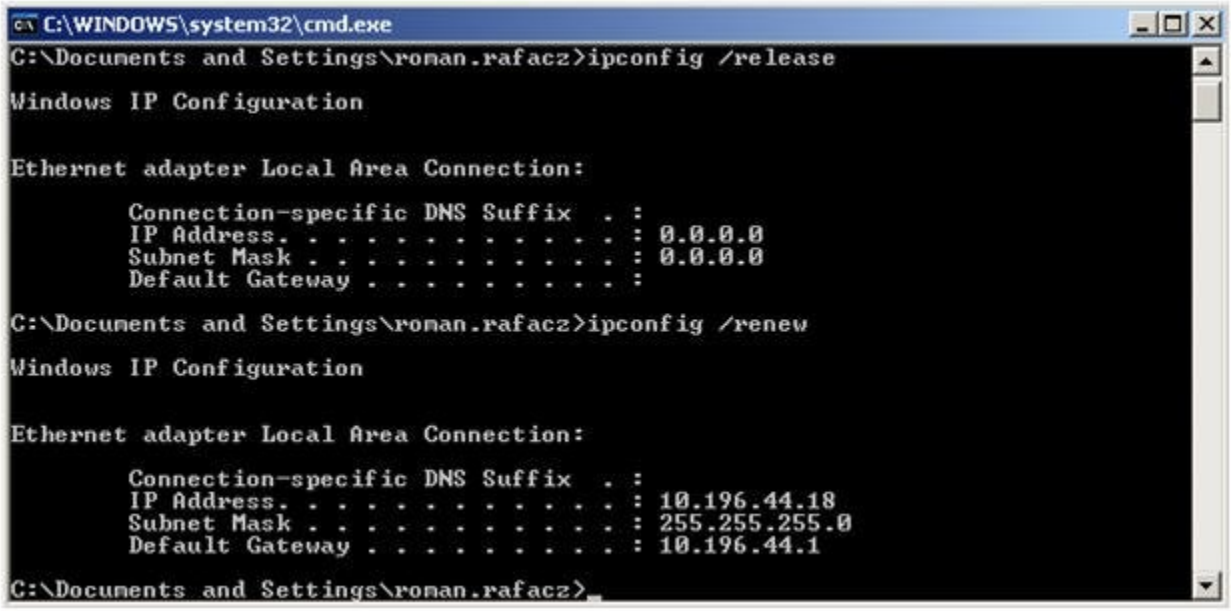


Unless you know your static IP address you'll want to stick to the option for automatically obtaining the IP address. If you have it set to automatic your computer will be issued an IP through a DHCP server. Dynamic Host Configuration Protocol (DHCP) is a network protocol that enables a server to automatically assign an IP address to a computer from a defined range of numbers (i.e., a scope) configured for a given network.

Let's look at what happens when we release our IP address.



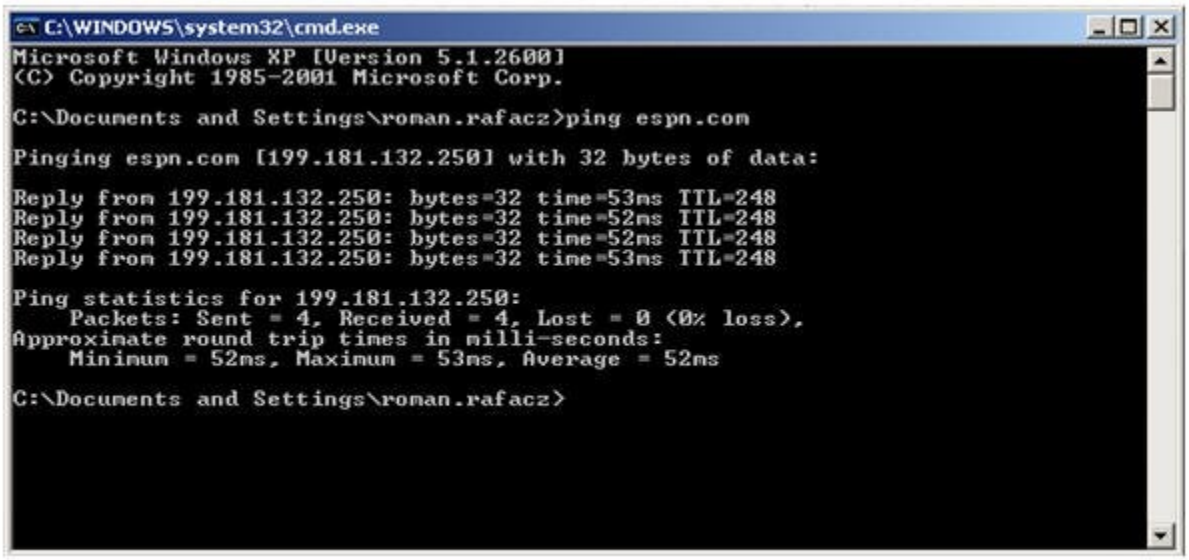
I have just lost internet connection and my IP address is 0.0.0.0. If I type ipconfig /renew this option re-establishes TCP/IP connections on all network adapters and I can resume my internet surfing.



Note: ipconfig /release renew won't work if you manually assigned your IP addresses.

Ping : The PING utility tests connectivity between two hosts. PING uses a special protocol called the **Internet Control Message Protocol (ICMP)** to determine whether the remote machine (website, server, etc.) can receive the test packet and reply. Also a great way to verify whether you have TCP/IP installed and your Network Card is working. We'll start by Pinging the loopback address (127.0.0.1) to verify that TCP/IP is installed and configured correctly on the local computer. **Type: PING 127.0.0.1**

This tells me that TCP/IP is working as well as my Network Card. To test out connectivity to a website all you have to do is type: **ping espn.com**



The results should tell you if the connection was successful or if you had any lost packets.

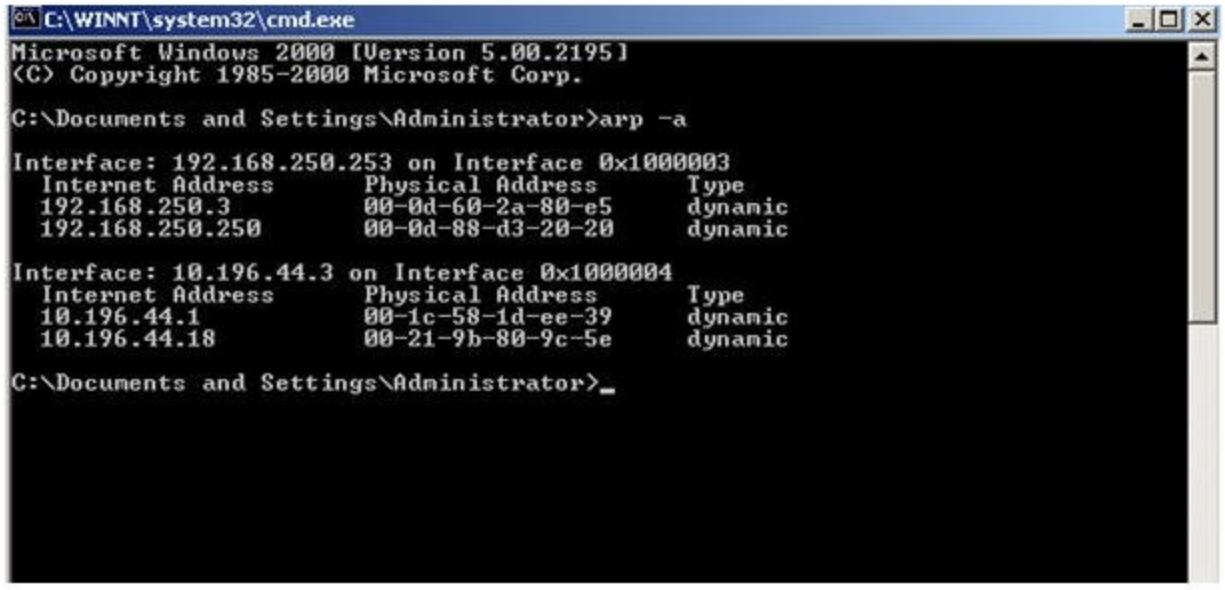
Packet loss describes a condition in which data packets appear to be transmitted correctly at one end of a connection, but never arrive at the other. The network connection might be poor and packets get damaged in transit or the packet was dropped at a router because of internet congestion. Some Internet Web servers may be configured to disregard ping requests for security purposes.

Note the IP address of espn.com -- 199.181.132.250. You can also ping this address and get the same result. However, Ping is not just used to test websites. It can also test connectivity to various servers: DNS, DHCP, your Print server, etc.

Traceroute : traceroute print the route packets take to network host. Destination host or IP is mandatory parameter to use this utility

```
[root@localhost ~]# traceroute geekflare.com
traceroute to geekflare.com (162.159.243.243), 30 hops max, 60 byte packets
1 172.16.179.2 (172.16.179.2) 0.154 ms 0.074 ms 0.074 ms
2 * * *
3 * * *
```

ARP : The ARP utility helps diagnose problems associated with the Address Resolution Protocol (ARP). TCP/IP hosts use ARP to determine the physical (MAC) address that corresponds with a specific IP address. Type **arp** with the **– a** option to display IP addresses that have been resolved to MAC addresses recently.



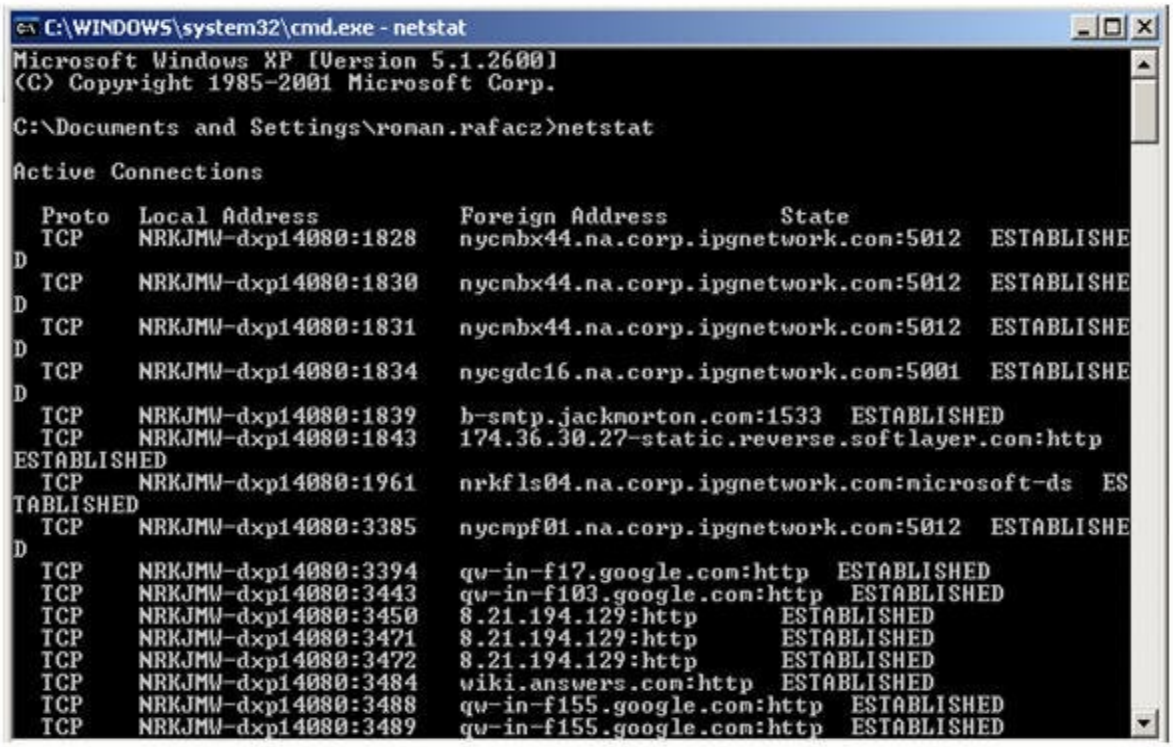
whois : You can use the *whois* command in Linux to find out information about a domain, such as the owner of the domain, the owner's contact information, and the nameservers that the domain is using. For example, to find out domain information of **linux-bible.com**, we can use the following command:

```
susel:~ # whois -H linux-bible.com

Whois Server Version 2.0

Domain names in the .com and .net domains can now be registered
with many different competing registrars. Go to http://www.internic.net
for detailed information.

   Domain Name: LINUX-BIBLE.COM
   Registrar: LAUNCHPAD.COM, INC.
   Whois Server: whois.launchpad.com
   Referral URL: http://www.launchpad.com
   Name Server: NS6175.HOSTGATOR.COM
   Name Server: NS6176.HOSTGATOR.COM
   Status: clientTransferProhibited
   Updated Date: 16-may-2014
   Creation Date: 16-may-2014
   Expiration Date: 16-may-2015
   Registrant Name: Antun Peicevic
   Registrant Organization: 1
   Registrant Street: Nova cesta 1
   Registrant City: Zagreb
   Registrant State/Province: Zagreb
   Registrant Postal Code: 10000
   Registrant Country: HR
   Registrant Phone: +385.921021346
   Registrant Phone Ext:
   Registrant Fax:
   Registrant Fax Ext:
   Registrant Email: antunpeicevic@gmail.com
```

Here is a brief description of the most important fields:

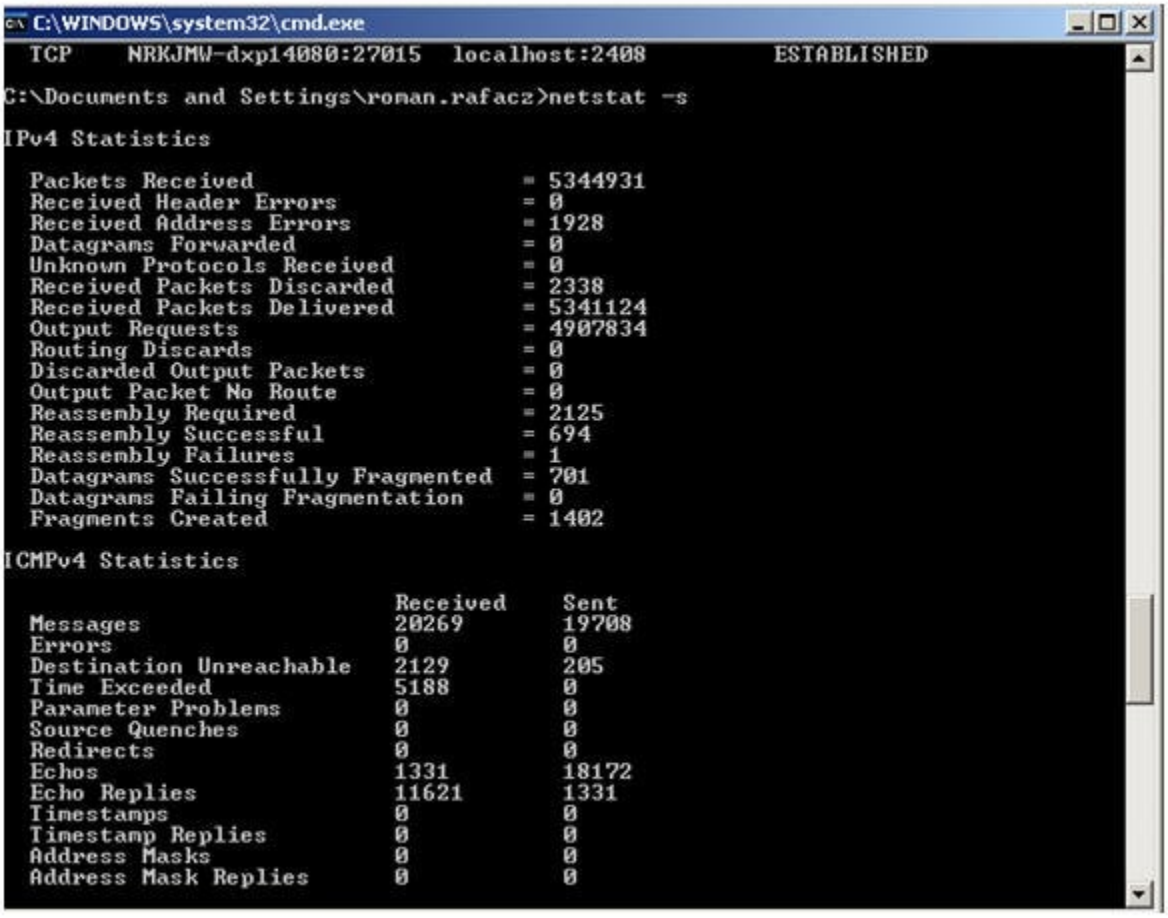
- **Registrar:** LAUNCHPAD.COM, INC. – the company that registered the domain on behalf of the domain's owner.
- **Name Servers:** NS6175.HOSTGATOR.COM, NS6176.HOSTGATOR.COM – the servers that control the domain's DNS.
- **Creation Date:** 16 May 2014 – the date the domain was originally registered.
- **Expiration Date:** 16 May 2015 – the date when the domain will expire.
- **Registrant Name, Address, City… :** publicly accessible information of the domain owner.

Netstat : Netstat (Network Statistics) displays network connections (both incoming and outgoing), routing tables, and a number of network interface statistics.

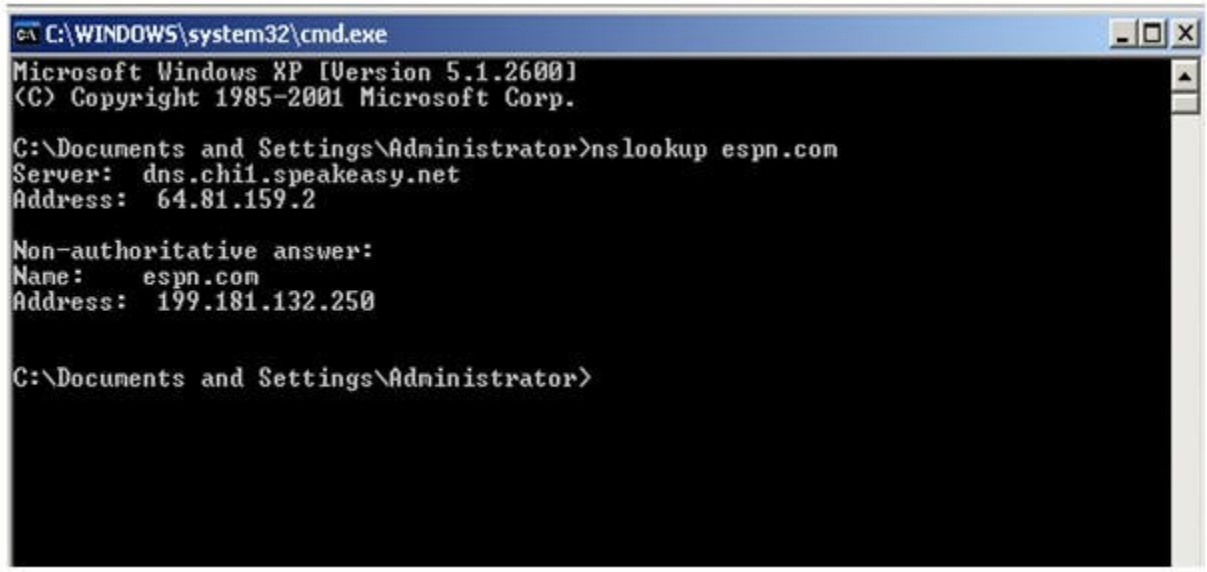


N

etstat – provides statistics about incoming and outgoing traffic.



NSLookup : NSLookup provides a command-line utility for diagnosing DNS problems. In its most basic usage, NSLookup returns the IP address with the matching host name.



A hostname command is used to view a computer's hostname and domain name (DNS) (Domain Name Service), and to display or set a computer's hostname or domain name.

A hostname is a name that is given to a computer that attached to the network that uniquely identifies over a network and thus allows it to be accessed without using its IP address.

The basic syntax for the hostname command is:

```
# hostname [options] [new_host_name]
```

In this short article, we will explain 5 useful hostname command examples for Linux beginners to view, set or change Linux system hostname from the Linux command-line interface. If you run hostname command without any options, it will displays the current host name and domain name of your Linux system.

```
$ hostname
```

```
tecmint
```

```
[root@tecmint ~]# hostname
tecmint.com
[root@tecmint ~]#
```

Show Linux Hostname

```
$ hostname -i

$ hostname -I
```

```
[root@tecmint ~]# hostname -i
192.168.0.1
[root@tecmint ~]# hostname -I
192.168.0.1 3a03:7b00::f13c:91ff:fedb:134560
[root@tecmint ~]#
```

Show Hostname IP Addresses

To view the name of the DNS domain and FQDN (Fully Qualified Domain NAme) of your machine, use the –f and –d switches respectively. The –A enables you to see all the FQDNs of the machine.

```
$ hostname -d

$ hostname -f

$ hostname -A
```

```
[root@tecmint ~]# hostname -d
com
[root@tecmint ~]# hostname -f
tecmint.com
[root@tecmint ~]# hostname -A
mail.tecmint.com
[root@tecmint ~]#
```

Show Host DNS Names

To display the alias name (i.e.substitute names), if used for the host name, use the –a flag.

```
$ hostname -a
```

To change or set hostname of your Linux system, simply run following command, remember to replace “NEW HOSTNAME” with the actual hostname that you wish to set or change.

```
$ sudo hostname NEW_HOSTNAME
```

```
tecmint@TecMint ~ $ sudo hostname TecMint.com
[sudo] password for tecmint:
tecmint@TecMint ~ $ hostname
TecMint.com
tecmint@TecMint ~ $
```

Set Linux System Hostname

Note that the changes made using the above command will only last until the next reboot. Under systemd – system and service manager, you can use the hostnamectl command to permanently set or change your system

Finger : finger **looks up and displays information about system users.**

finger syntax

```
finger [-lmsp] [user ...] [user@host ...]
```

dig : dig (Domain Information Groper) is a flexible tool for interrogating DNS name servers. It performs **DNS lookups** and displays the answers that are returned from the name servers.

```
[root@localhost ~]# dig geekflare.com
; <<>> DiG 9.9.4-RedHat-9.9.4-14.el7 <<>> geekflare.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 18699
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1
;; OPT PSEUDOSECTION:
```

```
; EDNS: version: 0, flags:; MBZ: 0005 , udp: 4000
;; QUESTION SECTION:
;geekflare.com.                 IN      A
;; ANSWER SECTION:
geekflare.com.          5       IN      A       162.159.244.243
geekflare.com.          5       IN      A       162.159.243.243
;; Query time: 6 msec
;; SERVER: 172.16.179.2#53(172.16.179.2)
;; WHEN: Sun May 01 23:28:19 PDT 2016
;; MSG SIZE rcvd: 74
[root@localhost ~]#
```

Telnet : telnet connect destination host:port via a telnet protocol if connection establishes means connectivity between two hosts is working fine.

```
[root@localhost ~]# telnet geekflare.com 443
Trying 162.159.244.243...
Connected to geekflare.com.
Escape character is '^]'.
```

nslookup

nslookup is a program to query Internet domain name servers.

```
[root@localhost ~]# nslookup geekflare.com
Server:                 172.16.179.2
Address:        172.16.179.2#53
Non-authoritative answer:
Name: geekflare.com
Address: 162.159.243.243
Name: geekflare.com
Address: 162.159.244.243
[root@localhost ~]#
```

Netstat : Netstat command allows you a simple way to review each of your network connections and open sockets. Netstat with head output is very helpful while performing web server troubleshooting.

```
[root@localhost ~]# netstat
```

```
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 172.16.179.135:58856    mirror.comp.nus.ed:http TIME_WAIT
tcp        0      0 172.16.179.135:34444    riksun.riken.go.jp:http ESTABLISHED
tcp        0      0 172.16.179.135:37948    mirrors.isu.net.sa:http TIME_WAIT
tcp        0      0 172.16.179.135:53128    ossm.utm.my:http        TIME_WAIT
tcp        0      0 172.16.179.135:59723    103.237.168.15:http     TIME_WAIT
tcp        0      0 172.16.179.135:60244    no-ptr.as20860.net:http TIME_WAIT
```

scp : scp allows you to secure copy files to and from another host in the network.

Ex:

```
scp $filename user@targethost:/$path
```

w : w prints a summary of the current activity on the system, including what each user is doing, and their processes.

Also list the logged in users and system load average for the past 1, 5, and 15 minutes.

```
[root@localhost ~]# w
23:32:48 up 2:52, 2 users, load average: 0.51, 0.36, 0.19
USER     TTY      LOGIN@   IDLE   JCPU   PCPU WHAT
chandan :0       20:41   ?xdm?   7:07 0.13s gdm-session-worker [pam/gdm-password]
chandan pts/0     20:42    0.00s 0.23s 3.42s /usr/libexec/gnome-terminal-server
[root@localhost ~]#
```

Nmap : nmap is a one of the powerful commands, which checks the **opened port** on the server.

Usage example:

```
nmap $server_name
```

Enable/Disable Network Interface

You can enable or disable the network interface by using ifup/ifdown commands with ethernet interface parameter.

To enable eth0

```
#ifup eth0
```

To disable eth0

```
#ifdown eth0
```

I hope above Linux Commands help you to gather network information or troubleshoot the networking issue.

- **Conclusion**

Successfully Studied and implemented TCP/IP utilities and Network Commands on Linux.

D Y PATIL INSTITUTE OF ENGINEERING & TECHNOLOGY AMBI, PUNE	netting and Super-netting of a given networks.	
DEPARTMENT OF INFORMATION TECHNOLOGY		
EXPERIMENT NO. : DYPIET/IT/TE/SL-IV	**SEMESTER** : VI (TE)	**PAGE:1-**

AIM: Using a Network Simulator Configure Sub-netting and Super-netting of a given networks.

OBJECTIVES:

To study,

Network Simulator tool

Configure sub-netting and Super-netting network.

THEORY:

Subnetting is the process of dividing an IP network in to sub divisions called subnets. Computers belonging to a sub network have a common group of most-significant bits in their IP addresses. So, this would break the IP address in to two parts (logically), as the network prefix and the rest field. Supernetting is the process of combining several sub networks, which have a common Classless Inter-Domain Routing (CIDR) routing prefix. Suppernetting is also called route aggregation or route summarization.

Supernetting is the process of combining several sub **networks**, which have a common Classless Inter-Domain Routing (**CIDR**) routing prefix. ... **Subnetting is** a process of dividing **network for** ease of monitoring whereas, **Supernetting is** a process of aggregating multiple **networks** into the single **network**.

What is Subnetting?

Process of dividing an IP network in to sub divisions is called subnetting. Subnetting divides an IP address in to two parts as the network (or routing prefix) and the rest field (which is used to identify a specific host). CIDR notation is used to write a routing prefix. This notation uses a slash (/) to separate the network starting address and the length of the network prefix (in bits).

For example, in IPv4, 192.60.128.0/22 indicates that 22 bits are allocated for the network prefix and the remaining 10 bits are reserved for the host address. In addition, routing prefix can also be represented using the subnet mask. 255.255.252.0 (11111111.11111111.11111100.00000000) is the subnet mask for 192.60.128.0/22. Separating the network portion and the subnet portion of an IP address is done by performing a bitwise AND operation between the IP address and the subnet mask. This would result in identifying the network prefix and the host identifier.

What is Supernetting?

Supernetting is the process of combining several IP networks with a common network prefix. Supernetting was introduced as a solution to the problem of increasing size in routing tables. Supernetting also simplifies the routing process. For example, the subnetworks 192.60.2.0/24 and 192.60.3.0/24 can be combined in to the supernetwork denoted by 192.60.2.0/23. In the supernet, the first 23 bits are the network part of the address and the other 9 bits are used as the host identifier. So, one address will represent several small networks and this would reduce the number of entries that should be included in the routing table. Typically, supernetting is used for class C IP addresses (addresses beginning with 192 to 223 in decimal), and most of the routing protocols support supernetting. Examples of such protocols are Border Gateway Protocol (BGP) and Open Shortest Path First (OSPF). But, protocols such as Exterior Gateway Protocol (EGP) and the Routing Information Protocol (RIP) do not support supernetting.

- **Conclusion :** Hence in this we have studied that Network Simulator Configure Sub-netting and Super-netting of a given networks.

AIM: Using a Network Simulator Configure A router using router commands and Access Control lists – Standard & Extended.

OBJECTIVES: To study,

Network Simulator tool

A router using router commands

Access Control lists – Standard & Extended.

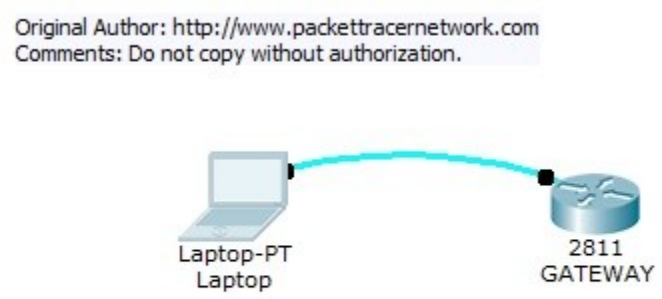
THEORY: A large number of commands are available on Cisco **routers**, as well as many different protocols and features that can be used to establish a **network**.

Basic router setup

When you first boot up your Cisco ISR router, some basic configuration has to be performed to secure adminitrative access to the router.

1. Configure correctly the LAPTOP terminal software and connect to the router console.
2. Configure the router hostname to "GATEWAY"
3. Configure the enable password and secret to "cisco"
4. Configure password encryption for this router
5. Configure the console access :
   - Login : yes
   - Password : "cisco"
   - History : 10 commands
   - Logging synchronous
   - Timeout : 2 minutes 45 seconds.

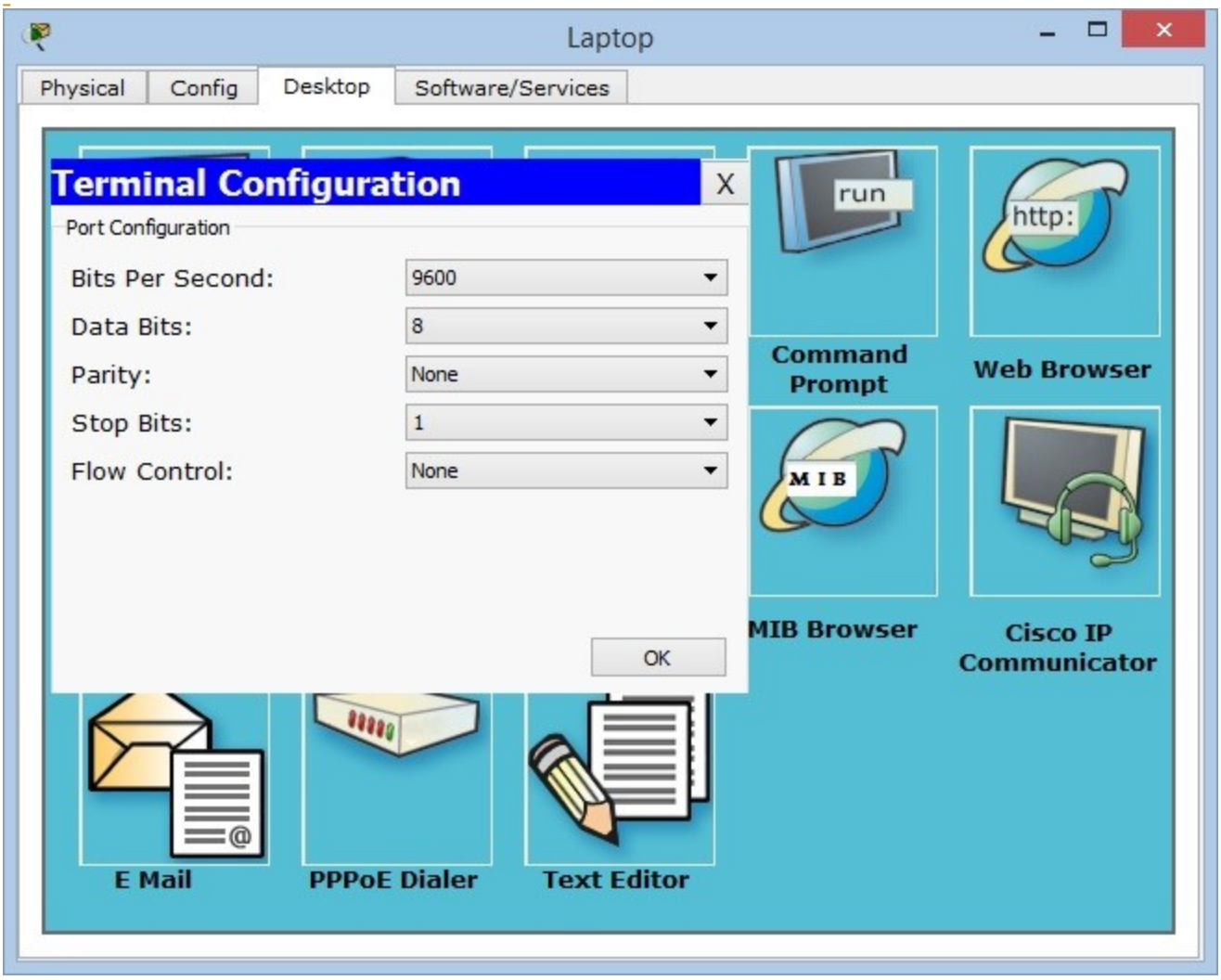
Network diagram



Solution

1. Configure the laptop terminal software

The terminal software in not correctly configured on the laptop. You have to change the settings to 9600 / 8 / None / 1 to connect to the router's console. Remerber this tip as it could help you answer CCENT questions or achieve CCENT simlet.



2. Configure the router' name

```
Router>enable
Router#configure tterminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#hostname GATEWAY
```

3. Configure the enable password and secret to "cisco"

```
GATEWAY(config)#enable password cisco
GATEWAY(config)#enable secret cisco
```

4. Configure password encryption for this router

```
GATEWAY(config)#service password-encryption
```

5. Configure the console access

```
GATEWAY(config)#line console 0
GATEWAY(config-line)#password cisco
GATEWAY(config-line)#login
GATEWAY(config-line)#logging synchronous
GATEWAY(config-line)#exec-timeout 2 45
GATEWAY(config-line)#history size 10
```

Configuring Default Route

Console Based | **GUI Based**

Console Based :

Description: ip default-gateway command is used when ip routing is disabled on a Cisco router. Use the **ip default-network** and **ip route 0.0.0.0 0.0.0.0** commands are used to set the gateway of last resort on Cisco routers that have ip routing enabled.

Command syntax:

1. ip default-gateway <ip address>
example: ip default-gateway 192.168.14.2

2. ip default-network <ip address>
example: ip default-network 192.168.1.0

3. ip route 0.0.0.0 0.0.0.0 <ip-address>
example: ip route 0.0.0.0 0.0.0.0 192.168.5.1

here 192.168.5.1 is the gateway of last resort to network 0.0.0.0

Instructions:

1. Enter into Global Configuration Mode

2. Set the default Network number as 192.168.17.0

```
R1>enable
R1#configure terminal
R1(config)#ip default-network 192.168.17.0
```

GUI Based :

Description: ip default-gateway command is used when ip routing is disabled on a Cisco router. The ip default-network and ip route 0.0.0.0 0.0.0.0 commands are used to set the gateway of last resort on Cisco routers that have ip routing enabled.

Command syntax:

1. ip default-gateway <ip address>
Ex: ip default-gateway 192.168.14.2

2. ip default-network <ip address>
Ex: ip default-network 192.168.1.0

3. ip route 0.0.0.0 0.0.0.0 <ip-address>
Ex: ip route 0.0.0.0 0.0.0.0 192.168.5.1

Here 192.168.5.1 is the gateway of last resort to network 0.0.0.0

Instructions:

1. Choose Configure > Router > Default Routing

2. Default routing window appears configure the next hop ip address as 192.168.1.5 and click OK
button.

Configure Standard Access Control List Step by Step Guide

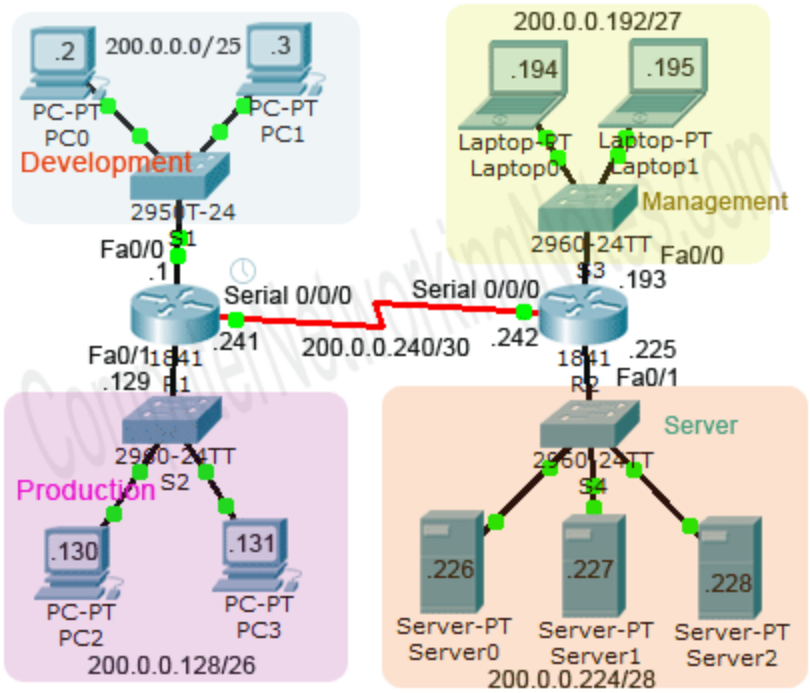
This tutorial explains how to create, enable and configure Standard Access Control List (Number and Named) in router step by step with examples. Learn how to create and implement Standard Access List statements and conditions with wildcard mask in easy language.

A standard ACL can be used for several purpose. In this tutorial we will see how it can be used in controlling the unwanted network traffic. With standard ACL, we can define certain conditions for the network traffic passing through the router. Once defined, Standard ACL works like a gate keeper that will allow only the authorized people (packets). All unwanted people (packets) are kicked out from the gate.

For demonstration purpose I will use packet tracer network simulator software. You can use it or can use any other network simulator software such as Boson, NetSim, GNS etc.

If require, you can download the latest as well as earlier version of Packet Tracer from here. Download Packet Tracer

Create a topology as illustrate in following figure.



This network is built with single class C IP address 200.0.0.0/24. Through VLSM network is divided in following sections:-

- Development (200.0.0.0/25)
- Production (200.0.0.128/26)
- Management (200.0.0.192/27)
- Server (200.0.0.224/28)

These sections are connected via two routers. Routers are running RIVv2 routing protocol.

VLSM Chart for Subnetted networks

Block size	Slash notation	Interface	Network address	Subnet mask
128	/25	Fa0/0 (R1)	200.0.0.0	255.255.255.
64	/26	Fa0/1 (R1)	200.0.0.128	255.255.255.
32	/27	Fa0/0 (R2)	200.0.0.192	255.255.255.
16	/28	Fa0/1 (R2)	200.0.0.224	255.255.255.
4	/30	Serial 0/0/0 (R1-R2)	200.0.0.240	255.255.255.

In VLSM we create multiple smaller networks from single large IP network. To learn more about VLAN check this tutorial. VLSM Tutorial with Examples

If you do not wish to start from scratch, download this pre-configured practice lab. This lab contains all initial configuration explained above. Just load this lab in packet tracer and start following this tutorial right from here. Download practice topology for Standard ACL configuration

In this network, at this moment all sections are connected with each other's. Users are able to access all resources from other sections as well as their own. You are hired to secure this network.

This network has following security requirements.

Section level requirement

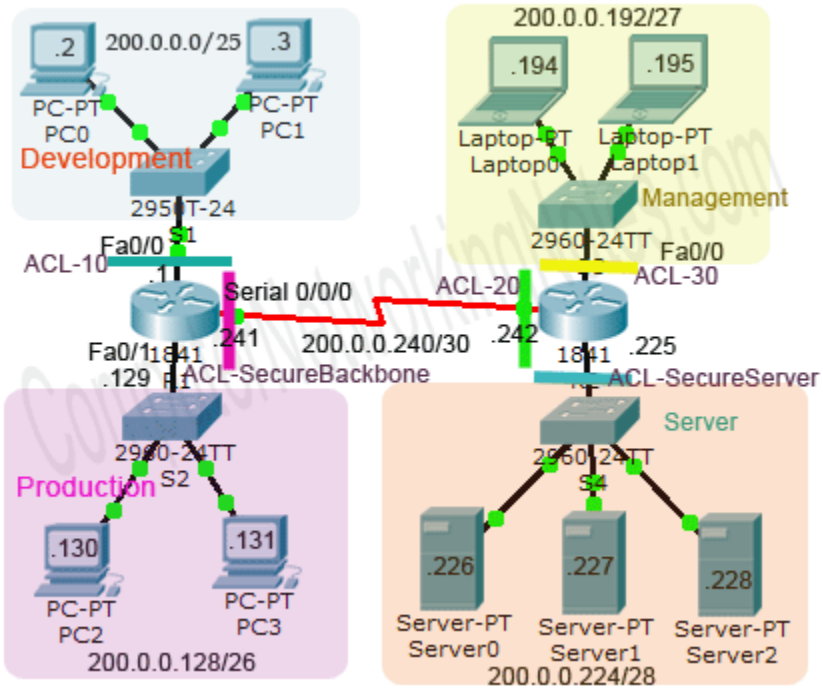
- Development section should be able to access only production section. It should not be able to access management section and server section.
- Production section should be able to access only development section. It should not be able to access management section and server section.

User level requirement

- One user (PC0) from development section should not be able to access anything except its own section.
- One user (PC2) from production section should also be able to access management section but not server section.
- One user (PC3) from production section should be able to access server section but not management section.
- One user (laptop0) from management section should be able to access only Server section not the development section and production section.

This tutorial is the third part of our article “Cisco IP ACL Configuration Guide”. You can read other parts of this article here:-
Access Control List Explained with Examples
This tutorial is the first part of this article. In this part I provided a brief introduction to Cisco IP ACLs such as what is ACL and how it works including ACLs direction and locations.
Standard ACL Configuration Commands Explained
This tutorial is the second part of this article. In this part I explained Standard Access Control List configuration commands and its parameters in detail with examples.
Extended ACL Configuration Commands Explained
This tutorial is the fourth part of this article. In this part I will explain Extended Access Control List configuration commands and its parameters in detail with examples.
Configure Extended Access Control List Step by Step Guide
This tutorial is the last part of this article. In this part I will provide a step by step configuration guide for Extended Access Control List.

For above requirements we need to secure five locations. For each location we need a separate ACL.



As you know we can create a standard ACL in three ways:-

1. Classic Numbered
2. Modern Numbered
3. Modern Named

To give you a better overview of these methods I will include all of them in this example.

ACL Number / Name	ACL Type	ACL Direction	Applied Interface
10	Classic Numbered	Inbound	R1's Fa0/0
20	Modern Numbered	Outbound	R2's Serial 0/0/0
30	Classic Numbered	Outbound	R2's Fa0/0
SecureBackbone	Modern Named	Outbound	R1's Serial 0/0/0
SecureServer	Modern Named	Outbound	R2's Fa0/1

Understanding ACL requirements

ACL is just like a double edge sword. We need to be extra careful while working with ACLs. A little mistake can mesh entire network data flow. Instead of creating ACL conditions directly in router, it's always a better idea to create them in paper first. This way we can update / reorder or remove conditions without recreating entire ACL.

For example our first requirement from section level requirements says "block production department from gaining access in management section". For this requirement we have to create a deny statement at section level. Suppose we created necessary condition for this requirement directly in router without reading remaining requirement. And later we came to know that one user from production section needs permission to access management section.

In this situation if we have created ACL directly in router using classical number method then the only way to allow this user is to delete the existing ACL and recreate it with allow statement prior to deny statement. But if we have created these conditions in paper then we could easily reorder / update /change them without recreating entire ACL. Once we are satisfy with conditions in paper, we can easily create them in router.

Okay let's create ACL conditions from section level requirements. Our requirements are

Development section should be able to access only production section. It should not be able to access management section and server section.

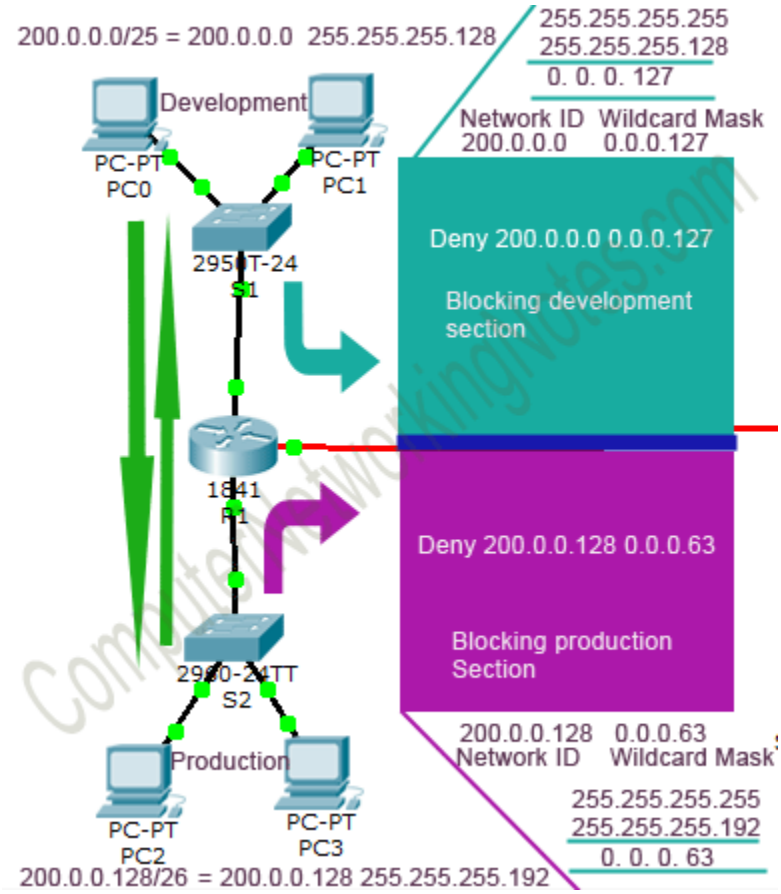
Production section should be able to access only development section. It should not be able to access management section and server section.

By default router does not filter any traffic unless we manually put an ACL. This behavior fulfills our half requirement. Production section and development section are able to access each other. We only need to control them from accessing management section and sever section.

In order to access Management section and Server section, both (Development and Production) section need to go through the Serial 0/0/0 interface. If we put deny condition in *SecureBackbone* ACL for development and production section, above requirements will be fulfilled.

ACL-SecureBackbone

- Deny 200.0.0.0 0.0.0.127 (*Blocking development section traffic from going outside*)
- Deny 200.0.0.128 0.0.0.63 (*Blocking production section traffic from going outside*)



Okay now let's see our user level requirement one by one from ACLs point of view.

Our first requirement is

One user (PC0) from development section should not be able to access anything except its own section.

This requirement needs Inbound ACL. As user only needs to access its own section which he can access through the LAN (switch) network. This user has nothing to access from other sections. We should drop the traffic from this user as soon as it enters in the interface (Fa0/0 of R1).

ACL-10

- deny 200.0.0.2 0.0.0.0 (*Blocking single user from development section*)
- permit any (*allowing all remaining traffic.*)

If we do not create ***permit any*** statement then router will block all traffic coming in this interface. As we know, as soon as we create our first statement, an Implicit Deny Statement would be added automatically in the end of ACL.

Our next requirement is

One user (PC2) from production section should also be able to access management section but not server section.

Let's see this requirement from ACL's point of view:-

User belongs to Production section. Being a member of production section:-

He should be able to access Development section (*Already doing, no action is required*).

He should not be able to access Management section and Server section. (*Here group level permission is restricting user from gaining access on management section and server section. But his individual permission is allowing him to access management section.*)

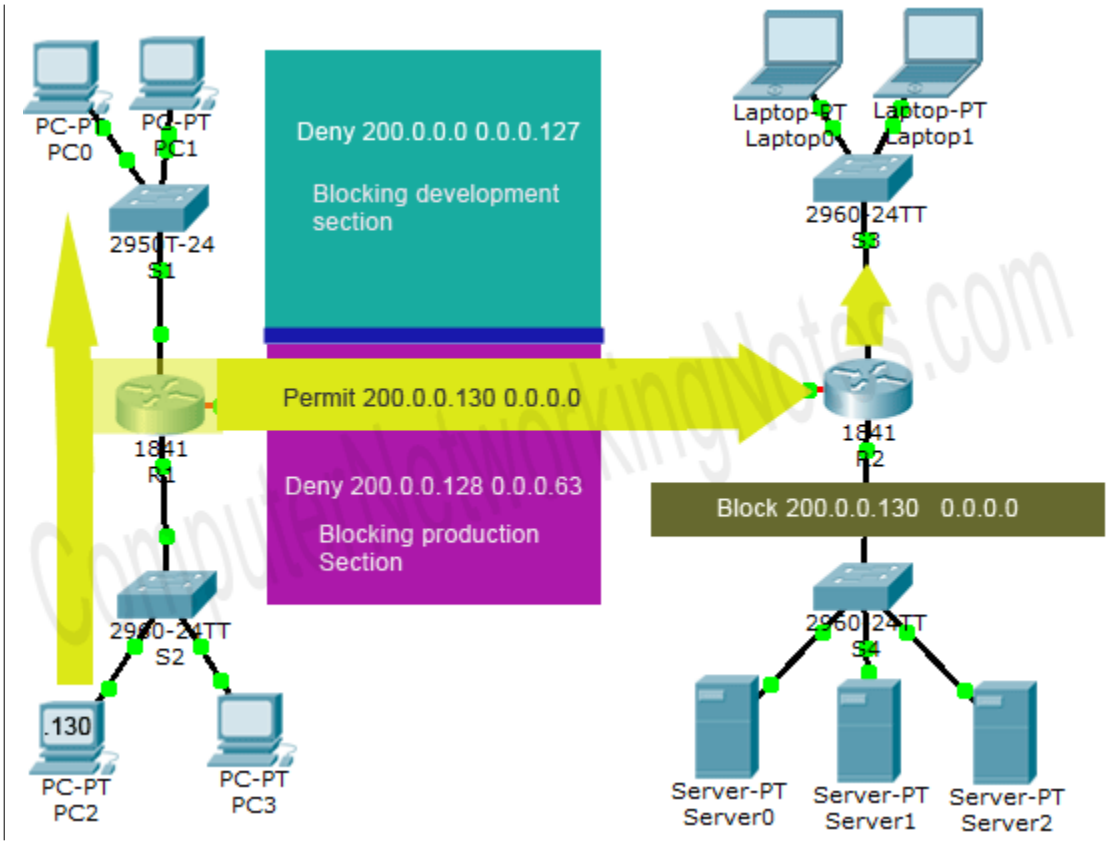
Whenever there is a conflict between User level permission and Group level permission, User level permission always override the Group level permission.

But wait…. we have already blocked group in SecureBackbone ACL at R1's Serial 0/0/0. So how could we allow single user from group while blocking the rest?

If you are reading this article from first, then answer should have already clicked in your mind. If you are guessing about ordering of ACL then you are absolutely right. With proper ordering, we can easily achieve this goal. As we know ACL conditions are processed from top to down without skipping. Once a match found, no further conditions are processed for that packet. So if we put permit condition for this host before the deny condition for the group then *SecureBackbone* ACL will do exactly what we want.

With permit condition, we will create a window for PC2 in SecureBackbone wall. Through this window, PC2 will be able to access the sections attached with R2.

R2 has two sections; Management and Server. PC2 will be able to access both sections. But as per requirement it should be allowed to access only Management section. We need to block it from accessing server section. For this goal we need to put a deny condition in *SecureServer* ACL.



Oaky lets update ACLs

ACL-SecureBackbone

- Deny 200.0.0.0 0.0.0.127 (*Blocking development section traffic from going outside*)
- Permit 200.0.0.130 0.0.0.0 (*Allowing single host traffic from production section*)
- Deny 200.0.0.128 0.0.0.63 (*Blocking production section traffic from going outside*)

ACL -SecureServer

- Deny 200.0.0.130 0.0.0.0 (*Blocking single host from accessing server section*)

Our next requirement is identically same as previous requirement

One user (PC3) from production section should be able to access server section but not management section.

For this requirement we need a permit condition is SecureBockbone ACL and one deny condition in ACL 30 for this PC3.

ACL-SecureBackbone

- Deny 200.0.0.0 0.0.0.127 (*Blocking development section traffic from going outside*)
- Permit 200.0.0.130 0.0.0.0 (*Allowing single host traffic from production section*)
- Permit 200.0.0.131 0.0.0.0 (*Allowing single host traffic from production section*)
- Deny 200.0.0.128 0.0.0.63 (*Blocking production section traffic from going outside*)

ACL -30

- Deny 200.0.0.131 0.0.0.0 (Blocking single host from accessing management section)

Our last requirement is fairly simple.

One user (laptop0) from management section should be able to access only Server section not the development section and production section.

Simply creating a block condition in ACL 20 (R2's Serial 0/0/0) will do this job.

- deny 200.0.0.194 0.0.0.0 (*Blocking single host from management section*)

We have gone through all the requirements. Let's have quick look on ACL conditions

ACL-10 (*Filtering incoming traffic on R1's Fa0/0*)

- deny 200.0.0.2 0.0.0.0 (*Blocking incoming traffic from single host*)
- permit any (*Allowing remaining all hosts.*)

ACL-SecureBackbone (*Filtering outgoing traffic on R1's Serial 0/0/0*)

- deny 200.0.0.0 0.0.0.127 (*Blocking development section*)
- permit 200.0.0.130 0.0.0.0 (*Allowing single host from production section*)
- permit 200.0.0.131 0.0.0.0 (*Allowing single host from production section*)
- deny 200.0.0.128 0.0.0.63 (*Blocking production section*)

ACL-20 (*Filtering outgoing traffic on R2's Serial 0/0/0*)

- deny 200.0.0.194 0.0.0.0 (*Blocking single host from management section*)
- permit any (*Allowing remaining traffic*)

ACL-30 (*Filtering traffic going from R2's Fa0/0*)

- deny 200.0.0.131 0.0.0.0 (*Blocking single user from production section from gaining unauthorized on management section.*)
- permit any (*Allowing remaining traffic*)

ACL-SecureServer (*Filtering traffic going from R2's Fa0/1*)

- deny 200.0.0.130 0.0.0.0 (*Blocking single user from production section from gaining unauthorized on server section.*)
- permit any (*Allowing remaining traffic*)

That's all paper work we need to do before creating real ACLs.

Well… you may be a little bit annoyed with all above preparation. But believe me friends; it will save a lot of time and effort in Cisco exams and as well as in job life.

Create Standard ACL

A standard ACL can be created in two ways:-

1. Classic numbered method
2. Modern numbered or named method

Classic numbered method uses following global configuration mode command

```
Router(config)# access-list ACL_Identifier_number permit/deny matching-parameters
```

Modern numbered or named method uses following global configuration mode commands

```
Router(config)#ip access-list standard ACL_Number / ACL_Name
Router(config-std-nacl)#permit / deny Source Address
Router(config-std-nacl)#exit
Router(config)#
```

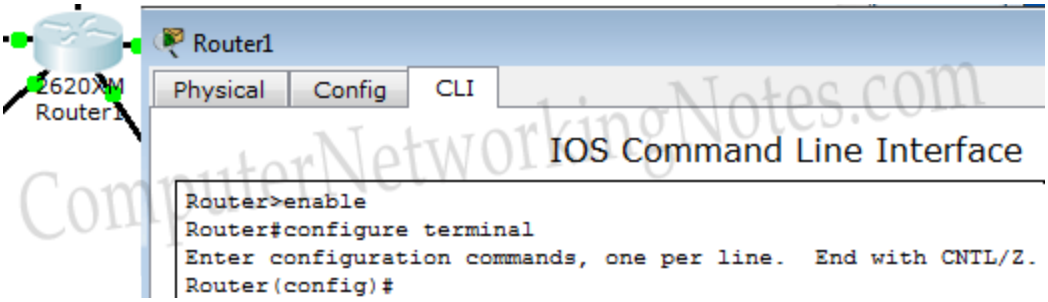
I have already explained above commands and parameters in detail with examples in previous part of this article. For this part I assume that you are familiar with above commands.

In our example we will create two ACLs (10 and SecureBackbone) in Router1 and three ACLs (20, 30 and SecureServer) in Router2.

Okay let's create them one by one

ACL-10 (Configuration style - ***Classical Numbered***)

Access CLI prompt of Router1 and enter in global configuration mode



Enter following commands

```
Router(config)#access-list 10 deny 200.0.0.2 0.0.0.0
Router(config)#access-list 10 permit any
Router(config)#
```

Great job, we have just created our first ACL with classic numbered method. Now let's create our second ACL, but this time use modern named method.

ACL-SecureBackbone (Configuration style – *Modern Named*)

```
Router(config)#ip access-list standard SecureBackbone
Router(config-std-nacl)#deny 200.0.0.0 0.0.0.127
Router(config-std-nacl)#permit 200.0.0.130 0.0.0.0
Router(config-std-nacl)#permit 200.0.0.131 0.0.0.0
Router(config-std-nacl)#deny 200.0.0.128 0.0.0.63
Router(config-std-nacl)#exit
Router(config)#
```

Good going, we have finished our ACL creation task or router R1. Now access the global configuration mode of router R2 and enter following commands to create ACL20

ACL-20 (Configuration style – *Classical Numbered*)

```
Router(config)#ip access-list standard 20
Router(config-std-nacl)#deny 200.0.0.194 0.0.0.0
Router(config-std-nacl)#permit any
Router(config)#
```

Following commands will create ACL-30

ACL-30 (Configuration style – *Modern Numbered*)

```
Router(config)#access-list 30 deny 200.0.0.131 0.0.0.0
Router(config)#access-list 30 permit any
Router(config)#
```

Finally use following commands to create our last ACL-SecureServer

ACL-SecureServer (Configuration style – *Modern Named*)

```
Router(config)#ip access-list standard SecureServer
Router(config-std-nacl)#deny 200.0.0.130 0.0.0.0
Router(config-std-nacl)#permit any
Router(config-std-nacl)#exit
Router(config)#
```

Now our security guards (ACLs) have an authorized persons (conditions) list. Right now they are just sitting in office (router). From here they will do nothing. We need to send them on their job place (interface) where they will perform their jobs (filtrations).

Assign Standard ACLs in interfaces

Regardless what method we used in creating the ACLs, assigning them in interfaces are the same steps process:-

```
Router(config)#interface  type [slot_#]  port_#
Router(config-if)#ip access-group  ACL_#  in|out
```

Commands and parameters are explained in previous part of this article. In this part we will use these commands in assigning the ACLs.

Let’s assign our ACLs in their respective interfaces

ACL-10 (R1’s Fa0/0 interface, *Inbound direction*)

```
Router(config)#interface fastethernet 0/0
Router(config-if)#ip access-group 10 in
Router(config-if)#exit
Router(config)#
```

ACL-SecureBackbone (R1’s Serial 0/0/0, *Outbound direction*)

```
Router(config)#interface serial 0/0/0
```

```
Router(config-if)#ip access-group SecureBackbone out
Router(config-if)#exit
Router(config)#
```

ACL-20 (R2's Serial 0/0/0 interface, *Outbound direction*)

```
Router(config)#interface serial 0/0/0
Router(config-if)#ip access-group 20 out
Router(config-if)#exit
Router(config)#
```

ACL-30 (R2's Fa0/0 interface – *Outbound direction*)

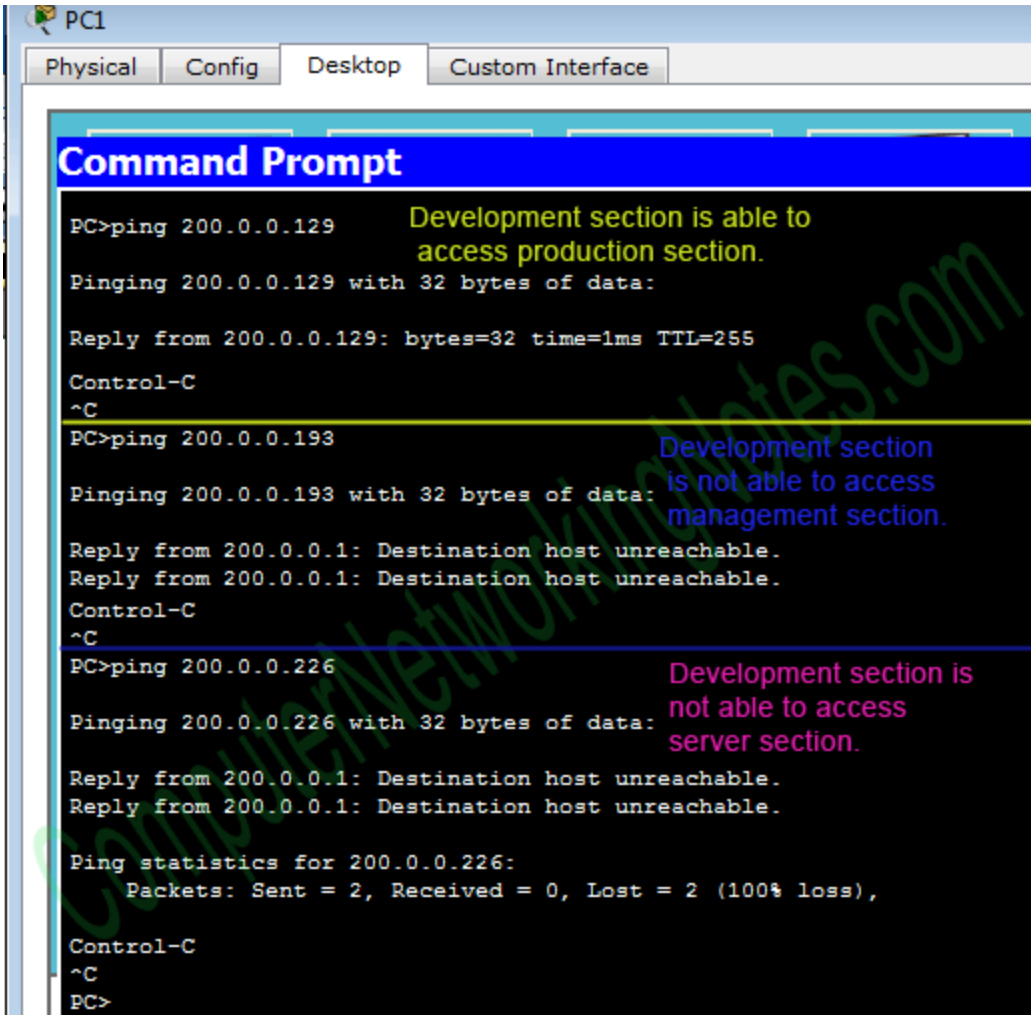
```
Router(config)#interface fastethernet 0/0
Router(config-if)#ip access-group 30 out
Router(config-if)#exit
Router(config)#
```

ACL-SecureServer (R2's Fa0/1 interface – *Outbound direction*)

```
Router(config)#interface fastethernet 0/1
Router(config-if)#ip access-group SecureServer out
Router(config-if)#exit
Router(config)#
```

Testing Standard ACLs

To verify the implementation, we can use **ping** command. **ping** command is used to test the connectivity between source and destination. For example in following figure I tested our first requirement from PC1 (belongs to development section).



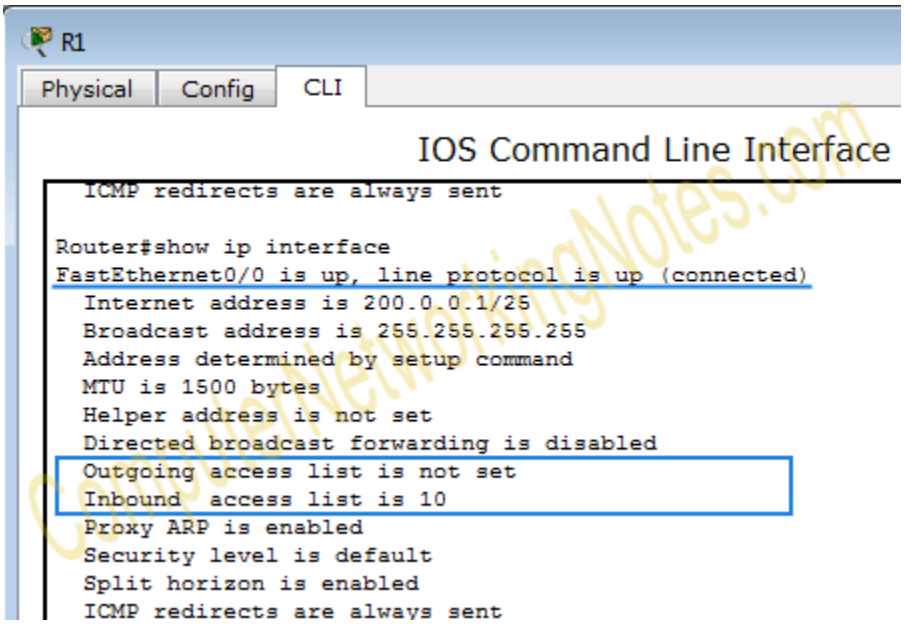
Now it's your turn to test remaining conditions. If you have followed all above steps then requirements should be fulfilled. If you are missing any requirement or not getting result as expected, use my practice topology for cross check. You can download my practice topology from here.

Download Standard ACL configuration topology configured

Verifying Standard Access List configuration

Once created and activated ACLs, we can verify them with following privilege exec mode commands.

To show which ACLs are activated on which interfaces in which direction, we can use ***show ip interface*** command



From output we can see that ACL-10 is applied in inbound direction on FastEthernet0/0. By default above command will list all interfaces. To view a single interface, we need to specify it in above command as command line option. For example, to view only serial interface use **show ip interface *serial 0/0/0*** command.

To view the conditions in ACL, we have two commands

Router# show access-lists ACL_Number_or_Name (Optional, used to see the specific ACL)

```
Router#show access-lists 10
Standard IP access list 10
    deny host 200.0.0.2
    permit any
Router#show access-lists SecureBackbone
Standard IP access list SecureBackbone
    deny 200.0.0.0 0.0.0.127
    permit host 200.0.0.130
    permit host 200.0.0.131
    deny 200.0.0.128 0.0.0.63
Router#
```

Router# show ip access-list ACL_Number_or_Name (Optional, used to see the specific ACL)

```
Router#show access-lists 10
Standard IP access list 10
    deny host 200.0.0.2
    permit any (8 match(es))
Router#show ip access-list 10
Standard IP access list 10
    deny host 200.0.0.2
    permit any (8 match(es))
Router#
```

Have you notice any difference between outputs? Second command provides more detailed information about modern style ACLs. It lists the sequence number of each condition in ACL. Sequence numbers are used to edit or delete any condition from ACL. Sequence numbers are available only when you create ACL from modern style.

Router keeps track of every match on every condition. To reset this counter, use ***clear*** command.

```
Router#show access-lists SecureBackbone
Standard IP access list SecureBackbone
    deny 200.0.0.0 0.0.0.127 (8 match(es))
    permit host 200.0.0.130
    permit host 200.0.0.131
    deny 200.0.0.128 0.0.0.63
Router#clear access-list counters SecureBackbone
Router#show access-lists SecureBackbone
Standard IP access list SecureBackbone
    deny 200.0.0.0 0.0.0.127
    permit host 200.0.0.130
    permit host 200.0.0.131
    deny 200.0.0.128 0.0.0.63
Router#
```

We can also view all running configuration including ACLs from ***show running-config*** command.

```
Router#show running-config
Building configuration...
interface FastEthernet0/0
 ip address 200.0.0.1 255.255.255.128
 ip access-group 10 in
 duplex auto
 speed auto
!
interface FastEthernet0/1
 ip address 200.0.0.129 255.255.255.192
 duplex auto
 speed auto
!
interface Serial0/0/0
 bandwidth 64
 ip address 200.0.0.241 255.255.255.252
 ip access-group SecureBackbone out
 clock rate 64000
router rip
 version 2
 network 200.0.0.0
access-list 10 deny host 200.0.0.2
access-list 10 permit any
ip access-list standard SecureBackbone
 deny 200.0.0.0 0.0.0.127
 permit host 200.0.0.130
 permit host 200.0.0.131
 deny 200.0.0.128 0.0.0.63
line vty 0 4
 login
end

Router#
```


Editing / Updating Standard ACLs

We can edit or update a standard ACL only if it is created from modern configuration style. If it is created from classical configuration style then we cannot edit or update it, we can only append it.

How will I know which ACL is created from which style?

ACLs created from modern way have sequence numbers. We can use ***show ip access-list*** command to know whether a specific ACL is created from classic style or modern style. If output of this command shows sequence numbers in front of conditions then that ACL is created from modern style. For example following figure illustrates the output of show ip access-list command from router R1.

```
Router#show access-lists 10
Standard IP access list 10
    deny host 200.0.0.2
    permit any (8 match(es))
Router#show ip access-list 10
Standard IP access list 10
    deny host 200.0.0.2
    permit any (8 match(es))
Router#
```

As we can see in output, ACL-10 has no sequence number while ACL-SecureBackbone has it. So ACL-10 is created from classical numbered approach while ACL-SecureBackbone is created from modern named style.

Okay now we know how to find out the configuration style of ACLs. Let's edit them. Suppose we have two tasks, one for each ACL:-

- For ACL-10 :- Deny host 200.0.0.3
- For ACL-SecureBackbone Deny host 200.0.0.130

For ACL-10

As we know that this ACL is created from classical numbered method, so it cannot be edited. We have only one option, delete existing ACL and create new one with requirement.

For ACL-SecureBackbone

This ACL is created from modern named method. We can edit it directly. We are asked to deny the **host 200.0.0.130**, which is currently allowed (*20 permit host 200.0.0.130*).

Okay let's update this ACL step by step.

Verify current status

```
Router#show ip access-list SecureBackbone
Standard IP access list SecureBackbone
    10 deny 200.0.0.0 0.0.0.127
    20 permit host 200.0.0.130
    30 permit host 200.0.0.131
    40 deny 200.0.0.128 0.0.0.63
```

Remove old permission

```
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#ip access-list standard SecureBackbone
Router(config-std-nacl)#no 20
Router(config-std-nacl)#exit
```

```
Router(config)#exit
```

Confirm removal

```
Router#show ip access-list SecureBackbone
Standard IP access list SecureBackbone
    10 deny 200.0.0.0 0.0.0.127
    30 permit host 200.0.0.131
    40 deny 200.0.0.128 0.0.0.63
```

Insert new condition in the place of old condition

```
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#ip access-list standard SecureBackbone
Router(config-std-nacl)#20 deny 200.0.0.130 0.0.0.0
Router(config-std-nacl)#exit
Router(config)#exit
```

Verify update

```
Router#show ip access-list SecureBackbone
Standard IP access list SecureBackbone
    10 deny 200.0.0.0 0.0.0.127
    20 deny host 200.0.0.130
    30 permit host 200.0.0.131
    40 deny 200.0.0.128 0.0.0.63
Router#
```

How to delete a Standard ACL

We have two commands to delete a standard ACL.

Router(config)#no access-list [ACL Number] *and* Router(config)#no ip access-list standard [ACL Number or Name]

First command is used to delete numbered ACL while second command is used to delete both numbered and named ACLs. Let's have an example of both commands.

Delete both ACLs from router R1.

```
Router(config)#no access-list 10
Router(config)#no ip access-list standard SecureBackbone
```

Conclusion : Hence in this we have studied that Network Simulator tool and Configure A router using router commands and Access Control lists Standard & Extended.

D Y PATIL INSTITUTE OF ENGINEERING & TECHNOLOGY AMBI, PUNE	**D Y PATIL INSTITUTE OF ENGINEERING & TECHNOLOGY, AMBI**	LABORATORY MANUAL
	EXPERIMENT TITLE: Using a Network Simulator Configure EIGRP , RIP, WLAN and DHCP.	
DEPARTMENT OF INFORMATION TECHNOLOGY		

RIP, WLAN and DHCP.

EXPERIMENT NO. : DYPIET/IT/TE/SL-IV	**SEMESTER** : VI(TE)	**PAGE:1-**

AIM: Using a Network Simulator Configure EIGRP , RIP, WLAN and DHCP.

OBJECTIVES:

To study,

Network Simulator tool

Configure EIGRP , RIP, WLAN and DHCP

THEORY:

Packet Tracer 5.0:

Cisco Packet Tracer is a powerful network simulation program that allows students to experin with network behavior and ask “what if” questions. As an integral part of the Networking Acad comprehensive learning experience, Packet Tracer provides simulation, visualization, author assessment, and collaboration capabilities and facilitates the teaching and learning of com technology concepts.

Configuration of Routing protocols:

Following are the protocols which are to be configured in Packet Tracer.

Static Routing: There are two basic methods of building a routing table:

- Static Routing
- Dynamic Routing

A static routing table is created, maintained, and updated by a network administrator, manuall static route to every network must be configured on every router for full connectivity. This provid granular level of

control over routing, but quickly becomes impractical on large networks.

Routers will not share static routes with each other, thus reducing CPU/RAM overhead and sa bandwidth. However, static routing is not fault-tolerant, as any change to the routing infrastruc (such as a link going down, or a new network added) requires manual intervention. Routers opera in a purely static environment cannot seamlessly choose a better route if a link becomes unavaila Static routes have an Administrative Distance (AD) of 1, and thus are always preferred over dyna routes, unless the default A D is changed. A static route with an adjusted AD is called a floating st route , and is covered in greater detail in another guide. The following briefly outlines the

advantages and disadvantages of static routing:

Advantages of Static Routing
•Minimal CPU/Memory overhead
•No bandwidth overhead (updates are not shared between routers)
•Granular control on how traffic is routed

Disadvantages of Static Routing

•Infrastructure changes must be manually adjusted
•No “dynamic” fault tolerance if a link goes down
•Impractical on large network

Configuration of Static routing:

b.RIPV2 Routing Protocol:

RIP (Routing Information Protocol)

RIP is a standardized Distance Vector protocol, designed for use on smaller networks. RIP was on
the first true Distance Vector routing protocols, and is supported on a wide variety of systems.

RIP adheres to the following Distance Vector characteristics:

•RIP sends out periodic routing updates (every 30 seconds)
•RIP sends out the full routing table every periodic update
•RIP uses a form of distance as its metric (in this case, hop count)
•RIP uses the Bellman-Ford Distance Vector algorithm to determine the best “path” to a partic
destination

Other characteristics of RIP include:

•RIP supports IP and IPX routing.
•RIP utilizes UDP port 520
•RIP routes have an administrative distance of 120
•RIP has a maximum hop count of 15 hops.

Any network that is 16 hops away or more is considered unreachable to RIP, thus the maxim
diameter of the network is 15 hops. A metric of 16 hops in RIP is considered a poison route or infi
metric.

If multiple paths exist to a particular destination , RIP will load balance between those paths
default, up to

4) Only if the metric (hop count) is equal. RIP uses a round-robin system of load-balancing betw
equal metric routes, which can lead to pinhole congestion.

For example, two paths might exist to a particular destination, one going through a 9600 baud
the other via a T1. If the metric (hop count) is equal, RIP will load-balance, sending an equal amo
of traffic down the 9600 baud link and the T1. This will (obviously) cause the slower link to bec
congested.

CONFIGURATION:

1. RIP

There is no big difference between RIP version 1 and version 2 when we are applying them in pa
tracer. In order to apply RIP version 2 on packet tracer.

Just make sure that the protocol is applied as an additional step and cannot replace the basic s

i.e. we have to assign IP addresses to the router's interfaces and PCs and also change the state of interfaces from down to UP and then we will go ahead and apply Protocol.

2. EIGRP

EIGRP (Enhanced Interior Gateway Routing Protocol)

EIGRP is a Cisco-proprietary Hybrid routing protocol, incorporating features of both Distance-Ve and Link-State routing protocols.

EIGRP adheres to the following Hybrid characteristics:

•EIGRP uses Diffusing Update Algorithm (DUAL) to determine the best path among all "feasi paths. DUAL also helps ensure a loop-free routing environment.

•EIGRP will form neighbor relationships with adjacent routers in the same Autonomous System (A

•EIGRP traffic is either sent as unicasts, or as multicasts on address 224.0.0.10, depending on EIGRP packet type.

•Reliable Transport Protocol (RTP) is used to ensure delivery of most EIGRP packets.

•EIGRP routers do not send periodic, full-table routing updates. Updates are sent when a cha occurs, and include only the change.

•EIGRP is a classless protocol, and thus supports VLSMs.

Other characteristics of EIGRP include:

•EIGRP supports IP, IPX, and AppleTalk routing.

•EIGRP applies an Administrative Distance of 90 for routes originating within the local Autonom System.

•EIGRP applies an Administrative Distance of 170 for external routes coming from outside the l Autonomous System

•EIGRP uses Bandwidth and Delay of the Line, by default, to calculate its distance metric. It supports

three other parameters to calculate its metric: Reliability, Load, and MTU.

•EIGRP has a maximum hop-count of 224, though the default maximum hop-count is set to 100.

EIGRP, much like OSPF, builds three separate tables

•Neighbor table

– list of all neighboring routers. Neighbors must

belong to the same

Autonomous System

•Topology table

– list of all routes in the Autonomous System

•Routing table

– contains the

Best route for each known network

3. OSPF

OSPF (Open Shortest Path First)

OSPF is a standardized Link-State routing protocol, designed to scale efficiently to support la networks.

OSPF adheres to the following Link State characteristics:

•OSPF employs a hierarchical network design using Areas.

•OSPF will form neighbor relationships with adjacent routers in the same Area.

•Instead of advertising the distance to connected networks, OSPF advertises the status of dire
connected
links using Link-State Advertisements (LSAs).

•OSPF sends updates (LSAs) when there is a change to one of its links, and will only send the cha
in the update. LSAs are additionally refreshed every 30 minutes.

•OSPF traffic is multicast either to address 224.0.0.5 (all OSPF routers) or 224.0.0.6(all Designa
Routers).

•OSPF uses the Dijkstra Shortest Path First algorithm to determine the shortest path.

•OSPF is a classless protocol, and thus supports VLS Ms. Other characteristics of OSPF include:

•OSPF supports only IP routing.

•OSPF routes have an administrative distance is 110.

•OSPF uses cost as its metric, which is computed based on the bandwidth of the link. OSPF ha
hop-count limit.

The OSPF process builds and maintains three separate tables:

• **Neighbor table** – contains a list of all neighboring routers.

• **Topology table**– contains a list of all possible routes to all known networks within an area.

•R**outing table**– contains the best route for each known network.

Here are the basic set of commands that we can apply on router CLI mode for RIP configuration

Commands	Descriptions
Router(config)#router rip	Enables RIP as a routing protocol
Router(config-router)#network w.x.y.z	w.x.y.z is the network number of the directly connected netw you want to advertise.
Router(config)#no router rip	Turns off the RIP routing process
Router(config-router)#no network w.x.y.z	Removes network w.x.y.z from the RIP routing process.
Router(config-router)#version 2	RIP will now send and receive RIPv2 packets globally.
Router(config-router)#version 1	RIP will now send and receive RIPv1 packets only
Router(config-router)#no auto-summary	RIPv2 summarizes networks at the classful boundary. This command turns autosummarization off.
Router(config-router)#passive-interface s0/0/0	RIP updates will not be sent out this interface.

Router(config-router)#no ip split-horizon	Turns off split horizon (on by default).
Router(config-router)#ip split-horizon	Re-enables split horizon
Router(config-router)#timers basic 30 90 180 270 360	Changes timers in RIP: 30 = Update timer (in seconds) 90 = Invalid timer (in seconds) 180 = Hold-down timer seconds) 270 = Flush timer (in seconds) 360 = Sleep time (in millisecon
Router#debug ip rip	Displays all RIP activity in real time
Router#show ip rip database	Displays contents of the RIP database

Here are the basic set of commands that we can apply on router CLI mode for OSPF configuration

Commands	Descriptions
Router(config)#router ospf 1	Starts OSPF process 1. The process ID is any positive integer value between 1 and 65,535.
Router(config-router)#network 172.16.0.0 0.0.255.255 area 0	OSPF advertises interfaces, not networks. Uses the wildcard mas determine which interfaces to advertise.
Router(config-if)#ip ospf hellointerval timer 20	Changes the Hello Interval timer to 20 seconds.
Router(config-if)#ip ospf deadinterval 80	Changes the Dead Interval timer to 80 seconds.
NOTE: Hello and Dead Interval timers must match for routers to become neighbors	
Router#show ip protocol	Displays parameters for all protocols running on the router
Router#show ip route	Displays a complete IP routing table
Router#show ip ospf	Displays basic information about OSPF routing processes
Router#show ip ospf interface	Displays OSPF info as it relates to all interfaces
Router#show ip ospf interface fastethernet 0/0	Displays OSPF information for interface fastethernet 0/0
Router#show ip ospf border-routers	Displays border and boundary router information
Router#show ip ospf neighbor	Lists all OSPF neighbors and their states

Router#show ip ospf neighbor detail	Displays a detailed list of neighbors
Router#clear ip route *	Clears entire routing table, forcing it to rebuild
Router#clear ip route a.b.c.d	Clears specific route to network a.b.c.d
Router#clear ip opsf counters	Resets OSPF counters
Router#clear ip ospf process	Resets entire OSPF process, forcing OSPF to re-create neighbors database, and routing table
Router#debug ip ospf events	Displays all OSPF events
Router#debug ip ospf adjacency	Displays various OSPF states and DR/ BDR election betw adjacent routers
Router#debug ip ospf packets	Displays OPSF packets

Here are the basic set of commands that we can apply on router CLI mode for EIGRP configuration

Commands	**Descriptions**
Router(config)#router eigrp 1	Turns on the EIGRP process. 1 is the autonomous system number, which can be a number between 1 and 65,535.
Note:- All routers in the same autonomous system must use the same autonomous system number.	
Router(config-router)#network 10.0.0.0	Specifies which network to advertise in EIGRP.
Router(config-if)#bandwidth x	Sets the bandwidth of this interface to x kilobits to allow EIGRP to make a better metric calculation
TIP: The bandwidth command is used for metric calculations only. It does not change interface performance.	
Router(config-router)#no network 10.0.0.0	Removes the network from the EIGRP process.
Router(config)#no router eigrp 1	Disables routing process 1
Router(config-router)#auto-	Enables auto-summarization for the EIGRP

summary	process.
Router(config-router)#no autosummary	Turns off the auto-summarization feature.
Router(config-router)#variance n	include routes with a metric less than or equal to n times the minimum metric route for that destination, where n is the number specified by the variance command
NOTE: If a path is not a feasible successor, it is not used in load balancing. EIGRP supports up to six unequal-cost paths.	
Router(config)#interface serial 0/0	Enters interface configuration mode.
Router(config-if)#bandwidth 256	Sets the bandwidth of this interface to 256 kilobits to allow EIGRP to make a better metric calculation.
Router#show ip eigrp neighbors	Displays the neighbor table.
Router#show ip eigrp neighbors detail	Displays a detailed neighbor table.
Router#show ip eigrp interfaces	Shows information for each interface
Router#show ip eigrp interfaces serial 0/0	Shows information for a specific interface
Router#show ip eigrp interfaces 1	Shows information for interfaces running process 1.
Router#show ip eigrp topology	Displays the topology table
Router#show ip eigrp traffic	Shows the number and type of packets sent and received
Router#show ip route eigrp	Shows a routing table with only EIGRP entries
Router#debug eigrp fsm	Displays events/actions related to EIGRP feasible successor metrics (FSM)
Router#debug eigrp packet	Displays events/actions related to EIGRP packets
Router#debug eigrp neighbor	Displays events/actions related to your EIGRP

	neighbors
Router#debug ip eigrp neighbor	Displays events/actions related to your EIGRP neighbors
Router#debug ip eigrp notifications	Displays EIGRP event notifications

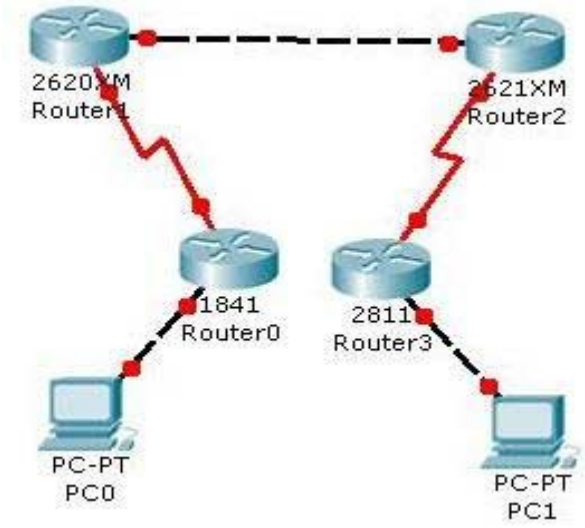
THEORY:

SIMULATION:

1. **Static Routing**

How to configure Static Route on router

In this article we will demonstrate an example of static route configurations. We will use four different series router so you can get familiar with all different platform Create a topology as shown in figure.



A static route is a manually configured route on your router. Static routes are typically used in smaller networks and when few networks or subnets exist, or with WAN links that have little available bandwidth. With a network that has hundreds of routes, static routes are not scalable, since you would have to configure each route and any redundant paths for that route on each router.

1841 Series Router0 (R1)		
	FastEthernet0/0	Serial0/0/0

IP address	10.0.0.1	20.0.0.1
Connected With	Pc0	R2 on Serial 0/0
2811 Series Router0 (R4)		
	FastEthernet0/0	Serial0/0/0
IP address	50.0.0.1	40.0.0.2
Connected With	Pc1	R3 on Serial 0/0
2621XM Series Router0 (R3)		
	FastEthernet0/0	Serial0/0/0
IP address	30.0.0.2	40.0.0.1
Connected With	FastEthernet0/0	R4 on Serial 0/0/0
2620XM Series Router1 (R2)		
	FastEthernet0/0	Serial0/0
IP address	30.0.0.1	20.0.0.2
Connected With	R3 on FastEthernet0/0	R1 on Serial 0/0/0
PC-PT PC0		
	FastEthernet0	Default Gateway
IP address	10.0.0.2	10.0.0.1
Connected With	R1 on FastEthernet0/0	
PC-PT PC1		
	FastEthernet0	Default Gateway
IP address	50.0.0.2	50.0.0.1

Connected With	R4 on FastEthernet0/0	

To configure any router double click on it and select CLI.To configure this topology use this step by step guide.

(1841Router0) Hostname R1

To configure and enable static routing on R1 follow these commands exactly.

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#hostname R1
R1(config)#interface fastethernet 0/0
R1(config-if)#ip address 10.0.0.1 255.0.0.0
R1(config-if)#no shutdown
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0,
changed state to up
R1(config-if)#exit
R1(config)#interface serial 0/0/0
R1(config-if)#ip address 20.0.0.1 255.0.0.0
R1(config-if)#clock rate 64000
R1(config-if)#bandwidth 64
R1(config-if)#no shutdown
%LINK-5-CHANGED: Interface Serial0/0/0, changed state to down
R1(config-if)#exit
%LINK-5-CHANGED: Interface Serial0/0/0, changed state to up
R1(config)#ip route 30.0.0.0 255.0.0.0 20.0.0.2
R1(config)#ip route 40.0.0.0 255.0.0.0 20.0.0.2
R1(config)#ip route 50.0.0.0 255.0.0.0 20.0.0.2
```

(2620XM-Router1) Hostname R2

To configure and enable static routing on R2 follow these commands exactly.

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#hostname R2
R2(config)#interface serial 0/0
R2(config-if)#ip address 20.0.0.2 255.0.0.0
R2(config-if)#no shutdown
%LINK-5-CHANGED: Interface Serial0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0, changed
```

```
state to up
R2(config-if)#exit
R2(config)#interface fastethernet 0/0
R2(config-if)#ip address 30.0.0.1 255.0.0.0
R2(config-if)#no shutdown
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up
R2(config-if)#exit
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0,
changed state to up
R2(config)#ip route 10.0.0.0 255.0.0.0 20.0.0.1
R2(config)#ip route 40.0.0.0 255.0.0.0 30.0.0.2
R2(config)#ip route 50.0.0.0 255.0.0.0 30.0.0.2
```

(2620XM-Router2)Hostname R3

To configure and enable static routing on R3 follow these commands exactly.

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#hostname R3
R3(config)#interface fastethernet 0/0
R3(config-if)#ip address 30.0.0.2 255.0.0.0
R3(config-if)#no shutdown
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0,
changed state to up
R3(config-if)#interface serial 0/0
R3(config-if)#ip address 40.0.0.1 255.0.0.0
R3(config-if)#clock rate 64000
R3(config-if)#bandwidth 64
R3(config-if)#no shutdown
%LINK-5-CHANGED: Interface Serial0/0, changed state to down
R3(config-if)#exit
%LINK-5-CHANGED: Interface Serial0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0, changed
state to up
R3(config)#ip route 10.0.0.0 255.0.0.0 30.0.0.1
R3(config)#ip route 20.0.0.0 255.0.0.0 30.0.0.1
R3(config)#ip route 50.0.0.0 255.0.0.0 40.0.0.2
```

(2811Router3) Hostname R4

To configure and enable static routing on R4 follow these commands exactly.

```
Router>enable
```

```
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface serial 0/0/0
Router(config-if)#ip address 40.0.0.2 255.0.0.0
Router(config-if)#no shutdown
%LINK-5-CHANGED: Interface Serial0/0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0/0,
changed state to up
Router(config-if)#exit
Router(config)#interface fastethernet 0/0
Router(config-if)#ip address 50.0.0.1 255.0.0.0
Router(config-if)#no shutdown
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0,
changed state to up
Router(config-if)#exit
Router(config)#ip route 10.0.0.0 255.0.0.0 40.0.0.1
Router(config)#ip route 20.0.0.0 255.0.0.0 40.0.0.1
Router(config)#ip route 30.0.0.0 255.0.0.0 40.0.0.1
```

PC-1

```
PC>ipconfig

IP Address.....................: 10.0.0.2
Subnet Mask....................: 255.0.0.0
Default Gateway................: 10.0.0.1

PC>ping 50.0.0.2

Pinging 50.0.0.2 with 32 bytes of data:

Reply from 50.0.0.2: bytes=32 time=156ms TTL=124
Reply from 50.0.0.2: bytes=32 time=127ms TTL=124
Reply from 50.0.0.2: bytes=32 time=156ms TTL=124
Reply from 50.0.0.2: bytes=32 time=140ms TTL=124

Ping statistics for 50.0.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 127ms, Maximum = 156ms, Average = 144ms
PC>
```

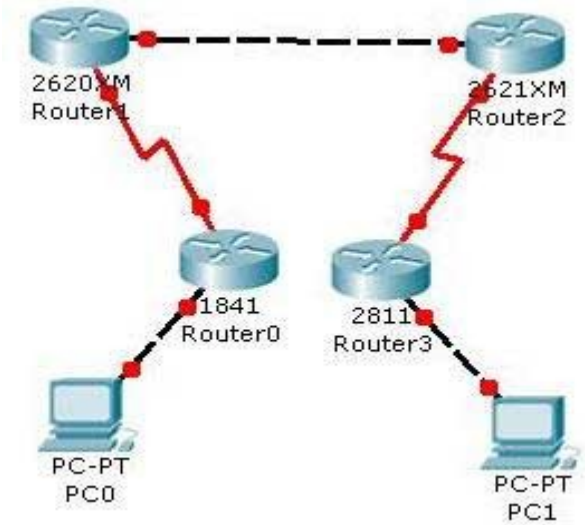
PC-2

```
PC>ipconfig

IP Address.....................: 50.0.0.2
Subnet Mask....................: 255.0.0.0
Default Gateway................: 50.0.0.1

PC>ping 10.0.0.2

Pinging 10.0.0.2 with 32 bytes of data:

Reply from 10.0.0.2: bytes=32 time=140ms TTL=124
Reply from 10.0.0.2: bytes=32 time=141ms TTL=124
Reply from 10.0.0.2: bytes=32 time=157ms TTL=124
Reply from 10.0.0.2: bytes=32 time=156ms TTL=124

Ping statistics for 10.0.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 140ms, Maximum = 157ms, Average = 148ms
```

To test static routing do ping from pc1 to pc2 and vice versa. If we get reply then you have successfully configured static routing but if we did not get reply double check this configuration and try to troubleshoot.

2. RIP

In this article we will demonstrate an example of **Rip Routing** configurations. We will use four different series router so you can get familiar with all different platform Create a topology as shown in figure.



IP RIP comes in two different versions: 1 and 2. Version 1 is a distance vector

protocol and is defined in RFC 1058. Version 2 is a hybrid protocol and is defined in RFCs 1721 and 1722. There are no major differences between RIPv1 or RIPv2 so far configurations concern.

1841 Series Router0 (R1)		
	FastEthernet0/0	Serial0/0/0
IP address	10.0.0.1	20.0.0.1
Connected With	Pc0	R2 on Serial 0/0
2811 Series Router0 (R4)		
	FastEthernet0/0	Serial0/0/0
IP address	50.0.0.1	40.0.0.2
Connected With	Pc1	R3 on Serial 0/0
2621XM Series Router0 (R3)		
	FastEthernet0/0	Serial0/0/0
IP address	30.0.0.2	40.0.0.1
Connected With	FastEthernet0/0	R4 on Serial 0/0/0
2620XM Series Router1 (R2)		
	FastEthernet0/0	Serial0/0
IP address	30.0.0.1	20.0.0.2
Connected With	R3 on FastEthernet0/0	R1 on Serial 0/0/0
PC-PT PC0		
	FastEthernet0	Default Gateway
IP address	10.0.0.2	10.0.0.1
Connected With	R1 on FastEthernet0/0	

PC-PT PC1		
	FastEthernet0	Default Gateway
IP address	50.0.0.2	50.0.0.1
Connected With	R4 on FastEthernet0/0	

To configure any router double click on it and select **CLI**. To configure this topology use this step by step guide.

(1841Router0) Hostname R1

To configure and enable rip routing on R1 follow these commands exactly.

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#hostname R1
R1(config)#interface fastethernet 0/0
R1(config-if)#ip address 10.0.0.1 255.0.0.0
R1(config-if)#no shutdown
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0,
changed state to up
R1(config-if)#exit
R1(config)#interface serial 0/0/0
R1(config-if)#ip address 20.0.0.1 255.0.0.0
R1(config-if)#clock rate 64000
R1(config-if)#bandwidth 64
R1(config-if)#no shutdown
%LINK-5-CHANGED: Interface Serial0/0/0, changed state to down
R1(config-if)#exit
%LINK-5-CHANGED: Interface Serial0/0/0, changed state to up
R1(config)#router rip
R1(config-router)#network 10.0.0.0
R1(config-router)#network 20.0.0.0
R1(config-router)#exit
R1(config)#
```

(2620XM-Router1) Hostname R2

To configure and enable rip routing on R2 follow these commands exactly.

```
Router>enable
```

```
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#hostname R2
R2(config)#interface serial 0/0
R2(config-if)#ip address 20.0.0.2 255.0.0.0
R2(config-if)#no shutdown
%LINK-5-CHANGED: Interface Serial0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0, changed
state to up
R2(config-if)#exit
R2(config)#interface fastethernet 0/0
R2(config-if)#ip address 30.0.0.1 255.0.0.0
R2(config-if)#no shutdown
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up
R2(config-if)#exit
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0,
changed state to up
R2(config)#router rip
R2(config-router)#network 20.0.0.0
R2(config-router)#network 30.0.0.0
R2(config-router)#exit
R2(config)#
```

(2620XM-Router2)Hostname R3

To configure and enable rip routing on R3 follow these commands exactly.

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#hostname R3
R3(config)#interface fastethernet 0/0
R3(config-if)#ip address 30.0.0.2 255.0.0.0
R3(config-if)#no shutdown
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0,
changed state to up
R3(config-if)#interface serial 0/0
R3(config-if)#ip address 40.0.0.1 255.0.0.0
R3(config-if)#clock rate 64000
R3(config-if)#bandwidth 64
R3(config-if)#no shutdown
%LINK-5-CHANGED: Interface Serial0/0, changed state to down
```

```
R3(config-if)#exit
%LINK-5-CHANGED: Interface Serial0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0, changed
state to up
R3(config)#router rip
R3(config-router)#network 30.0.0.0
R3(config-router)#network 40.0.0.0
R3(config-router)#exit
R3(config)#
```

(2811Router3) Hostname R4

To configure and enable rip routing on R4 follow these commands exactly.

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface serial 0/0/0
Router(config-if)#ip address 40.0.0.2 255.0.0.0
Router(config-if)#no shutdown
%LINK-5-CHANGED: Interface Serial0/0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0/0,
changed state to up
Router(config-if)#exit
Router(config)#interface fastethernet 0/0
Router(config-if)#ip address 50.0.0.1 255.0.0.0
Router(config-if)#no shutdown
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0,
changed state to up
Router(config-if)#exit
R4(config)#router rip
R4(config-router)#network 40.0.0.0
R4(config-router)#network 50.0.0.0
R4(config-router)#exit
R4(config)#
```

PC-1

```
PC>ipconfig

IP Address.....................: 10.0.0.2
Subnet Mask....................: 255.0.0.0
Default Gateway................: 10.0.0.1
```

```
PC>ping 50.0.0.2

Pinging 50.0.0.2 with 32 bytes of data:

Reply from 50.0.0.2: bytes=32 time=156ms TTL=124
Reply from 50.0.0.2: bytes=32 time=127ms TTL=124
Reply from 50.0.0.2: bytes=32 time=156ms TTL=124
Reply from 50.0.0.2: bytes=32 time=140ms TTL=124

Ping statistics for 50.0.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 127ms, Maximum = 156ms, Average = 144ms
PC>
```

PC-2

```
PC>ipconfig

IP Address.....................: 50.0.0.2
Subnet Mask....................: 255.0.0.0
Default Gateway................: 50.0.0.1

PC>ping 10.0.0.2

Pinging 10.0.0.2 with 32 bytes of data:

Reply from 10.0.0.2: bytes=32 time=140ms TTL=124
Reply from 10.0.0.2: bytes=32 time=141ms TTL=124
Reply from 10.0.0.2: bytes=32 time=157ms TTL=124
Reply from 10.0.0.2: bytes=32 time=156ms TTL=124

Ping statistics for 10.0.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 140ms, Maximum = 157ms, Average = 148ms
```

We can verify that RIP is running successfully via **show ip protocols** command in privilege mode.

```
R1#show ip protocols
Routing Protocol is "rip"
Sending updates every 30 seconds, next due in 2 seconds
Invalid after 180 seconds, hold down 180, flushed after 240
```

```
Outgoing update filter list for all interfaces is not set
Incoming update filter list for all interfaces is not set
Redistributing: rip
Default version control: send version 1, receive any version
  Interface             Send  Recv  Triggered RIP  Key-chain
  FastEthernet0/0       1     2 1
  Serial0/0/0           1     2 1
Automatic network summarization is in effect
Maximum path: 4
Routing for Networks:
      10.0.0.0
      20.0.0.0
Passive Interface(s):
Routing Information Sources:
      Gateway         Distance      Last Update
      20.0.0.2             120      00:00:20
Distance: (default is 120)
R1#
```

You can use **show ip route** command to troubleshoot rip network. If you did not see information about any route checks the router attached with that network.

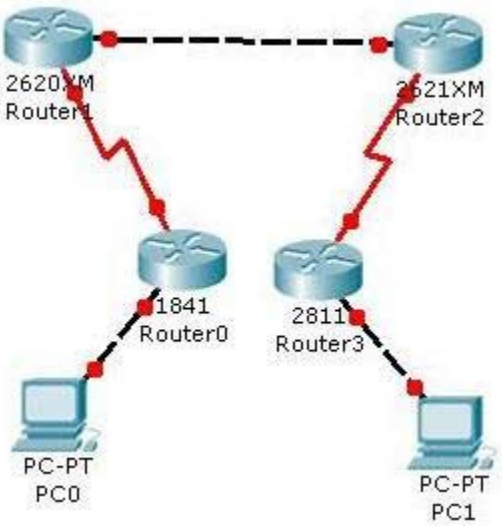
```
R1#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B
- BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter
area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type
2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E -
EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-
IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

C    10.0.0.0/8 is directly connected, FastEthernet0/0
C    20.0.0.0/8 is directly connected, Serial0/0/0
R    30.0.0.0/8 [120/1] via 20.0.0.2, 00:00:01, Serial0/0/0
R    40.0.0.0/8 [120/2] via 20.0.0.2, 00:00:01, Serial0/0/0
```

```
R    50.0.0.0/8 [120/3] via 20.0.0.2, 00:00:01, Serial0/0/0
R1#
```

To test rip routing do ping from pc1 to pc2 and vice versa. If we get reply then you have successfully configured rip routing but if we did not get replay double check this configuration and try to troubleshoot.

2. EIGRP

- **EIGRP** is a Cisco-proprietary routing protocol for TCP/IP. It's actually based on Cisco's proprietary IGRP routing protocol, with many enhancements built into it. Because it has its roots in IGRP, the configuration is similar to IGRP; however, it has many link state characteristics that were added to it to allow EIGRP to scale to enterprise network sizes. To know these characteristics read our previous article.
In this article I will demonstrate an example of **EIGRP Routing** configurations. Create a topology as shown in figure.



1841 Series Router0 (R1)		
	FastEthernet0/0	Serial0/0/0
IP address	10.0.0.1	20.0.0.1
Connected With	Pc0	R2 on Serial 0/0
2811 Series Router0 (R4)		
	FastEthernet0/0	Serial0/0/0
IP address	50.0.0.1	40.0.0.2

Connected With	Pc1	R3 on Serial 0/0
2621XM Series Router0 (R3)		
	FastEthernet0/0	Serial0/0/0
IP address	30.0.0.2	40.0.0.1
Connected With	FastEthernet0/0	R4 on Serial 0/0/0
2620XM Series Router1 (R2)		
	FastEthernet0/0	Serial0/0
IP address	30.0.0.1	20.0.0.2
Connected With	R3 on FastEthernet0/0	R1 on Serial 0/0/0
PC-PT PC0		
	FastEthernet0	Default Gateway
IP address	10.0.0.2	10.0.0.1
Connected With	R1 on FastEthernet0/0	
PC-PT PC1		
	FastEthernet0	Default Gateway
IP address	50.0.0.2	50.0.0.1
Connected With	R4 on FastEthernet0/0	

To configure any router double click on it and select CLI.To configure this topology use this step by step guide.

(1841Router0) Hostname R1

To configure and enable eigrp routing on R1 follow these commands exactly.

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
```

```
Router(config)#hostname R1
R1(config)#interface fastethernet 0/0
R1(config-if)#ip address 10.0.0.1 255.0.0.0
R1(config-if)#no shutdown
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0,
changed state to up
R1(config-if)#exit
R1(config)#interface serial 0/0/0
R1(config-if)#ip address 20.0.0.1 255.0.0.0
R1(config-if)#clock rate 64000
R1(config-if)#bandwidth 64
R1(config-if)#no shutdown
%LINK-5-CHANGED: Interface Serial0/0/0, changed state to down
R1(config-if)#exit
%LINK-5-CHANGED: Interface Serial0/0/0, changed state to up
R1(config)#router eigrp 1
R1(config-router)#network 10.0.0.0
R1(config-router)#network 20.0.0.0
R1(config-router)#exit
R1(config)#
```

(2620XM-Router1) Hostname R2

To configure and enable eigrp routing on R2 follow these commands exactly.

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#hostname R2
R2(config)#interface serial 0/0
R2(config-if)#ip address 20.0.0.2 255.0.0.0
R2(config-if)#no shutdown
%LINK-5-CHANGED: Interface Serial0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0, changed
state to up
R2(config-if)#exit
R2(config)#interface fastethernet 0/0
R2(config-if)#ip address 30.0.0.1 255.0.0.0
R2(config-if)#no shutdown
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up
R2(config-if)#exit
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0,
```

```
changed state to up
R2(config)#router eigrp 1
R2(config-router)#network 20.0.0.0
R2(config-router)#network 30.0.0.0
R2(config-router)#exit
R2(config)#
```

(2620XM-Router2)Hostname R3

To configure and enable eigrp routing on R3 follow these commands exactly.

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#hostname R3
R3(config)#interface fastethernet 0/0
R3(config-if)#ip address 30.0.0.2 255.0.0.0
R3(config-if)#no shutdown
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0,
changed state to up
R3(config-if)#interface serial 0/0
R3(config-if)#ip address 40.0.0.1 255.0.0.0
R3(config-if)#clock rate 64000
R3(config-if)#bandwidth 64
R3(config-if)#no shutdown
%LINK-5-CHANGED: Interface Serial0/0, changed state to down
R3(config-if)#exit
%LINK-5-CHANGED: Interface Serial0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0, changed
state to up
R3(config)#router eigrp 1
R3(config-router)#network 30.0.0.0
R3(config-router)#network 40.0.0.0
R3(config-router)#exit
R3(config)#
```

(2811Router3) Hostname R4

To configure and enable eigrp routing on R4 follow these commands exactly.

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface serial 0/0/0
Router(config-if)#ip address 40.0.0.2 255.0.0.0
```

```
Router(config-if)#no shutdown
%LINK-5-CHANGED: Interface Serial0/0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0/0,
changed state to up
Router(config-if)#exit
Router(config)#interface fastethernet 0/0
Router(config-if)#ip address 50.0.0.1 255.0.0.0
Router(config-if)#no shutdown
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0,
changed state to up
Router(config-if)#exit
R3(config)#router eigrp 1
R3(config-router)#network 30.0.0.0
R3(config-router)#network 40.0.0.0
R3(config-router)#exit
R3(config)#
```

PC-1

```
PC>ipconfig

IP Address.....................: 10.0.0.2
Subnet Mask....................: 255.0.0.0
Default Gateway................: 10.0.0.1

PC>ping 50.0.0.2

Pinging 50.0.0.2 with 32 bytes of data:

Reply from 50.0.0.2: bytes=32 time=156ms TTL=124
Reply from 50.0.0.2: bytes=32 time=127ms TTL=124
Reply from 50.0.0.2: bytes=32 time=156ms TTL=124
Reply from 50.0.0.2: bytes=32 time=140ms TTL=124

Ping statistics for 50.0.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 127ms, Maximum = 156ms, Average = 144ms
PC>
```

PC-2

```
PC>ipconfig
```

```
IP Address.....................: 50.0.0.2
Subnet Mask....................: 255.0.0.0
Default Gateway................: 50.0.0.1

PC>ping 10.0.0.2

Pinging 10.0.0.2 with 32 bytes of data:

Reply from 10.0.0.2: bytes=32 time=140ms TTL=124
Reply from 10.0.0.2: bytes=32 time=141ms TTL=124
Reply from 10.0.0.2: bytes=32 time=157ms TTL=124
Reply from 10.0.0.2: bytes=32 time=156ms TTL=124

Ping statistics for 10.0.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 140ms, Maximum = 157ms, Average = 148ms
```

You can verify that eigrp is running successfully via **show ip protocols** command in privilege mode.

```
R4#show ip protocols

Routing Protocol is "ospf 4"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Router ID 50.0.0.1
  Number of areas in this router is 1. 1 normal 0 stub 0 nssa
  Maximum path: 4
  Routing for Networks:
    50.0.0.0 0.255.255.255 area 0
    40.0.0.0 0.255.255.255 area 0
  Routing Information Sources:
    Gateway         Distance      Last Update
    40.0.0.1             110      00:01:26
  Distance: (default is 110)

R4#
```

You can use **show ip route** command to troubleshoot eigrp network. If you did not see information about any route checks the router attached with that network.

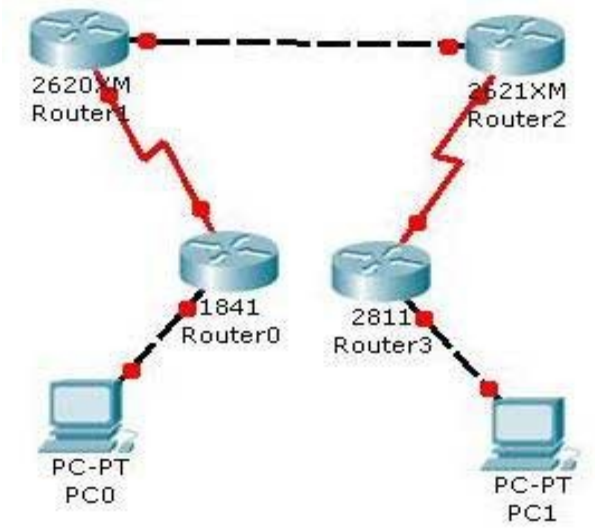
```
R4#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B
```

```
- BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter
area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type
2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E -
EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-
IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

O    10.0.0.0/8 [110/1564] via 40.0.0.1, 00:02:37, Serial0/0/0
O    20.0.0.0/8 [110/1563] via 40.0.0.1, 00:02:37, Serial0/0/0
O    30.0.0.0/8 [110/782] via 40.0.0.1, 00:02:37, Serial0/0/0
C    40.0.0.0/8 is directly connected, Serial0/0/0
C    50.0.0.0/8 is directly connected, FastEthernet0/0
R4#
```

To test eigrp routing do ping from pc1 to pc2 and vice versa. If you get replay then you have successfully configured eigrp routing but if you did not get replay double check this configuration and try to troubleshoot.

3. OSPF

In this article we will demonstrate an example of **OSPF Routing** configuration.Create a topology as shown in figure.



1841 Series Router0 (R1)		
	FastEthernet0/0	Serial0/0/0
IP address	10.0.0.1	20.0.0.1
Connected With	Pc0	R2 on Serial 0/0
2811 Series Router0 (R4)		
	FastEthernet0/0	Serial0/0/0
IP address	50.0.0.1	40.0.0.2
Connected With	Pc1	R3 on Serial 0/0

2621XM Series Router0 (R3)		
	FastEthernet0/0	Serial0/0/0
IP address	30.0.0.2	40.0.0.1
Connected With	FastEthernet0/0	R4 on Serial 0/0/0
2620XM Series Router1 (R2)		
	FastEthernet0/0	Serial0/0
IP address	30.0.0.1	20.0.0.2
Connected With	R3 on FastEthernet0/0	R1 on Serial 0/0/0

PC-PT PC0		
	FastEthernet0	Default Gateway
IP address	10.0.0.2	10.0.0.1
Connected With	R1 on FastEthernet0/0	
PC-PT PC1		
	FastEthernet0	Default

		Gateway
IP address	50.0.0.2	50.0.0.1
Connected With	R4 on FastEthernet0/0	

Configuring OSPF is slightly different from configuring RIP. When configuring OSPF, use the following syntax:

```
Router(config)# router ospf process_ID
Router(config-router)# network IP_address wildcard_mask area area_#
```

The process_ID is locally significant and is used to differentiate between OSPF processes running on the same router. Your router might be a boundary router between two OSPF autonomous systems, and to differentiate them on your router, you will give them unique process IDs. Note that these numbers do not need to match between different routers so they have nothing to do with autonomous system numbers.

To configure any router double click on it and select CLI.To configure this topology use this step by step guide.

(1841Router0) Hostname R1

To configure and enable ospf routing on R1 follow these commands exactly.

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#hostname R1
R1(config)#interface fastethernet 0/0
R1(config-if)#ip address 10.0.0.1 255.0.0.0
R1(config-if)#no shutdown
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0,
changed state to up
R1(config-if)#exit
R1(config)#interface serial 0/0/0
R1(config-if)#ip address 20.0.0.1 255.0.0.0
R1(config-if)#clock rate 64000
R1(config-if)#bandwidth 64
R1(config-if)#no shutdown
%LINK-5-CHANGED: Interface Serial0/0/0, changed state to down
R1(config-if)#exit
%LINK-5-CHANGED: Interface Serial0/0/0, changed state to up
R1(config)#router ospf 1
```

```
R1(config-router)#network 10.0.0.0 0.255.255.255 area 0
R1(config-router)#network 20.0.0.0 0.255.255.255 area 0
R1(config-router)#exit
R1(config)#
```

(2620XM-Router1) Hostname R2

To configure and enable ospf routing on R2 follow these commands exactly.

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#hostname R2
R2(config)#interface serial 0/0
R2(config-if)#ip address 20.0.0.2 255.0.0.0
R2(config-if)#no shutdown
%LINK-5-CHANGED: Interface Serial0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0, changed
state to up
R2(config-if)#exit
R2(config)#interface fastethernet 0/0
R2(config-if)#ip address 30.0.0.1 255.0.0.0
R2(config-if)#no shutdown
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up
R2(config-if)#exit
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0,
changed state to up
R2(config)#router ospf 2
R2(config-router)#network 20.0.0.0 0.255.255.255 area 0
R2(config-router)#network 3
00:03:10: %OSPF-5-ADJCHG: Process 2, Nbr 20.0.0.1 on Serial0/0 from
 LOADING to FULL, Loading Done0.0.0.0 0.255.255.255 area 0
R2(config-router)#network 30.0.0.0 0.255.255.255 area 0
R2(config-router)#exit
R2(config)#
```

(2620XM-Router2)Hostname R3

To configure and enable ospf routing on R3 follow these commands exactly.

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#hostname R3
R3(config)#interface fastethernet 0/0
```

```
R3(config-if)#ip address 30.0.0.2 255.0.0.0
R3(config-if)#no shutdown
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0,
changed state to up
R3(config-if)#interface serial 0/0
R3(config-if)#ip address 40.0.0.1 255.0.0.0
R3(config-if)#clock rate 64000
R3(config-if)#bandwidth 64
R3(config-if)#no shutdown
%LINK-5-CHANGED: Interface Serial0/0, changed state to down
R3(config-if)#exit
%LINK-5-CHANGED: Interface Serial0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0, changed
state to up
R3(config)#router ospf 3
R3(config-router)#network 40.0.0.0 0.255.255.255 area 0
R3(config-router)#network 30.0.0.0 0.255.255.255 area 0
00:04:53: %OSPF-5-ADJCHG: Process 3, Nbr 30.0.0.1 on
FastEthernet0/0 from
 LOADING to FULL, Loading D
R3(config-router)#exit
R3(config)#
%SYS-5-CONFIG_I: Configured from console by console
R3#
```

(2811Router3) Hostname R4

To configure and enable ospf routing on R4 follow these commands exactly.

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface serial 0/0/0
Router(config-if)#ip address 40.0.0.2 255.0.0.0
Router(config-if)#no shutdown
%LINK-5-CHANGED: Interface Serial0/0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0/0,
changed state to up
Router(config-if)#exit
Router(config)#interface fastethernet 0/0
Router(config-if)#ip address 50.0.0.1 255.0.0.0
```

```
Router(config-if)#no shutdown
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0,
changed state to up
Router(config-if)#exit
R4(config)#router ospf 4
R4(config-router)#network 50.0.0.0 0.255.255.255 area 0
R4(config-router)#network 40.0.0.0 0.255.255.255 area 0
R4(config-router)#
00:06:32: %OSPF-5-ADJCHG: Process 4, Nbr 40.0.0.1 on Serial0/0/0
from
LOADING to FULL, Loading Done
R4(config-router)#exit
R4(config)#
```

PC-1

```
PC>ipconfig

IP Address.....................: 10.0.0.2
Subnet Mask....................: 255.0.0.0
Default Gateway................: 10.0.0.1

PC>ping 50.0.0.2

Pinging 50.0.0.2 with 32 bytes of data:

Reply from 50.0.0.2: bytes=32 time=156ms TTL=124
Reply from 50.0.0.2: bytes=32 time=127ms TTL=124
Reply from 50.0.0.2: bytes=32 time=156ms TTL=124
Reply from 50.0.0.2: bytes=32 time=140ms TTL=124

Ping statistics for 50.0.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 127ms, Maximum = 156ms, Average = 144ms
PC>
```

PC-2

```
PC>ipconfig

IP Address.....................: 50.0.0.2
```

```
Subnet Mask.....................: 255.0.0.0
Default Gateway.................: 50.0.0.1

PC>ping 10.0.0.2

Pinging 10.0.0.2 with 32 bytes of data:

Reply from 10.0.0.2: bytes=32 time=140ms TTL=124
Reply from 10.0.0.2: bytes=32 time=141ms TTL=124
Reply from 10.0.0.2: bytes=32 time=157ms TTL=124
Reply from 10.0.0.2: bytes=32 time=156ms TTL=124

Ping statistics for 10.0.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 140ms, Maximum = 157ms, Average = 148ms
```

You can verify that ospf is running successfully via **show ip protocols** command in privilege mode.

```
R4#show ip protocols

Routing Protocol is "ospf 4"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Router ID 50.0.0.1
  Number of areas in this router is 1. 1 normal 0 stub 0 nssa
  Maximum path: 4
  Routing for Networks:
    50.0.0.0 0.255.255.255 area 0
    40.0.0.0 0.255.255.255 area 0
  Routing Information Sources:
    Gateway         Distance      Last Update
    40.0.0.1             110      00:01:26
  Distance: (default is 110)

R4#
```

You can use **show ip oute** command to troubleshoot ospf network. If you did not see information about any route checks the router attached with that network.

```
R4#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B
- BGP
```

```
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter
area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type
2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E -
EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-
IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

O    10.0.0.0/8 [110/1564] via 40.0.0.1, 00:02:37, Serial0/0/0
O    20.0.0.0/8 [110/1563] via 40.0.0.1, 00:02:37, Serial0/0/0
O    30.0.0.0/8 [110/782] via 40.0.0.1, 00:02:37, Serial0/0/0
C    40.0.0.0/8 is directly connected, Serial0/0/0
C    50.0.0.0/8 is directly connected, FastEthernet0/0
R4#
```

To test ospf routing do ping from pc1 to pc2 and vice versa. If you get replay then you have successfully configured ospf routing but if you did not get replay double check this configuration and try to troubleshoot. I have uploaded a configured and tested topology in case you are unable to locate the problem spot then download this configuration file. And try to find out where have you committed mistake

CONCLUSION: Thus we have simulated RIP, OSPF, EIGRP routing protocols

.

AIM: Using a Network Simulator Configure Using a Network Simulator Configure VLAN, OSPF and NAT

OBJECTIVE : To study,

VLAN,

OSPF and

NAT

THEORY:

THEORY:

VLAN: In simple terms, a VLAN is a set of workstations within a LAN that can communicate with each other as though they were on a single, isolated LAN.

The purpose of VLANs

The basic reason for splitting a network into VLANs is to reduce congestion on a large LAN.

Purpose of VLAN

Initially LANs were very flat—all the workstations were connected to a single piece of coaxial cable, or to sets of chained hubs. In a flat LAN, every packet that any device puts onto the wire gets sent to **every** other device on the LAN.

As the number of workstations on the typical LAN grew, they started to become hopelessly congested; there were just too many collisions, because most of the time when a workstation tried to send a packet, it would find that the wire was already occupied by a packet sent by some other device.

Redistribution of between multiple protocols

It is the way how routers exchange routing information if **two or more different protocols are interconnected**. Simply while running multiple routing protocols on same network. For example a company that is running EIGRP and you just bring another company and their network is running RIP, such conditions are solved by Redistribution. Actually we are going to **MIX different protocols** by redistribution command.

WEP:- Short for Wired Equivalent Privacy (or Wireless Encryption Protocol), WEP is part of the IEEE 802.11 wireless networking standard and was designed to provide the same level of security as that of a wired LAN. Because wireless networks broadcast messages using radio, they are susceptible to eavesdropping. WEP provides security by encrypting data over radio waves so that it is protected as it is transmitted from one end point to another.

Step 5: In laptop's configuration go to PC wireless, then click on connect, connect, and give the WEP as shown, it will the wireless connection as shown below. Check the LAN by pinging each other.

Simulation:

Step 1: Create 3 different VLANs Engg, Acc and Mgt. Below figure shows the IP configuration of VLAN 10 ie Engg. Similarly Create 2 other VLANs. VLAN 20 and 30

VLAN 20

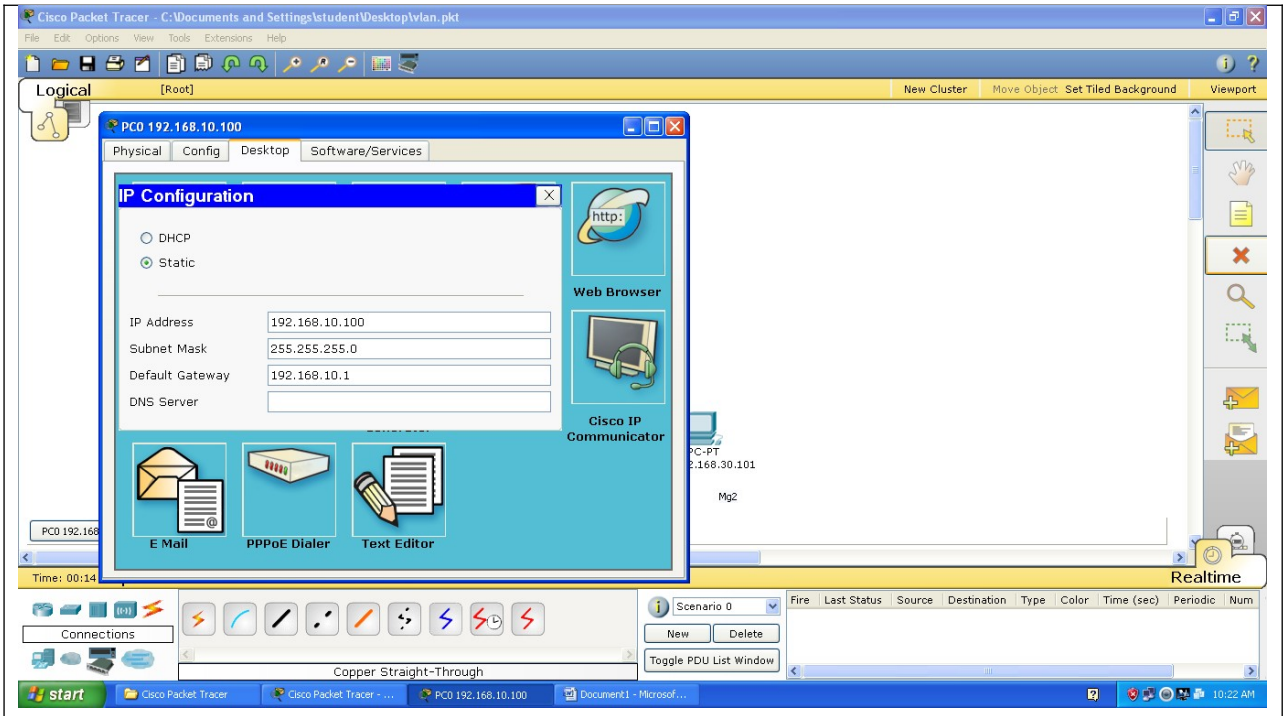
IP Address:192.168.20.100

Gateway:192.168.20.1

VLAN 30

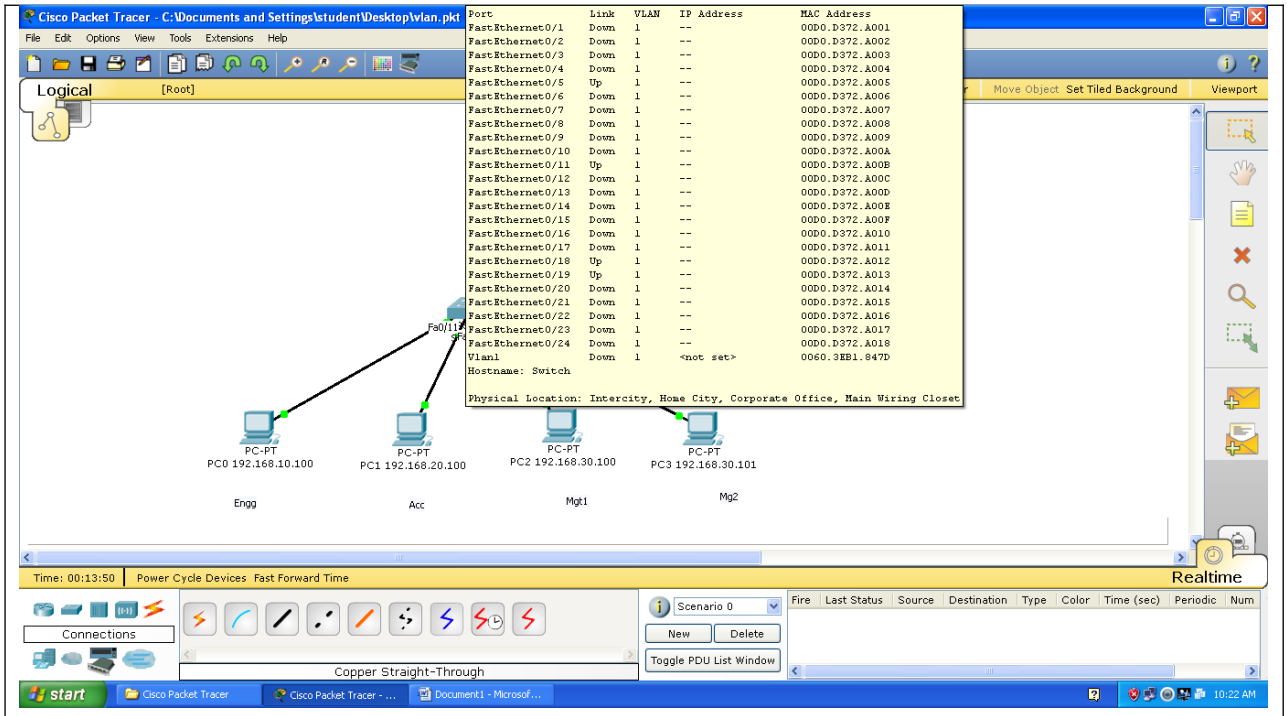
IP Address:192.168.30.100 and 192.168.30.101

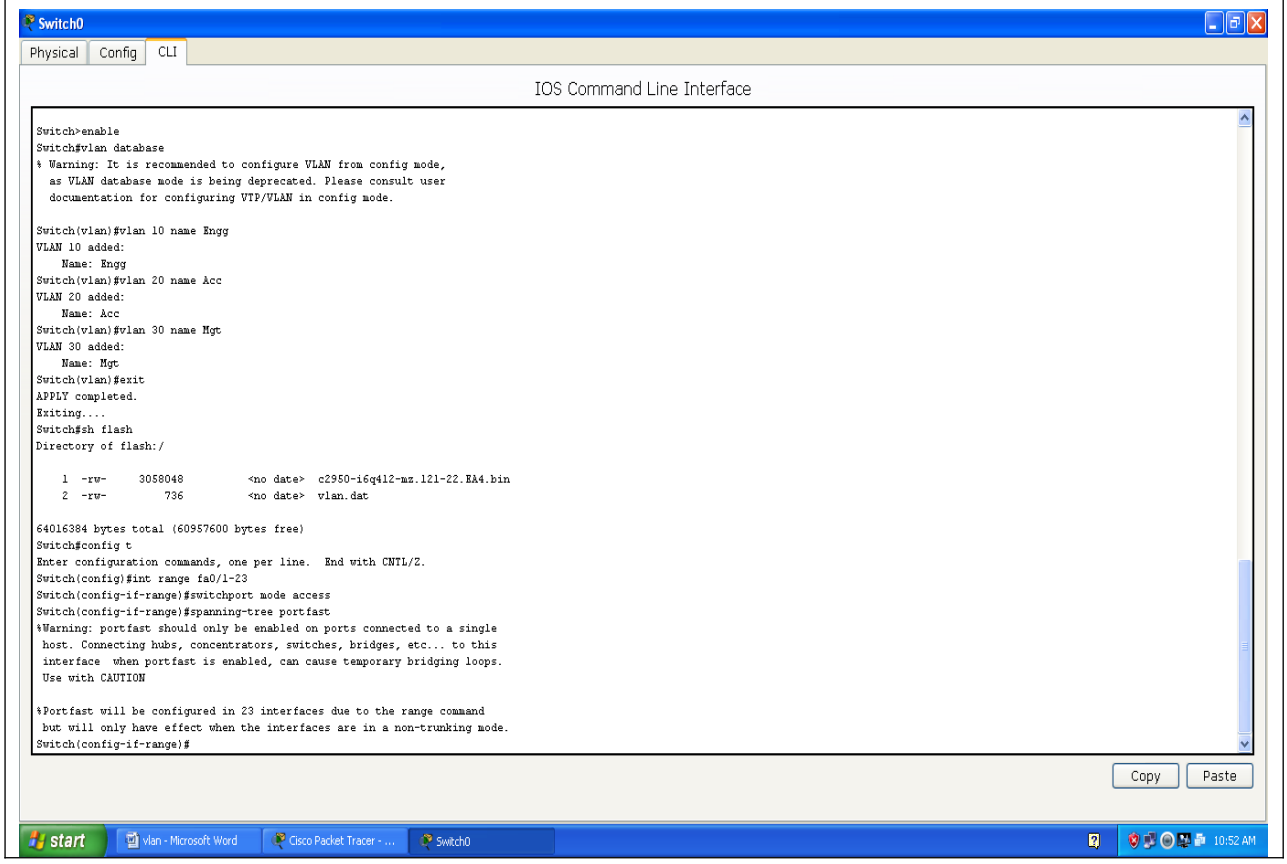
Gateway:192.168.30.1

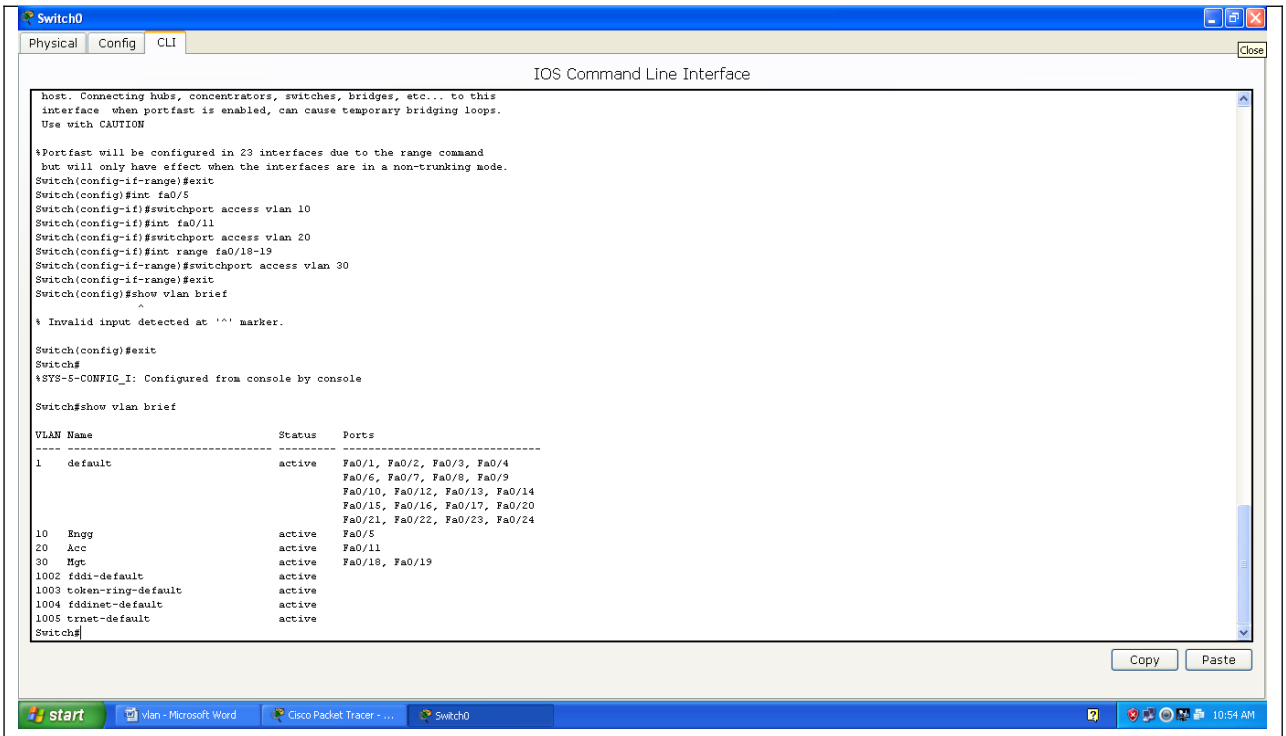


Step2 :Configuration of switch

A. Steps are given in the snapshots. In switch ports 5, 11,18 and 19 are used to connect different VLANs.

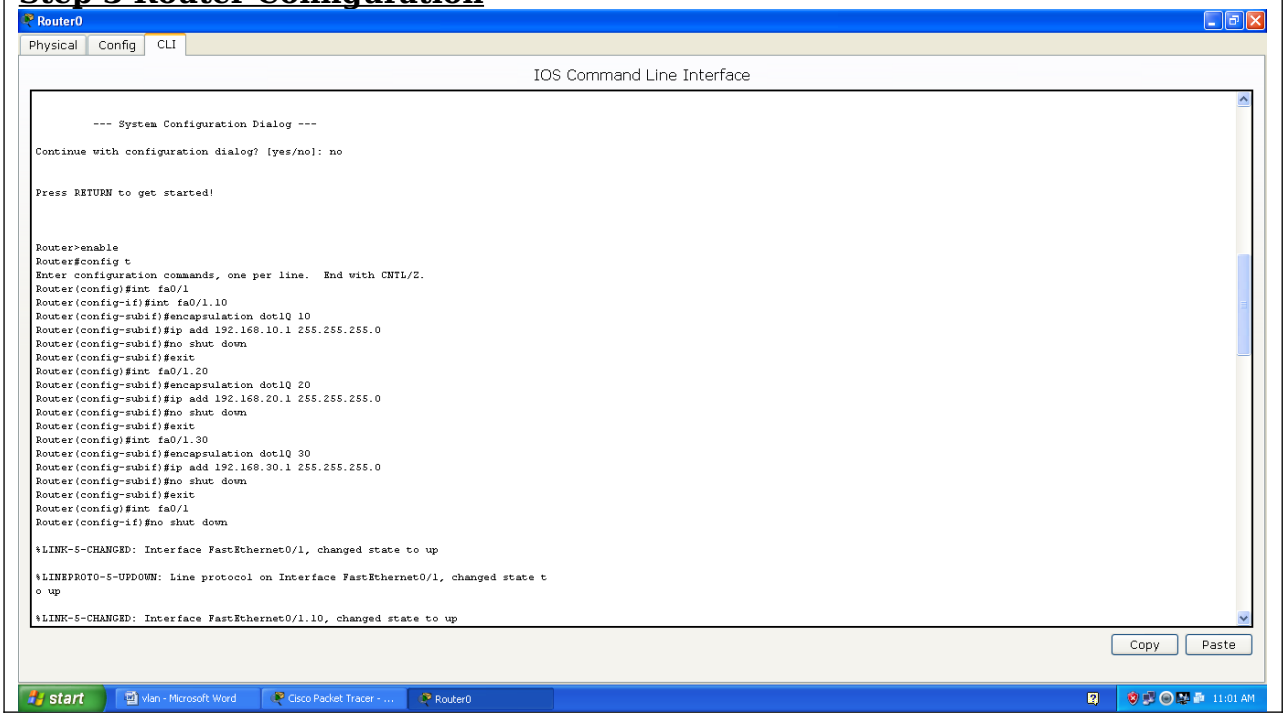

Cisco Packet Tracer - C:\Documents and Settings\student\Desktop\vlan.pkt
File Edit Options View Tools Extensions Help
Logical [Root]
Move Object Set Tiled Background Viewport
Port Link VLAN IP Address MAC Address
FastEthernet0/1 Down 1 -- 00D0.D372.A001
FastEthernet0/2 Down 1 -- 00D0.D372.A002
FastEthernet0/3 Down 1 -- 00D0.D372.A003
FastEthernet0/4 Down 1 -- 00D0.D372.A004
FastEthernet0/5 Up 1 -- 00D0.D372.A005
FastEthernet0/6 Down 1 -- 00D0.D372.A006
FastEthernet0/7 Down 1 -- 00D0.D372.A007
FastEthernet0/8 Down 1 -- 00D0.D372.A008
FastEthernet0/9 Down 1 -- 00D0.D372.A009
FastEthernet0/10 Down 1 -- 00D0.D372.A00A
FastEthernet0/11 Up 1 -- 00D0.D372.A00B
FastEthernet0/12 Down 1 -- 00D0.D372.A00C
FastEthernet0/13 Down 1 -- 00D0.D372.A00D
FastEthernet0/14 Down 1 -- 00D0.D372.A00E
FastEthernet0/15 Down 1 -- 00D0.D372.A00F
FastEthernet0/16 Down 1 -- 00D0.D372.A010
FastEthernet0/17 Down 1 -- 00D0.D372.A011
FastEthernet0/18 Up 1 -- 00D0.D372.A012
FastEthernet0/19 Up 1 -- 00D0.D372.A013
FastEthernet0/20 Down 1 -- 00D0.D372.A014
FastEthernet0/21 Down 1 -- 00D0.D372.A015
FastEthernet0/22 Down 1 -- 00D0.D372.A016
FastEthernet0/23 Down 1 -- 00D0.D372.A017
FastEthernet0/24 Down 1 -- 00D0.D372.A018
Vlan1 Down 1 <not set> 0060.3EB1.847D
Hostname: Switch
Physical Location: Intercity, Home City, Corporate Office, Main Wiring Closet
PC-PT PC0 192.168.10.100 Engg
PC-PT PC1 192.168.20.100 Acc
PC-PT PC2 192.168.30.100 Mgt1
PC-PT PC3 192.168.30.101 Mgt2
Time: 00:13:50 Power Cycle Devices Fast Forward Time
Realtime
Connections
Copper Straight-Through
Scenario 0 New Delete Toggle PDU List Window
Fire Last Status Source Destination Type Color Time (sec) Periodic Num
start Cisco Packet Tracer Cisco Packet Tracer - ... Document1 - Microsof... 10:22 AM

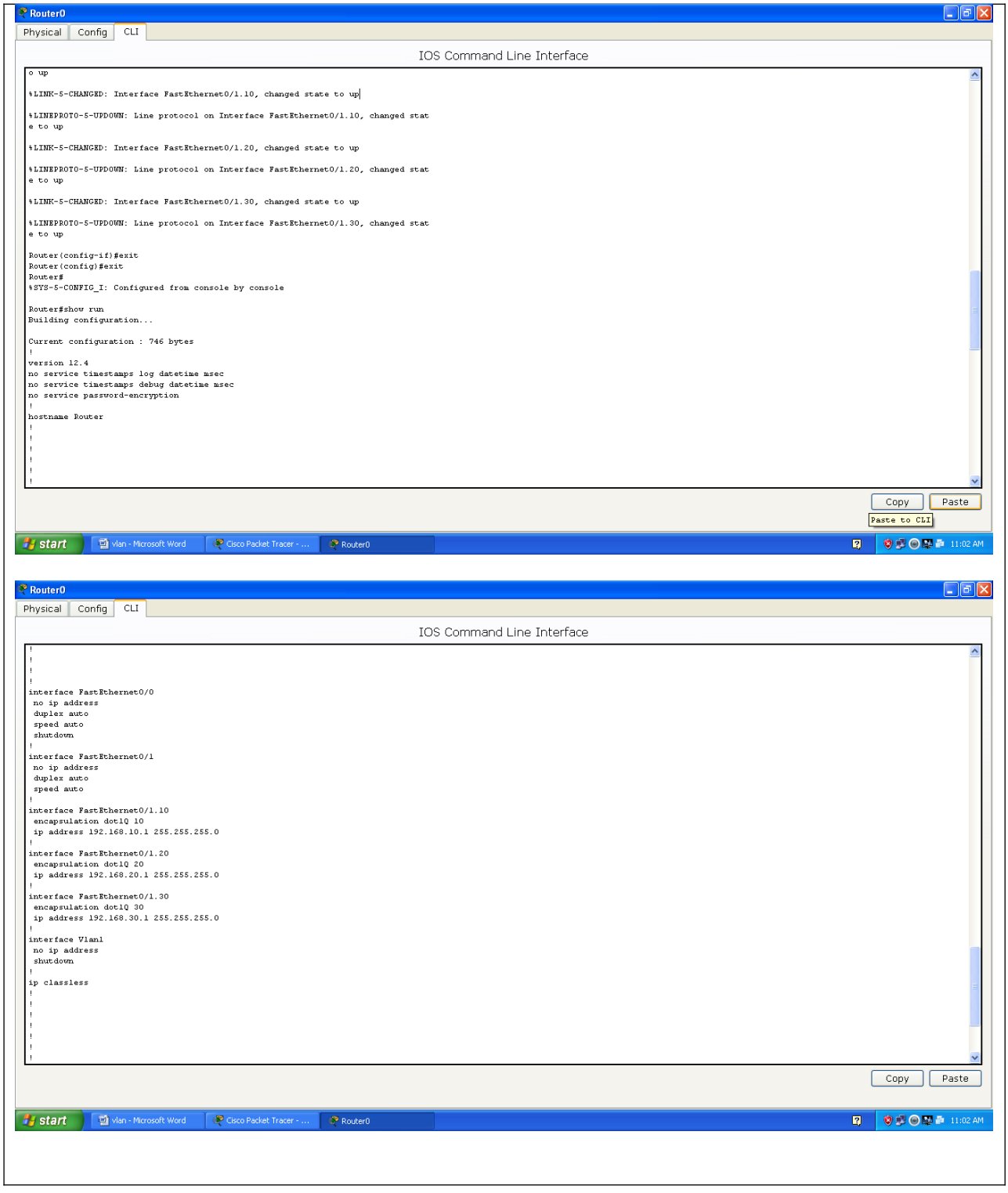

Switch0
Physical Config CLI
IOS Command Line Interface
Switch>enable
Switch#vlan database
% Warning: It is recommended to configure VLAN from config mode,
as VLAN database mode is being deprecated. Please consult user
documentation for configuring VTP/VLAN in config mode.
Switch(vlan)#vlan 10 name Engg
VLAN 10 added:
Name: Engg
Switch(vlan)#vlan 20 name Acc
VLAN 20 added:
Name: Acc
Switch(vlan)#vlan 30 name Mgt
VLAN 30 added:
Name: Mgt
Switch(vlan)#exit
APPLY completed.
Exiting....
Switch#sh flash
Directory of flash:/
1 -rw- 3058048 <no date> c2950-i6q412-mz.121-22.EA4.bin
2 -rw- 736 <no date> vlan.dat
64016384 bytes total (60957600 bytes free)
Switch#config t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#int range fa0/1-23
Switch(config-if-range)#switchport mode access
Switch(config-if-range)#spanning-tree portfast
%Warning: portfast should only be enabled on ports connected to a single
host. Connecting hubs, concentrators, switches, bridges, etc... to this
interface when portfast is enabled, can cause temporary bridging loops.
Use with CAUTION
%Portfast will be configured in 23 interfaces due to the range command
but will only have effect when the interfaces are in a non-trunking mode.
Switch(config-if-range)#
Copy Paste
start vlan - Microsoft Word Cisco Packet Tracer - ... Switch0 10:52 AM

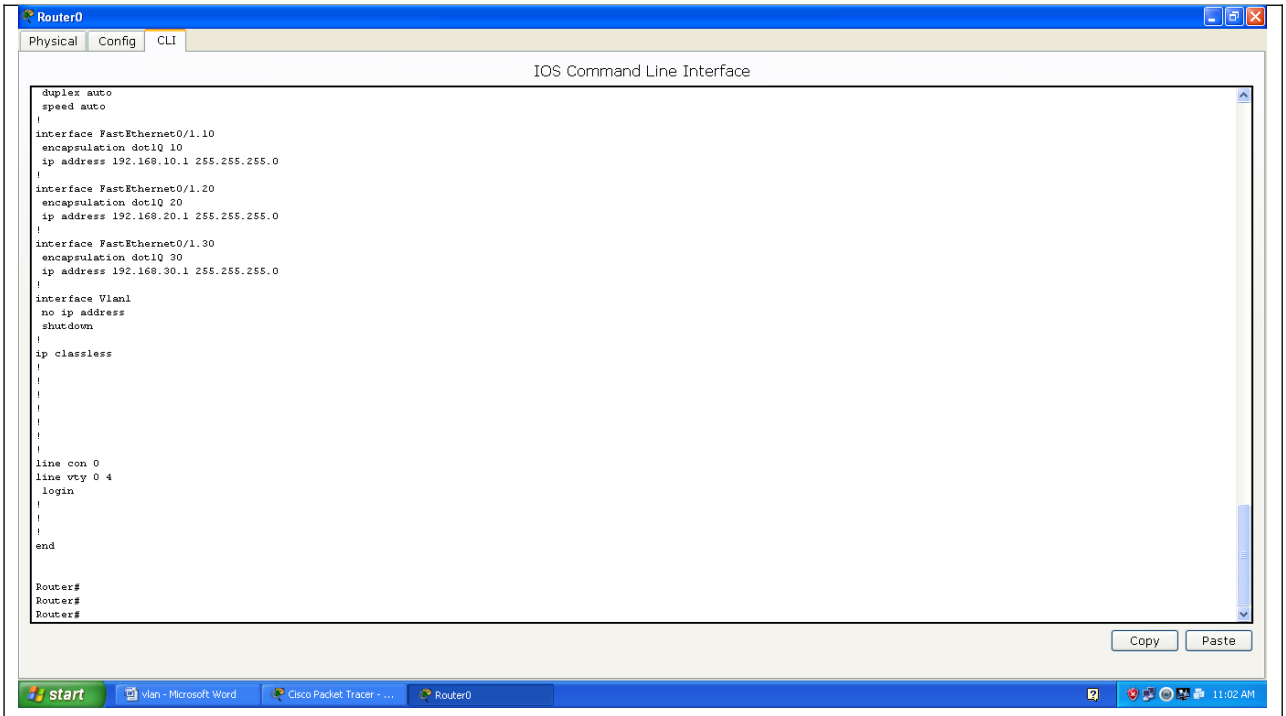


Still 192.168.10.100 can not ping to 192.168.20.100. For that there should be use of router.

Step 3 Router Configuration

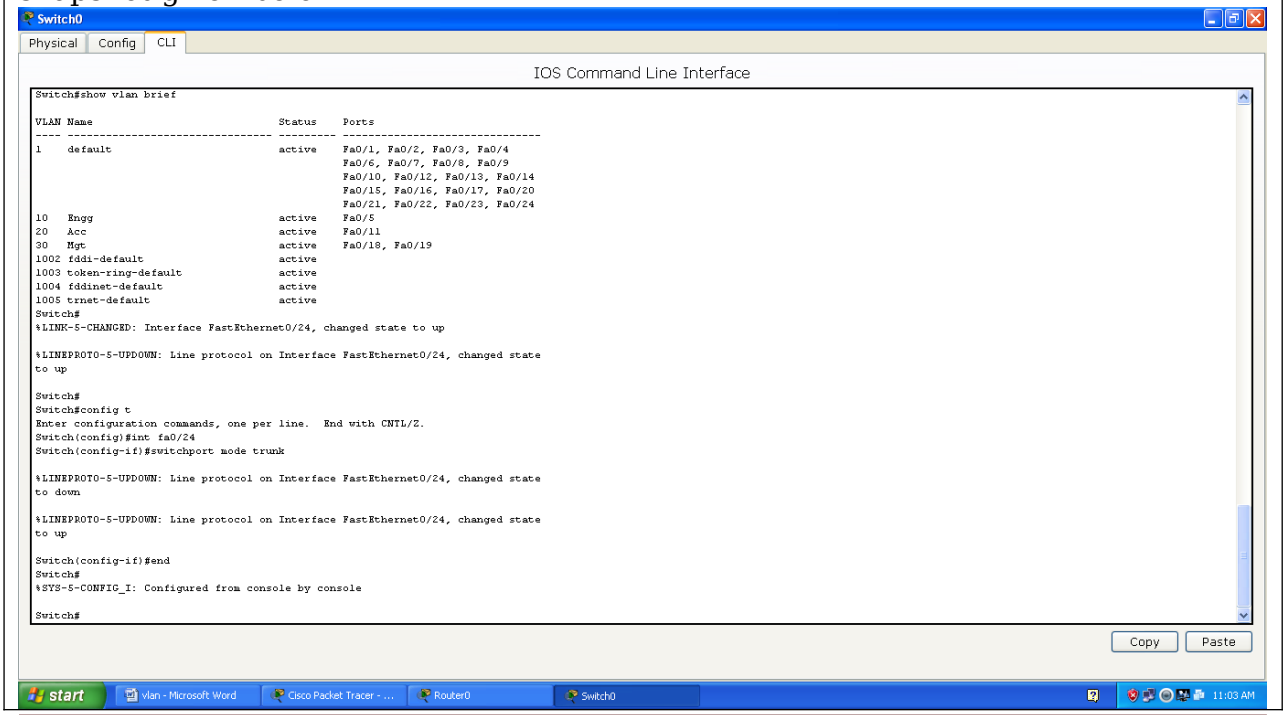



Router0
Physical
Config
CLI
IOS Command Line Interface
o up
%LINK-5-CHANGED: Interface FastEthernet0/1.10, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1.10, changed stat
e to up
%LINK-5-CHANGED: Interface FastEthernet0/1.20, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1.20, changed stat
e to up
%LINK-5-CHANGED: Interface FastEthernet0/1.30, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1.30, changed stat
e to up
Router(config-if)#exit
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console
Router#show run
Building configuration...
Current configuration : 746 bytes
!
version 12.4
no service timestamps log datetime msec
no service timestamps debug datetime msec
no service password-encryption
!
hostname Router
!
!
!
!
!
!
Copy
Paste
Paste to CLI
start
vlan - Microsoft Word
Cisco Packet Tracer - ...
Router0
11:02 AM
Router0
Physical
Config
CLI
IOS Command Line Interface
!
!
!
!
interface FastEthernet0/0
 no ip address
 duplex auto
 speed auto
 shutdown
!
interface FastEthernet0/1
 no ip address
 duplex auto
 speed auto
!
interface FastEthernet0/1.10
 encapsulation dot1Q 10
 ip address 192.168.10.1 255.255.255.0
!
interface FastEthernet0/1.20
 encapsulation dot1Q 20
 ip address 192.168.20.1 255.255.255.0
!
interface FastEthernet0/1.30
 encapsulation dot1Q 30
 ip address 192.168.30.1 255.255.255.0
!
interface Vlan1
 no ip address
 shutdown
!
ip classless
!
!
!
!
!
!
Copy
Paste
start
vlan - Microsoft Word
Cisco Packet Tracer - ...
Router0
11:02 AM

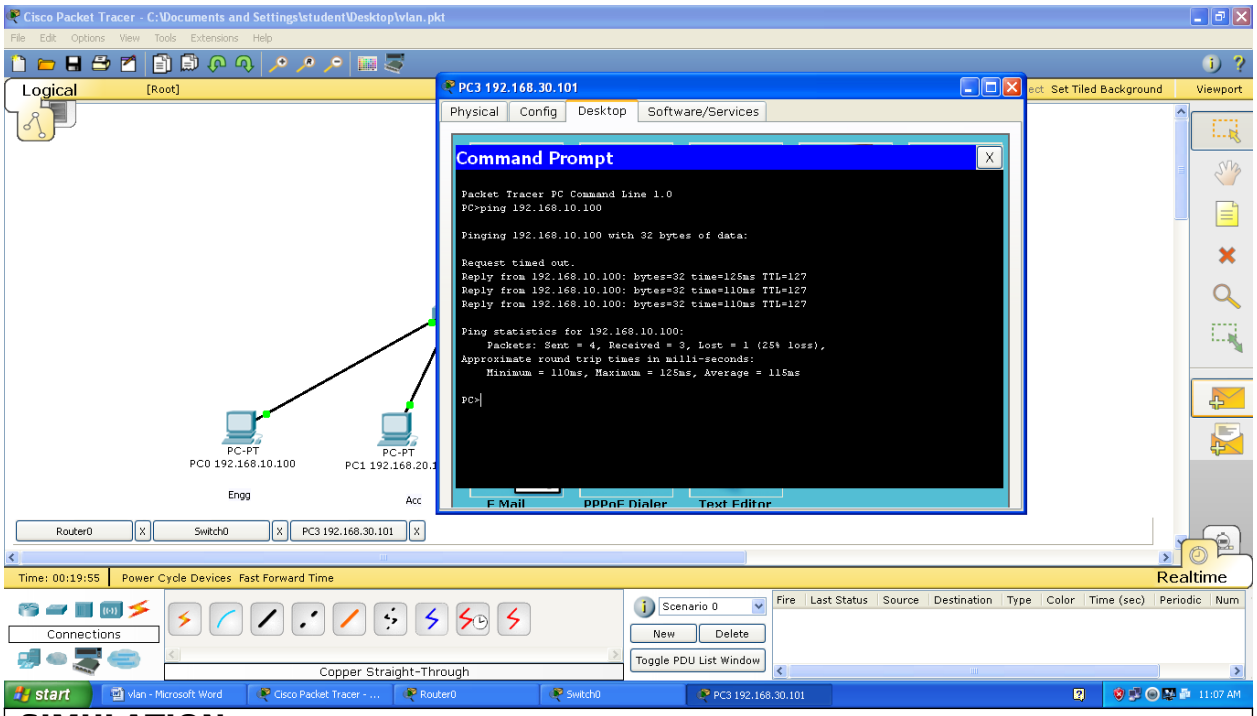


Step 4: Configuration of trunk in switch

We used port 24 for connecting switch to router. The command are shown in the snapshot given below



Here check the VLAN configuration working or not. As getting the reply from VLAN 10 so it's working

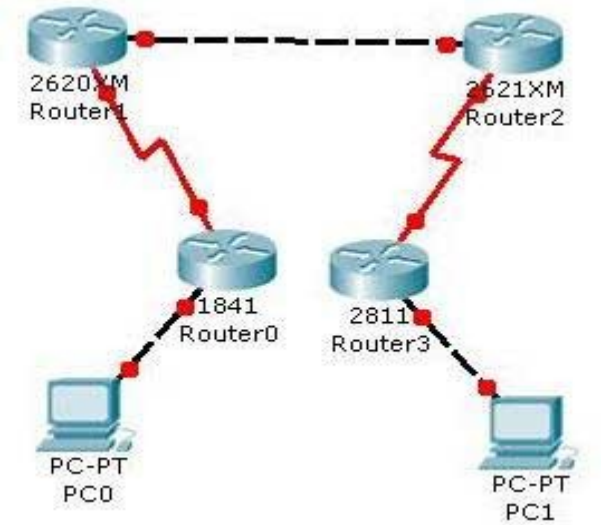


SIMULATION:

4. Static Routing

[How to configure Static Route on router]

In this article we will demonstrate an example of static route configurations. We will use four different series router so you can get familiar with all different platform Create a topology as shown in figure.



A static route is a manually configured route on your router. Static routes are

typically used in smaller networks and when few networks or subnets exist, or with WAN links that have little available bandwidth. With a network that has hundreds of routes, static routes are not scalable, since you would have to configure each route and any redundant paths for that route on each router.

1841 Series Router0 (R1)		
	FastEthernet0/0	Serial0/0/0
IP address	10.0.0.1	20.0.0.1
Connected With	Pc0	R2 on Serial 0/0
2811 Series Router0 (R4)		
	FastEthernet0/0	Serial0/0/0
IP address	50.0.0.1	40.0.0.2
Connected With	Pc1	R3 on Serial 0/0
2621XM Series Router0 (R3)		
	FastEthernet0/0	Serial0/0/0
IP address	30.0.0.2	40.0.0.1
Connected With	FastEthernet0/0	R4 on Serial 0/0/0
2620XM Series Router1 (R2)		
	FastEthernet0/0	Serial0/0
IP address	30.0.0.1	20.0.0.2
Connected With	R3 on FastEthernet0/0	R1 on Serial 0/0/0
PC-PT PC0		
	FastEthernet0	Default Gateway
IP address	10.0.0.2	10.0.0.1
Connected	R1 on	

With	FastEthernet0/0	
PC-PT PC1		
	FastEthernet0	Default Gateway
IP address	50.0.0.2	50.0.0.1
Connected With	R4 on FastEthernet0/0	

To configure any router double click on it and select CLI.To configure this topology use this step by step guide.

(1841Router0) Hostname R1

To configure and enable static routing on R1 follow these commands exactly.

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#hostname R1
R1(config)#interface fastethernet 0/0
R1(config-if)#ip address 10.0.0.1 255.0.0.0
R1(config-if)#no shutdown
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0,
changed state to up
R1(config-if)#exit
R1(config)#interface serial 0/0/0
R1(config-if)#ip address 20.0.0.1 255.0.0.0
R1(config-if)#clock rate 64000
R1(config-if)#bandwidth 64
R1(config-if)#no shutdown
%LINK-5-CHANGED: Interface Serial0/0/0, changed state to down
R1(config-if)#exit
%LINK-5-CHANGED: Interface Serial0/0/0, changed state to up
R1(config)#ip route 30.0.0.0 255.0.0.0 20.0.0.2
R1(config)#ip route 40.0.0.0 255.0.0.0 20.0.0.2
R1(config)#ip route 50.0.0.0 255.0.0.0 20.0.0.2
```

(2620XM-Router1) Hostname R2

To configure and enable static routing on R2 follow these commands exactly.

```
Router>enable
Router#configure terminal
```

```
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#hostname R2
R2(config)#interface serial 0/0
R2(config-if)#ip address 20.0.0.2 255.0.0.0
R2(config-if)#no shutdown
%LINK-5-CHANGED: Interface Serial0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0, changed
state to up
R2(config-if)#exit
R2(config)#interface fastethernet 0/0
R2(config-if)#ip address 30.0.0.1 255.0.0.0
R2(config-if)#no shutdown
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up
R2(config-if)#exit
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0,
changed state to up
R2(config)#ip route 10.0.0.0 255.0.0.0 20.0.0.1
R2(config)#ip route 40.0.0.0 255.0.0.0 30.0.0.2
R2(config)#ip route 50.0.0.0 255.0.0.0 30.0.0.2
```

(2620XM-Router2)Hostname R3

To configure and enable static routing on R3 follow these commands exactly.

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#hostname R3
R3(config)#interface fastethernet 0/0
R3(config-if)#ip address 30.0.0.2 255.0.0.0
R3(config-if)#no shutdown
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0,
changed state to up
R3(config-if)#interface serial 0/0
R3(config-if)#ip address 40.0.0.1 255.0.0.0
R3(config-if)#clock rate 64000
R3(config-if)#bandwidth 64
R3(config-if)#no shutdown
%LINK-5-CHANGED: Interface Serial0/0, changed state to down
R3(config-if)#exit
%LINK-5-CHANGED: Interface Serial0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0, changed
```

```
state to up
R3(config)#ip route 10.0.0.0 255.0.0.0 30.0.0.1
R3(config)#ip route 20.0.0.0 255.0.0.0 30.0.0.1
R3(config)#ip route 50.0.0.0 255.0.0.0 40.0.0.2
```

(2811Router3) Hostname R4

To configure and enable static routing on R4 follow these commands exactly.

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface serial 0/0/0
Router(config-if)#ip address 40.0.0.2 255.0.0.0
Router(config-if)#no shutdown
%LINK-5-CHANGED: Interface Serial0/0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0/0,
changed state to up
Router(config-if)#exit
Router(config)#interface fastethernet 0/0
Router(config-if)#ip address 50.0.0.1 255.0.0.0
Router(config-if)#no shutdown
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0,
changed state to up
Router(config-if)#exit
Router(config)#ip route 10.0.0.0 255.0.0.0 40.0.0.1
Router(config)#ip route 20.0.0.0 255.0.0.0 40.0.0.1
Router(config)#ip route 30.0.0.0 255.0.0.0 40.0.0.1
```

PC-1

```
PC>ipconfig

IP Address.....................: 10.0.0.2
Subnet Mask....................: 255.0.0.0
Default Gateway................: 10.0.0.1

PC>ping 50.0.0.2

Pinging 50.0.0.2 with 32 bytes of data:

Reply from 50.0.0.2: bytes=32 time=156ms TTL=124
Reply from 50.0.0.2: bytes=32 time=127ms TTL=124
Reply from 50.0.0.2: bytes=32 time=156ms TTL=124
```

```
Reply from 50.0.0.2: bytes=32 time=140ms TTL=124

Ping statistics for 50.0.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 127ms, Maximum = 156ms, Average = 144ms
PC>
```

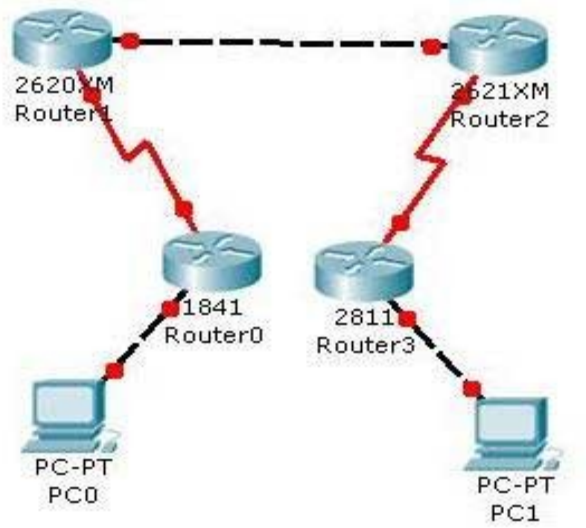
PC-2

```
PC>ipconfig

IP Address.....................: 50.0.0.2
Subnet Mask....................: 255.0.0.0
Default Gateway................: 50.0.0.1

PC>ping 10.0.0.2

Pinging 10.0.0.2 with 32 bytes of data:

Reply from 10.0.0.2: bytes=32 time=140ms TTL=124
Reply from 10.0.0.2: bytes=32 time=141ms TTL=124
Reply from 10.0.0.2: bytes=32 time=157ms TTL=124
Reply from 10.0.0.2: bytes=32 time=156ms TTL=124

Ping statistics for 10.0.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 140ms, Maximum = 157ms, Average = 148ms
```

To test static routing do ping from pc1 to pc2 and vice versa. If we get reply then you have successfully configured static routing but if we did not get reply double check this configuration and try to troubleshoot.

5. RIP

In this article we will demonstrate an example of **Rip Routing** configurations. We will use four different series router so you can get familiar with all different platform Create a topology as shown in figure.



IP RIP comes in two different versions: 1 and 2. Version 1 is a distance vector protocol and is defined in RFC 1058. Version 2 is a hybrid protocol and is defined in RFCs 1721 and 1722. There are no major differences between RIPv1 or RIPv2 so far configurations concern.

1841 Series Router0 (R1)		
	FastEthernet0/0	Serial0/0/0
IP address	10.0.0.1	20.0.0.1
Connected With	Pc0	R2 on Serial 0/0
2811 Series Router0 (R4)		
	FastEthernet0/0	Serial0/0/0
IP address	50.0.0.1	40.0.0.2
Connected With	Pc1	R3 on Serial 0/0
2621XM Series Router0 (R3)		
	FastEthernet0/0	Serial0/0/0
IP address	30.0.0.2	40.0.0.1
Connected With	FastEthernet0/0	R4 on Serial 0/0/0
2620XM Series Router1 (R2)		

	FastEthernet0/0	Serial0/0
IP address	30.0.0.1	20.0.0.2
Connected With	R3 on FastEthernet0/0	R1 on Serial 0/0/0

PC-PT PC0		
	FastEthernet0	Default Gateway
IP address	10.0.0.2	10.0.0.1
Connected With	R1 on FastEthernet0/0	

PC-PT PC1		
	FastEthernet0	Default Gateway
IP address	50.0.0.2	50.0.0.1
Connected With	R4 on FastEthernet0/0	

To configure any router double click on it and select **CLI**. To configure this topology use this step by step guide.

(1841Router0) Hostname R1

To configure and enable rip routing on R1 follow these commands exactly.

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#hostname R1
R1(config)#interface fastethernet 0/0
R1(config-if)#ip address 10.0.0.1 255.0.0.0
R1(config-if)#no shutdown
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0,
changed state to up
R1(config-if)#exit
R1(config)#interface serial 0/0/0
R1(config-if)#ip address 20.0.0.1 255.0.0.0
R1(config-if)#clock rate 64000
```

```
R1(config-if)#bandwidth 64
R1(config-if)#no shutdown
%LINK-5-CHANGED: Interface Serial0/0/0, changed state to down
R1(config-if)#exit
%LINK-5-CHANGED: Interface Serial0/0/0, changed state to up
R1(config)#router rip
R1(config-router)#network 10.0.0.0
R1(config-router)#network 20.0.0.0
R1(config-router)#exit
R1(config)#
```

(2620XM-Router1) Hostname R2

To configure and enable rip routing on R2 follow these commands exactly.

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#hostname R2
R2(config)#interface serial 0/0
R2(config-if)#ip address 20.0.0.2 255.0.0.0
R2(config-if)#no shutdown
%LINK-5-CHANGED: Interface Serial0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0, changed
state to up
R2(config-if)#exit
R2(config)#interface fastethernet 0/0
R2(config-if)#ip address 30.0.0.1 255.0.0.0
R2(config-if)#no shutdown
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up
R2(config-if)#exit
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0,
changed state to up
R2(config)#router rip
R2(config-router)#network 20.0.0.0
R2(config-router)#network 30.0.0.0
R2(config-router)#exit
R2(config)#
```

(2620XM-Router2)Hostname R3

To configure and enable rip routing on R3 follow these commands exactly.

```
Router>enable
Router#configure terminal
```

```
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#hostname R3
R3(config)#interface fastethernet 0/0
R3(config-if)#ip address 30.0.0.2 255.0.0.0
R3(config-if)#no shutdown
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0,
changed state to up
R3(config-if)#interface serial 0/0
R3(config-if)#ip address 40.0.0.1 255.0.0.0
R3(config-if)#clock rate 64000
R3(config-if)#bandwidth 64
R3(config-if)#no shutdown
%LINK-5-CHANGED: Interface Serial0/0, changed state to down
R3(config-if)#exit
%LINK-5-CHANGED: Interface Serial0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0, changed
state to up
R3(config)#router rip
R3(config-router)#network 30.0.0.0
R3(config-router)#network 40.0.0.0
R3(config-router)#exit
R3(config)#
```

(2811Router3) Hostname R4

To configure and enable rip routing on R4 follow these commands exactly.

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface serial 0/0/0
Router(config-if)#ip address 40.0.0.2 255.0.0.0
Router(config-if)#no shutdown
%LINK-5-CHANGED: Interface Serial0/0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0/0,
changed state to up
Router(config-if)#exit
Router(config)#interface fastethernet 0/0
Router(config-if)#ip address 50.0.0.1 255.0.0.0
Router(config-if)#no shutdown
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0,
```

```
changed state to up
Router(config-if)#exit
R4(config)#router rip
R4(config-router)#network 40.0.0.0
R4(config-router)#network 50.0.0.0
R4(config-router)#exit
R4(config)#
```

PC-1

```
PC>ipconfig

IP Address.....................: 10.0.0.2
Subnet Mask....................: 255.0.0.0
Default Gateway................: 10.0.0.1

PC>ping 50.0.0.2

Pinging 50.0.0.2 with 32 bytes of data:

Reply from 50.0.0.2: bytes=32 time=156ms TTL=124
Reply from 50.0.0.2: bytes=32 time=127ms TTL=124
Reply from 50.0.0.2: bytes=32 time=156ms TTL=124
Reply from 50.0.0.2: bytes=32 time=140ms TTL=124

Ping statistics for 50.0.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 127ms, Maximum = 156ms, Average = 144ms
PC>
```

PC-2

```
PC>ipconfig

IP Address.....................: 50.0.0.2
Subnet Mask....................: 255.0.0.0
Default Gateway................: 50.0.0.1

PC>ping 10.0.0.2

Pinging 10.0.0.2 with 32 bytes of data:
```

```
Reply from 10.0.0.2: bytes=32 time=140ms TTL=124
Reply from 10.0.0.2: bytes=32 time=141ms TTL=124
Reply from 10.0.0.2: bytes=32 time=157ms TTL=124
Reply from 10.0.0.2: bytes=32 time=156ms TTL=124

Ping statistics for 10.0.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 140ms, Maximum = 157ms, Average = 148ms
```

We can verify that RIP is running successfully via **show ip protocols** command in privilege mode.

```
R1#show ip protocols
Routing Protocol is "rip"
Sending updates every 30 seconds, next due in 2 seconds
Invalid after 180 seconds, hold down 180, flushed after 240
Outgoing update filter list for all interfaces is not set
Incoming update filter list for all interfaces is not set
Redistributing: rip
Default version control: send version 1, receive any version
  Interface             Send  Recv  Triggered RIP  Key-chain
  FastEthernet0/0       1     2 1
  Serial0/0/0           1     2 1
Automatic network summarization is in effect
Maximum path: 4
Routing for Networks:
      10.0.0.0
      20.0.0.0
Passive Interface(s):
Routing Information Sources:
      Gateway         Distance      Last Update
      20.0.0.2             120      00:00:20
Distance: (default is 120)
R1#
```

You can use **show ip route** command to troubleshoot rip network. If you did not see information about any route checks the router attached with that network.

```
R1#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B
- BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter
area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external
```

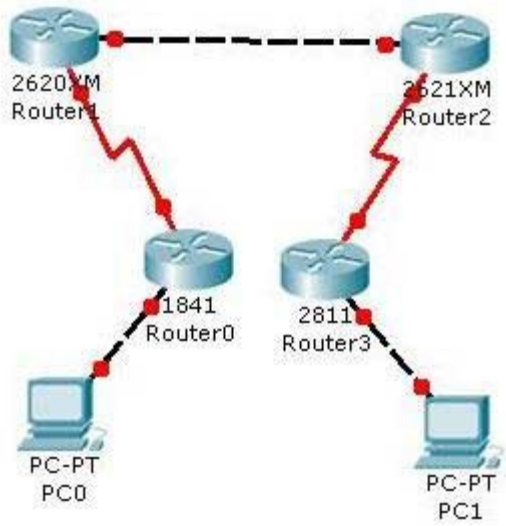
```
type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E -
EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-
IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

C    10.0.0.0/8 is directly connected, FastEthernet0/0
C    20.0.0.0/8 is directly connected, Serial0/0/0
R    30.0.0.0/8 [120/1] via 20.0.0.2, 00:00:01, Serial0/0/0
R    40.0.0.0/8 [120/2] via 20.0.0.2, 00:00:01, Serial0/0/0
R    50.0.0.0/8 [120/3] via 20.0.0.2, 00:00:01, Serial0/0/0
R1#
```

To test rip routing do ping from pc1 to pc2 and vice versa. If we get reply then you have successfully configured rip routing but if we did not get replay double check this configuration and try to troubleshoot.

2. EIGRP

EIGRP is a Cisco-proprietary routing protocol for TCP/IP. It's actually based on Cisco's proprietary IGRP routing protocol, with many enhancements built into it. Because it has its roots in IGRP, the configuration is similar to IGRP; however, it has many link state characteristics that were added to it to allow EIGRP to scale to enterprise network sizes. To know these characteristics read our previous article.

In this article I will demonstrate an example of **EIGRP Routing** configurations. Create a topology as shown in figure.



1841 Series Router0 (R1)		
	FastEthernet0/0	Serial0/0/0
IP address	10.0.0.1	20.0.0.1
Connected With	Pc0	R2 on Serial 0/0
2811 Series Router0 (R4)		
	FastEthernet0/0	Serial0/0/0
IP address	50.0.0.1	40.0.0.2
Connected With	Pc1	R3 on Serial 0/0
2621XM Series Router0 (R3)		
	FastEthernet0/0	Serial0/0/0
IP address	30.0.0.2	40.0.0.1
Connected With	FastEthernet0/0	R4 on Serial 0/0/0
2620XM Series Router1 (R2)		
	FastEthernet0/0	Serial0/0

IP address	30.0.0.1	20.0.0.2
Connected With	R3 on FastEthernet0/0	R1 on Serial 0/0/0
PC-PT PC0		
	FastEthernet0	Default Gateway
IP address	10.0.0.2	10.0.0.1
Connected With	R1 on FastEthernet0/0	
PC-PT PC1		
	FastEthernet0	Default Gateway
IP address	50.0.0.2	50.0.0.1
Connected With	R4 on FastEthernet0/0	

To configure any router double click on it and select CLI.To configure this topology use this step by step guide.

(1841Router0) Hostname R1

To configure and enable eigrp routing on R1 follow these commands exactly.

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#hostname R1
R1(config)#interface fastethernet 0/0
R1(config-if)#ip address 10.0.0.1 255.0.0.0
R1(config-if)#no shutdown
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0,
changed state to up
R1(config-if)#exit
R1(config)#interface serial 0/0/0
R1(config-if)#ip address 20.0.0.1 255.0.0.0
R1(config-if)#clock rate 64000
R1(config-if)#bandwidth 64
R1(config-if)#no shutdown
```

```
%LINK-5-CHANGED: Interface Serial0/0/0, changed state to down
R1(config-if)#exit
%LINK-5-CHANGED: Interface Serial0/0/0, changed state to up
R1(config)#router eigrp 1
R1(config-router)#network 10.0.0.0
R1(config-router)#network 20.0.0.0
R1(config-router)#exit
R1(config)#
```

(2620XM-Router1) Hostname R2

To configure and enable eigrp routing on R2 follow these commands exactly.

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#hostname R2
R2(config)#interface serial 0/0
R2(config-if)#ip address 20.0.0.2 255.0.0.0
R2(config-if)#no shutdown
%LINK-5-CHANGED: Interface Serial0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0, changed
state to up
R2(config-if)#exit
R2(config)#interface fastethernet 0/0
R2(config-if)#ip address 30.0.0.1 255.0.0.0
R2(config-if)#no shutdown
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up
R2(config-if)#exit
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0,
changed state to up
R2(config)#router eigrp 1
R2(config-router)#network 20.0.0.0
R2(config-router)#network 30.0.0.0
R2(config-router)#exit
R2(config)#
```

(2620XM-Router2)Hostname R3

To configure and enable eigrp routing on R3 follow these commands exactly.

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#hostname R3
R3(config)#interface fastethernet 0/0
```

```
R3(config-if)#ip address 30.0.0.2 255.0.0.0
R3(config-if)#no shutdown
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0,
changed state to up
R3(config-if)#interface serial 0/0
R3(config-if)#ip address 40.0.0.1 255.0.0.0
R3(config-if)#clock rate 64000
R3(config-if)#bandwidth 64
R3(config-if)#no shutdown
%LINK-5-CHANGED: Interface Serial0/0, changed state to down
R3(config-if)#exit
%LINK-5-CHANGED: Interface Serial0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0, changed
state to up
R3(config)#router eigrp 1
R3(config-router)#network 30.0.0.0
R3(config-router)#network 40.0.0.0
R3(config-router)#exit
R3(config)#
```

(2811Router3) Hostname R4

To configure and enable eigrp routing on R4 follow these commands exactly.

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface serial 0/0/0
Router(config-if)#ip address 40.0.0.2 255.0.0.0
Router(config-if)#no shutdown
%LINK-5-CHANGED: Interface Serial0/0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0/0,
changed state to up
Router(config-if)#exit
Router(config)#interface fastethernet 0/0
Router(config-if)#ip address 50.0.0.1 255.0.0.0
Router(config-if)#no shutdown
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0,
changed state to up
Router(config-if)#exit
R3(config)#router eigrp 1
```

```
R3(config-router)#network 30.0.0.0
R3(config-router)#network 40.0.0.0
R3(config-router)#exit
R3(config)#
```

PC-1

```
PC>ipconfig

IP Address.....................: 10.0.0.2
Subnet Mask....................: 255.0.0.0
Default Gateway................: 10.0.0.1

PC>ping 50.0.0.2

Pinging 50.0.0.2 with 32 bytes of data:

Reply from 50.0.0.2: bytes=32 time=156ms TTL=124
Reply from 50.0.0.2: bytes=32 time=127ms TTL=124
Reply from 50.0.0.2: bytes=32 time=156ms TTL=124
Reply from 50.0.0.2: bytes=32 time=140ms TTL=124

Ping statistics for 50.0.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 127ms, Maximum = 156ms, Average = 144ms
PC>
```

PC-2

```
PC>ipconfig

IP Address.....................: 50.0.0.2
Subnet Mask....................: 255.0.0.0
Default Gateway................: 50.0.0.1

PC>ping 10.0.0.2

Pinging 10.0.0.2 with 32 bytes of data:

Reply from 10.0.0.2: bytes=32 time=140ms TTL=124
Reply from 10.0.0.2: bytes=32 time=141ms TTL=124
Reply from 10.0.0.2: bytes=32 time=157ms TTL=124
```

```
Reply from 10.0.0.2: bytes=32 time=156ms TTL=124

Ping statistics for 10.0.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 140ms, Maximum = 157ms, Average = 148ms
```

You can verify that eigrp is running successfully via **show ip protocols** command in privilege mode.

```
R4#show ip protocols

Routing Protocol is "ospf 4"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Router ID 50.0.0.1
  Number of areas in this router is 1. 1 normal 0 stub 0 nssa
  Maximum path: 4
  Routing for Networks:
    50.0.0.0 0.255.255.255 area 0
    40.0.0.0 0.255.255.255 area 0
  Routing Information Sources:
    Gateway         Distance      Last Update
    40.0.0.1             110      00:01:26
  Distance: (default is 110)

R4#
```

You can use **show ip route** command to troubleshoot eigrp network. If you did not see information about any route checks the router attached with that network.

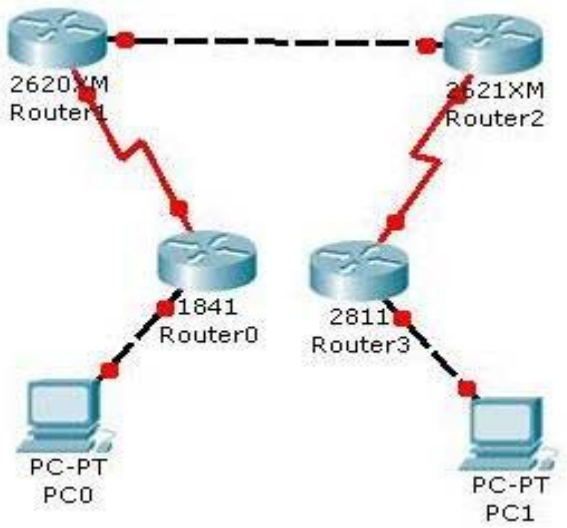
```
R4#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B
- BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter
area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external
type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E -
EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-
IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route
```

```
Gateway of last resort is not set

O    10.0.0.0/8 [110/1564] via 40.0.0.1, 00:02:37, Serial0/0/0
O    20.0.0.0/8 [110/1563] via 40.0.0.1, 00:02:37, Serial0/0/0
O    30.0.0.0/8 [110/782] via 40.0.0.1, 00:02:37, Serial0/0/0
C    40.0.0.0/8 is directly connected, Serial0/0/0
C    50.0.0.0/8 is directly connected, FastEthernet0/0
R4#
```

To test eigrp routing do ping from pc1 to pc2 and vice versa. If you get replay then you have successfully configured eigrp routing but if you did not get replay double check this configuration and try to troubleshoot.

6. OSPF

In this article we will demonstrate an example of **OSPF Routing** configuration.Create a topology as shown in figure.



1841 Series Router0 (R1)		
	FastEthernet0/0	Serial0/0/0
IP address	10.0.0.1	20.0.0.1
Connected With	Pc0	R2 on Serial 0/0
2811 Series Router0 (R4)		
	FastEthernet0/0	Serial0/0/0
IP address	50.0.0.1	40.0.0.2

Connected With	Pc1	R3 on Serial 0/0
2621XM Series Router0 (R3)		
	FastEthernet0/0	Serial0/0/0
IP address	30.0.0.2	40.0.0.1
Connected With	FastEthernet0/0	R4 on Serial 0/0/0
2620XM Series Router1 (R2)		
	FastEthernet0/0	Serial0/0
IP address	30.0.0.1	20.0.0.2
Connected With	R3 on FastEthernet0/0	R1 on Serial 0/0/0
PC-PT PC0		
	FastEthernet0	Default Gateway
IP address	10.0.0.2	10.0.0.1
Connected With	R1 on FastEthernet0/0	
PC-PT PC1		
	FastEthernet0	Default Gateway
IP address	50.0.0.2	50.0.0.1
Connected With	R4 on FastEthernet0/0	

Configuring OSPF is slightly different from configuring RIP. When configuring OSPF, use the following syntax:

```
Router(config)# router ospf process_ID
Router(config-router)# network IP_address wildcard_mask area
area_#
```

The process_ID is locally significant and is used to differentiate between OSPF processes running on the same router. Your router might be a boundary router

between two OSPF autonomous systems, and to differentiate them on your router, you will give them unique process IDs. Note that these numbers do not need to match between different routers so they have nothing to do with autonomous system numbers.

To configure any router double click on it and select CLI.To configure this topology use this step by step guide.

(1841Router0) Hostname R1

To configure and enable ospf routing on R1 follow these commands exactly.

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#hostname R1
R1(config)#interface fastethernet 0/0
R1(config-if)#ip address 10.0.0.1 255.0.0.0
R1(config-if)#no shutdown
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0,
changed state to up
R1(config-if)#exit
R1(config)#interface serial 0/0/0
R1(config-if)#ip address 20.0.0.1 255.0.0.0
R1(config-if)#clock rate 64000
R1(config-if)#bandwidth 64
R1(config-if)#no shutdown
%LINK-5-CHANGED: Interface Serial0/0/0, changed state to down
R1(config-if)#exit
%LINK-5-CHANGED: Interface Serial0/0/0, changed state to up
R1(config)#router ospf 1
R1(config-router)#network 10.0.0.0 0.255.255.255 area 0
R1(config-router)#network 20.0.0.0 0.255.255.255 area 0
R1(config-router)#exit
R1(config)#
```

(2620XM-Router1) Hostname R2

To configure and enable ospf routing on R2 follow these commands exactly.

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#hostname R2
R2(config)#interface serial 0/0
```

```
R2(config-if)#ip address 20.0.0.2 255.0.0.0
R2(config-if)#no shutdown
%LINK-5-CHANGED: Interface Serial0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0, changed
state to up
R2(config-if)#exit
R2(config)#interface fastethernet 0/0
R2(config-if)#ip address 30.0.0.1 255.0.0.0
R2(config-if)#no shutdown
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up
R2(config-if)#exit
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0,
changed state to up
R2(config)#router ospf 2
R2(config-router)#network 20.0.0.0 0.255.255.255 area 0
R2(config-router)#network 3
00:03:10: %OSPF-5-ADJCHG: Process 2, Nbr 20.0.0.1 on Serial0/0
from
 LOADING to FULL, Loading Done0.0.0.0 0.255.255.255 area 0
R2(config-router)#network 30.0.0.0 0.255.255.255 area 0
R2(config-router)#exit
R2(config)#
```

(2620XM-Router2)Hostname R3

To configure and enable ospf routing on R3 follow these commands exactly.

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#hostname R3
R3(config)#interface fastethernet 0/0
R3(config-if)#ip address 30.0.0.2 255.0.0.0
R3(config-if)#no shutdown
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0,
changed state to up
R3(config-if)#interface serial 0/0
R3(config-if)#ip address 40.0.0.1 255.0.0.0
R3(config-if)#clock rate 64000
R3(config-if)#bandwidth 64
R3(config-if)#no shutdown
```

```
%LINK-5-CHANGED: Interface Serial0/0, changed state to down
R3(config-if)#exit
%LINK-5-CHANGED: Interface Serial0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0, changed
state to up
R3(config)#router ospf 3
R3(config-router)#network 40.0.0.0 0.255.255.255 area 0
R3(config-router)#network 30.0.0.0 0.255.255.255 area 0
00:04:53: %OSPF-5-ADJCHG: Process 3, Nbr 30.0.0.1 on
FastEthernet0/0 from
 LOADING to FULL, Loading D
R3(config-router)#exit
R3(config)#
%SYS-5-CONFIG_I: Configured from console by console
R3#
```

(2811Router3) Hostname R4

To configure and enable ospf routing on R4 follow these commands exactly.

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface serial 0/0/0
Router(config-if)#ip address 40.0.0.2 255.0.0.0
Router(config-if)#no shutdown
%LINK-5-CHANGED: Interface Serial0/0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0/0,
changed state to up
Router(config-if)#exit
Router(config)#interface fastethernet 0/0
Router(config-if)#ip address 50.0.0.1 255.0.0.0
Router(config-if)#no shutdown
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0,
changed state to up
Router(config-if)#exit
R4(config)#router ospf 4
R4(config-router)#network 50.0.0.0 0.255.255.255 area 0
R4(config-router)#network 40.0.0.0 0.255.255.255 area 0
R4(config-router)#
00:06:32: %OSPF-5-ADJCHG: Process 4, Nbr 40.0.0.1 on Serial0/0/0
```

```
from
LOADING to FULL, Loading Done
R4(config-router)#exit
R4(config)#
```

PC-1

```
PC>ipconfig

IP Address.....................: 10.0.0.2
Subnet Mask....................: 255.0.0.0
Default Gateway................: 10.0.0.1

PC>ping 50.0.0.2

Pinging 50.0.0.2 with 32 bytes of data:

Reply from 50.0.0.2: bytes=32 time=156ms TTL=124
Reply from 50.0.0.2: bytes=32 time=127ms TTL=124
Reply from 50.0.0.2: bytes=32 time=156ms TTL=124
Reply from 50.0.0.2: bytes=32 time=140ms TTL=124

Ping statistics for 50.0.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 127ms, Maximum = 156ms, Average = 144ms
PC>
```

PC-2

```
PC>ipconfig

IP Address.....................: 50.0.0.2
Subnet Mask....................: 255.0.0.0
Default Gateway................: 50.0.0.1

PC>ping 10.0.0.2

Pinging 10.0.0.2 with 32 bytes of data:

Reply from 10.0.0.2: bytes=32 time=140ms TTL=124
Reply from 10.0.0.2: bytes=32 time=141ms TTL=124
Reply from 10.0.0.2: bytes=32 time=157ms TTL=124
```

```
Reply from 10.0.0.2: bytes=32 time=156ms TTL=124

Ping statistics for 10.0.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 140ms, Maximum = 157ms, Average = 148ms
```

You can verify that ospf is running successfully via **show ip protocols** command in privilege mode.

```
R4#show ip protocols

Routing Protocol is "ospf 4"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Router ID 50.0.0.1
  Number of areas in this router is 1. 1 normal 0 stub 0 nssa
  Maximum path: 4
  Routing for Networks:
    50.0.0.0 0.255.255.255 area 0
    40.0.0.0 0.255.255.255 area 0
  Routing Information Sources:
    Gateway         Distance      Last Update
    40.0.0.1             110      00:01:26
  Distance: (default is 110)

R4#
```

You can use **show ip oute** command to troubleshoot ospf network. If you did not see information about any route checks the router attached with that network.

```
R4#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B
- BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter
area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external
type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E -
EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-
IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route
```

```
Gateway of last resort is not set

O    10.0.0.0/8 [110/1564] via 40.0.0.1, 00:02:37, Serial0/0/0
O    20.0.0.0/8 [110/1563] via 40.0.0.1, 00:02:37, Serial0/0/0
O    30.0.0.0/8 [110/782] via 40.0.0.1, 00:02:37, Serial0/0/0
C    40.0.0.0/8 is directly connected, Serial0/0/0
C    50.0.0.0/8 is directly connected, FastEthernet0/0
R4#
```

To test ospf routing do ping from pc1 to pc2 and vice versa. If you get replay then you have successfully configured ospf routing but if you did not get replay double check this configuration and try to troubleshoot. I have uploaded a configured and tested topology in case you are unable to locate the problem spot then download this configuration file. And try to find out where have you committed mistake

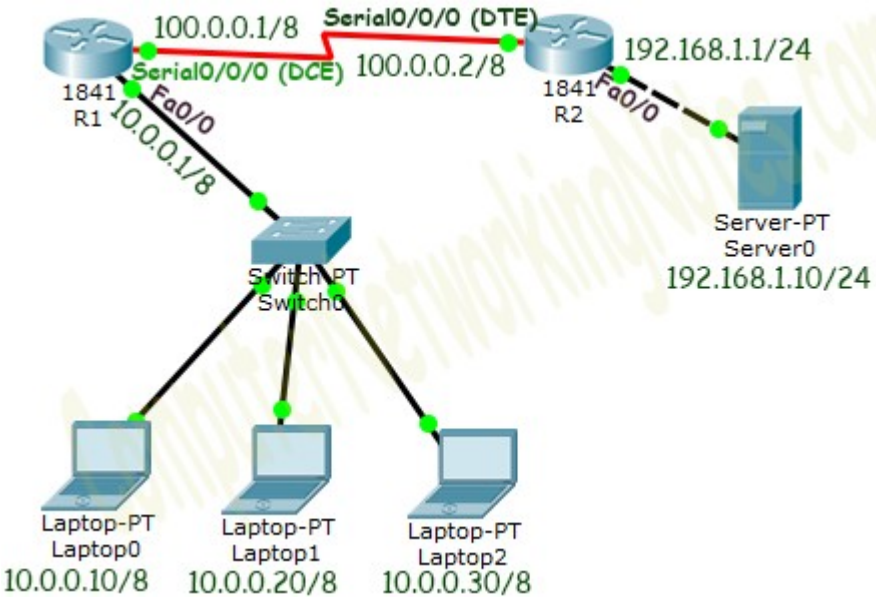
How to Configure Static NAT in Cisco Router

This tutorial explains Static NAT configuration in detail. Learn how configure static NAT, map address (inside local address, outside local address, inside global address and outside global address), debug and verify Static NAT translation step by step with practical examples in packet tracer.

To explain Static NAT Configuration, I will use packet tracer network simulator software. You can use any network simulator software or can use real Cisco devices to follow this guide. There is no difference in output as long as your selected software contains the commands explained in this tutorial.

Create a practice lab as shown in following figure or download this pre-created practice lab and load in packet tracer

Download NAT Practice LAB with initial IP configuration



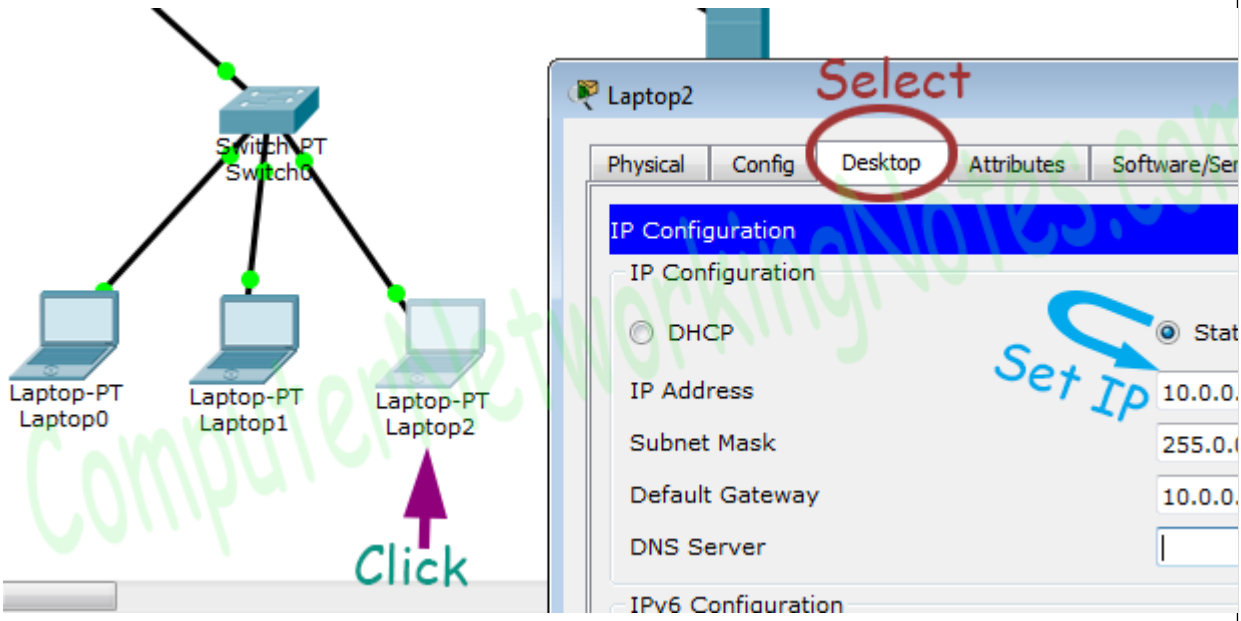
If require, you can download the latest as well as earlier version of Packet Tracer from here. Download Packet Tracer

Initial IP Configuration

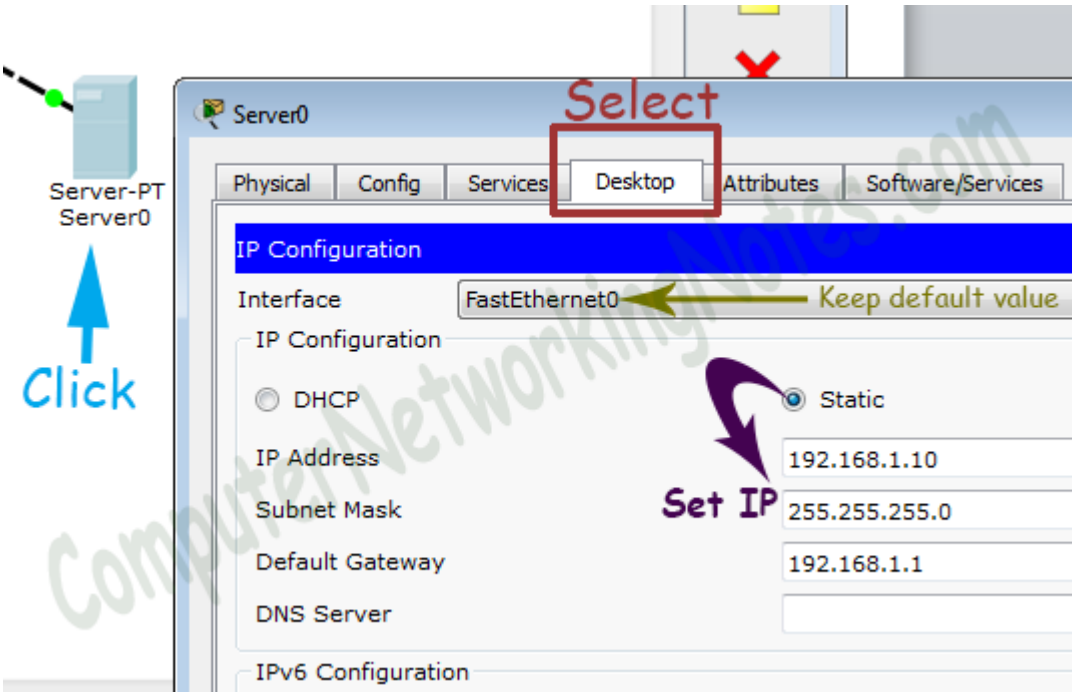
Device / Interface	IP Address
Laotop0	10.0.0.10/8
Laptop1	10.0.0.20/8
Laptop2	10.0.0.30/8
Server0	192.168.1.10/24
Serial 0/0/0 of R1	100.0.0.1/8
Serial 0/0/0 of R2	100.0.0.2/8

If you are following this tutorial on my practice topology, skip this IP configuration section as that topology is already configured with this initial IP configuration

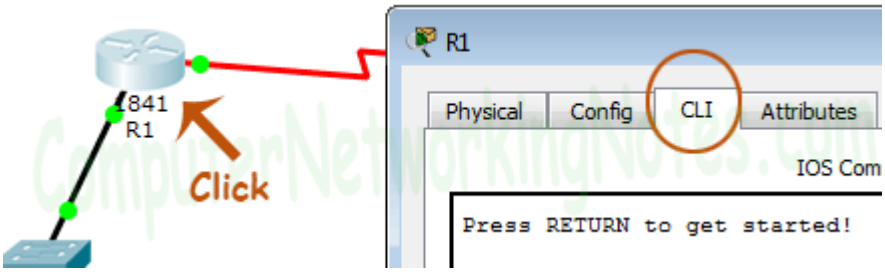
To assign IP address in Laptop click **Laptop** and click **Desktop** and **IP configuration** and Select **Static** and **set IP address** as given in above table.



Following same way configure IP address in Server.



To configure IP address in Router1 click **Router1** and select **CLI** and press **Enter key**.



Two interfaces of Router1 are used in topology; FastEthernet0/0 and Serial 0/0/0.

By default interfaces on router are remain administratively down during the start up. We need to configure IP address and other parameters on interfaces before we could actually use them for routing. Interface mode is used to assign the IP address and other parameters. Interface mode can be accessed from global configuration mode. Following commands are used to access the global configuration mode.

```
Router>enable
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#
```

Before we configure IP address in interfaces let's assign a unique descriptive name to router.

```
Router(config)#hostname R1
R1#
```

Now execute the following commands to set IP address in FastEthernet 0/0 interface.

```
R1(config)#interface FastEthernet0/0
R1(config-if)#ip address 10.0.0.1 255.0.0.0
R1(config-if)#no shutdown
R1(config-if)#exit
```

interface FastEthernet 0/0 command is used to enter in interface mode.

ip address 10.0.0.1 255.0.0.0 command assigns IP address to

interface.

no shutdown command is used to bring the interface up.

exit command is used to return in global configuration mode.

Serial interface needs two additional parameters clock rate and bandwidth. Every serial cable has two ends DTE and DCE. These parameters are always configured at DCE end.

We can use show controllers interface command from privilege mode to check the cable's end.

```
R1(config)#exit
R1#show controllers serial 0/0/0
Interface Serial0/0/0
Hardware is PowerQUICC MPC860
DCE V.35, clock rate 2000000
[Output omitted]
```

Fourth line of output confirms that DCE end of serial cable is attached. If you see DTE here instead of DCE skip these parameters.

Now we have necessary information let's assign IP address to serial interface.

```
R1#configure terminal
R1(config)#interface Serial0/0/0
R1(config-if)#ip address 100.0.0.1 255.0.0.0
R1(config-if)#clock rate 64000
R1(config-if)#bandwidth 64
R1(config-if)#no shutdown
R1(config-if)#exit
R1(config)#
```

Router#configure terminal Command is used to enter in global configuration mode.

Router(config)#interface serial 0/0/0 Command is used to enter in interface mode.

Router(config-if)#ip address 100.0.0.1 255.0.0.0 Command assigns IP address to interface.

Router(config-if)#clock rate 64000

In real life environment this parameter controls the data flow between serial links and need to be set at service provider's end. In lab environment we need not to worry about this value. We can use any valid rate here.

Router(config-if)#bandwidth 64

Bandwidth works as an influencer. It is used to influence the metric calculation of EIGRP or any other routing protocol which uses bandwidth parameter in route selection process.

Router(config-if)#no shutdown Command brings interface up.

Router(config-if)#exit Command is used to return in global configuration mode.

We will use same commands to assign IP addresses on interfaces of Router2. We need to provided clock rate and bandwidth only on DCE side of serial interface. Following command will assign IP addresses on interface of Router2.

Initial IP configuration in R2

```
Router>enable
Router#configure terminal
Router(config)#hostname R2
R2(config)#interface FastEthernet0/0
R2(config-if)#ip address 192.168.1.1 255.255.255.0
R2(config-if)#no shutdown
R2(config-if)#exit
R2(config)#interface Serial0/0/0
R2(config-if)#ip address 100.0.0.2 255.0.0.0
R2(config-if)#no shutdown
R2(config-if)#exit
R2(config)#
```

That's all initial IP configuration we need. Now this topology is ready for the practice of static nat.

Configure Static NAT

Static NAT configuration requires three steps: -

1. Define IP address mapping
2. Define inside local interface
3. Define inside global interface

Since static NAT use manual translation, we have to map each inside local IP address (which needs a translation) with inside global IP address. Following command is used to map the inside local IP address with inside global IP address.

```
Router(config)#ip nat inside source static [inside local ip address] [inside global IP address]
```

For example in our lab Laptop1 is configured with IP address 10.0.0.10. To map it with 50.0.0.10 IP address we will use following command

```
Router(config)#ip nat inside source static 10.0.0.10 50.0.0.10
```

In second step we have to define which interface is connected with local the network. On both routers interface Fa0/0 is connected with the local network which need IP translation.

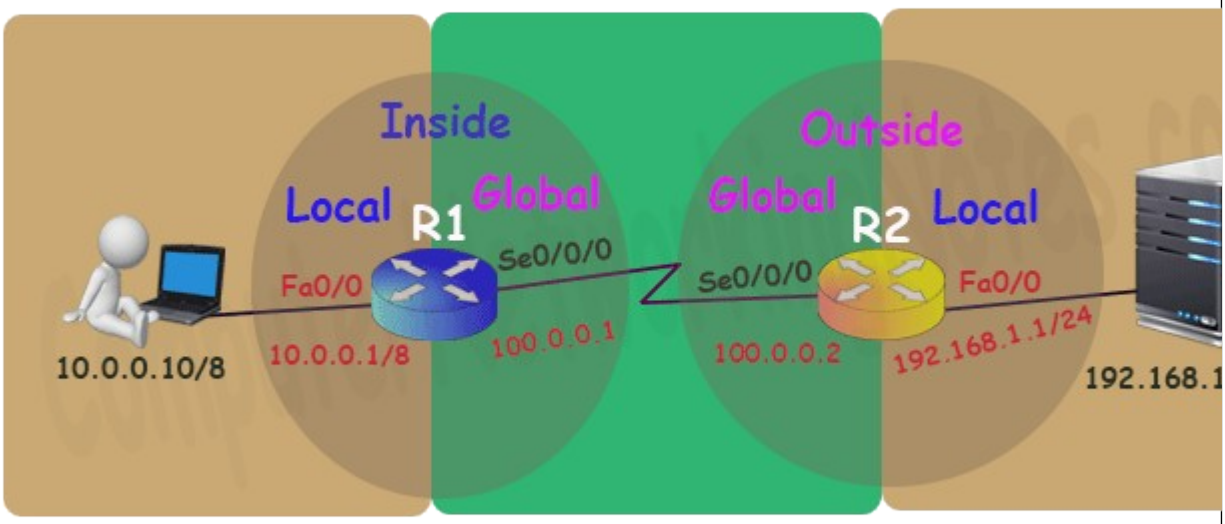
Following command will define interface Fa0/0 as inside local.

```
Router(config-if)#ip nat inside
```

In third step we have to define which interface is connected with the global network. On both routers serial 0/0/0 interface is connected with the global network. Following command will define interface Serial0/0/0 as inside global.

```
Router(config-if)#ip nat outside
```

Following figure illustrates these terms.



Let's implement all these commands together and configure the static NAT.

R1 Static NAT Configuration

```
R1(config)#ip nat inside source static 10.0.0.10 50.0.0.10
R1(config)#interface FastEthernet 0/0
R1(config-if)#ip nat inside
R1(config-if)#exit
R1(config)#
R1(config)#interface Serial 0/0/0
R1(config-if)#ip nat outside
R1(config-if)#exit
```

For testing purpose I configured only one static translation. You may use following commands to configure the translation for remaining address.

```
R1(config)#ip nat inside source static 10.0.0.20 50.0.0.20
R1(config)#ip nat inside source static 10.0.0.30 50.0.0.30
```

R2 Static NAT Configuration

```
R2(config)#ip nat inside source static 192.168.1.10 200.0.0.10
R2(config)#interface FastEthernet 0/0
R2(config-if)#ip nat inside
R2(config-if)#exit
R2(config)#
R2(config)#interface Serial 0/0/0
R2(config-if)#ip nat outside
```

```
R2(config-if)#exit
```

Before we test this lab we need to configure the IP routing. IP routing is the process which allows router to route the packet between different networks. Following tutorial explain routing in detail with examples

Routing concepts Explained with Examples

Configure static routing in R1

```
R1(config)#ip route 200.0.0.0 255.255.255.0 100.0.0.2
```

Configure static routing in R2

```
R2(config)#ip route 50.0.0.0 255.0.0.0 100.0.0.1
```

Testing Static NAT Configuration

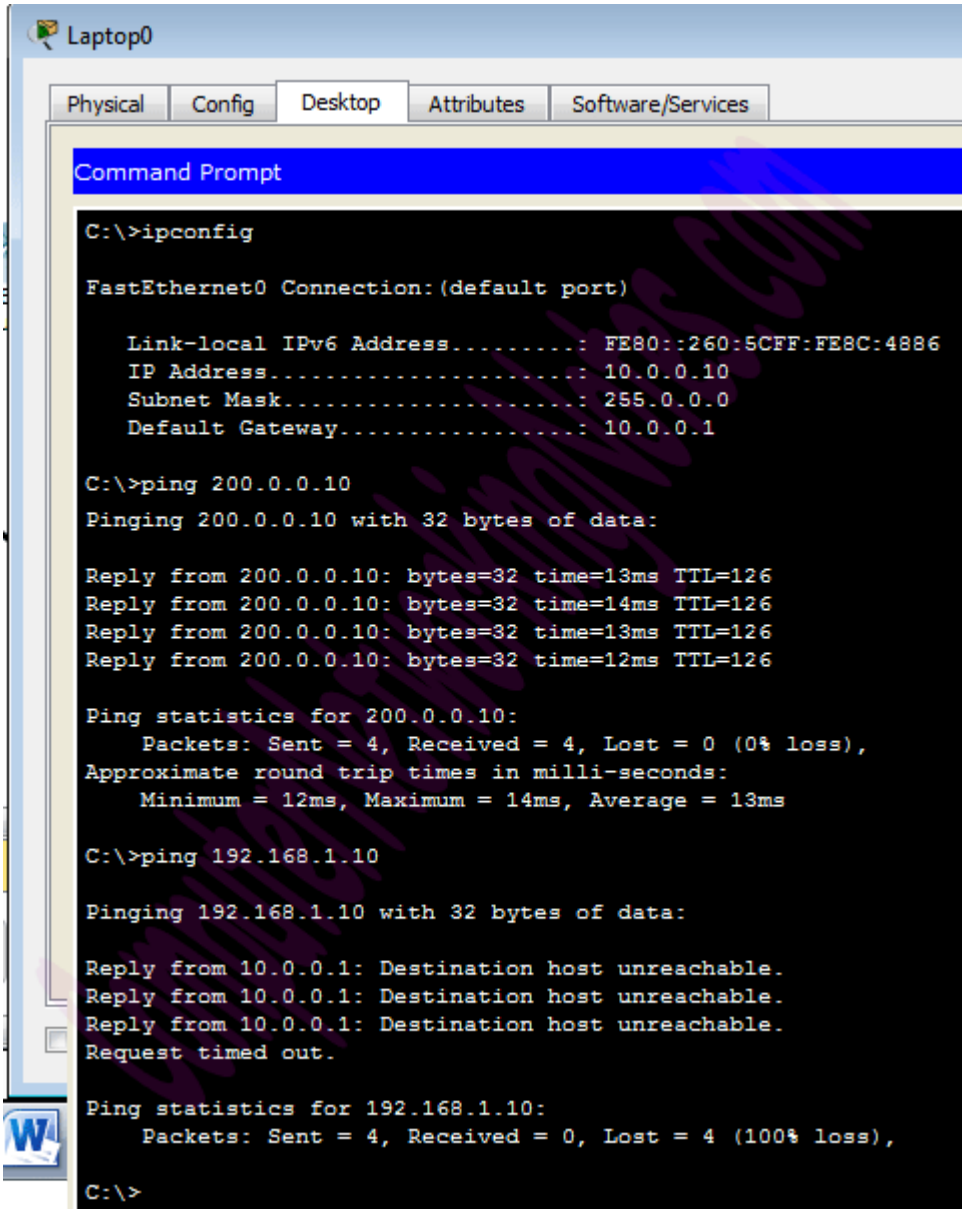
In this lab we configured static NAT on R1 and R2. On R1 we mapped inside local IP address 10.0.0.10 with inside global address 50.0.0.10 while on R2 we mapped inside local IP address 192.168.1.10 with inside global IP address 200.0.0.10.

Device	Inside Local IP Address	Inside Global IP Addres
Laptop0	10.0.0.10	50.0.0.10
Server	192.168.1.10	200.0.0.10

To test this setup click Laptop0 and Desktop and click Command Prompt.

Run **ipconfig** command.
Run **ping 200.0.0.10** command.

Run **ping 192.168.1.10** command.



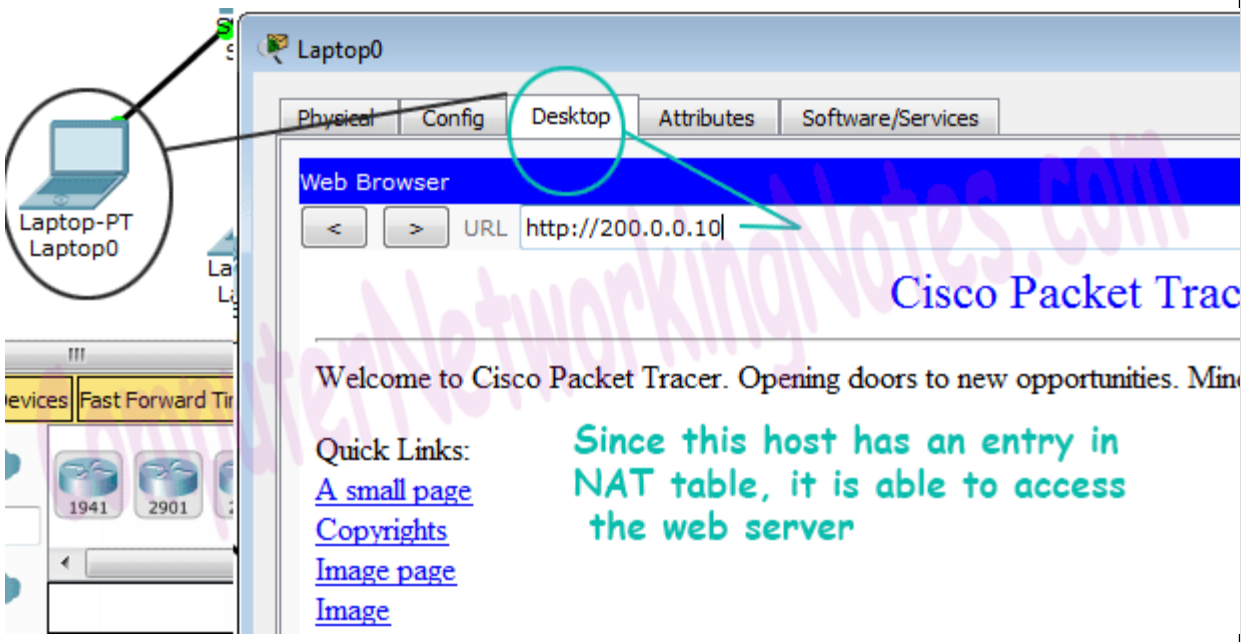
First command verifies that we are testing from correct NAT device.

Second command checks whether we are able to access the remote device or not. A ping reply confirms that we are able to connect with remote device on this IP address.

Third command checks whether we are able to access the remote device on its actual IP address or not. A ping error confirms that we are not able to

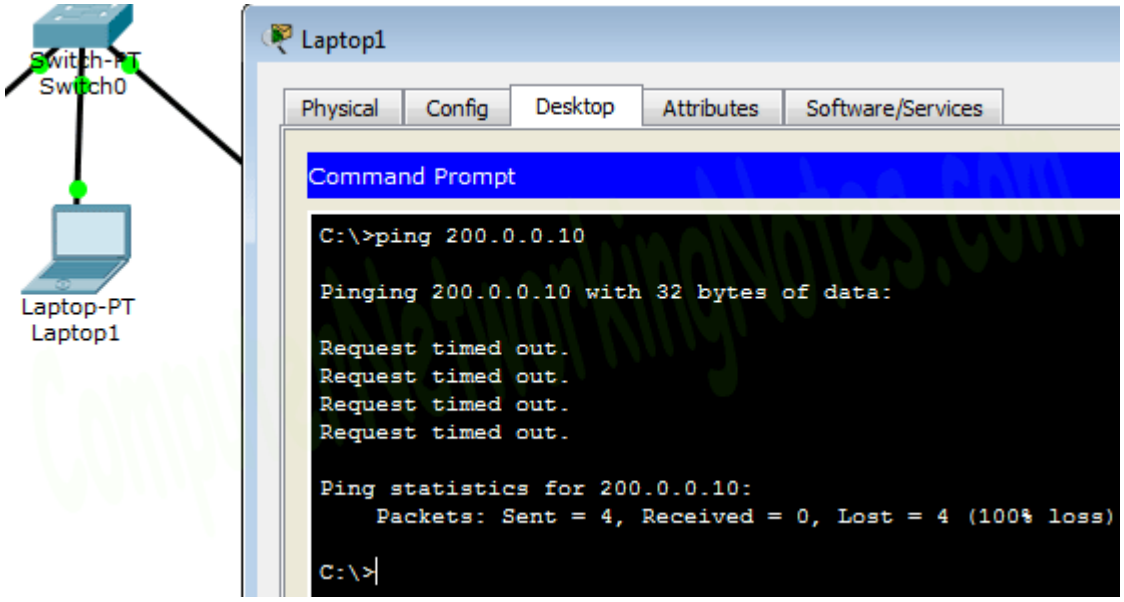
connect with remote device on this IP address.

Let's do one more testing. Click **Laptop0** and click **Desktop** and click **Web Browser** and access 200.0.0.10.



Above figure confirms that host 10.0.0.10 is able to access the 200.0.0.10.

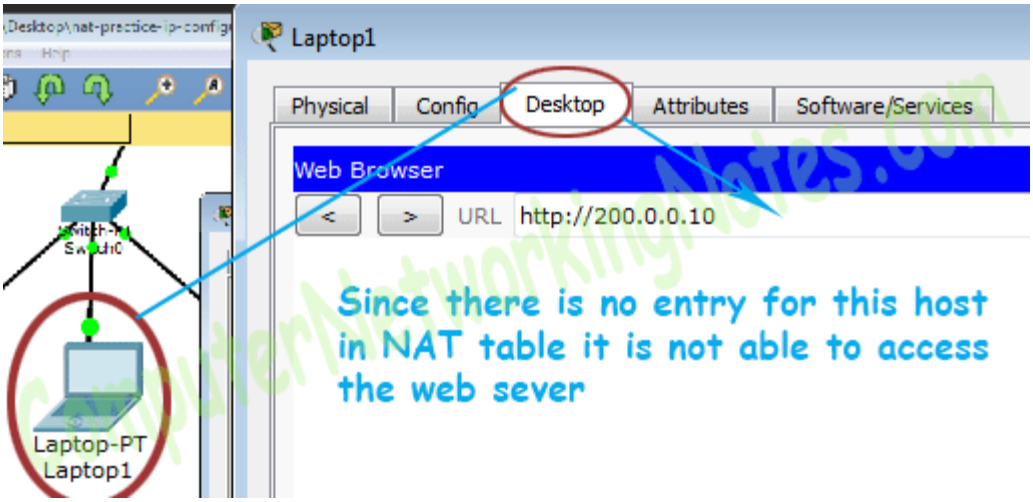
Now run **ping 200.0.0.10** command from Laptop1.



Why we are not able to connect with the remote device from this host?

Because we configured NAT only for one host (Laptop0) which IP address is 10.0.0.10. So only the host 10.0.0.10 will be able to access the remote device.

To confirm it again, let's try to access web service from this host.



If you followed this tutorial step by step, you should get the same output of testing. Although it's very rare but some time you may get different output. To figure out what went wrong you can use my practice topology with all

above configuration. Download my practice topology

Download NAT Practice LAB with Static NAT configuration

We can also verify this translation on router with ***show ip nat translation*** command.

Following figure illustrate this translation on router R1.

```
R1#show ip nat translations
Pro  Inside global     Inside local       Outside local      Outside global
icmp 50.0.0.10:13      10.0.0.10:13       200.0.0.10:13      200.0.0.10:13
icmp 50.0.0.10:14      10.0.0.10:14       200.0.0.10:14      200.0.0.10:14
icmp 50.0.0.10:15      10.0.0.10:15       200.0.0.10:15      200.0.0.10:15
icmp 50.0.0.10:16      10.0.0.10:16       200.0.0.10:16      200.0.0.10:16
tcp 50.0.0.10:1030     10.0.0.10:1030     200.0.0.10:80      200.0.0.10:80
tcp 50.0.0.10:1031     10.0.0.10:1031     200.0.0.10:80      200.0.0.10:80

R1#
```

Following figure illustrate this translation on router R2

```
R2#show ip nat translations
Pro  Inside global     Inside local       Outside local      Outside global
icmp 200.0.0.10:13     192.168.1.10:13    50.0.0.10:13       50.0.0.10:13
icmp 200.0.0.10:14     192.168.1.10:14    50.0.0.10:14       50.0.0.10:14
icmp 200.0.0.10:15     192.168.1.10:15    50.0.0.10:15       50.0.0.10:15
icmp 200.0.0.10:16     192.168.1.10:16    50.0.0.10:16       50.0.0.10:16
tcp 200.0.0.10:80      192.168.1.10:80    50.0.0.10:1030     50.0.0.10:1030
tcp 200.0.0.10:80      192.168.1.10:80    50.0.0.10:1031     50.0.0.10:1031

R2#
```

Pay a little bit extra attention on outside local address filed. Have you noticed one interesting feature of NAT in above output? Why actual outside local IP address is not listed in this filed?

The actual IP address is not listed here because router is receiving packets after the translation. From R1's point of view remote device's IP address is 200.0.0.10 while from R2's point of view end device's IP address is 50.0.0.10.

This way if NAT is enabled we would not be able to trace the actual end device.

That's all for this tutorial. In next part we will learn dynamic NAT

configuration step by step with examples.

CONCLUSION: Thus we have simulated RIP, OSPF, EIGRP routing protocols

Configuration on Router R1

we are going to assign ip address on both ethernet port and serial port,hostname,up port,set clock rate on serial port,assign bandwith on router R1.

```
Router>enable
Router#config terminal
Enter configuration commands, one per line.  End with CNTL/Z.

R!(config)#hostname R1
R1(config)#int fa0/0
R1(config-if)#ip address 10.0.0.1 255.0.0.0
R1(config-if)#no shutdown

%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up

R1(config-if)#exit


R1(config)#int se0/0/0
R1(config-if)#ip address  20.0.0.1 255.0.0.0
R1(config-if)#clock rate 64000
R1(config-if)#bandwidth 64
R1(config-if)#no shutdown

%LINK-5-CHANGED: Interface Serial0/0/0, changed state to down
R1(config-if)#exit

R1(config)#router rip
R1(config-router)#version 2
R1(config-router)#network 10.0.0.0
```

```
R1(config-router)#network 20.0.0.0
R1(config-router)#exit
```

Configuration on Router R2

assign ip address on both ethernet and serial port,assign bandwith,clock rate on router R2.

```
Router>enable
Router#config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#hostname R2
R2(config)#interface se0/0/0
R2(config-if)#ip address 20.0.0.2 255.0.0.0
R2(config-if)#bandwidth 64
R2(config-if)#no shutdown

%LINK-5-CHANGED: Interface Serial0/0/0, changed state to up

R2(config-if)#exit
```

Now Configure Router RIP and EIGRP on Router R2.

```
R2(config)#router rip
R2(config-router)#version 2
R2(config-router)#network 20.0.0.0
R2(config-router)#network 30.0.0.0
R2(config-router)#exit

R2(config)#router eigrp 1
R2(config-router)#network 20.0.0.0
R2(config-router)#network 30.0.0.0
R2(config-router)#exit

R2(config)#interface se0/0/1
R2(config-if)#ip address 30.0.0.1 255.0.0.0
R2(config-if)#bandwidth 64
R2(config-if)#clock rate 64000
R2(config-if)#no shutdown
```

Configuration on Router R3

We assigned hostname and address firts and then we up the network interface.

```
Router>enable
Router#config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#hostname R3
R3(config)#int fastEthernet 0/0

R3(config-if)#ip address 40.0.0.1 255.0.0.0
R3(config-if)#no shutdown

%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state
to up

R3(config-if)#exit
R3(config)#int se0/0/0
R3(config-if)#ip address 30.0.0.2 255.0.0.0

R3(config-if)#bandwidth 64

R3(config-if)#clock rate 64000
R3(config-if)#no shutdown

%LINK-5-CHANGED: Interface Serial0/0/0, changed state to up

R3(config-if)#router rip
R3(config-router)#version 2
R3(config-router)#network 30.0.0.0
R3(config-router)#network 40.0.0.0
R3(config-router)#exit

R3(config)#router eigrp 1
R3(config-router)#network 30.0.0.0
R3(config-router)#
%DUAL-5-NBRCHANGE: IP-EIGRP 1: Neighbor 30.0.0.1 (Serial0/0/0) is up: new
adjacency

R3(config-router)#network 40.0.0.0
R3(config-router)#exit
```

Troubleshooting PART:

Troubleshooting is the major part of a network admin life or a system admin life.To find error and solve it you must be good in trobleshooting.Once you have successfully configured routing on a router you need to verify or troubleshoot

whether your configured task is working properly or not.

```
R1#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

C    10.0.0.0/8 is directly connected, FastEthernet0/0
C    20.0.0.0/8 is directly connected, Serial0/0/0
R    30.0.0.0/8 [120/1] via 20.0.0.2, 00:00:12, Serial0/0/0
R    40.0.0.0/8 [120/2] via 20.0.0.2, 00:00:12, Serial0/0/0
```

how to see neighbour Table?

```
R2#show ip eigrp neighbors
IP-EIGRP neighbors for process 1
H   Address         Interface     Hold Uptime    SRTT   RTO   Q   Seq
                                  (sec)          (ms)       Cnt  Num
0   30.0.0.2        Se0/0/1       13   00:30:07  40     1000  0   3
```

How to see the type and and number of packets sent or recieved?

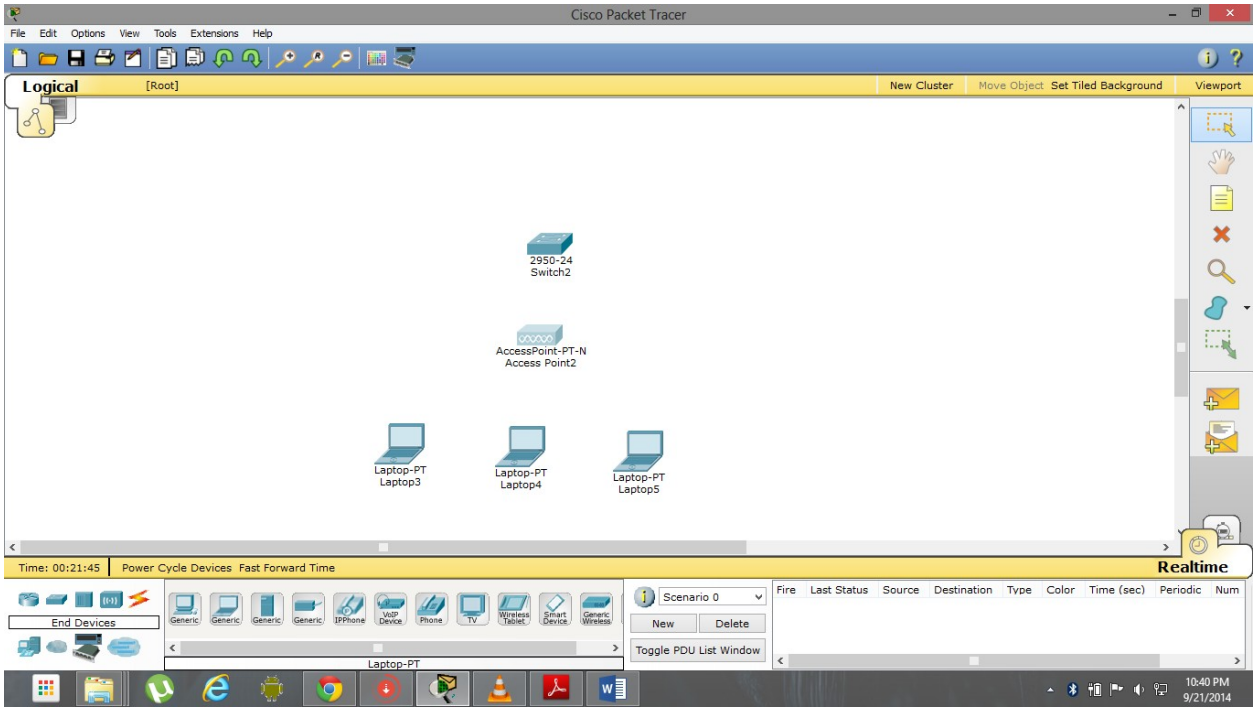
```
R2#show ip eigrp traffic
IP-EIGRP Traffic Statistics for process 1
  Hellos sent/received: 835/416
  Updates sent/received: 4/2
  Queries sent/received: 0/0
  Replies sent/received:  0/0
  Acks sent/received:  2/3
  Input queue high water mark 1, 0 drops
  SIA-Queries sent/received: 0/0
  SIA-Replies sent/received: 0/0
```

how to see the information of running proceess on interfaces? Below command will show you detail information on interfaces taking part in eigrp process.

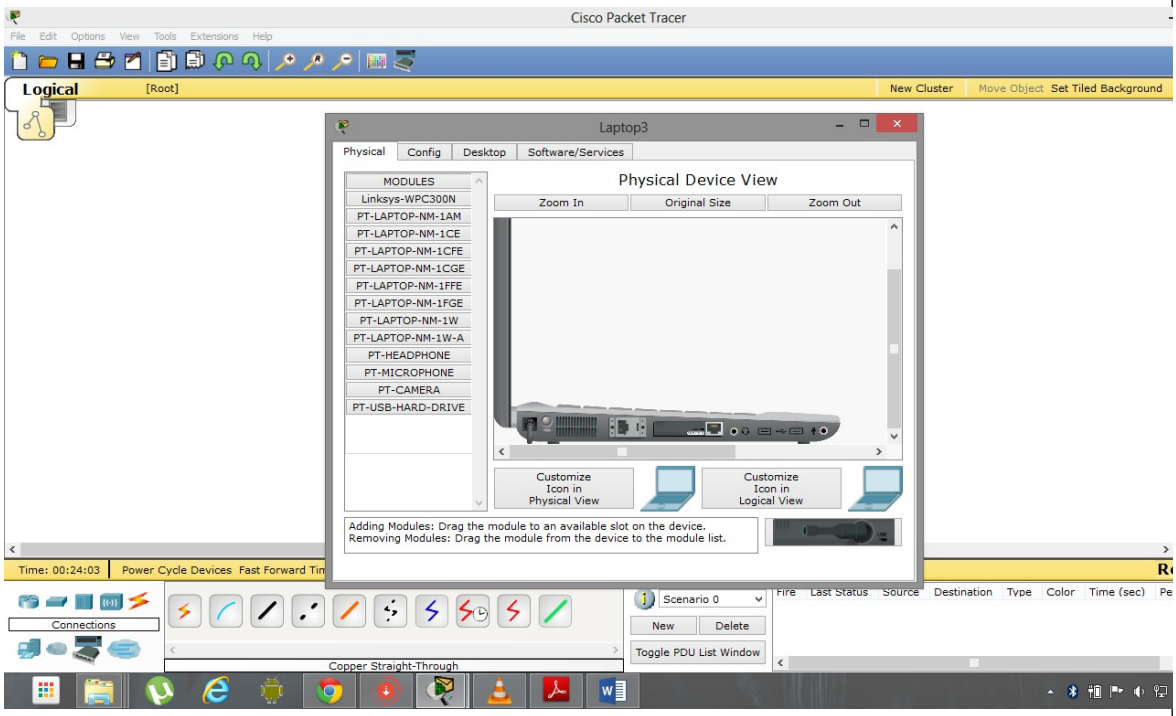
```
R2#show ip eigrp  interfaces
IP-EIGRP interfaces for process 1

                  Xmit Queue   Mean   Pacing Time   Multicast    Pending
Interface         Peers  Un/Reliable  SRTT  Un/Reliable  Flow Timer  Routes
Se0/0/1             1       0/0      1236       0/10          0          0
Se0/0/0             0       0/0      1236       0/10          0
```

B. CWLAN with static IP addressing and DHCP with MAC security and filters

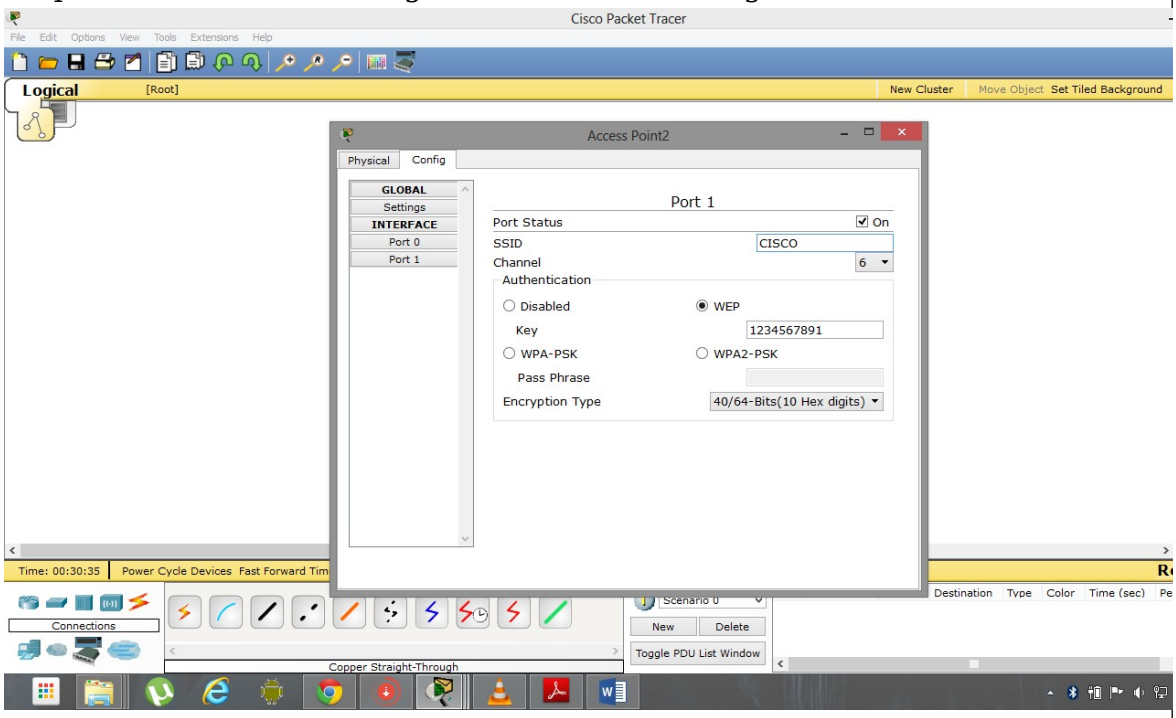
Step 1: To create star topology, take one wireless Access point, switch and three laptops

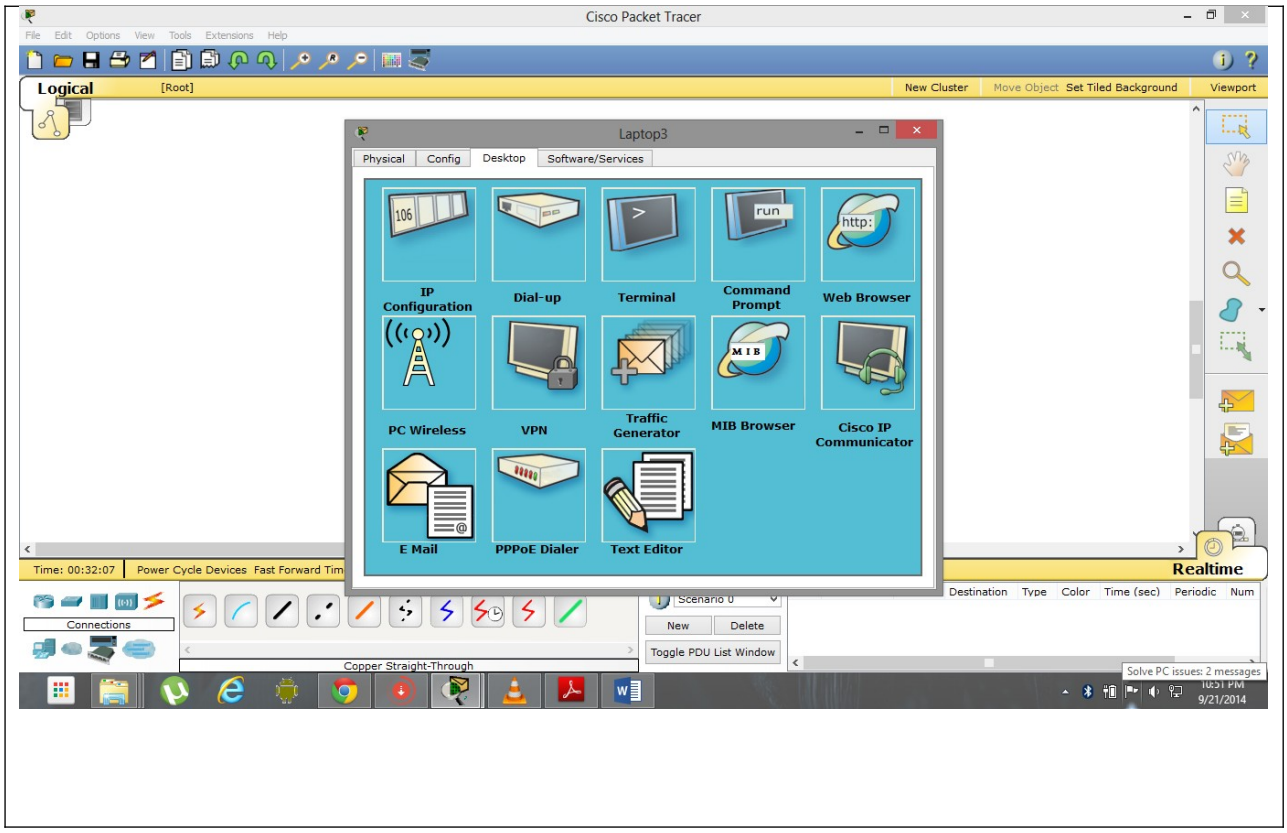


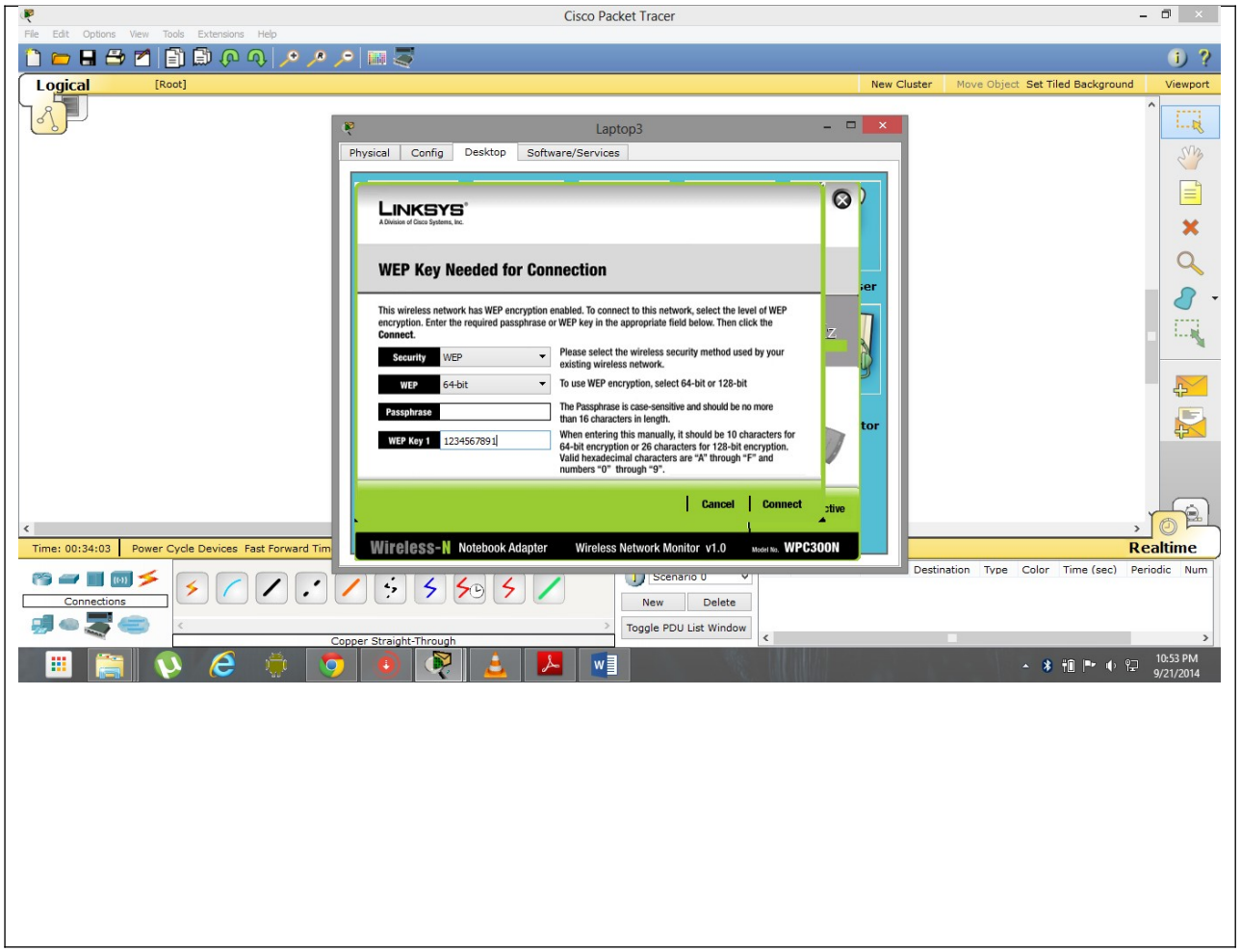
Step 2: Connect switch and AP by straight thru cable
Step 3: go to laptop's physical mode, make the button off as shown in below diagram and remove physical port and connect wireless port LINKAGE-WPC300N(In modules it is given. And switch on the button. Give the IP address statically to all the laptops.

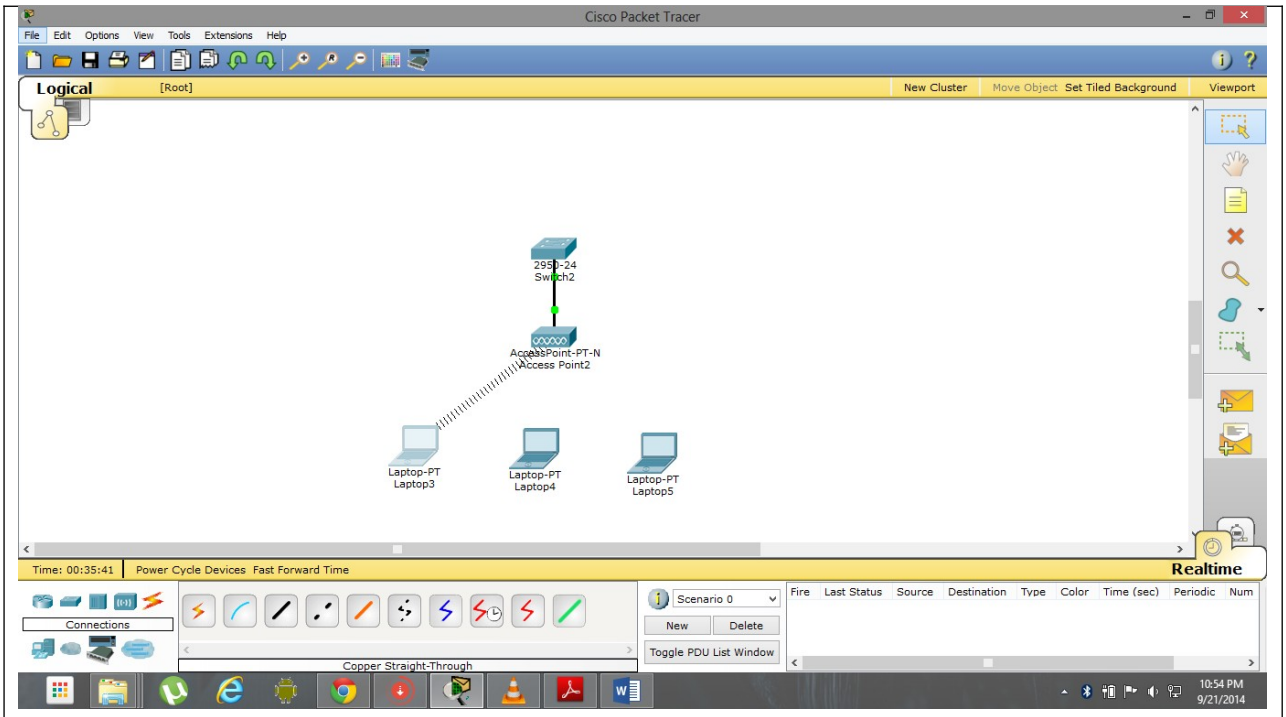


Step 4: In AP , do the configuration as shown in figure



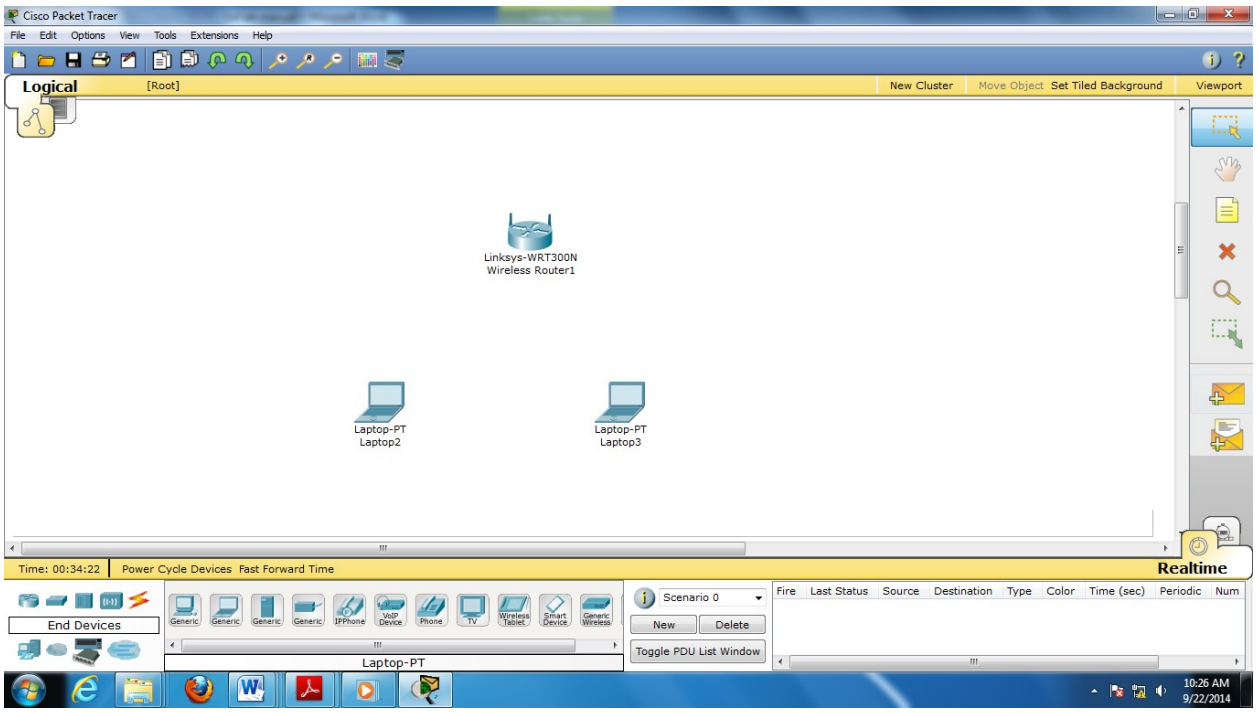

Cisco Packet Tracer
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Laptop3
Physical
Config
Desktop
Software/Services
IP Configuration
Dial-up
Terminal
Command Prompt
Web Browser
PC Wireless
VPN
Traffic Generator
MIB Browser
Cisco IP Communicator
E Mail
PPPoE Dialer
Text Editor
Time: 00:32:07
Power Cycle Devices
Fast Forward Tim
Realtime
Connections
Copper Straight-Through
Scenario 0
New
Delete
Toggle PDU List Window
Destination
Type
Color
Time (sec)
Periodic
Num
Solve PC issues: 2 messages
9/21/2014


Cisco Packet Tracer
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Laptop3
Physical
Config
Desktop
Software/Services
LINKSYS
A Division of Cisco Systems, Inc.
WEP Key Needed for Connection
This wireless network has WEP encryption enabled. To connect to this network, select the level of WEP encryption. Enter the required passphrase or WEP key in the appropriate field below. Then click the Connect.
Security
WEP
Please select the wireless security method used by your existing wireless network.
WEP
64-bit
To use WEP encryption, select 64-bit or 128-bit
Passphrase
The Passphrase is case-sensitive and should be no more than 16 characters in length.
WEP Key 1
1234567891
When entering this manually, it should be 10 characters for 64-bit encryption or 26 characters for 128-bit encryption. Valid hexadecimal characters are "A" through "F" and numbers "0" through "9".
Cancel
Connect
Wireless-N Notebook Adapter
Wireless Network Monitor v1.0
Model No. WPC300N
Time: 00:34:03
Power Cycle Devices
Fast Forward Tim
Realtime
Connections
Copper Straight-Through
Scenario 0
New
Delete
Toggle PDU List Window
Destination
Type
Color
Time (sec)
Periodic
Num
10:53 PM
9/21/2014

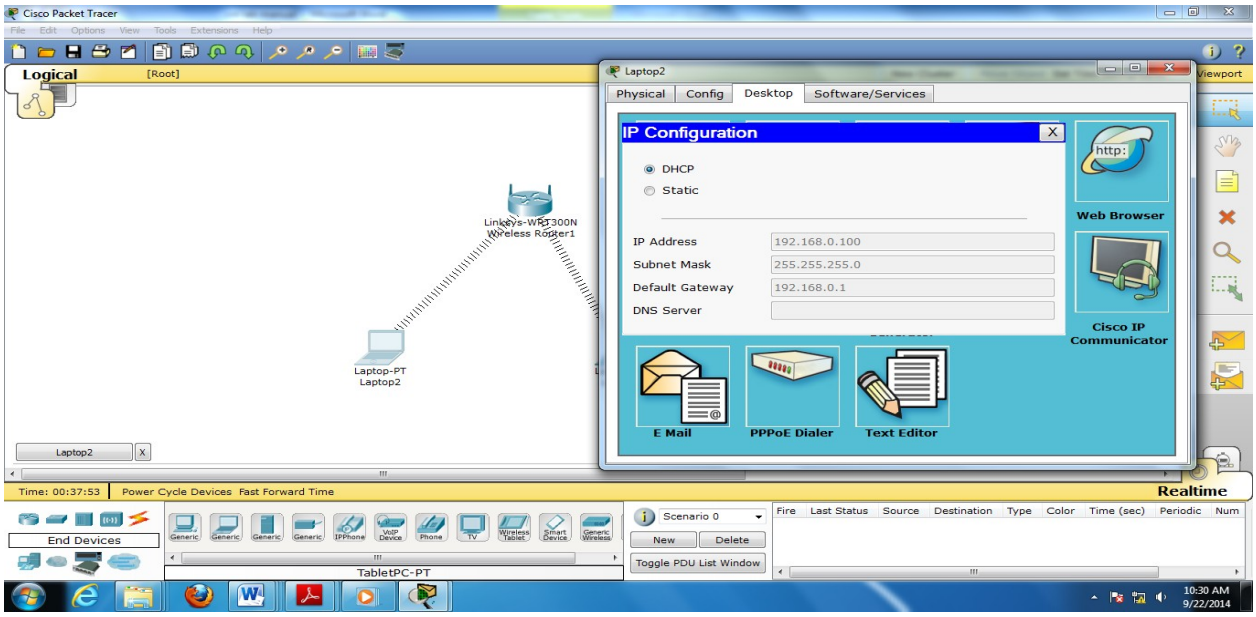


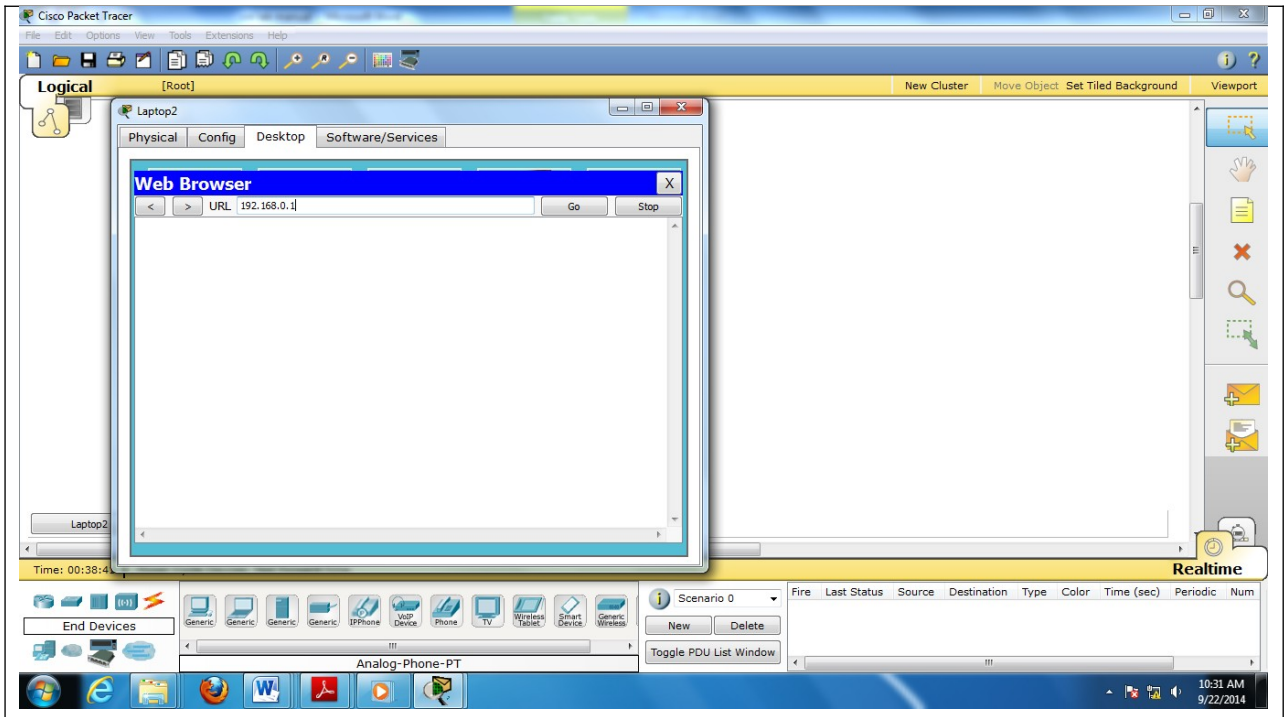
DHCP with MAC security and filters

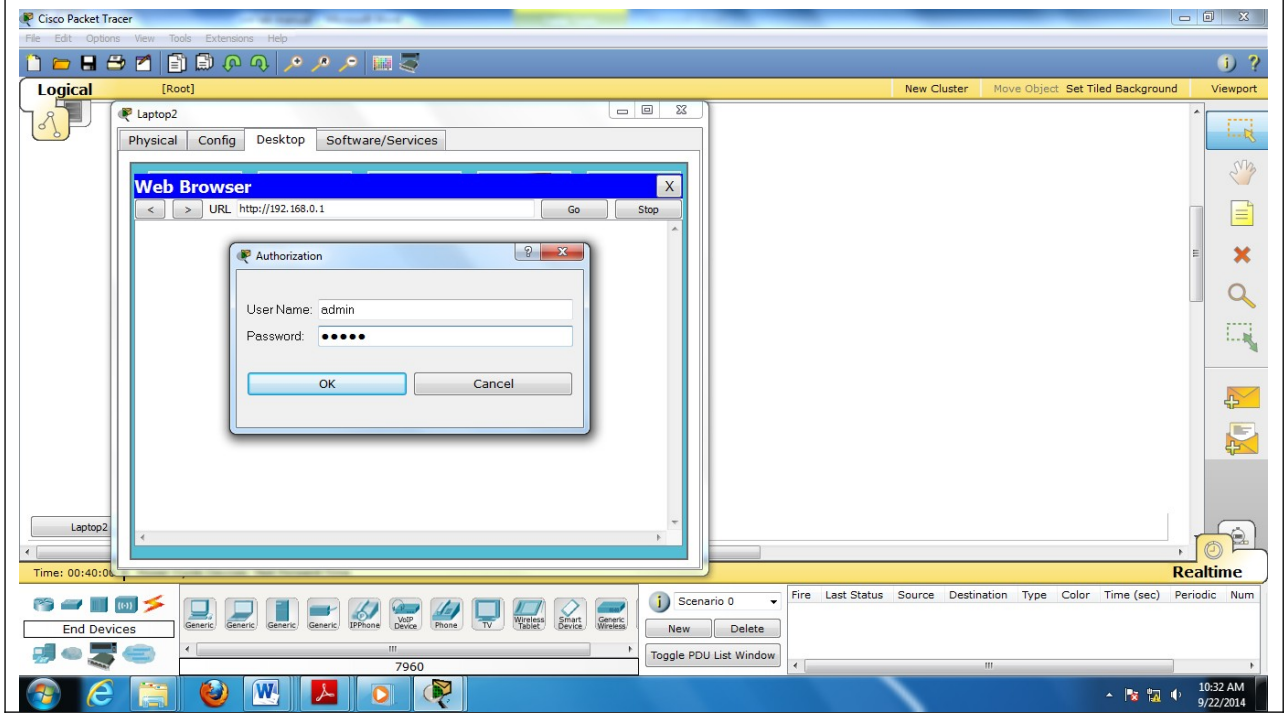
Step1: Create the topology as shown in below.

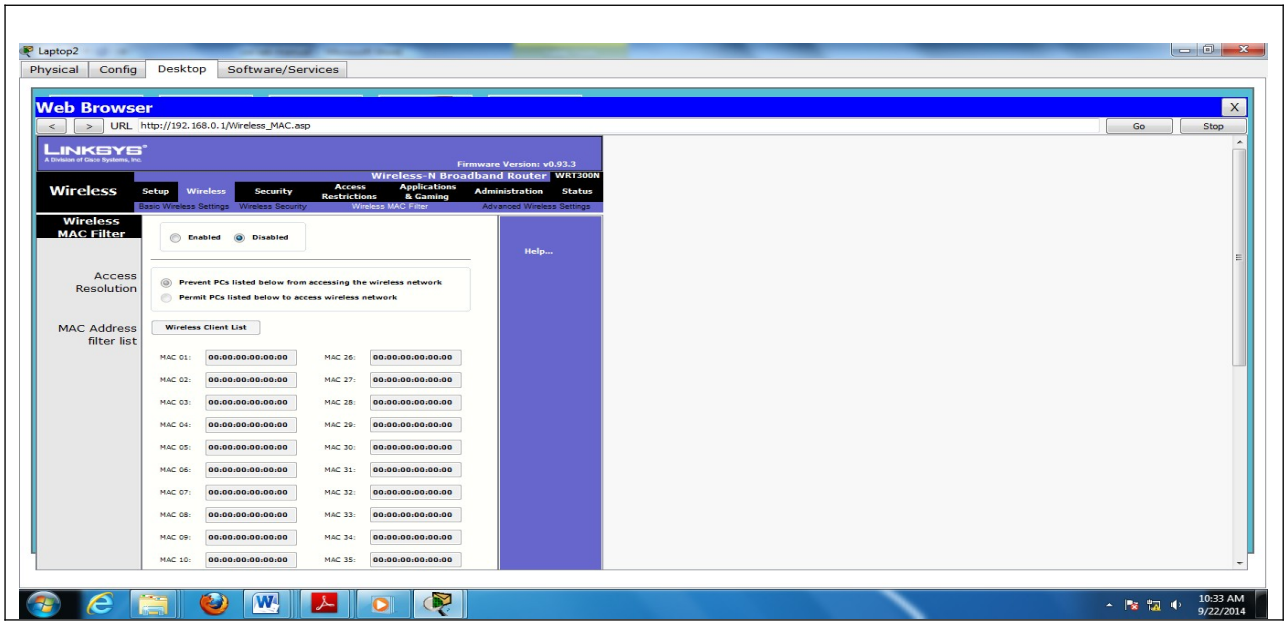


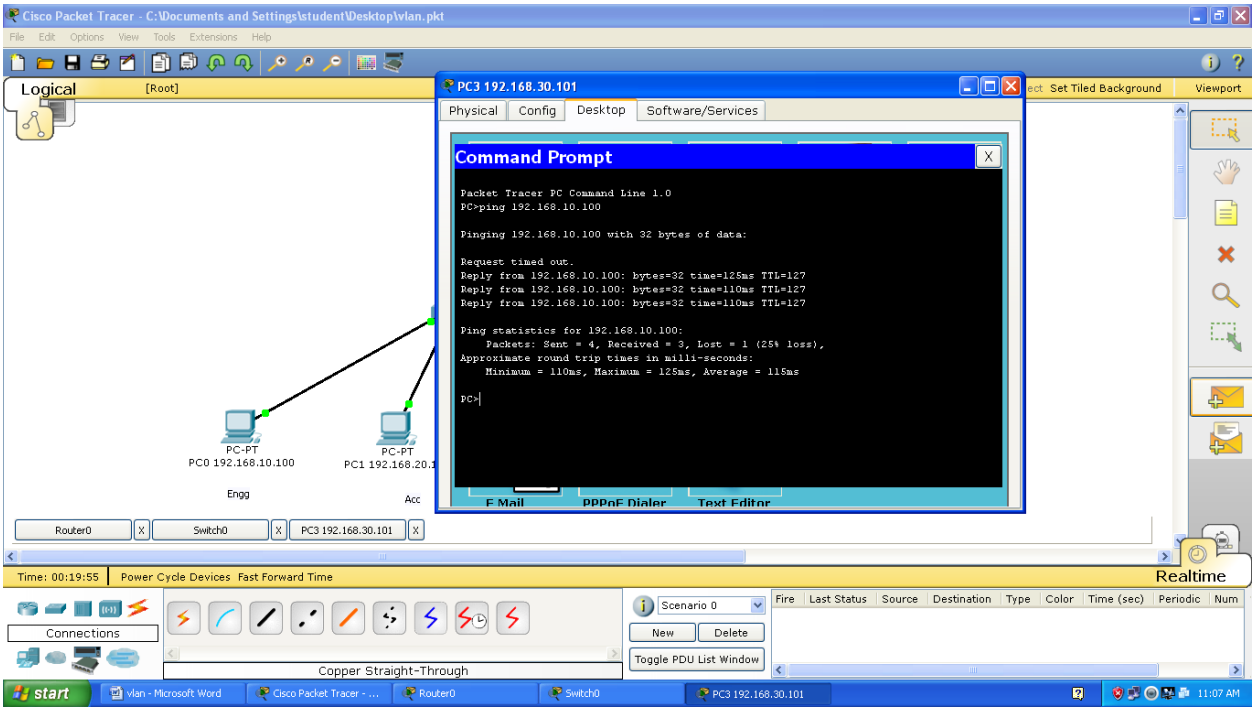
Step 2: It will give the IP addresses automatically using DHCP
Step3: In laptop's web server type the IP address of router i.e 192.168.0.1
Step 4: click on Go. User name and password is admin in this case
Step 5: go to wirless-> wirless MAC filters
Step6: give the MAC address in the field wireless client list. The MAC address is permanent address of particular system. By using ipconfig /all we can get the MAC address given in snap
Step7: If the MAC is 0007.EC51.D93B
Make the format 00:07:EC:51:D9:3B
Save the setting.
By using this procedure security to the wirless network is achieved using MAC filtering.
Step 8: Last snap shows how the laptop having MAC 00:07:EC:51:D9:3B
is disconnected.




Cisco Packet Tracer
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Laptop2
Physical
Config
Desktop
Software/Services
Web Browser
URL 192.168.0.1
Go
Stop
Realtime
End Devices
Analog-Phone-PT
Scenario 0
New
Delete
Toggle PDU List Window
Fire
Last Status
Source
Destination
Type
Color
Time (sec)
Periodic
Num
10:31 AM
9/22/2014


Cisco Packet Tracer
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Laptop2
Physical
Config
Desktop
Software/Services
Web Browser
URL http://192.168.0.1
Go
Stop
Authorization
User Name: admin
Password:
OK
Cancel
Realtime
End Devices
7960
Scenario 0
New
Delete
Toggle PDU List Window
Fire
Last Status
Source
Destination
Type
Color
Time (sec)
Periodic
Num
10:32 AM
9/22/2014


Laptop2
Physical
Config
Desktop
Software/Services
Web Browser
URL http://192.168.0.1/Wireless_MAC.asp
Go
Stop
LINKSYS
Wireless-N Broadband Router
WRT300N
Wireless
Setup
Wireless
Security
Access Restrictions
Applications & Gaming
Administration
Status
Wireless MAC Filter
Enabled
Disabled
Help...
Access Resolution
Prevent PCs listed below from accessing the wireless network
Permit PCs listed below to access wireless network
MAC Address filter list
Wireless Client List
10:33 AM
9/22/2014


Cisco Packet Tracer - C:\Documents and Settings\student\Desktop\vlan.pkt
Logical
[Root]
PC3 192.168.30.101
Physical
Config
Desktop
Software/Services
Command Prompt
Packet Tracer PC Command Line 1.0
PC>ping 192.168.10.100
Pinging 192.168.10.100 with 32 bytes of data:
Request timed out.
Reply from 192.168.10.100: bytes=32 time=125ms TTL=127
Reply from 192.168.10.100: bytes=32 time=110ms TTL=127
Reply from 192.168.10.100: bytes=32 time=110ms TTL=127
Ping statistics for 192.168.10.100:
Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
Minimum = 110ms, Maximum = 125ms, Average = 115ms
PC>
PC-PT
PC0 192.168.10.100
PC-PT
PC1 192.168.20.1
Engg
Acc
Router0
Switch0
PC3 192.168.30.101
Time: 00:19:55
Power Cycle Devices
Fast Forward Time
Realtime
Connections
Copper Straight-Through
Scenario 0
New
Delete
Toggle PDU List Window
Fire
Last Status
Source
Destination
Type
Color
Time (sec)
Periodic
Num
start
11:07 AM

D Y PATIL INSTITUTE OF ENGINEERING & TECHNOLOGY AMBI, PUNE	**EXPERIMENT TITLE:** Using a Network Simulator Configure VLAN,OSPF and NAT	
DEPARTMENT OF INFORMATION TECHNOLOGY		
EXPERIMENT NO. : DYPIET/IT/	**SEMESTER** : VI (TE)	**PAGE:**

How to Configure Dynamic NAT in Cisco Router

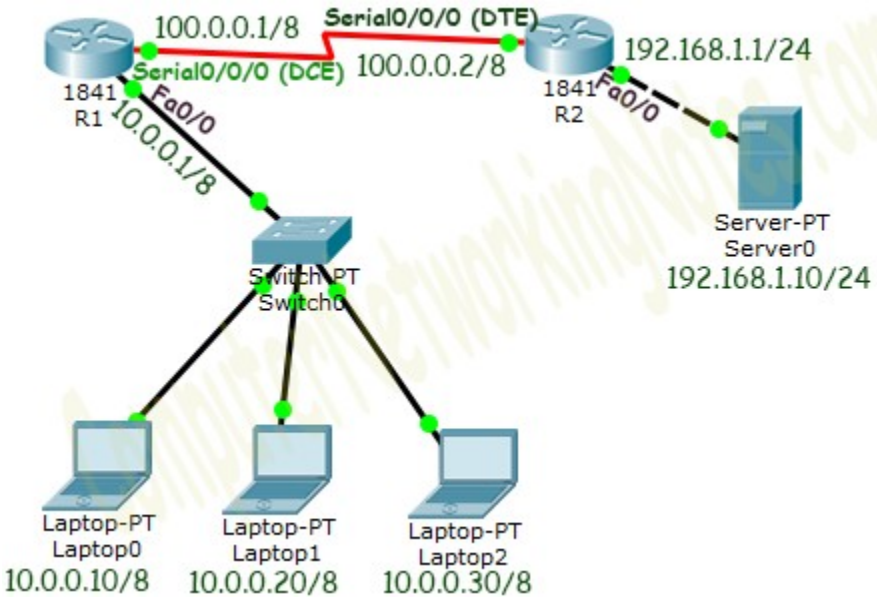
This tutorial explains Dynamic NAT configuration (creating an access list of IP addresses which need translation, creating a pool of available IP address, mapping access list with pool and defining inside and outside interfaces) in detail. Learn how to configure, manage, verify and debug dynamic NAT step by step with packet tracer examples.

To explain Dynamic NAT configuration, I will use packet tracer network simulator software. You can use any network simulator software to follow this guide. There is no difference in output as long as your selected software contains the commands explained in this tutorial.

Create a practice lab as shown in following figure or download this pre-created practice lab and load in packet tracer

Download NAT Practice LAB with initial IP configuration

If require, you can download the latest as well as earlier version of Packet Tracer from here. Download Packet Tracer



This tutorial is the third part of our article "**Learn NAT (Network Address Translation) Step by Step in Easy Language with Examples**". You can read other parts of this article here.

Basic Concepts of NAT Exaplained in Easy Language
This tutorial is the first part of this article. This tutorial explains basic concepts of static nat, dynamic nat, pat, inside local, outside local, inside global and outside global in detail with examples.
How to Configure Static NAT in Cisco Router
This tutorial is the second part of this article. This tutorial explains how to configure Static NAT (Network Address Translation) in Cisco Router step by step with packet tracer examples.
Configure PAT in Cisco Router with Examples
This tutorial is the last part of this article. This tutorial explains how to configure PAT (Port Address Translation) in Cisco Router step by step with packet tracer examples.

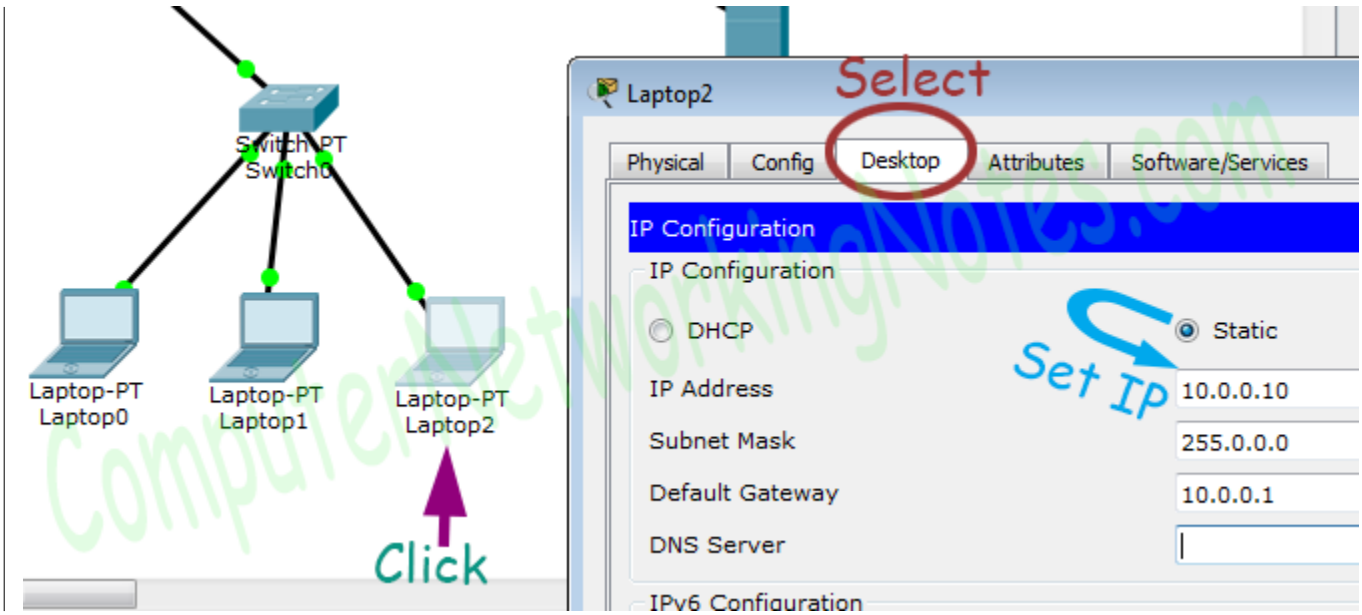
Initial IP Configuration

Device / Interface	IP Address	Connected
Laotop0	10.0.0.10/8	Fa0/0 of R0
Laptop1	10.0.0.20/8	Fa0/0 of R0
Laptop2	10.0.0.30/8	Fa0/0 of R0
Server0	192.168.1.10/24	Fa0/0 of R1
Serial 0/0/0 of R1	100.0.0.1/8	Serial 0/0/0

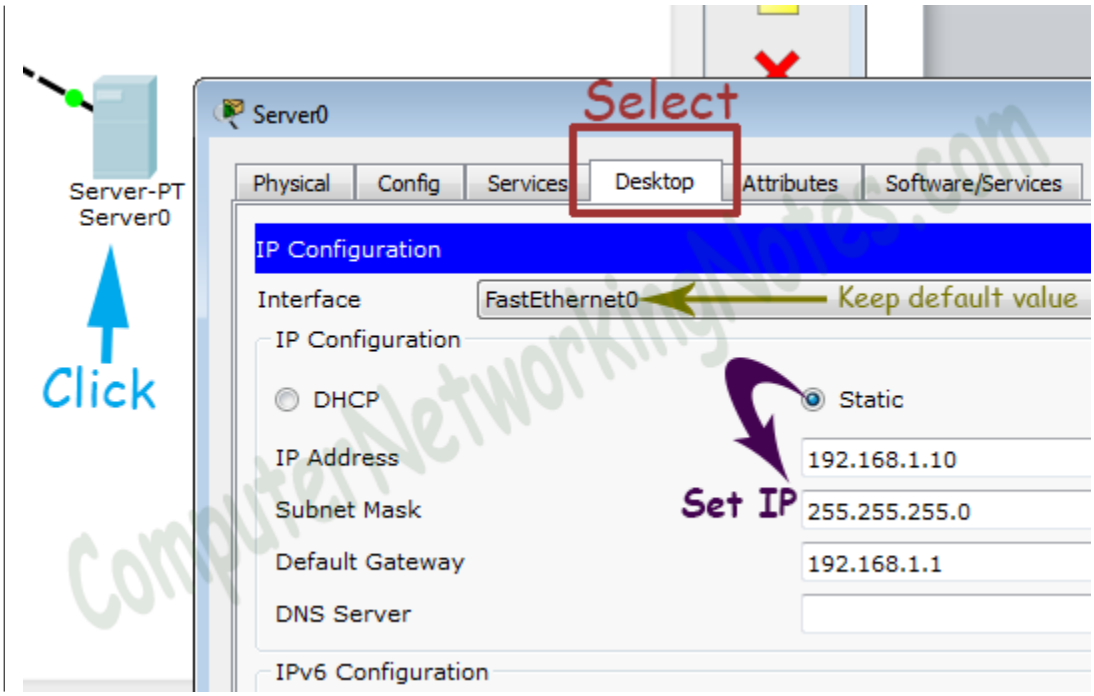
Serial 0/0/0 of R2	100.0.0.2/8	Serial 0/0/0

If you are following this tutorial on my practice topology, skip this IP configuration section as that topology is already configured with this initial IP configuration.

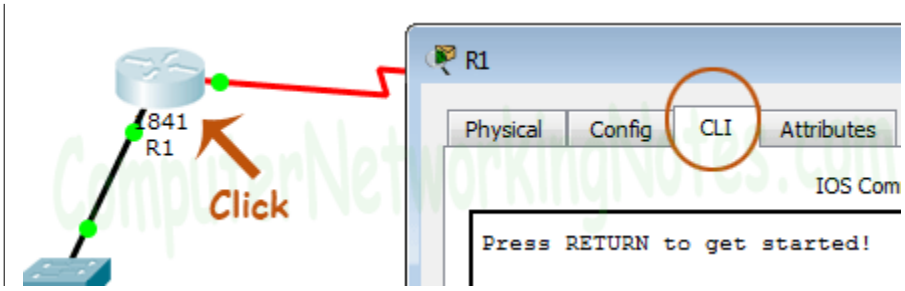
To assign IP address in Laptop click **Laptop** and click **Desktop** and click **IP configuration** and Select **Static** and set **IP address** as given in above table.



Following same way configure IP address in Server.



To configure IP address in Router1 click **Router1** and select **CLI** and press **Enter key**.



Run following commands to set IP address and hostname.

```
Router>enable
Router# configure terminal
Router(config)#
Router(config)#hostname R1
R1(config)#interface FastEthernet0/0
R1(config-if)#ip address 10.0.0.1 255.0.0.0
R1(config-if)#no shutdown
R1(config-if)#exit
R1(config)#interface Serial0/0/0
R1(config-if)#ip address 100.0.0.1 255.0.0.0
```

```
R1(config-if)#clock rate 64000
R1(config-if)#bandwidth 64
R1(config-if)#no shutdown
R1(config-if)#exit
R1(config)#
```

Same way access the command prompt of R2 and run following commands to set IP address and hostname.

```
Router>enable
Router#configure terminal
Router(config)#hostname R2
R2(config)#interface FastEthernet0/0
R2(config-if)#ip address 192.168.1.1 255.255.255.0
R2(config-if)#no shutdown
R2(config-if)#exit
R2(config)#interface Serial0/0/0
R2(config-if)#ip address 100.0.0.2 255.0.0.0
R2(config-if)#no shutdown
R2(config-if)#exit
R2(config)#
```

That's all initial IP configuration we need. Now this topology is ready for the practice of dynamic nat.

Configure Dynamic NAT

Dynamic NAT configuration requires four steps: -

1. Create an access list of IP addresses which need translation
2. Create a pool of all IP address which are available for translation
3. Map access list with pool
4. Define inside and outside interfaces

In first step we will create a standard access list which defines which inside local addresses are permitted to map with inside global address.

To create a standard numbered ACL following global configuration mode command is used:-

```
Router(config)# access-list ACL_Identifier_number permit/deny
matching-parameters
```

Let’s understand this command and its options in detail.

Router(config)#

This command prompt indicates that we are in global configuration mode.

access-list

Through this parameter we tell router that we are creating or accessing an access list.

ACL_Identifier_number

With this parameter we specify the type of access list. We have two types of access list; standard and extended. Both lists have their own unique identifier numbers. Standard ACL uses numbers range 1 to 99 and 1300 to 1999. We can pick any number from this range to tell the router that we are working with standard ACL. This number is used in groping the conditions under a single ACL. This number is also a unique identifier for this ACL in router.

permit/deny

An ACL condition has two actions; permit and deny. If we use permit keyword, ACL will allow all packets from the source address specified in next parameter. If we use deny keyword, ACL will drop all packets from the source address specified in next parameter.

matching-parameters

This parameter allows us to specify the contents of packet that we want to match. In a standard ACL condition it could be a single source address or a range of addresses. We have three options to specify the source address.

- Any
- host
- A.B.C.D

Any

Any keyword is used to match all sources. Every packet compared against this condition would be matched.

Host

Host keyword is used to match a specific host. To match a particular host, type the keyword host and then the IP address of host.

A.B.C.D

Through this option we can match a single address or a range of addresses. To match a single address, simply type its address. To match a range of addresses, we need to use wildcard mask.

Wildcard mask

Just like subnet mask, wildcard mask is also used to draw a boundary in IP address. Where subnet mask is used to separate network address from host address, wildcard mask is used to distinguish the matching portion from the rest. Wildcard mask is the invert of Subnet mask. Wildcard can be calculated in decimal or in binary from subnet mask.

We have three hosts in lab. Let's create a standard access list which allows two hosts and denies one host.

```
R1(config)#access-list 1 permit 10.0.0.10 0.0.0.0
R1(config)#access-list 1 permit 10.0.0.20 0.0.0.0
R1(config)#access-list 1 deny any
```

To learn standard ACL in detail you can use following tutorial.

Standard ACL Configuration Explained

In second step we define a pool of inside global addresses which are available for translation.

Following command is used to define the NAT pool.

```
Router(config)#ip nat pool [Pool Name] [Start IP address] [End
IP address] netmask [Subnet mask]
```

This command accepts four options pool name, start IP address, end IP address and Subnet mask.

Pool Name: - This is the name of pool. We can choose any descriptive name here.

Start IP Address: - First IP address from the IP range which is available for translation.

End IP Address: - Last IP address from the IP range which is available for translation. There is no minimum or maximum criteria for IP range for example we can have a range of single IP address or we can have a range of all IP address from a subnet.

Subnet Mask: - Subnet mask of IP range.

Let's create a pool named ccna with an IP range of two addresses.

```
R1(config)#ip nat pool ccna 50.0.0.1 50.0.0.2 netmask 255.0.0.0
```

This pool consist two class A IP address 50.0.0.1 and 50.0.0.2.

In third step we map access list with pool. Following command will map the access list with pool and configure the dynamic NAT.

```
Router(config)#ip nat inside source list [access list name or
number] pool [pool name]
```

This command accepts two options.

Access list name or number: - Name or number the access list which we created in first step.

Pool Name: - Name of pool which we created in second step.

In first step we created a standard access list with number **1** and in second step we created a pool named **ccna**. To configure a dynamic NAT with these options we will use following command.

```
R1(config)#ip nat inside source list 1 pool ccna
```

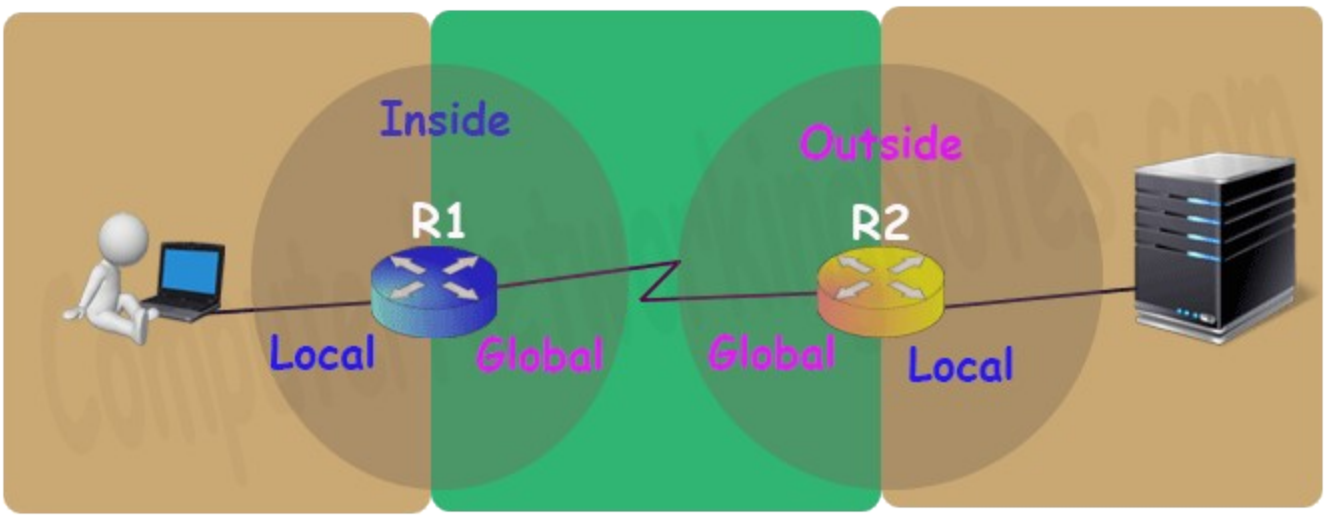
Finally we have to define which interface is connected with local network and which interface is connected with global network.

To define an inside local we use following command

```
Router(config-if)#ip nat inside
```

Following command defines inside global

```
Router(config-if)#ip nat outside
```



Let's implement all these commands together and configure the dynamic NAT.

R1 Dynamic NAT Configuration

```
R1#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#access-list 1 permit 10.0.0.10 0.0.0.0
R1(config)#access-list 1 permit 10.0.0.20 0.0.0.0
R1(config)#access-list 1 deny any
R1(config)#ip nat pool ccna 50.0.0.1 50.0.0.2 netmask 255.0.0.0
R1(config)#ip nat inside source list 1 pool ccna
R1(config)#interface FastEthernet 0/0
R1(config-if)#ip nat inside
R1(config-if)#exit
R1(config)#interface Serial0/0/0
R1(config-if)#ip nat outside
R1(config-if)#exit
```

```
R1(config)#
_
```

For testing purpose I configured dynamic translations for two addresses only.

On R2 we can keep standard configuration or can configure dynamic NAT as we just did in R1 or can configure static NAT as we learnt in pervious part of this article.

Let’s do a quick recap of what we learnt in previous part and configure static NAT on R2.

```
R2>enable
R2#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R2(config)#ip nat inside source static 192.168.1.10 200.0.0.10
R2(config)#interface Serial 0/0/0
R2(config-if)#ip nat outside
R2(config-if)#exit
R2(config)#interface FastEthernet 0/0
R2(config-if)#ip nat inside
R2(config-if)#exit
R2(config)#
```

To understand above commands in detail please see the second part of this tutorial.

Before we test this lab we need to configure the IP routing. IP routing is the process which allows router to route the packet between different networks. Following tutorial explain routing in detail with examples

Routing Protocols Explained in details

Configure static routing in R1

```
R1(config)#ip route 200.0.0.0 255.255.255.0 100.0.0.2
```

Configure static routing in R2

```
R2(config)#ip route 50.0.0.0 255.0.0.0 100.0.0.1
```

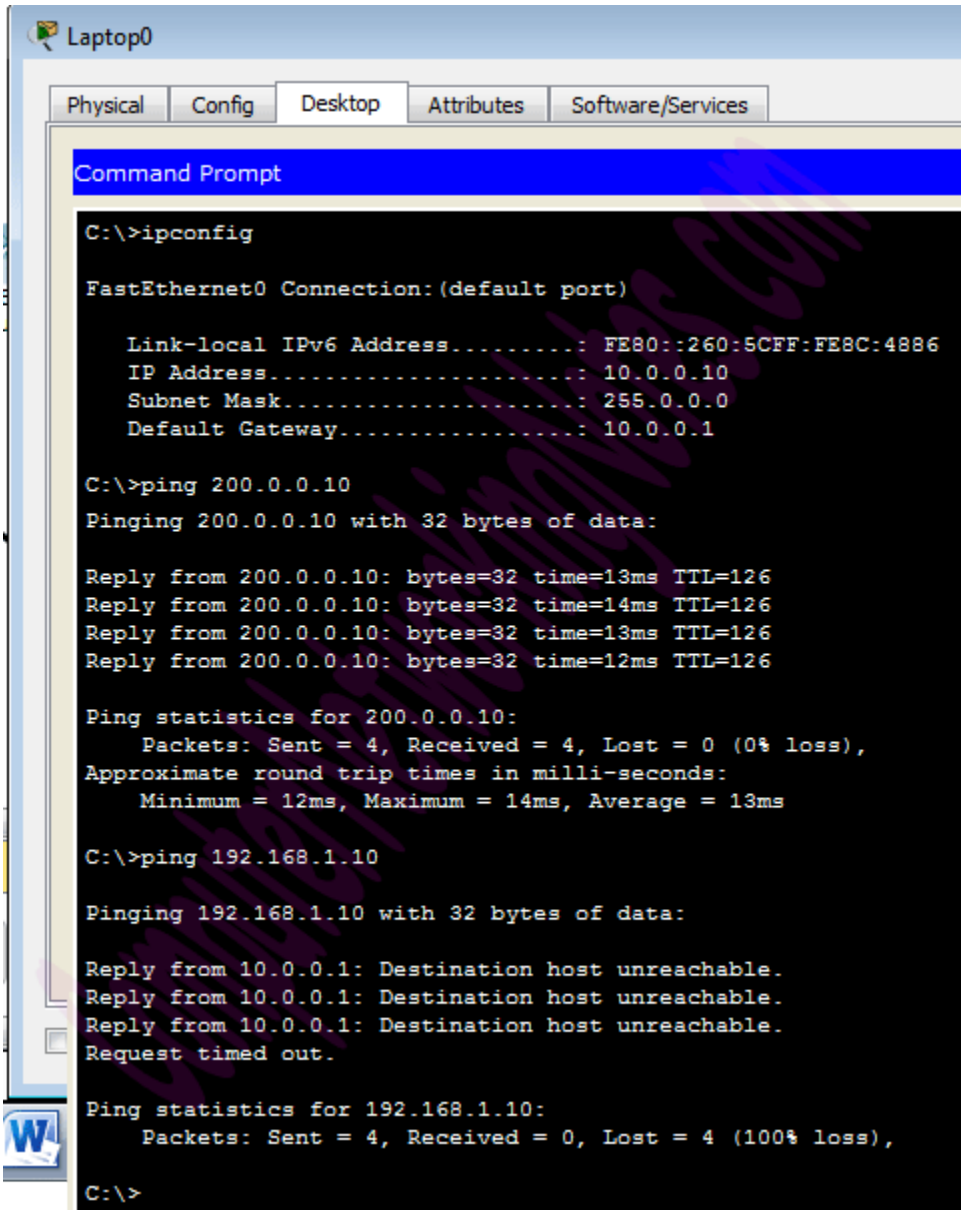
Testing Dynamic NAT Configuration

In this lab we configured dynamic NAT on R1for 10.0.0.10 and 10.0.0.20 and static NAT on R2 for 192.168.1.10.

Device	Inside Local IP Address	Inside Global IP Address
Laptop0	10.0.0.10	50.0.0.1
Laptop1	10.0.0.20	50.0.0.2
Server	192.168.1.10	200.0.0.10

To test this setup click **Laptop0** and **Desktop** and click **Command Prompt**.

- Run **ipconfig** command.
- Run **ping 200.0.0.10** command.
- Run **ping 192.168.1.10** command.

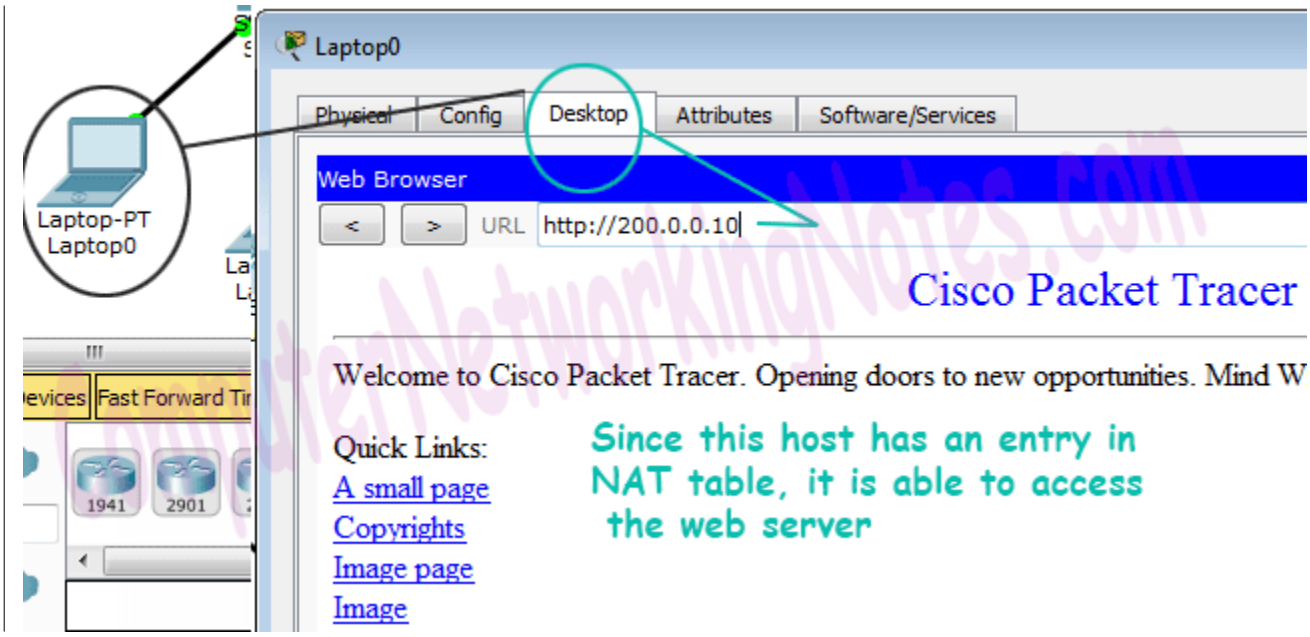


First command verifies that we are testing from correct NAT device.

Second command checks whether we are able to access the remote device or not. A ping reply confirms that we are able to connect with remote device on this IP address.

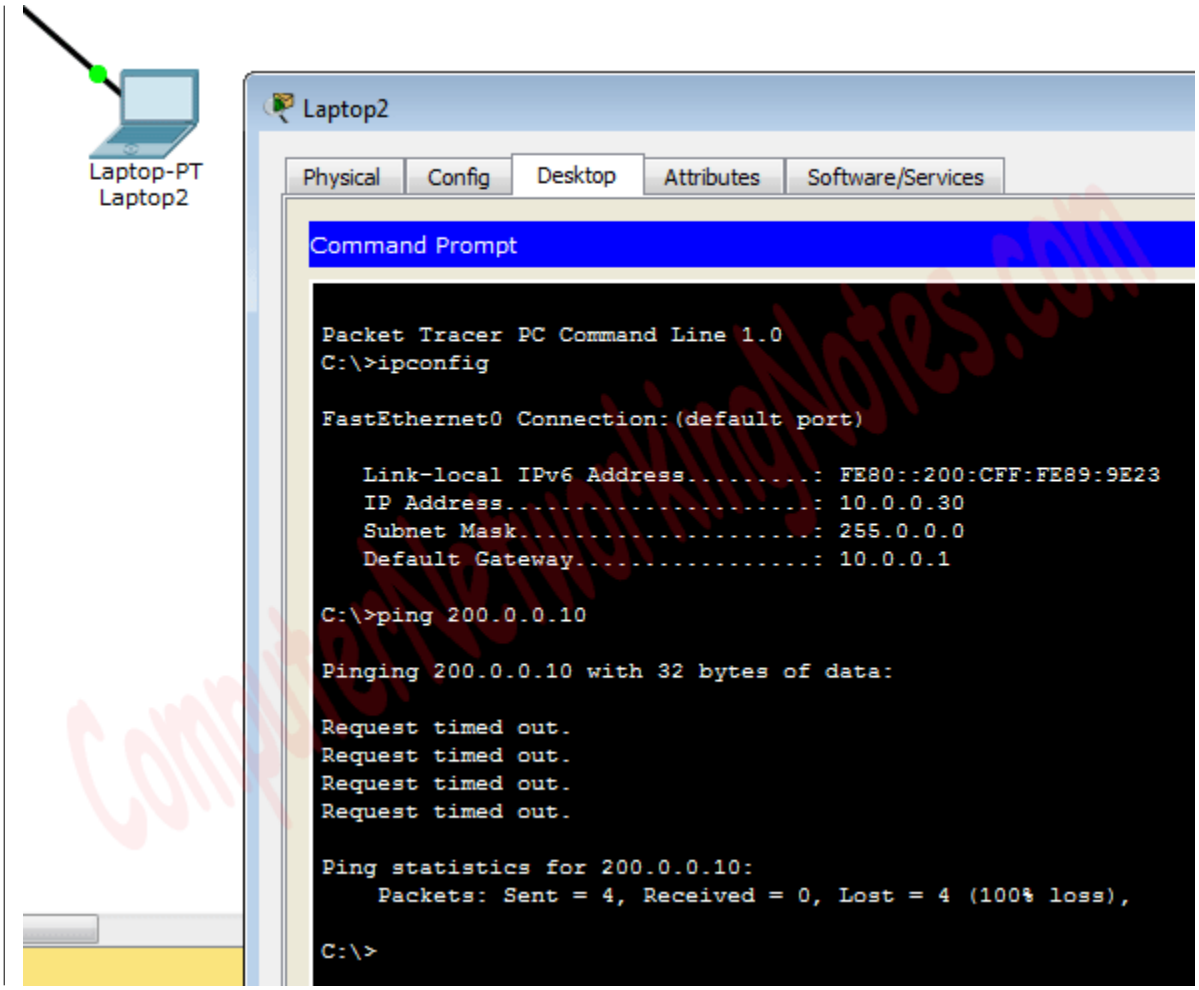
Third command checks whether we are able to access the remote device on its actual IP address or not. A ping error confirms that we are not able to connect with remote device on this IP address.

Let's do one more testing. Close the command prompt and click web server and access 200.0.0.10.

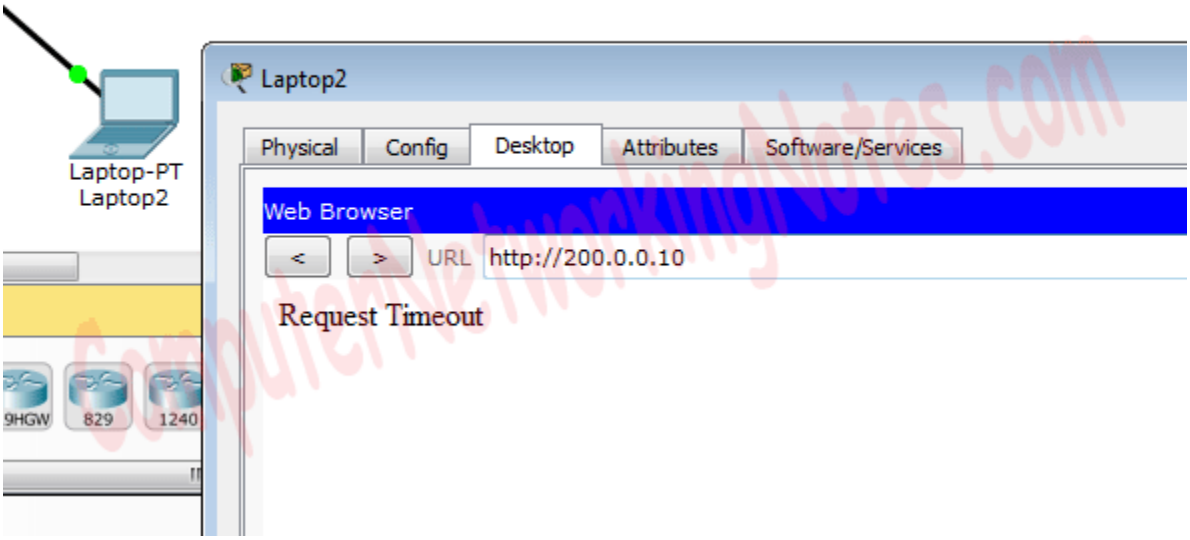


Above figure confirms that host 10.0.0.10 is able to access the 200.0.0.10. You can also do the same testing from Laptop1, result will be same.

Now run ping 200.0.0.10 command from Laptop2.



Close the command prompt and access web server from this host.



Why we are not able to connect with the remote device from this host?

Because we configured NAT only for two hosts (Laptop0 and Laptop1) which IP addresses are 10.0.0.10 and 10.0.0.20. So only the host 10.0.0.10 and 10.0.0.20 will be able to access the remote device.

If you followed this tutorial step by step, you should get the same output of testing. Although it's very rare but some time you may get different output. To figure out what went wrong you can use my practice topology with all above configuration. Download my practice topology.

Download NAT Practice LAB with Dynamic NAT configuration

We can also verify this translation on router with *show ip nat translation* command.

Following figure illustrates this translation on router R1.

```
R1>en
R1#show ip nat translations
Pro  Inside global     Inside local       Outside local      Outside global
tcp 50.0.0.1:1025      10.0.0.10:1025     200.0.0.10:80      200.0.0.10:80
tcp 50.0.0.2:1025      10.0.0.20:1025     200.0.0.10:80      200.0.0.10:80

R1#
```

We did three tests one from each host, but why only two tests are listed here? Remember in first step we created an access list. Access list filters the unwanted traffic before it reaches to the NAT. We can see how many packets are blocked by ACL with following command

```
R1#show ip access-lists 1
```

```
R1#show ip access-lists 1
Standard IP access list 1
    permit host 10.0.0.10 (8 match(es))
    permit host 10.0.0.20 (2 match(es))
    deny any (3 match(es))

R1#
```

Basically it is access list which filters the traffic. NAT does not filter any traffic it only translate the address.

Following figure illustrate NAT translation on router R2

```
R2>enable
R2#show ip nat translations
Pro  Inside global     Inside local       Outside local      Outside global
---  200.0.0.10        192.168.1.10       ---                ---
tcp 200.0.0.10:80      192.168.1.10:80    50.0.0.1:1025      50.0.0.1:1025
tcp 200.0.0.10:80      192.168.1.10:80    50.0.0.2:1025      50.0.0.2:1025

R2#
```


Configure PAT in Cisco Router with Examples

This tutorial explains how to configure port address translation (PAT) in router step by step with examples. Learn how to connect multiple devices with remote network from single IP address through PAT or NAT Overload, verify and troubleshoot PAT configuration view PAT address translation from show commands.

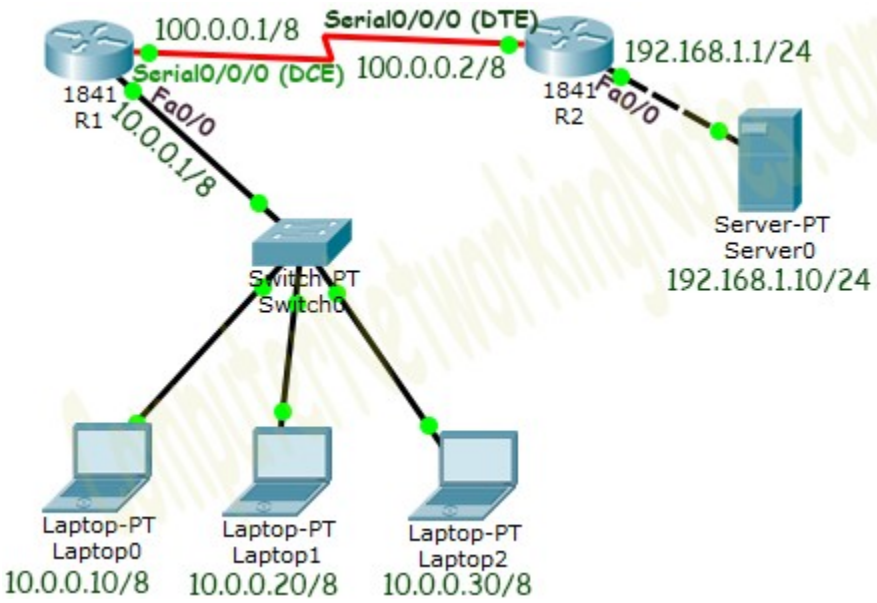
PAT (NAT Overload) Practice LAB Setup

In this tutorial I will use Packet Tracer network simulator software for demonstration.

If require, you can download the latest as well as earlier version of Packet Tracer from here. Download Packet Tracer

Create a practice lab as shown in following figure or download this pre-created practice lab and load in packet tracer

Download NAT Practice LAB with initial IP configuration



This tutorial is the last part of our article "**Learn NAT (Network Address Translation) Step by Step in Easy Language with Examples**". You can read other parts of this article here.

Basic Concepts of NAT Exaplained in Easy Language
This tutorial is the first part of this article. This tutorial explains basic concepts of static nat, dynamic nat, pat, inside local, outside local, inside global and outside global in detail with examples.

How to Configure Static NAT in Cisco Router
This tutorial is the second part of this article. This tutorial explains how to configure Static NAT (Network Address Translation) in Cisco Router step by step with packet tracer examples.

How to Configure Dynamic NAT in Cisco Router
This tutorial is the third part of this article. This tutorial explains how to configure Dynamic NAT (Network Address Translation) in Cisco Router step by step with packet tracer examples.

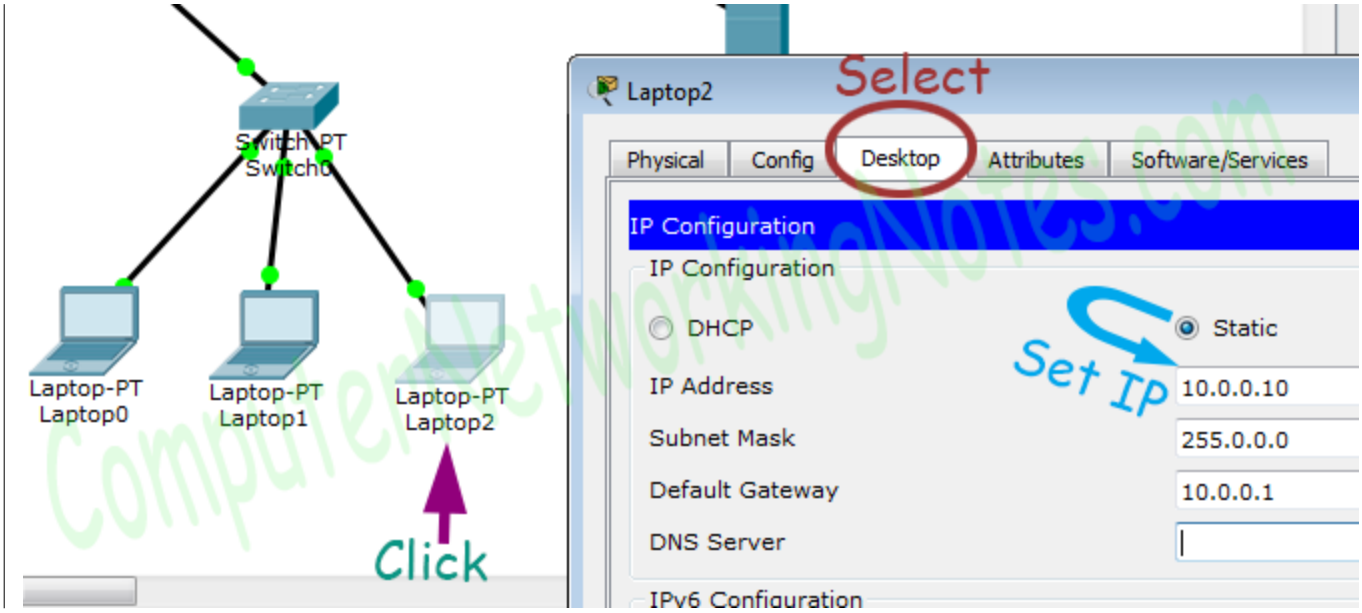
Initial IP Configuration

Device / Interface	IP Address	Connected
Laotop0	10.0.0.10/8	Fa0/0 of R0
Laptop1	10.0.0.20/8	Fa0/0 of R0
Laptop2	10.0.0.30/8	Fa0/0 of R0

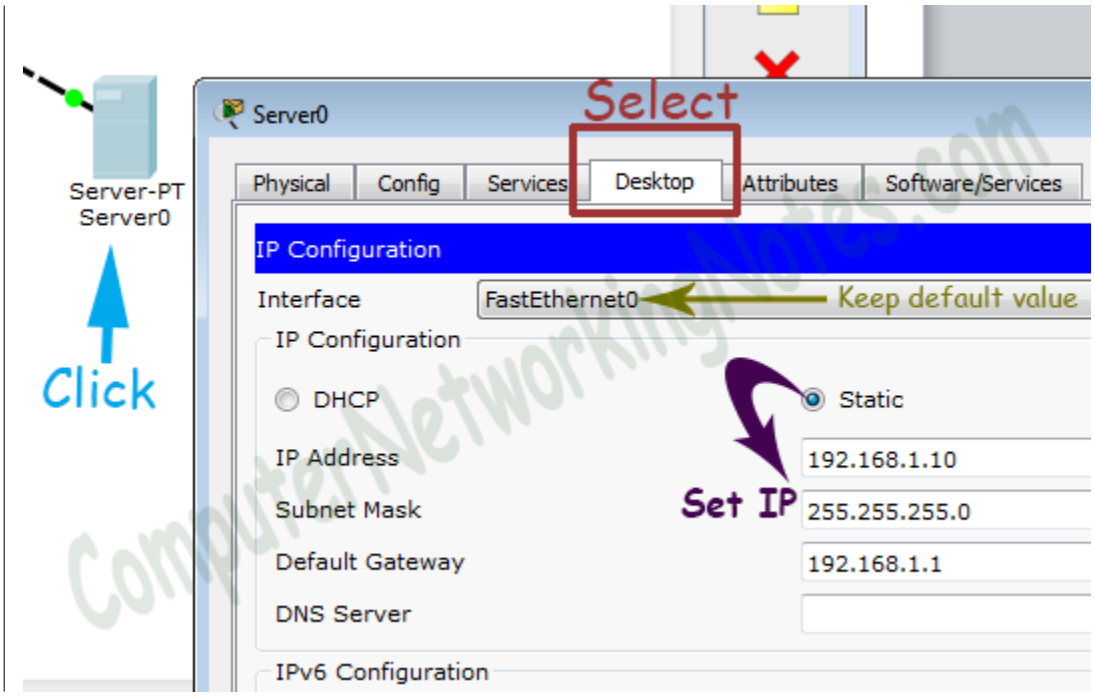
Server0	192.168.1.10/24	Fa0/0 of R1
Serial 0/0/0 of R1	100.0.0.1/8	Serial 0/0/0
Serial 0/0/0 of R2	100.0.0.2/8	Serial 0/0/0

If you are following this tutorial on my practice topology, skip this IP configuration section as that topology is already configured with this initial IP configuration.

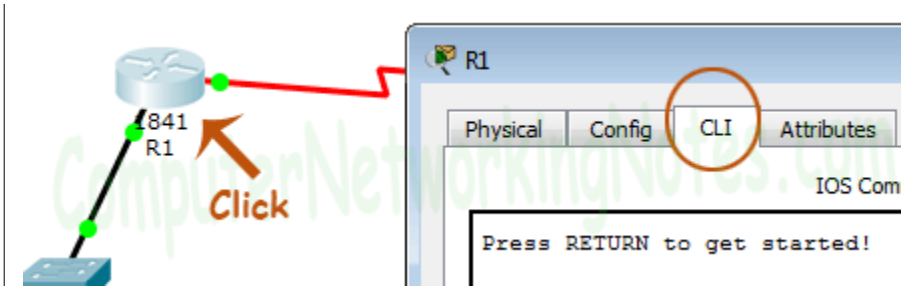
To assign IP address in Laptop click **Laptop** and click **Desktop** and click **IP configuration** and Select **Static** and set **IP address** as given in above table.



Following same way configure IP address in Server.



To configure IP address in Router1 click **Router1** and select **CLI** and press **Enter key**.



Run following commands to set IP address and hostname.

```
Router>enable
Router# configure terminal
Router(config)#
Router(config)#hostname R1
R1(config)#interface FastEthernet0/0
R1(config-if)#ip address 10.0.0.1 255.0.0.0
R1(config-if)#no shutdown
R1(config-if)#exit
R1(config)#interface Serial0/0/0
R1(config-if)#ip address 100.0.0.1 255.0.0.0
```

```
R1(config-if)#clock rate 64000
R1(config-if)#bandwidth 64
R1(config-if)#no shutdown
R1(config-if)#exit
R1(config)#
```

Same way access the command prompt of R2 and run following commands to set IP address and hostname.

```
Router>enable
Router#configure terminal
Router(config)#hostname R2
R2(config)#interface FastEthernet0/0
R2(config-if)#ip address 192.168.1.1 255.255.255.0
R2(config-if)#no shutdown
R2(config-if)#exit
R2(config)#interface Serial0/0/0
R2(config-if)#ip address 100.0.0.2 255.0.0.0
R2(config-if)#no shutdown
R2(config-if)#exit
R2(config)#
```

That's all initial IP configuration we need. Now this topology is ready for the practice of pat.

Configure PAT (NAT Overload)

PAT configuration requires four steps: -

1. Create an access list of IP addresses which need translation
2. Create a pool of all IP address which are available for translation
3. Map access list with pool
4. Define inside and outside interfaces

In first step we will create a standard access list which defines which inside local addresses are permitted to map with inside global address.

To create a standard numbered ACL following global configuration mode command is used:-

```
Router(config)# access-list ACL_Identifier_number permit/deny
matching-parameters
```

Let’s understand this command and its options in detail.

Router(config)#

This command prompt indicates that we are in global configuration mode.

access-list

Through this parameter we tell router that we are creating or accessing an access list.

ACL_Identifier_number

With this parameter we specify the type of access list. We have two types of access list; standard and extended. Both lists have their own unique identifier numbers. Standard ACL uses numbers range 1 to 99 and 1300 to 1999. We can pick any number from this range to tell the router that we are working with standard ACL. This number is used in groping the conditions under a single ACL. This number is also a unique identifier for this ACL in router.

permit/deny

An ACL condition has two actions; permit and deny. If we use permit keyword, ACL will allow all packets from the source address specified in next parameter. If we use deny keyword, ACL will drop all packets from the source address specified in next parameter.

matching-parameters

This parameter allows us to specify the contents of packet that we want to match. In a standard ACL condition it could be a single source address or a range of addresses. We have three options to specify the source address.

- Any
- host
- A.B.C.D

Any

Any keyword is used to match all sources. Every packet compared against this condition would be matched.

Host

Host keyword is used to match a specific host. To match a particular host, type the keyword host and then the IP address of host.

A.B.C.D

Through this option we can match a single address or a range of addresses. To match a single address, simply type its address. To match a range of addresses, we need to use wildcard mask.

Wildcard mask

Just like subnet mask, wildcard mask is also used to draw a boundary in IP address. Where subnet mask is used to separate network address from host address, wildcard mask is used to distinguish the matching portion from the rest. Wildcard mask is the invert of Subnet mask. Wildcard can be calculated in decimal or in binary from subnet mask.

We have three hosts in lab. Let's create a standard access list which allows two hosts and denies one host.

```
R1(config)#access-list 1 permit 10.0.0.10 0.0.0.0
R1(config)#access-list 1 permit 10.0.0.20 0.0.0.0
R1(config)#access-list 1 deny any
```

To learn standard ACL in detail you can use following tutorial.

Standard ACL Explained with Examples

In second step we define a pool of inside global addresses which are available for translation.

Following command is used to define the NAT pool.

```
Router(config)#ip nat pool [Pool Name] [Start IP address] [End
IP address] netmask [Subnet mask]
```

This command accepts four options pool name, start IP address, end IP address and Subnet mask.

Pool Name: - This is the name of pool. We can choose any descriptive name here.

Start IP Address: - First IP address from the IP range which is available for translation.

End IP Address: - Last IP address from the IP range which is available for translation. There is no minimum or maximum criteria for IP range for example we can have a range of single IP address or we can have a range of all IP address from a subnet.

Subnet Mask: - Subnet mask of IP range.

Let's create a pool named ccna with a single IP address.

```
R1(config)#ip nat pool ccna 50.0.0.1 50.0.0.1 netmask 255.0.0.0
```

In third step we map access list with pool. Following command will map the access list with pool and configure the PAT.

```
Router(config)#ip nat inside source list [access list name or number] pool [pool name]overload
```

This command accepts two options.

Access list name or number: - Name or number the access list which we created in first step.

Pool Name: - Name of pool which we created in second step.

In first step we created a standard access list with number 1 and in second step we created a pool named ccna. To configure a PAT with these options we will use following command.

```
R1(config)#ip nat inside source list 1 pool ccna overload
```

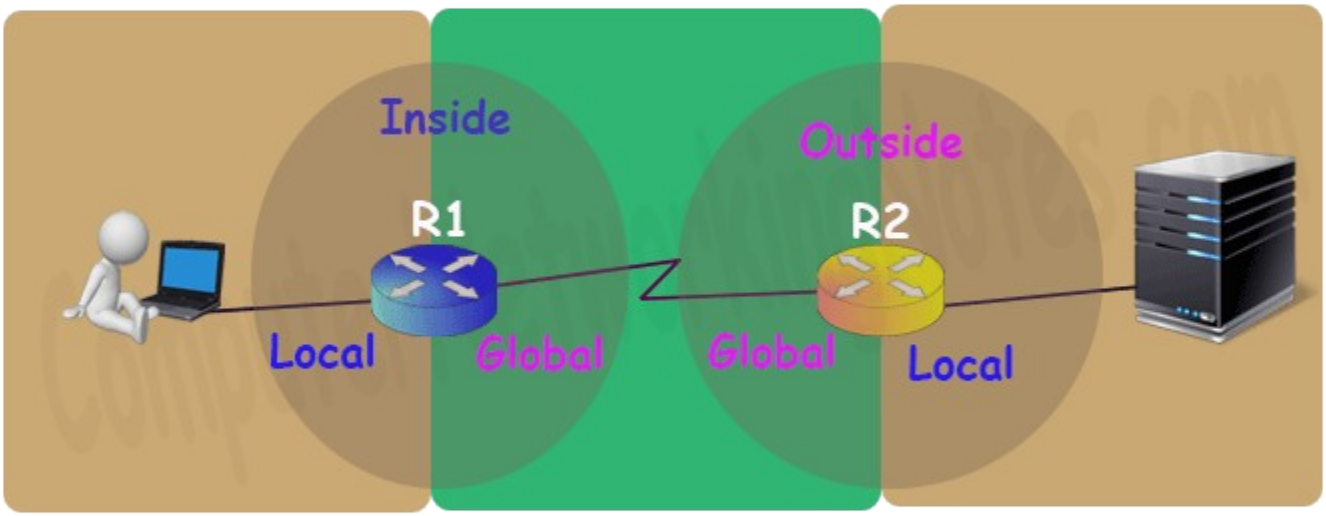
Finally we have to define which interface is connected with local network and which interface is connected with global network.

To define an inside local we use following command

```
Router(config-if)#ip nat inside
```

Following command defines inside global

```
Router(config-if)#ip nat outside
```



Let's implement all these commands together and configure the PAT.

R1 PAT (NAT Overload) Configuration

```
R1>enable
R1#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#access-list 1 permit 10.0.0.10 0.0.0.0
R1(config)#access-list 1 permit 10.0.0.20 0.0.0.0
R1(config)#access-list 1 deny any
R1(config)#ip nat pool ccna 50.0.0.1 50.0.0.1 netmask 255.0.0.0
R1(config)#ip nat inside source list 1 pool ccna overload
R1(config)#interface FastEthernet 0/0
R1(config-if)#ip nat inside
R1(config-if)#exit
R1(config)#interface Serial 0/0/0
R1(config-if)#ip nat outside
R1(config-if)#exit
R1(config)#
```

For testing purpose I configured pat translations for two addresses only.

On R2 we can keep standard configuration or can configure dynamic NAT or can configure static NAT as we learnt in pervious parts of this article.

Let's do a quick recap of what we learnt in previous part and configure static NAT on R2.

```
R2>enable
R2#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R2(config)#ip nat inside source static 192.168.1.10 200.0.0.10
R2(config)#interface Serial 0/0/0
R2(config-if)#ip nat outside
R2(config-if)#exit
R2(config)#interface FastEthernet 0/0
R2(config-if)#ip nat inside
R2(config-if)#exit
R2(config)#
```

To understand above commands in detail please see the second part of this tutorial.

Before we test this lab we need to configure the IP routing. IP routing is the process which allows router to route the packet between different networks. Following tutorial explain routing in detail with examples

Routing Protocol Explained

Configure static routing in R1

```
R1(config)#ip route 200.0.0.0 255.255.255.0 100.0.0.2
```

Configure static routing in R2

```
R2(config)#ip route 50.0.0.0 255.0.0.0 100.0.0.1
```

Testing PAT Configuration

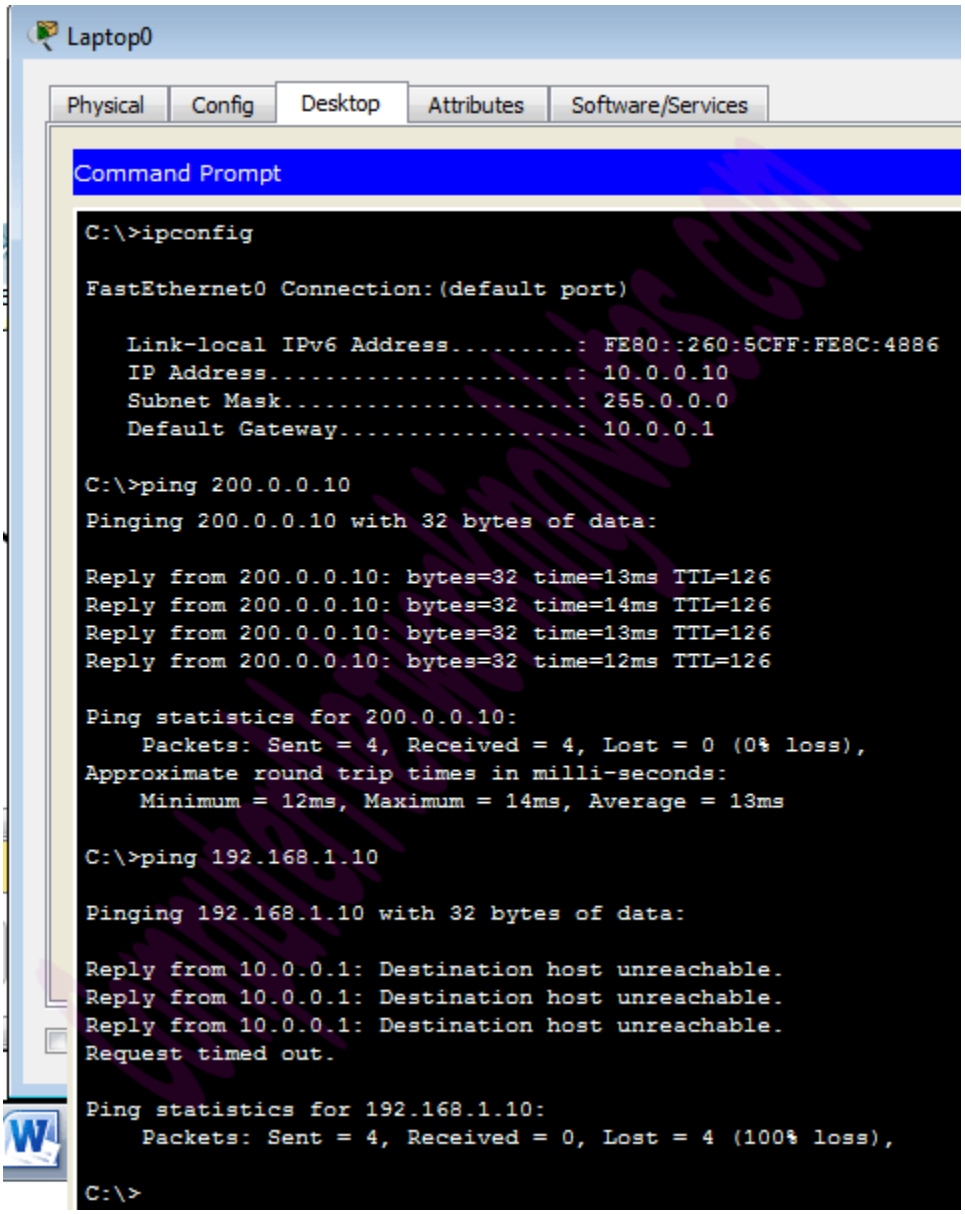
In this lab we configured PAT on R1for 10.0.0.10 and 10.0.0.20 and static NAT on R2 for 192.168.1.10.

Device	Inside Local IP Address	Inside Global IP Address
Laptop0	10.0.0.10	50.0.0.1

Laptop1	10.0.0.20	50.0.0.2
Server	192.168.1.10	200.0.0.10

To test this setup click **Laptop0** and **Desktop** and click **Command Prompt**.

- Run **ipconfig** command.
- Run **ping 200.0.0.10** command.
- Run **ping 192.168.1.10** command.

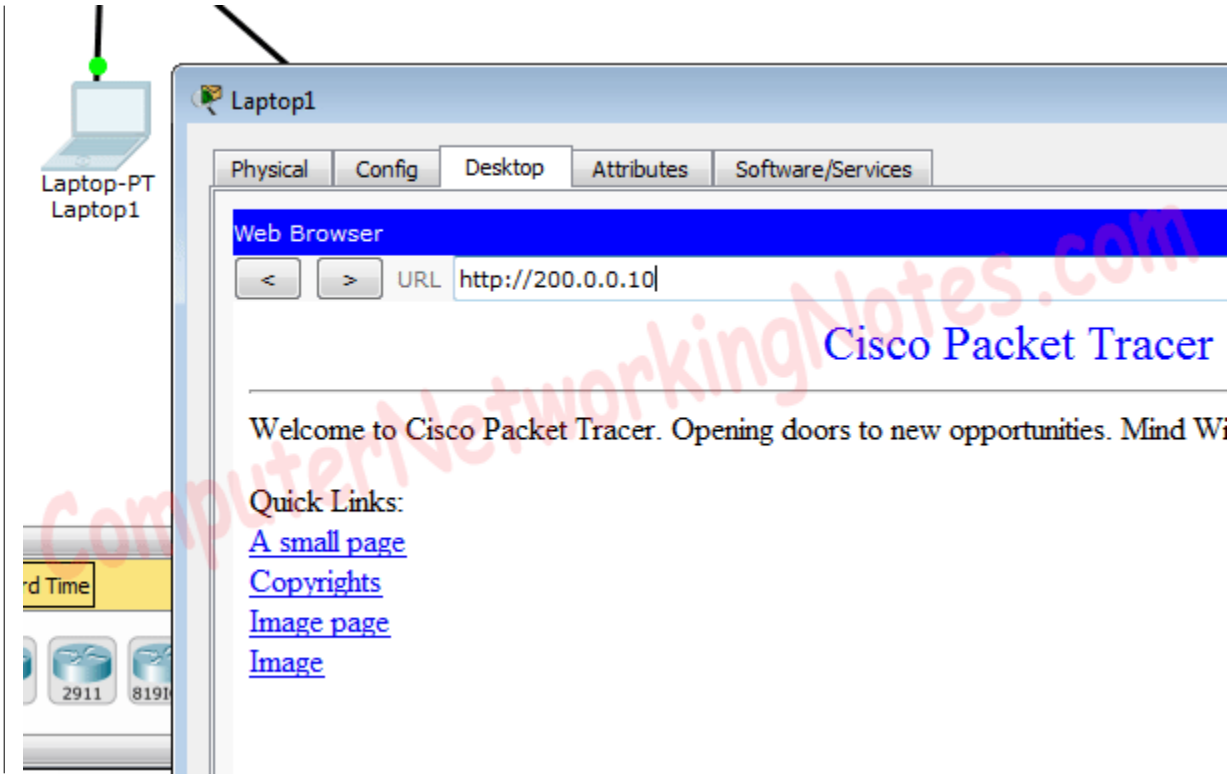


First command verifies that we are testing from correct NAT device.

Second command checks whether we are able to access the remote device or not. A ping reply confirms that we are able to connect with remote device on this IP address.

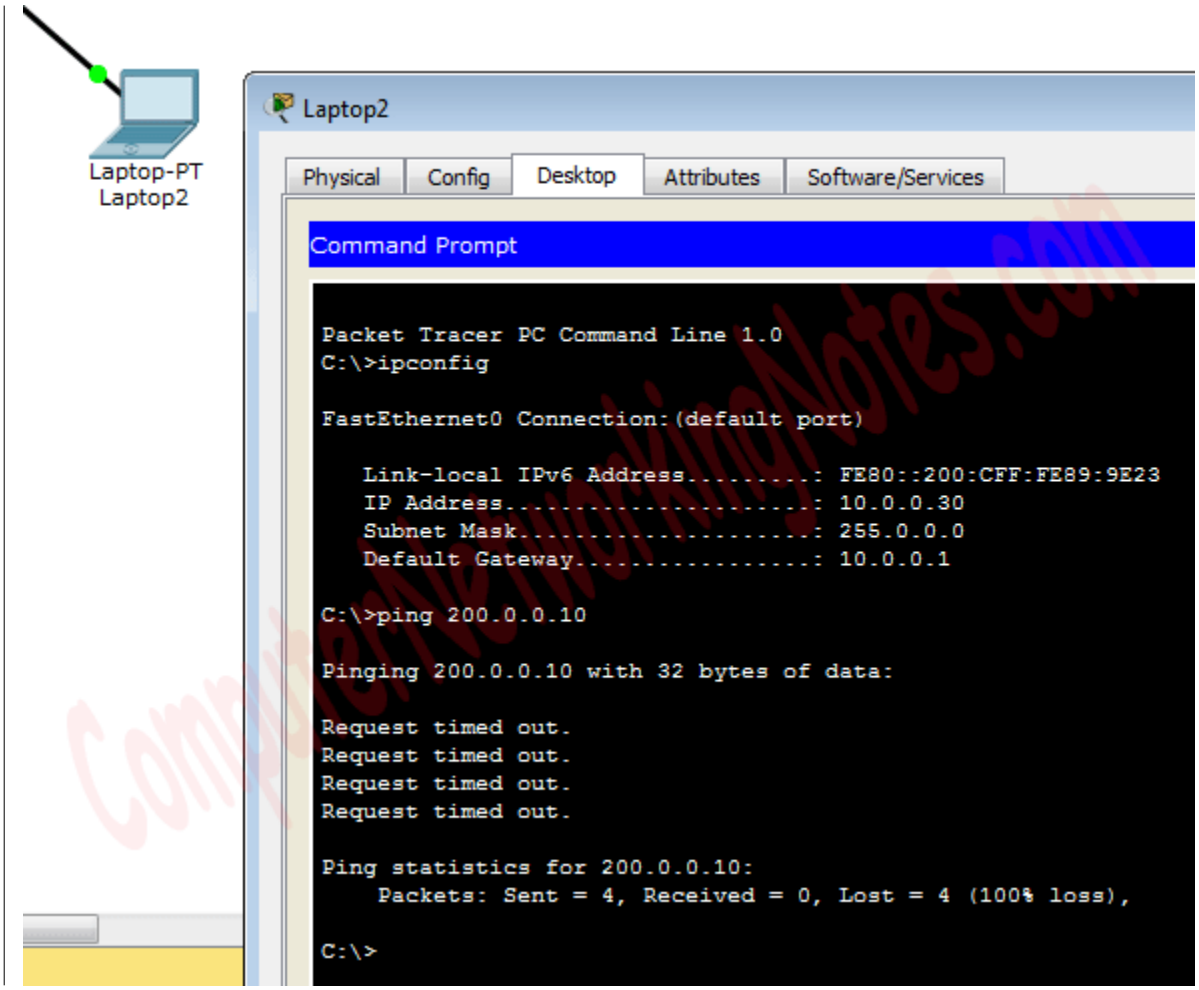
Third command checks whether we are able to access the remote device on its actual IP address or not. A ping error confirms that we are not able to connect with remote device on this IP address.

Let's do one more testing. Close the command prompt and click web server and access 200.0.0.10.

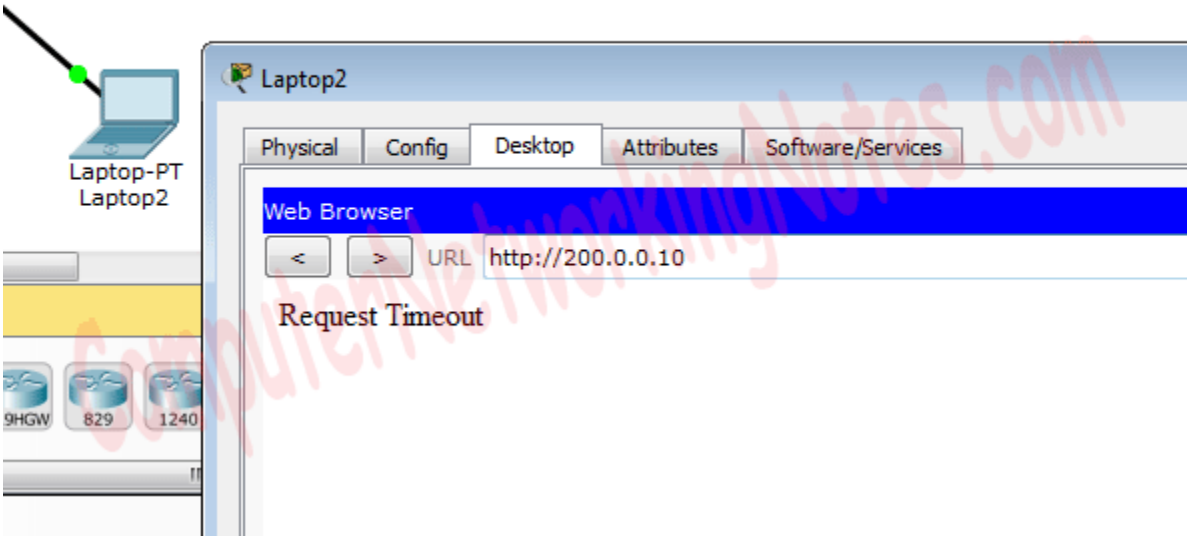


Above figure confirms that host 10.0.0.10 is able to access the 200.0.0.10. You can also do the same testing from Laptop1, result will be same.

Now run **ping 200.0.0.10** command from Laptop2.



Close the command prompt and access web server from this host.



Why we are not able to connect with the remote device from this host?

Because we configured PAT only for two hosts (Laptop0 and Laptop1) which IP addresses are 10.0.0.10 and 10.0.0.20. So only the host 10.0.0.10 and 10.0.0.20 will be able to access the remote device.

If you followed this tutorial step by step, you should get the same output of testing. Although it's very rare but some time you may get different output. To figure out what went wrong you can use my practice topology with all above configuration. Download my practice topology

Download NAT Practice LAB with PAT configuration

We can also verify this translation on router with ***show ip nat translation*** command.

Following figure illustrate this translation on router R1.

```
R1#show ip nat translation
Pro  Inside global     Inside local       Outside local      Outside global
icmp 50.0.0.1:1        10.0.0.20:1        200.0.0.10:1       200.0.0.10:1
icmp 50.0.0.1:2        10.0.0.20:2        200.0.0.10:2       200.0.0.10:2
icmp 50.0.0.1:3        10.0.0.20:3        200.0.0.10:3       200.0.0.10:3
icmp 50.0.0.1:4        10.0.0.20:4        200.0.0.10:4       200.0.0.10:4
tcp 50.0.0.1:1024      10.0.0.10:1025     200.0.0.10:80      200.0.0.10:80
tcp 50.0.0.1:1025      10.0.0.20:1025     200.0.0.10:80      200.0.0.10:80

R1#
```

As we can see in above output same inside global IP address is used to translate all the inside local IP addresses. For each inside local IP address a unique port number is used.

Following figure illustrate NAT translation on router R2

```
R2#show ip nat translation
Pro  Inside global     Inside local       Outside local      Outside global
icmp 200.0.0.10:1      192.168.1.10:1     50.0.0.1:1         50.0.0.1:1
icmp 200.0.0.10:2      192.168.1.10:2     50.0.0.1:2         50.0.0.1:2
icmp 200.0.0.10:3      192.168.1.10:3     50.0.0.1:3         50.0.0.1:3
icmp 200.0.0.10:4      192.168.1.10:4     50.0.0.1:4         50.0.0.1:4
---  200.0.0.10        192.168.1.10       ---                ---
tcp 200.0.0.10:80      192.168.1.10:80    50.0.0.1:1024      50.0.0.1:1024
tcp 200.0.0.10:80      192.168.1.10:80    50.0.0.1:1025      50.0.0.1:1025

R2#
```

In above output the Outside global field also confirms that all packets are coming from single IP address.

CONCLUSION: Hence in this we have studied that Configuration of VLAN,OSPF and NAT

D Y PATIL INSTITUTE OF ENGINEERING & TECHNOLOGY AMBI, PUNE	**D Y PATIL INSTITUTE OF ENGINEERING & TECHNOLOGY, AMBI**	LABORATORY MANUAL
	EXPERIMENT TITLE: Socket Programming in C on Linux using TCP and UDP protocol	
DEPARTMENT OF INFORMATION TECHNOLOGY		
EXPERIMENT NO. : DYPIET/IT/TE/SL-IV	**SEMESTER** : VI(TE)	**PAGE:1-**

AIM: Socket Programming in C on Linux using TCP and UDP protocol

OBJECTIVES: To study,

TCP , UDP Protocol

Client Server programming

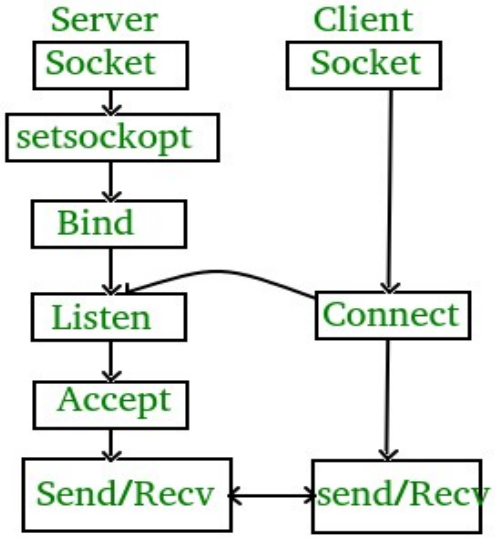
Socket programming primitives with syntax

THEORY:

What is socket programming?

Socket programming is a way of connecting two nodes on a network to communicate with each other. One socket(node) listens on a particular port at an IP, while other socket reaches out to the other to form a connection. Server forms the listener socket while client reaches out to the server.

State diagram for server and client model



Stages for server

- **Socket creation:**

```
int sockfd = socket(domain, type, protocol)
```

sockfd: socket descriptor, an integer (like a file-handle)
domain: integer, communication domain e.g., AF_INET (IPv4 protocol) , AF_INET6 (IPv6 protocol)
type: communication type
SOCK_STREAM: TCP(reliable, connection oriented)
SOCK_DGRAM: UDP(unreliable, connectionless)
protocol: Protocol value for Internet Protocol(IP), which is 0. This is the same number which appears on protocol field in the IP header of a packet.(man protocols for more details)

- **Setsockopt:**

```
int setsockopt(int sockfd, int level, int optname,
                    const void *optval, socklen_t optlen);
```

This helps in manipulating options for the socket referred by the file descriptor sockfd. This is completely optional, but it helps in reuse of address and port. Prevents error such as: “address already in use”.

- **Bind:**

```
int bind(int sockfd, const struct sockaddr *addr,
```

```
                                socklen_t addrlen);
```

After creation of the socket, bind function binds the socket to the address and port number specified in addr(custom data structure). In the example code, we bind the server to the localhost, hence we use INADDR_ANY to specify the IP address.

- **Listen:**

```
int listen(int sockfd, int backlog);
```

It puts the server socket in a passive mode, where it waits for the client to approach the server to make a connection. The backlog, defines the maximum length to which the queue of pending connections for sockfd may grow. If a connection request arrives when the queue is full, the client may receive an error with an indication of ECONNREFUSED.

- **Accept:**

```
int new_socket= accept(int sockfd, struct sockaddr *addr,
socklen_t *addrlen);
```

It extracts the first connection request on the queue of pending connections for the listening socket, sockfd, creates a new connected socket, and returns a new file descriptor referring to that socket. At this point, connection is established between client and server, and they are ready to transfer data.

Stages for Client

- **Socket connection:** Exactly same as that of server's socket creation
- **Connect:**
-

```
int connect(int sockfd, const struct sockaddr *addr,
                                socklen_t addrlen);
```

The connect() system call connects the socket referred to by the file descriptor sockfd to the address specified by addr. Server's address and port is specified in addr.

Implementation

Here we are exchanging one hello message between server and client to demonstrate the client/server model.

- server.c
- client.c

filter_none

edit

play_arrow
brightness_4

```
// Client side C/C++ program to demonstrate Socket programming
#include <stdio.h>
#include <sys/socket.h>
#include <stdlib.h>
#include <netinet/in.h>
#include <string.h>
#define PORT 8080

int main(int argc, char const *argv[])
{
    struct sockaddr_in address;
    int sock = 0, valread;
    struct sockaddr_in serv_addr;
    char *hello = "Hello from client";
    char buffer[1024] = {0};
    if ((sock = socket(AF_INET, SOCK_STREAM, 0)) < 0)
    {
        printf("\n Socket creation error \n");
        return -1;
    }

    memset(&serv_addr, '0', sizeof(serv_addr));

    serv_addr.sin_family = AF_INET;
    serv_addr.sin_port = htons(PORT);

    // Convert IPv4 and IPv6 addresses from text to binary form
    if(inet_pton(AF_INET, "127.0.0.1", &serv_addr.sin_addr)<=0)
    {
        printf("\nInvalid address/ Address not supported \n");
        return -1;
    }

    if (connect(sock, (struct sockaddr *)&serv_addr, sizeof(serv_addr)) < 0)
    {
        printf("\nConnection Failed \n");
        return -1;
    }
    send(sock , hello , strlen(hello) , 0 );
    printf("Hello message sent\n");
    valread = read( sock , buffer, 1024);
    printf("%s\n",buffer );
    return 0;
}
```

Compiling:

gcc client.c -o client
gcc server.c -o server

Output:

```
Client:Hello message sent
Hello from server
Server:Hello from client
Hello message sent
```

```
// Server side C/C++ program to demonstrate Socket programming
#include <unistd.h>
#include <stdio.h>
#include <sys/socket.h>
#include <stdlib.h>
#include <netinet/in.h>
#include <string.h>
#define PORT 8080
int main(int argc, char const *argv[])
{
    int server_fd, new_socket, valread;
    struct sockaddr_in address;
    int opt = 1;
    int addrlen = sizeof(address);
    char buffer[1024] = {0};
    char *hello = "Hello from server";

    // Creating socket file descriptor
    if ((server_fd = socket(AF_INET, SOCK_STREAM, 0)) == 0)
    {
```

```
        perror("socket failed");
        exit(EXIT_FAILURE);
    }

    // Forcefully attaching socket to the port 8080
    if (setsockopt(server_fd, SOL_SOCKET, SO_REUSEADDR | SO_REUSEPORT,
                                                  &opt, sizeof(opt)))
    {
        perror("setsockopt");
        exit(EXIT_FAILURE);
    }
    address.sin_family = AF_INET;
    address.sin_addr.s_addr = INADDR_ANY;
    address.sin_port = htons( PORT );

    // Forcefully attaching socket to the port 8080
    if (bind(server_fd, (struct sockaddr *)&address,
                                sizeof(address))<0)
    {
        perror("bind failed");
        exit(EXIT_FAILURE);
    }
    if (listen(server_fd, 3) < 0)
    {
        perror("listen");
        exit(EXIT_FAILURE);
```

```
    }
    if ((new_socket = accept(server_fd, (struct sockaddr *)&address,
                        (socklen_t*)&addrlen))<0)
    {
        perror("accept");
        exit(EXIT_FAILURE);
    }
    valread = read( new_socket , buffer, 1024);
    printf("%s\n",buffer );
    send(new_socket , hello , strlen(hello) , 0 );
    printf("Hello message sent\n");
    return 0;
}
```

Compiling:
gcc client.c -o client
gcc server.c -o server

Output:

```
Client:Hello message sent
Hello from server
Server:Hello from client
Hello message sent
```

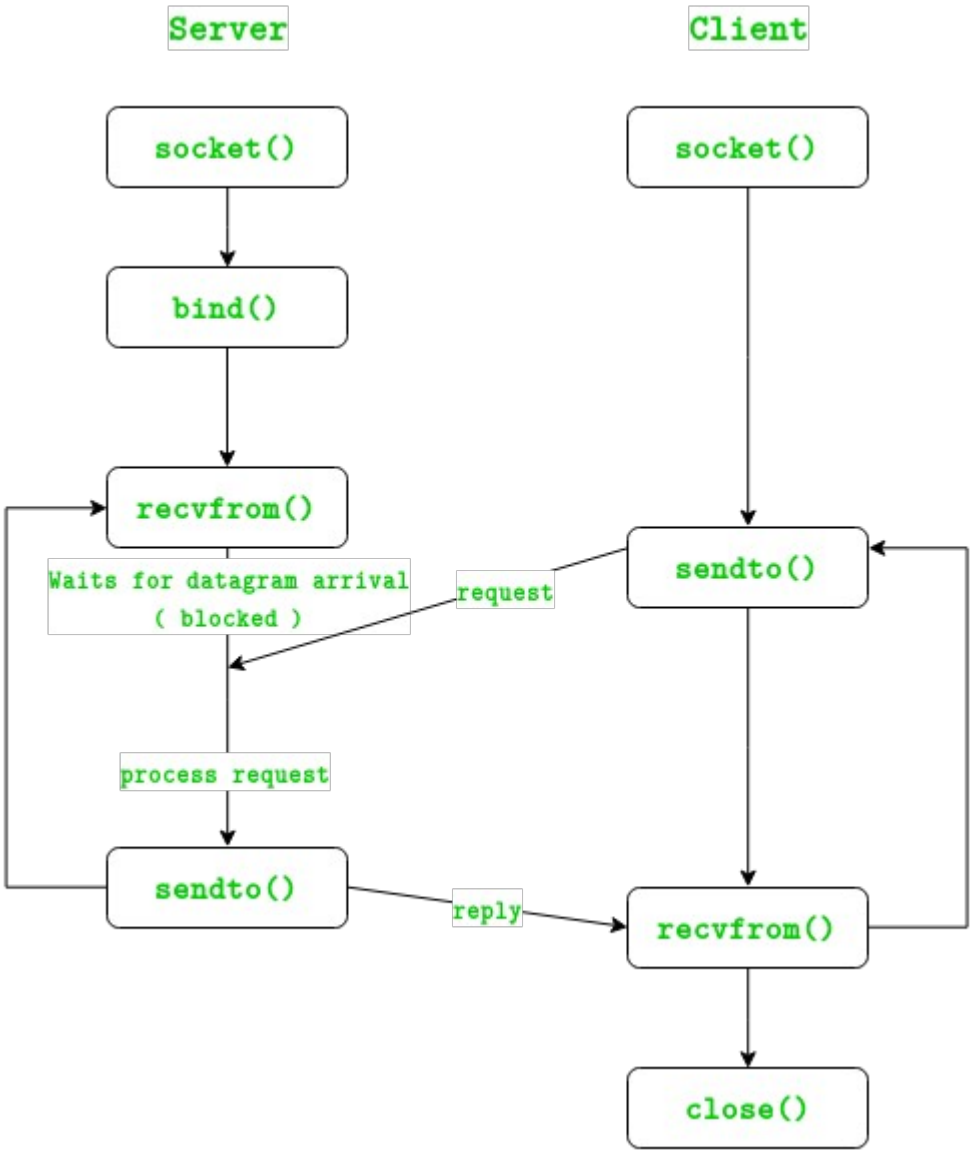
UDP Server-Client implementation in C

There are two major transport layer protocols to communicate between hosts : **TCP** and **UDP**. Creating TCP Server/Client was discussed in a previous post.
Prerequisite : Creating TCP Server/Client

Theory

In UDP, the client does not form a connection with the server like in TCP and instead

just sends a datagram. Similarly, the server need not accept a connection and just waits for datagrams to arrive. Datagrams upon arrival contain the address of sender which the server uses to send data to the correct client.



The entire process can be broken down into following steps :

UDP Server :

1. Create UDP socket.
2. Bind the socket to server address.
3. Wait until datagram packet arrives from client.
4. Process the datagram packet and send a reply to client.
5. Go back to Step 3.

UDP Client :

1. Create UDP socket.
2. Send message to server.
3. Wait until response from server is recieved.
4. Process reply and go back to step 2, if necessary.
5. Close socket descriptor and exit.

Necessary Functions :

```
int socket(int domain, int type, int protocol)

Creates an unbound socket in the specified domain.

Returns socket file descriptor.
```

Arguments :
domain – Specifies the communication
domain (AF_INET for IPv4/ AF_INET6 for IPv6)
type – Type of socket to be created
(SOCK_STREAM for TCP / SOCK_DGRAM for UDP)
protocol – Protocol to be used by socket.
0 means use default protocol for the address family.

```
int bind(int sockfd, const struct sockaddr *addr, socklen_t
addrlen)

Assigns address to the unbound socket.
```

Arguments :
sockfd – File descriptor of socket to be binded
addr – Structure in which address to be binded to is specified
addrlen – Size of *addr* structure

```
ssize_t sendto(int sockfd, const void *buf, size_t len, int flags,

               const struct sockaddr *dest_addr, socklen_t addrlen)

Send a message on the socket
```

Arguments :
sockfd – File descriptor of socket
buf – Application buffer containing the data to be sent
len – Size of *buf* application buffer
flags – Bitwise OR of flags to modify socket behaviour
dest_addr – Structure containing address of destination
addrlen – Size of *dest_addr* structure

```
ssize_t recvfrom(int sockfd, void *buf, size_t len, int flags,

                 struct sockaddr *src_addr, socklen_t *addrlen)
```

Receive a message from the socket.

Arguments :
sockfd – File descriptor of socket
buf – Application buffer in which to receive data
len – Size of *buf* application buffer
flags – Bitwise OR of flags to modify socket behaviour
src_addr – Structure containing source address is returned
addrlen – Variable in which size of *src_addr* structure is returned

int close(int fd)

Close a file descriptor

Arguments :
fd – File descriptor
In the below code, exchange of one hello message between server and client is shown to demonstrate the model.

- UDPServer.c
- UDPClient.c

filter_none

edit
play_arrow
brightness_4

```
// Client side implementation of UDP client-server model
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <arpa/inet.h>
#include <netinet/in.h>

#define PORT     8080
#define MAXLINE 1024

// Driver code
int main() {
    int sockfd;
    char buffer[MAXLINE];
    char *hello = "Hello from client";
    struct sockaddr_in     servaddr;

    // Creating socket file descriptor
```

```
    if ( (sockfd = socket(AF_INET, SOCK_DGRAM, 0)) < 0 ) {
        perror("socket creation failed");
        exit(EXIT_FAILURE);
    }

    memset(&servaddr, 0, sizeof(servaddr));

    // Filling server information
    servaddr.sin_family = AF_INET;
    servaddr.sin_port = htons(PORT);
    servaddr.sin_addr.s_addr = INADDR_ANY;

    int n, len;

    sendto(sockfd, (const char *)hello, strlen(hello),
        MSG_CONFIRM, (const struct sockaddr *) &servaddr,
            sizeof(servaddr));
    printf("Hello message sent.\n");

    n = recvfrom(sockfd, (char *)buffer, MAXLINE,
                MSG_WAITALL, (struct sockaddr *) &servaddr,
                &len);
    buffer[n] = '\0';
    printf("Server : %s\n", buffer);

    close(sockfd);
    return 0;
}
```

Output :

```
$ ./server
Client : Hello from client
Hello message sent.
$ ./client
Hello message sent.
Server : Hello from server
```

```
// Server side implementation of UDP client-server model

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <string.h>
```

```
#include <sys/types.h>
#include <sys/socket.h>
#include <arpa/inet.h>
#include <netinet/in.h>

#define PORT     8080
#define MAXLINE 1024

// Driver code
int main() {
    int sockfd;
    char buffer[MAXLINE];
    char *hello = "Hello from server";
    struct sockaddr_in servaddr, cliaddr;

    // Creating socket file descriptor
    if ( (sockfd = socket(AF_INET, SOCK_DGRAM, 0)) < 0 ) {
        perror("socket creation failed");
        exit(EXIT_FAILURE);
    }

    memset(&servaddr, 0, sizeof(servaddr));
    memset(&cliaddr, 0, sizeof(cliaddr));

    // Filling server information
```

```
    servaddr.sin_family    = AF_INET; // IPv4
    servaddr.sin_addr.s_addr = INADDR_ANY;
    servaddr.sin_port = htons(PORT);

    // Bind the socket with the server address
    if ( bind(sockfd, (const struct sockaddr *)&servaddr,
            sizeof(servaddr)) < 0 )
    {
        perror("bind failed");
        exit(EXIT_FAILURE);
    }

    int len, n;
    n = recvfrom(sockfd, (char *)buffer, MAXLINE,
                MSG_WAITALL, ( struct sockaddr *) &cliaddr,
                &len);
    buffer[n] = '\0';
    printf("Client : %s\n", buffer);
    sendto(sockfd, (const char *)hello, strlen(hello),
        MSG_CONFIRM, (const struct sockaddr *) &cliaddr,
            len);
    printf("Hello message sent.\n");

    return 0;
}
```

Output :

```
$ ./server
Client : Hello from client
Hello message sent.
$ ./client
Hello message sent.
Server : Hello from server
```

Conclusion: Thus in this we have studied that socket programming using TCP and UDP protocol.

D Y PATIL INSTITUTE OF ENGINEERING & TECHNOLOGY AMBI, PUNE	Configure Server FTP, DHCP, Telnet	
DEPARTMENT OF INFORMATION TECHNOLOGY		
EXPERIMENT NO. : DYPIET/IT/TE/SL-IV	**SEMESTER** : VI(TE)	**PAGE:1-**

AIM: Introduction to server administration and Configure Server FTP, DHCP, Telnet

OBJECTIVE: To study,

FTP,DHCP, Telnet protocol

Configuration of the same protocol

THEORY:

FTP:FTP is built on client-server architecture and uses separate control and data connections between the client and the server. FTP users may authenticate themselves using a clear-text sign-in protocol, normally in the form of a username and password, but can connect anonymously if the server is configured to allow it. For secure transmission that protects the username and password, and encrypts the content, FTP is often secured with SSL/TLS (FTPS). SSH File Transfer Protocol (SFTP) is sometimes also used instead, but is technologically different. FTP may run in active or passive mode, which determines how the data connection is established.
DHCP: With DHCP, computers request IP addresses and networking parameters automatically from a DHCP server, reducing the need for a network administrator or a user to configure these settings manually. The protocol operates based on the client-server model. DHCP is very common in all modern networks ranging in size from home networks to large campus networks and regional Internet service provider networks. Most residential network routers receive a globally unique IP address within the provider network. Within a local network, DHCP assigns a local IP address to devices connected to the local network.

Telnet: Telnet was developed in 1969 beginning with RFC 15, extended in RFC 854, and standardized as Internet Engineering Task Force (IETF) Internet Standard STD 8, one of the first Internet standards. Telnet provided access to a command-line interface (usually, of an operating system) on a remote host. Most network equipment and operating systems with a TCP/IP stack support a Telnet service for remote configuration (including systems based on Windows NT). The term telnet

may also refer to the software that implements the client part of the protocol. Telnet client applications are available for virtually all computer platforms.

CONFIGURATION/SIMULATION:

The configuration is completed on Windows server 2008 operating system. The steps are mentioned as below.

Install Telnet Server on Windows Server 2008 R2 and Windows Server 2008

1. Start Server Manager. Click Start, right-click Computer, and then click Manage.
2. If the User Account Control dialog box appears, confirm that the action it displays is what you want, and then click Continue.
3. In the Features Summary section, click Add features.
4. On the Select Features page, select Telnet Server. You can also select Telnet Client if you want.
5. Click next, and then on the Confirm Installation Options page, click Install.
6. On the Installation Results page, click Close.
7. Close Server Manager.

Install Telnet Client on Windows 7 or Windows Vista

1. Click Start, and then click Control Panel.
2. On the Control Panel Home page, click Programs.
3. In the Programs and Features section, click Turn Windows features on or off.
4. If the User Account Control dialog box appears, confirm that the action it displays is what you want, and then click Continue.
5. In the Windows Features list, select Telnet Client, and then click OK.

Installing and enabling IIS and FTP on Windows Server 2008 R2

1. Open Server Manager, go to Roles and click “Add Roles”
2. In the Add Role Wizard, select Web Server (IIS) role to install
3. Click Next until you reach Select Role Services page, leave the default and check FTP Server, FTP Service and FTP Extensibility at the bottom. Click Next, follow the wizard and finish the role installation.
4. Now open IIS Manager from Start > Administrative Tools, expand the server, right click Sites, and click Add FTP Site, give it a site name and configure the physical path as needed.
5. Configure Binding and SSL. In our case, we’d like to bind to all unassigned IP addresses and do not use SSL.
6. Enable Basic Authentication and configure authorization. In our case I’ll start with allowing All users both Read and Write permission as long as all users on the server are password protected.
7. Click Finish to finish the configuration.
8. Open Windows Firewall with Advanced Security from Start > Administrative Tools, go to Inbound Rules in the left pane, and create a new rule by clicking New Rule in the Action Pane, select Port and click next.
9. Apply this rule to TCP port 21, and click Next
10. Keep the default configure for the rest of steps to Allow the connection and apply it to all profiles, name the rule and finish the wizard.
11. Now the FTP should be up and running, please test the connection to confirm.

How to Install and Configure Windows Server 2008 DHCP Server

1. To start the DHCP installation process, you can click Add Roles from the Initial Configuration Tasks window or from Server Manager à Roles à Add Roles.
2. When the Add Roles Wizard comes up, you can click Next on that screen.
3. Next, select that you want to add the DHCP Server Role, and click next.

Follow the on screen instructions. The installation process will get complete.

CONCLUSION: Thus in this we have studied that FTP,DHCP and Telnet and configuration on server.

D Y PATIL INSTITUTE OF ENGINEERING & TECHNOLOGY AMBI, PUNE	**D Y PATIL INSTITUTE OF ENGINEERING & TECHNOLOGY, AMBI**	LABORATORY MANUAL
	EXPERIMENT TITLE: Using Network Simulator, Implement MANET / Wireless Sensor Network	
DEPARTMENT OF INFORMATION TECHNOLOGY		
EXPERIMENT NO. : DYPIET/IT/TE/SL-IV	**SEMESTER** : VI(TE)	**PAGE:1-**

AIM: Using Network Simulator, Implement MANET / Wireless Sensor Network

OBJECTIVES: Understand about the basics of Mobile and Adhoc network, various standards and different routing protocols including proactive and reactive Identifying and Simulating the MANET protocols using AODV and DSR with the network simulator NS2

Analyze the trace file and they can check the entire time duration of the simulation

THEORY:

Ad Hoc Network

In Latin, adhoc means "for this", further meaning "for this purpose only". Adhoc networks are temporary network, setup anywhere without any need of external infrastructure like wires or base stations.

Mobile Adhoc NETwork (MANET)

MANET is a self-configuring network of mobile devices connected by wireless links. Each devices moves indepedently in any direction. Each node acts as a router.Some devices will detach from some devices in that area and attached or make link with other devices.'

A typical MANET is shown in the figure-01 below

Mobile Adhoc Network MANET

Routing

Routing is selecting a path or route in a network for forwarding packets. The objective of routing packets in a network is to determine the best possible path in terms of minimizing the number of hops (path length), delay, packet loss, cost etc.

Routing in MANET

MANETs are formed dynamically by collecting arbitrary wireless mobile nodes, without use of existing network infrastructure. So routing in MANET is different from traditional routing. In MANET each node acts as both host and router. The nodes transmit and receive their own packets. The nodes also take part in forwarding packets for other nodes. Therefore MANET provides limited physical security as compared to the traditional network.

Routing protocols for MANET

Routing protocols for a MANET can be classified as:

Proactive (table-driven) : DSDV, OLSR etc.
Reactive (on-demand): AODV, DSR etc.

Hybrid: ZRP

Proactive routing protocols determine path in advance and periodically exchange routing data to maintain the path. Reactive routing protocols, on the other hand, determine a route to some destination node only when it is required to send some data to that node. If at any time a path fails, an alternative path is determined again. Hybrid routing takes the advantages of both table driven and on-demand algorithms.

Destination-Sequenced Distance-Vector(DSDV) algorithm:

The procedure for DSDV [1] is :

1. Each mobile node maintains a routing table with an entry of routing information from all its neighbors.
2. Each routing information in a routing table specifies

   a)the destination identifier
   b)the next hop on the route to the destination
   c)the distance(in terms of hops) to the destination
   d)a sequence number by monotonically increasing each time the node sends an update message to its neighbors. A route will be replaced only when the destination sequence number is less than the new one or two routes have the same sequence number but one has a lower metric.
3. After generating a new routing table, each node broadcasts this table to all its neighbors.
4. Based on the recieved tables, each mobile node update thier tables, until routing information is stable.

Dynamic source routing (DSR)

The DSR [v] protocol is composed of two mechanisms that work together to allow the discovery and maintenance of source routes in the ad hoc network:

1.**Route Discovery:** Route discovery is used only when source wants to send a packet to destination and does not know a route to destination.A mobile node A wants to send a packet to a destination node B, then obtain a source route to B.

2.**Route Maintenance** : Route Maintenance is the mechanism by which a source node A is able to detect, while using a source route to B. If the network topology has changed and the route is broken then the source route attempts to use any other route to destination if it exist or can invoke route discovery again to find a new route. Route Maintenance is used only when source is actually sending packets to destination.

Both Route Discovery and Route Maintenance each operate entirely on demand. When the destination node is reached, it returns a reply containing the route to the source node. The reply then travels in the reverse direction of the discovered route or on a path already known by the destination, to the source. The source node, on receiving the reply, will place the route in its route cache.

Application of MANET

The applications of MANET are:[iv]
- Military or police exercises.
- Disaster relief operations.
- Mine site operations.
- Robot data acquisition. etc.

Advantages

The following are the advantages of MANETs:[iv]
-They provide access to information and services regardless of geographic position.
-These networks can be set up at any place and time.
-These networks work without any pre-existing infrastructure.

Disadvantages

Some of the disadvantages of MANETs are:[iv]
-Limited resources.
-Limited physical security.
-Intrinsic mutual trust vulnerable to attacks.
-Lack of authorization facilities.
-Volatile network topology makes it hard to detect malicious nodes.

-Security protocols for wired networks cannot work for ad hoc networks.

SIMULATION:

The Network Simulator simulate following steps for MANET:

1. The command required to configure mobile a node :

```
$ns_ node-config -adhocRouting $val(rp)
-llType $val(ll) # LinkLayer
-macType $val(mac) #MAC type
-ifqType $val(ifq) #interface queue type
-ifqLen $val(ifqlen) #interface queue length
-antType $val(ant) #antenna type
-propType $val(prop) #propagation model
-phyType $val(netif) #network interface type
-topoInstance $topo #topography instance
-agentTrace ON #tracing at agent level
-routerTrace ON #tracing at router level
-macTrace ON #tracing at mac level
-movementTrace ON #mobile node movement
-channel $chan_1_
```

The four last option in node configuration can either be ON or OFF based on the condition of the mobile nodes. The agent trace will give the trace of TCP,

routerTrace provides tracing of packets used in routing, macTrace is used to trace MAC protocol packets and movementTrace is used to allow tracing the motion of nodes for nam.
Create some mobile nodes and assign them to the channels:
for {set i 0} {$i <$val(nn)} {incr i} {
set node_(i)[ns_ node]
$node_$(i) random-motion 0;
}

2. Assigning mobility to the node.
3. Specifying routing protocols.
   set val(rp) AODV; #for AODV
   set val(rp) DSR; #for DSR

CONCLUSION: Thus we have studied about MANET and WSN

D Y PATIL INSTITUTE OF ENGINEERING & TECHNOLOGY AMBI,PUNE	**D Y PATIL INSTITUTE OF ENGINEERING & TECHNOLOGY, AMBI**	LABORATORY MANUAL
	EXPERIMENT TITLE: Write a program using Arduino / Rasberry Pi Kit for Demonstration of IOT Application on any one of the following Topics. • Appliance Remote Control • Time Lapse Camera Controller • Security / Automation Sensors • The Traffic Light Controller Temperature Controller	

DEPARTMENT OF INFORMATION TECHNOLOGY	
EXPERIMENT NO. : DYPIET/IT/TE/SL-IV	**SEMESTER** : VI(TE)	**PAGE:1-**

AIM:

OBJECTIVES:

THEORY :

Raspberry Pi

Raspberry Pi is actually a system-on-a-chip, or SOC, for short. It runs a full version of Linux, such as Raspbian, and is designed to help teach you as you go. Arduino, on the other hand, is more of a micro-controller than a computer that has a massive support community as well as hundreds of expansion options.

When the Raspberry Pi was first released, it seemed to some that Arduino might now be obsolete. This, however, is not really fair as they perform different tasks. Whether the Raspberry Pi or the Arduino is your weapon of choice, the fact remains that each has distinct advantages and disadvantages over each other.

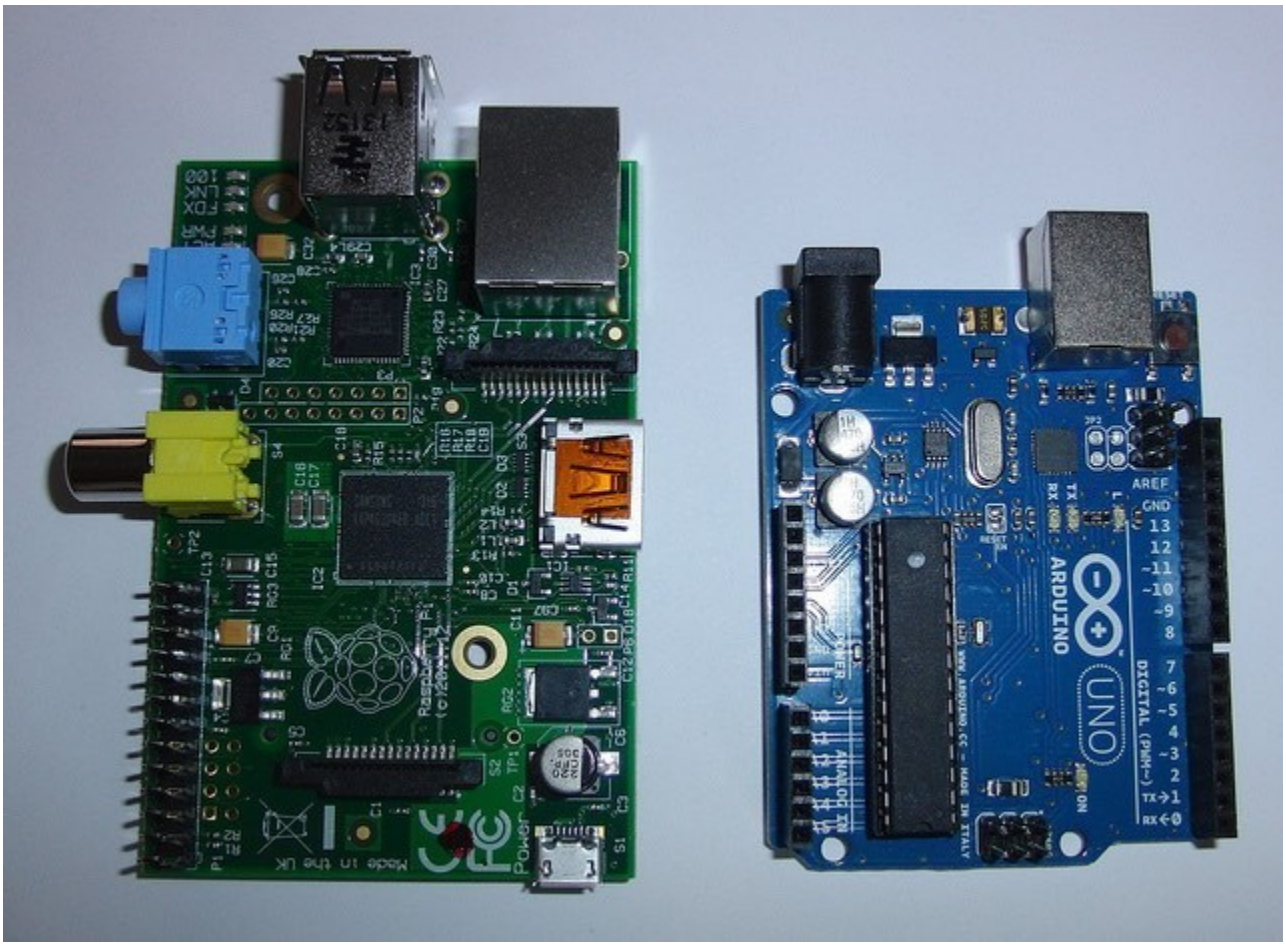


Fig:Raspberry Pi Model B und Arduino Uno

Arduino

Unlike the Raspberry Pi, Arduino boards are actually micro-controllers rather than 'full' computers. Arduino lacks a full operating system but can run written code that is interpreted by its firmware.

Because of this, you do lose access to basic tools that an OS would provide but you gain the flexibility of executing code directly with no OS overhead.

Arduino has no API and cannot provide user interactivity as there is no operating system. It basically runs code on 'bare metal'

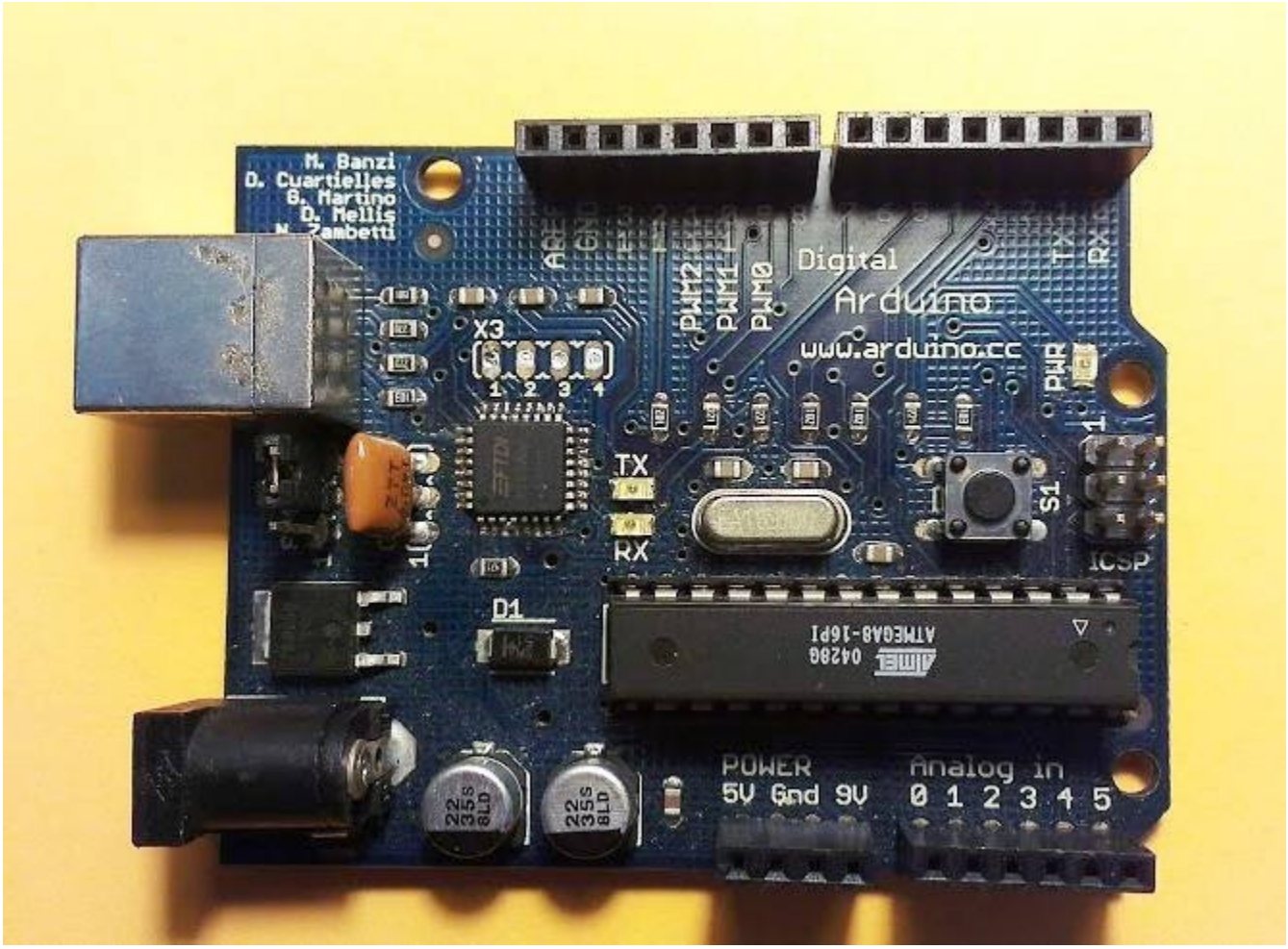


Fig: The first Arduino board, circa 2005. Source: *Ales9000/Wikimedia commons*

Arduino is really better suited as an interface for sensors and other devices. In this capacity its brilliant for hardware projects where you want something to respond to sensor readings or other inputs.

Pros of Arduino

-Arduino is easier to get started with

-Best used for real-time applications of hardware, software and IDE is open source

-You don't need a lot of programming knowledge for basic applications

-Very easy to extend and has a lot of contributed shields and libraries.

Cons of Arduino

-Not as powerful as the Raspberry Pi

-Can only be programmed using Arduino or C/C++

-Connection to the internet is more tricky than the Pi but is possible. You can pass data using YQL or JSON

What is the use of Raspberry Pi in IOT?

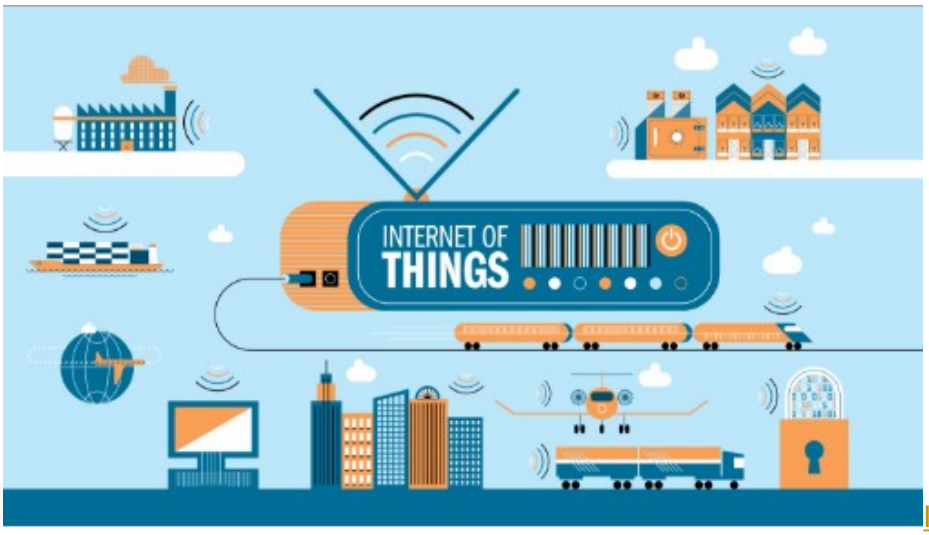
The Raspberry Pi is a powerful and inexpensive embedded computing platform that has great community support. It is powerful enough to run a full Linux operating **system**, comes with Java SE pre-**installed**, and has digital input/output ports that you can use to interact with LEDs, buttons, sensors, and motors.

IOT using Raspberry Pi

IOT using raspberry pi mainly incldue what is an IOT, Raspberry pi, IOT design methodology, etc.

What is Internet of Things ?

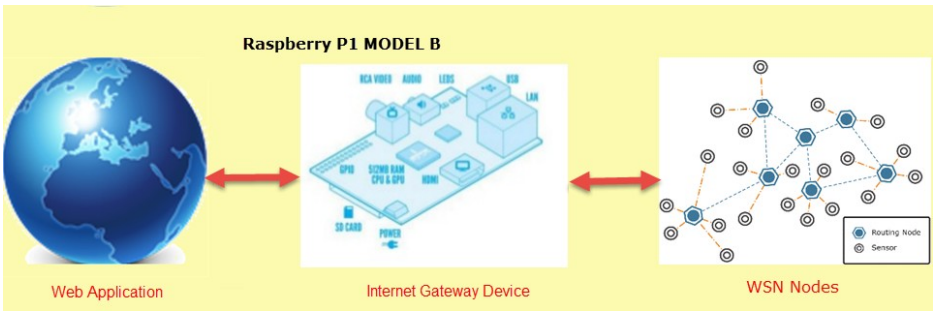
The Internet of Things (IoT) is a scenario in which objects, animals or people are provided with single identifiers and the capability to automatically transfer and the capability to automatically transfer data more to a network without requiring human-to-human or human-to-computer communication. IOT has evolved from the meeting of wireless technologies, micro-electromechanical systems (MEMS) and the internet.



Internet of Things

IOT Design Methodology

All web application is developed natively in Java Programming Language. It includes java technologies similar to JSP, servelts, hibernate, and web services etc., latest version of net beans IDE is basically used for web applications development. Additional technologies like bootstrap, java script, jQuery etc are used to handle UI and client side validations. Cisco provided APIs are used to develop application related to Cisco IP phones.



IOT uisng Raspberry Pi

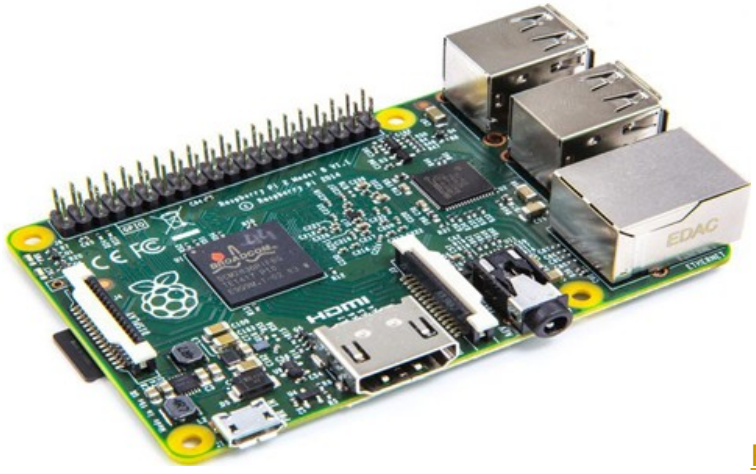
Five steps are used in web applications

- Installing Apache Web server
- Create My SQL database system
- Developed web application For the GUI (Graphical User Interface)

- Write lots of PHP, JAVA script, CSS and Python Programs for the Web Application
- Host Web application on our Web server

Raspberry Pi

The history of the Raspberry Pi was basically introduced in 2006. Its main concept is based on Atmel ATmega644 which is particularly designed for educational use and intended for Python. A Raspberry Pi is of small size i.e., of a credit card sized single board computer, which is developed in the United Kingdom(U.K) by a foundation called Raspberry Pi. The main motto of this foundation is to promote the teaching of basic computer science in the education institutes and also in developing countries. The first generation of Raspberry (Pi 1) was released in the year 2012, that has two types of models namely model A and model B.



Raspberry Pi

In the subsequent year A+ and B+ models were released. Again in 2015, Raspberry Pi2 model B was released and a immediate year Raspberry Pi3 model B was released in the market.

Raspberry Pi can be plugged into a TV, computer monitor, and it uses a standard keyboard and mouse. It is user friendly as can be handled by all the age groups. It does everything you would expect a desktop computer to do like word-processing, browsing the internet spreadsheets, playing games to playing high definition videos. It is used in

many applications like in a wide array of digital maker projects, music machines, parent detectors to the weather station and tweeting birdhouses with infrared cameras.

All models feature on a broadcom system on a chip (SOC), which includes chip graphics processing unit GPU(a Video Core IV), an ARM compatible and CPU. The CPU speed ranges from 700 MHz to 1.2 GHz for the Pi 3 and on board memory range from 256 MB to 1 GB RAM. An operating system is stored in the secured digital SD cards and program memory in either the MicroSDHC or SDHC sizes. Most boards have one to four USB slots, composite video output, HDMI and a 3.5 mm phone jack for audio. Some models have WiFi and Bluetooth.

The Raspberry Pi Foundation provides Arch Linux ARM and Debian distributions for download, and promotes Python as the main programming language, with support for BBC BASIC, Java, C, Perl,Ruby, PHP, Squeak Smalltalk, C++, etc.

The following are essential to get started

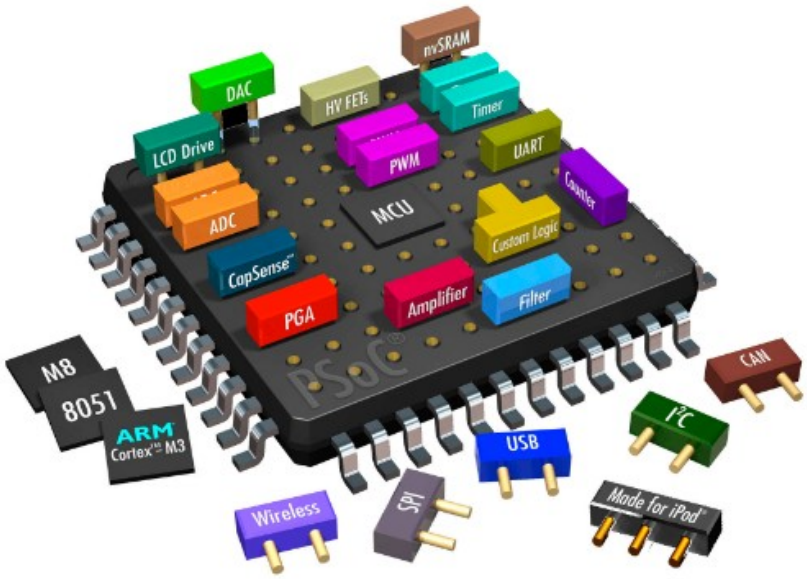
- Video cable to suit the TV or monitor used
- SD card containing Linux Operating system
- Power supply (see Section 1.6 below)
- USB keyboard
- TV or monitor (with DVI, HDMI, Composite or SCART input)

Recommended optional extras include

- Internet connection, Model B only: LAN (Ethernet) cable
- USB mouse
- Powered USB hub
- Internet connection, Model A or B: USB WiFi adaptor

What is a System on Chip?

A system on chip is a complex IC that integrates the functional elements into a single chip or chipset. It is a programmable processor on a chip memory, accelerating function hardware, software, hardware and analog components.



System on Chip

Benefits of SoC

- Lower power consumption
- Reduces size
- Reduces overall system cost
- Increases performance

Internet Gateway Device

Internet Gateway Device has the ability to route data approaching from the WSN network to the internet and Send data coming from the internet to WSN network. It is like Wi-Fi router for Internet of Things. In the internet gateway device we use raspberry pi model B, it features a quad-core ARM Cortex- A7 CPU is running at 900MHz (for a 6x presentation improve on the first generation Raspberry Pi Model B+) and 1GB of LPDDR2 SDRAM (for a 2x memory increase). And yes, there is total compatibility with Raspberry Pi1 we are secured. Broadcom's new SoC, the BCM2836, is the key factor. Five steps we are using Internet Gateway Device

- Port Linux operating system on raspberry Pi
- Modify Linux to work with Our Prototype
- Developed Python Library for Communication of RPI with Xbee ZB
- Wrote Program from sensors and Device controlling
- Create WI-FI functionality on RPI for Internet Connection

WSN Nodes

A wireless sensor network (WSN) consists of three main components: nodes, gateways, and software. The spatially dispersed measurement nodes interface with the sensors to monitor assets or their surroundings.The acquired information is wirelessly transmitted to the gateway, which provides a connection to the wired globe where you can collect, procedure, analyze, and present your measurement information using the software. Routers are an individual type of dimension node that you can use to expand the distance and dependability in a WSN. Sensors can be dispersed on the roads, vehicles, hospitals, buildings, people and allow dissimilar applications such as medical services, battlefield operations, disaster response, disaster relief and environmental monitoring.

IOT Applications

- Weather security and temperature cam
- Working doctor who props with raspberry pi
- Sensually an air quality monitoring hat
- Beer and wine fridge of awesomeness
- Raspberry pi Internet doorbell
- Internet of things toilet
- Train your rat behavioral science at home
- Pebbly smart doorbell
- The raspberry pi microwave

This is all about IOT using Raspberry Pi. Currently, IOT is made up of a loose collection of different, purpose-built networks. Today's cars, intended for example, have multiple networks to control engine function, safety features, communication systems, and so on. Commercial and residential buildings also have various control systems for heating, venting, and air condition (HVAC), telephone service, security, and lighting.

As IOT evolves, these networks, and a lot of others will be connected with additional security, analytics, and management capabilities. This will allow IOT to become even more powerful in what it can help people achieve. Furthermore, any queries regarding this concept or electrical and electronics projects, please give your valuable suggestions by commenting in the comment section below.

CONCLUSION: Thus in this we have studied that