

SCTR's
PUNE INSTITUTE OF COMPUTER TECHNOLOGY



THIRD YEAR
Information technology
(2015 Course)

LABORATORY MANUAL
For

SOFTWARE LABORATORY - IV

SEMESTER - II

[Subject code: 314455]

[Prepared By]

Mr. Tushar A. Rane
Mr. Naman V. Buradkar

**PUNE INSTITUTE OF COMPUTER TECHNOLOGY
DEPARTMENT OF INFORMATION TECHNOLOGY**

LABORATORY MANUAL

AY 2018-19

**314455: SOFTWARE LAB – iv
THIRD YEAR – INFORMATION TECHNOLOGY
SEMESTER-II**

TEACHING SCHEME

Lectures: 3 Hrs/Week

Theory: 70 Marks

Practical: 2 Hrs/Week

In-Sem: 30 Marks

Practical: 25 Marks

Term Work: 25 Marks

::|| PrePared By ||::

Mr. Tushar a. rane

Mr. naman v. buradkar

314455 : SOFTWARE LABORATORY – IV

Teaching Scheme:	Credits	Examination Scheme:
Practical : 2 Hours/Week	01	Term Work : 25 Marks Oral : 25 Marks

Prerequisites:

1. Fundamentals of computer Networks.

Course Objectives :

1. To design and implement small size network and to understand various networking commands
2. To provide the knowledge of various networking tools and their related concepts
3. To understand various application layer protocols for its implementation in client/server environment
4. To understand network layer protocols and its implementations.
5. To explore and understand various simulations tools for network applications.
6. To understand the fundamentals of wireless networks and standards.

Course Outcomes :

1. To implement small size network and its use of various networking commands.
2. To understand and use various networking and simulations tools.
3. To configure various client/server environments to use application layer protocols
4. To understand the protocol design at various layers.
5. To explore use of protocols in various wired and wireless applications.
6. To develop applications on emerging trends.

Guidelines for Instructor's Manual

1. The faculty member should prepare the laboratory manual for all the experiments and it should be made available to students and laboratory instructor/Assistant

Guidelines for Student's Lab Journal

1. Student should submit term work in the form of handwritten journal based on specified list of assignments.
2. Practical Examination will be based on the term work.
3. Candidate is expected to know the theory involved in the experiment.
4. The practical examination should be conducted if and only if the journal of the candidate is complete in all respects.

Guidelines for Lab /TW Assessment

1. Examiners will assess the term work based on performance of students considering the parameters such as timely conduction of practical assignment, methodology adopted for implementation of practical assignment, timely submission of assignment in the form of handwritten write-up along with results of implemented assignment, attendance etc.
2. Examiners will judge the understanding of the practical performed in the examination by asking some questions related to theory & implementation of experiments he/she has carried out.
3. Appropriate knowledge of usage of software and hardware related to respective laboratory should be checked by the concerned faculty member.

As a conscious effort and little contribution towards Green IT and environment awareness, attaching printed papers of the program in journal may be avoided. There must be hand-written write-ups for every assignment in the journal. The DVD/CD containing students programs should be attached to the journal by

every student and same to be maintained by department/lab In-charge is highly encouraged. For reference one or two journals may be maintained with program prints at Laboratory.

Suggested List of Laboratory Assignments

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

- | | |
|------------------------|---------------------------------|
| a) Ping | g) Tracert/Traceroute/Tracepath |
| b) ipconfig / ifconfig | h) NSlookup |
| c) Hostname | i) Arp |
| d) Whois | j) Finger |
| e) Netstat | k) Port Scan / nmap |
| f) Route | |

2. Using a Network Simulator (e.g. packet tracer) Configure

- Sub-netting of a given network
- Super-netting of a given networks.

3. Using a Network Simulator (e.g. packet tracer) Configure

- A router using router commands,
- Access Control lists – Standard & Extended.

4. Using a Network Simulator (e.g. packet tracer) Configure

- EIGRP – Explore Neighbor-ship Requirements and Conditions, its K Values Metrics Assignment and Calculation,
- RIPv2 and EIGRP on same network.
- WLAN with static IP addressing and DHCP with MAC security and filters

5. Using a Network Simulator (e.g. packet tracer) Configure

- VLAN, Dynamic trunk protocol and spanning tree protocol
- OSPF – Explore Neighbor-ship Condition and Requirement, Neighbor-ship states, OSPF Metric Cost Calculation.
- Network Address Translation : Static, Dynamic & PAT (Port Address Translation)

6. Socket Programming in C/C++ on Linux.

- TCP Client , TCP Server
- UDP Client , UDP Server

7. Introduction to server administration (server administration commands and their applications) and configuration any three of below Server : (Study/Demonstration Only)

- FTP, Web Server, DHCP, Telnet, Mail, DNS

8. Using any open source Network Simulator, Implement

- MANET / Wireless Sensor Network

9. Write a program using Arduino / Raspberry 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

References

1. Andrew S. Tanenbaum, David J. Wethrall, Computer Network, Pearson Education, ISBN : 978-0-13-212695-3.
2. Kurose Ross, Computer Networking: A Top Down Approach Featuring the Internet, Pearson Education, ISBN :978-81-7758-878-1.

3. Charles E. Perkins, Adhoc Networking, Pearson Education, 978-81-317-2096-7.
4. Andrea Goldsmith, Wireless Communication, Cambridge University Press, ISBN:978-0-521-83716-3.
5. Mayank Dave, Computer Network, Cengage Learning, ISBN :978-81-315-0986-9.
6. C. K. Toh, Ad Hoc Mobile Wireless Networks Protocols and Systems, Prentice Hall, ISBN:978-01-324-42046.
7. Paul Goransson, Chuck Black, Software Defined Networks: A Comprehensive Approach, Morgan Kaufmann, ISBN:978-0124166752.

Basic Networking Commands

This assignment explains basic networking commands such as ipconfig, ping, route, tracert, traceroute, arp, netstat, NetBIOS, nslookup, finger, nmap/port scan in detail with examples.

Ipconfig

Displays all current TCP/IP network configuration values and refreshes Dynamic Host Configuration Protocol (DHCP) and Domain Name System (DNS) settings. This command is most useful on computers that are configured to obtain an IP address automatically. This enables users to determine which TCP/IP configuration values have been configured by DHCP, Automatic Private IP Addressing (APIPA), or an alternate configuration.

- If the Adapter name contains any spaces, use quotation marks around the adapter name (that is, "Adapter Name").
- For adapter names, ipconfig supports the use of the asterisk (*) wildcard character to specify either adapters with names that begin with a specified string or adapters with names that contain a specified string.
- For example, **Local*** matches all adapters that start with the string Local and ***Con*** matches all adapters that contain the string Con.

Syntax

```
ipconfig [/all] [/renew [Adapter]] [/release [Adapter]] [/flushdns] [/displaydns] [/registerdns]
[/showclassid Adapter] [/setclassid Adapter [ClassID]]
```

Parameters

Used without parameters, ipconfig displays the IP address, subnet mask, and default gateway for all adapters.

/all Displays the full TCP/IP configuration for all adapters. Without this parameter, ipconfig displays only the IP address, subnet mask, and default gateway values for each adapter. Adapters can represent physical interfaces, such as installed network adapters, or logical interfaces, such as dial-up connections.

/renew [Adapter] Renews DHCP configuration for all adapters (if an adapter is not specified) or for a specific adapter if the Adapter parameter is included. This parameter is available only on computers with adapters that are configured to obtain an IP address automatically. To specify an adapter name, type the adapter name that appears when you use ipconfig without parameters.

/release [Adapter] Sends a DHCPRELEASE message to the DHCP server to release the current DHCP configuration and discard the IP address configuration for either all adapters (if an adapter is not specified) or for a specific adapter if the Adapter parameter is included. This parameter disables TCP/IP for adapters configured to obtain an IP address automatically. To specify an adapter name, type the adapter name that appears when you use ipconfig without parameters.

/flushdns Flushes and resets the contents of the DNS client resolver cache. During DNS troubleshooting, you can use this procedure to discard negative cache entries from the cache, as well as any other entries that have been added dynamically.

/displaydns Displays the contents of the DNS client resolver cache, which includes both entries preloaded from the local Hosts file and any recently obtained resource records for name queries resolved by the computer. The DNS Client service uses this information to resolve frequently queried names quickly, before querying its configured DNS servers.

/registerdns Initiates manual dynamic registration for the DNS names and IP addresses that are configured at a computer. You can use this parameter to troubleshoot a failed DNS name registration or resolve a dynamic update problem between a client and the DNS server without rebooting the client computer. The DNS settings in the advanced properties of the TCP/IP protocol determine which names are registered in DNS.

/showclassid Adapter Displays the DHCP class ID for a specified adapter. To see the DHCP class ID for all adapters, use the asterisk (*) wildcard character in place of Adapter. This parameter is available only on computers with adapters that are configured to obtain an IP address automatically.

/setclassid Adapter [ClassID] Configures the DHCP class ID for a specified adapter. To set the DHCP class ID for all adapters, use the asterisk (*) wildcard character in place of Adapter. This parameter is available only on computers with adapters that are configured to obtain an IP address automatically. If a DHCP class ID is not specified, the current class ID is removed.

Examples:

To display the basic TCP/IP configuration for all adapters, type:

- ipconfig

To display the full TCP/IP configuration for all adapters, type:

- ipconfig /all

To renew a DHCP-assigned IP address configuration for only the Local Area Connection adapter, type:

- ipconfig /renew "Local Area Connection"

To flush the DNS resolver cache when troubleshooting DNS name resolution problems, type:

- ipconfig /flushdns

To display the DHCP class ID for all adapters with names that start with Local, type:

- ipconfig /showclassid Local

To set the DHCP class ID for the Local Area Connection adapter to TEST, type:

- ipconfig /setclassid "Local Area Connection" TEST

winipcfg

This utility allows users or administrators to see the current IP address and other useful information about your network configuration. You can reset one or more IP addresses. The Release or Renew buttons allow you to release or renew one IP address. If you want to release or renew all IP addresses click Release All or Renew All. When one of these buttons is clicked, a new IP address is obtained from either the DHCP service or from the computer assigning itself an automatic private IP address. **To use the winipcfg utility:**

- Click Start, and then click Run and type **winipcfg**
- Click More Info.
- To see the addresses of the DNS servers the computer is configured to use, click the ellipsis (...) button to the right of DNS Servers.
- To see address information for your network adapter(s), select an adapter from the list in Ethernet Adapter Information.

Ping

Verifies IP-level connectivity to another TCP/IP computer by sending Internet Control Message Protocol (ICMP) Echo Request messages. The receipt of corresponding Echo Reply messages are displayed, along with round-trip times. Ping is the primary TCP/IP command used to troubleshoot connectivity, reachability, and name resolution.

```
K:\WINNT\system32\cmd.exe
C:\>
C:\>ping 192.168.0.1

Pinging 192.168.0.1 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 192.168.0.1:
  Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
Approximate round trip times in milli-seconds:
  Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>ping 192.168.1.1

Pinging 192.168.1.1 with 32 bytes of data:
Reply from 192.168.1.1: bytes=32 time<10ms TTL=64

Ping statistics for 192.168.1.1:
  Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
  Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>
```

You can use ping to test both the computer name and the IP address of the computer. If pinging the IP address is successful, but pinging the computer name is not, you might have a name resolution problem. In this case, ensure that the computer name you are specifying can be resolved through the local Hosts file, by using Domain Name System (DNS) queries, or through NetBIOS name resolution techniques.

To test a TCP/IP configuration by using the ping command:

- To quickly obtain the TCP/IP configuration of a computer, open Command Prompt, and then type **ipconfig**. From the display of the ipconfig command, ensure that the network adapter for the TCP/IP configuration you are testing is not in a Media disconnected state.
- At the command prompt, ping the loopback address by typing **ping 127.0.0.1**.
- Ping the IP address of the computer.
- Ping the IP address of the default gateway. If the ping command fails, verify that the default gateway IP address is correct and that the gateway (router) is operational.
- Ping the IP address of a remote host (a host that is on a different subnet). If the ping command fails, verify that the remote host IP address is correct, that the remote host is operational, and that all of the gateways (routers) between this computer and the remote host are operational.

- Ping the IP address of the DNS server. If the ping command fails, verify that the DNS server IP address is correct, that the DNS server is operational, and that all of the gateways (routers) between this computer and the DNS server are operational.

hostname

The **hostname** command shows or sets the system hostname.

hostname is used to display the system's DNS name, and to display or set its hostname or NIS (Network Information Services) domain name.

When called without any arguments, **hostname** will display the name of the system as returned by the `gethostname` function.

When called with one argument or with the `--file` option, **hostname** will set the system's host name using the `sethostname` function. Only the superuser can set the host name.

The host name is usually set once at system startup in the script `/etc/init.d/hostname.sh` normally by reading the contents of a file which contains the host name, e.g., `/etc/hostname`.

hostname syntax

```
hostname [-v] [-a|--alias] [-d|--domain] [-f|--fqdn|--long] [-A|--all-fqdns]
         [-i|--ip-address] [-l|--all-ip-addresses] [-s|--short] [-y|--yp|--nis]
hostname [-v] [-b|--boot] [-F|--file file name] [hostname]
hostname [-v] [-h|--help] [-V|--version]
```

Options

-a, --alias	Display the alias name of the host (if used). This option is deprecated and should not be used anymore.
-A, --all-fqdns	Displays every FQDN of the machine. This option enumerates all configured network addresses on all configured network interfaces, and translates them to DNS domain names. Addresses that cannot be translated (i.e. because they do not have an appropriate reverse DNS entry) are skipped. Note that different addresses may resolve to the same name, therefore the output may contain duplicate entries. Do not make any assumptions about the order of the output.
-b, --boot	Always set a hostname; this allows the file specified by <code>-F</code> to be non-existent or empty, in which case the default hostname

	localhost will be used if none is yet set.
-d, --domain	Display the name of the DNS domain. Don't use the command domainname to get the DNS domain name because it will show the NIS domain name and not the DNS domain name. Use dnsdomainname instead. See the warnings in section The FQDN, and avoid using this option if at all possible.
-f, --fqdn, --long	Display the FQDN (Fully Qualified Domain Name). A FQDN consists of a short host name and the DNS domain name. Unless you are using bind (Berkeley Internet Domain Name) or NIS for host lookups, you can change the FQDN and the DNS domain name (which is part of the FQDN) in the /etc/hosts file. See the warnings in section The FQDN, and avoid using this option if at all possible; use hostname --all-fqdns instead.
-F, --file <i>file name</i>	Read the host name from the specified file. Comments (lines starting with a '#') are ignored.
-i, --ip-address	Display the network address(es) of the host name. Note that this works only if the host name can be resolved. Avoid using this option if at all possible; use hostname --all-ip-addresses instead.
-l, --all-ip-addresses	Display all network addresses of the host. This option enumerates all configured addresses on all network interfaces. The loopback interface and IPv6 link-local addresses are omitted. Contrary to option -i , this option does not depend on name resolution. Do not make any assumptions about the order of the output.
-s, --short	Display the short host name. This is the host name cut at the first dot.
-v, --verbose	Be verbose with all output.
-V, --version	Print version information on standard output and exit successfully.
-y, --yp, --nis	Display the NIS domain name. If a parameter is given (or --file name) then root (the superuser) can also set a new NIS domain.
-h, --help	Print a help message and exit.

Netstat

Displays active TCP connections, ports on which the computer is listening, Ethernet statistics, the IP routing table, IPv4 statistics (for the IP, ICMP, TCP, and UDP protocols), and IPv6 statistics (for the IPv6, ICMPv6, TCP over IPv6, and UDP over IPv6 protocols).

```

Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\yongmo.FSMY>netstat -nb
Active Connections

  Proto  Local Address          Foreign Address        State      PID
  TCP    10.70.1.71:1306        10.70.0.10:1910      ESTABLISHED 2580
                                                [Communicator.exe]
  TCP    10.70.1.71:1308        10.70.0.10:1910      ESTABLISHED 2580
                                                [Communicator.exe]
  TCP    10.70.1.71:1319        10.70.0.10:1910      ESTABLISHED 2580
                                                [Communicator.exe]
  TCP    10.70.1.71:1334        10.70.0.10:1910      ESTABLISHED 2580
                                                [Communicator.exe]
  TCP    10.70.1.71:1581        10.70.0.10:1910      ESTABLISHED 2800
                                                [OUTLOOK.EXE]
  TCP    10.70.1.71:1854        10.70.0.10:1910      ESTABLISHED 2580
                                                [Communicator.exe]
  TCP    10.70.1.71:2109        10.70.0.10:1910      ESTABLISHED 2580
                                                [Communicator.exe]

```

Netstat provides statistics for the following:

- Proto - The name of the protocol (TCP or UDP).
- Local Address - The IP address of the local computer and the port number being used. The name of the local computer that corresponds to the IP address and the name of the port is shown unless the -n parameter is specified. If the port is not yet established, the port number is shown as an asterisk (*).
- Foreign Address - The IP address and port number of the remote computer to which the socket is connected. The names that corresponds to the IP address and the port are shown unless the -n parameter is specified. If the port is not yet established, the port number is shown as an asterisk (*)�.

(state) Indicates the state of a TCP connection. The possible states are as follows:

- CLOSE_WAIT
- CLOSED
- ESTABLISHED
- FIN_WAIT_1
- FIN_WAIT_2
- LAST_ACK
- LISTEN
- SYN_RECEIVED
- SYN_SEND
- TIMED_WAIT

Syntax

```
netstat [-a] [-e] [-n] [-o] [-p Protocol] [-r] [-s] [Interval]
```

Parameters

Used without parameters, netstat displays active TCP connections.

-a Displays all active TCP connections and the TCP and UDP ports on which the computer is listening.

-e Displays Ethernet statistics, such as the number of bytes and packets sent and received. This parameter can be combined with -s.

-n Displays active TCP connections, however, addresses and port numbers are expressed numerically and no attempt is made to determine names.

-o Displays active TCP connections and includes the process ID (PID) for each connection. You can find the application based on the PID on the Processes tab in Windows Task Manager. This parameter can be combined with -a, -n, and -p.

-p Shows connections for the protocol specified by Protocol. In this case, the Protocol can be tcp, udp, tcpv6, or udpv6. If this parameter is used with -s to display statistics by protocol, Protocol can be tcp, udp, icmp, ip, tcpv6, udpv6, icmpv6, or ipv6.

-s Displays statistics by protocol. By default, statistics are shown for the TCP, UDP, ICMP, and IP protocols. If the IPv6 protocol for Windows XP is installed, statistics are shown for the TCP over IPv6, UDP over IPv6, ICMPv6, and IPv6 protocols. The -p parameter can be used to specify a set of protocols.

-r Displays the contents of the IP routing table. This is equivalent to the route print command.

Interval Redisplays the selected information every Interval seconds. Press CTRL+C to stop the redisplay. If this parameter is omitted, netstat prints the selected information only once.

/? - Displays help at the command prompt.

Nbtstat

Displays NetBIOS over TCP/IP (NetBT) protocol statistics,



NetBIOS

NetBIOS name tables for both the local computer and remote computers, and the NetBIOS name cache. Nbtstat allows a refresh of the NetBIOS name cache and the names registered with Windows Internet Name Service (WINS).

Nbtstat command-line parameters are case-sensitive.

Syntax

```
nbtstat [-a RemoteName] [-A IPAddress] [-c] [-n] [-r] [-R] [-RR] [-s] [-S] [Interval]
```

Parameters

Used without parameters, nbtstat displays help.

-a RemoteName Displays the NetBIOS name table of a remote computer, where RemoteName is the NetBIOS computer name of the remote computer. The NetBIOS name table is the list of NetBIOS names that corresponds to NetBIOS applications running on that computer.

-A IPAddress Displays the NetBIOS name table of a remote computer, specified by the IP address (in dotted decimal notation) of the remote computer.

-c Displays the contents of the NetBIOS name cache, the table of NetBIOS names and their resolved IP addresses.

-n Displays the NetBIOS name table of the local computer. The status of Registered indicates that the name is registered either by broadcast or with a WINS server.

-r Displays NetBIOS name resolution statistics. On a Windows XP computer that is configured to use WINS, this parameter returns the number of names that have been resolved and registered using broadcast and WINS.

-R Purges the contents of the NetBIOS name cache and then reloads the #PRE-tagged entries from the Lmhosts file.

-RR Releases and then refreshes NetBIOS names for the local computer that is registered with WINS servers.

-s Displays NetBIOS client and server sessions, attempting to convert the destination IP address to a name.

-S Displays NetBIOS client and server sessions, listing the remote computers by destination IP address only.

Interval Redisplays selected statistics, pausing the number of seconds specified in Interval between each display. Press CTRL+C to stop redisplaying statistics. If this parameter is omitted, nbtstat prints the current configuration information only once.

/? - Displays help at the command prompt.

Route

route - show / manipulate the IP routing table

Tag	Description
route	[-v] [-A family] add [-net -host] target [netmask Nm] [gw Gw] [metric N] [mss M] [window W] [irtt I] [reject] [mod] [dyn] [reinstate] [[dev] If]
route	[-v] [-A family] del [-net -host] target [gw Gw] [netmask Nm] [metric N] [[dev] If]
route	[-V] [--version] [-h] [-help]

Route manipulates the kernel's IP routing tables. Its primary use is to set up static routes to specific hosts or networks via an interface after it has been configured with the ifconfig(8) program.

When the add or del options are used, route modifies the routing tables. Without these options, route displays the current contents of the routing tables.

OPTIONS

Tag	Description
-A family	
	use the specified address family (eg 'inet'; use 'route --help' for a full list).
-F	operate on the kernel's FIB (Forwarding Information Base) routing table. This is the default.
-C	operate on the kernel's routing cache.
-v	select verbose operation.
-n	show numerical addresses instead of trying to determine symbolic host names. This is useful if you are trying to determine why the route to your nameserver has vanished.
-e	use netstat(8)-format for displaying the routing table. -ee will generate a very long line with all parameters from the routing table.
del	delete a route.
add	add a new route.
target	the destination network or host. You can provide IP addresses in dotted decimal or host/network names.
-net	the target is a network.
-host	the target is a host.
netmask NM	
	when adding a network route, the netmask to be used.
gw GW	route packets via a gateway. NOTE: The specified gateway must be reachable first. This usually means that you have to set up a static route to the gateway beforehand. If you specify the address of one of your local interfaces, it will be used to decide about the interface to which the packets should be routed to. This is a BSDism compatibility hack.

metric M	
	set the metric field in the routing table (used by routing daemons) to M.
mss M	set the TCP Maximum Segment Size (MSS) for connections over this route to M bytes. The default is the device MTU minus headers, or a lower MTU when path mtu discovery occurred. This setting can be used to force smaller TCP packets on the other end when path mtu discovery does not work (usually because of misconfigured firewalls that block ICMP Fragmentation Needed)
window W	
	set the TCP window size for connections over this route to W bytes. This is typically only used on AX.25 networks and with drivers unable to handle back to back frames.
irtt I	set the initial round trip time (irtt) for TCP connections over this route to I milliseconds (1-12000). This is typically only used on AX.25 networks. If omitted the RFC 1122 default of 300ms is used.
reject	install a blocking route, which will force a route lookup to fail. This is for example used to mask out networks before using the default route. This is NOT for firewalling.
mod, dyn, reinstate	
	install a dynamic or modified route. These flags are for diagnostic purposes, and are generally only set by routing daemons.
dev If	<p>force the route to be associated with the specified device, as the kernel will otherwise try to determine the device on its own (by checking already existing routes and device specifications, and where the route is added to). In most normal networks you won't need this.</p> <p>If dev If is the last option on the command line, the word dev may be omitted, as it's the default. Otherwise the order of the route modifiers (metric - netmask - gw - dev) doesn't matter.</p>

EXAMPLES

Tag	Description
route add -net 127.0.0.0	

	adds the normal loopback entry, using netmask 255.0.0.0 (class A net, determined from the destination address) and associated with the "lo" device (assuming this device was previously set up correctly with ifconfig(8)).
route add -net 192.56.76.0 netmask 255.255.255.0 dev eth0	
	adds a route to the network 192.56.76.x via "eth0". The Class C netmask modifier is not really necessary here because 192.* is a Class C IP address. The word "dev" can be omitted here.
route add default gw mango-gw	
	adds a default route (which will be used if no other route matches). All packets using this route will be gatewayed through "mango-gw". The device which will actually be used for that route depends on how we can reach "mango-gw" - the static route to "mango-gw" will have to be set up before.
route add ipx4 sl0	
	Adds the route to the "ipx4" host via the SLIP interface (assuming that "ipx4" is the SLIP host).
route add -net 192.57.66.0 netmask 255.255.255.0 gw ipx4	
	This command adds the net "192.57.66.x" to be gatewayed through the former route to the SLIP interface.
route add -net 224.0.0.0 netmask 240.0.0.0 dev eth0	
	This is an obscure one documented so people know how to do it. This sets all of the class D (multicast) IP routes to go via "eth0". This is the correct normal configuration line with a multicasting kernel.
route add -net 10.0.0.0 netmask 255.0.0.0 reject	
	This installs a rejecting route for the private network "10.x.x.x."

Tracert / traceroute

Tracert: Determines the path taken to a destination by sending Internet Control Message Protocol (ICMP) Echo Request messages to the destination with incrementally increasing Time to Live (TTL) field values. The path displayed is the list of near-side router interfaces of the routers in the path between a source host and a destination. The near-side interface is the interface of the

router that is closest to the sending host in the path. Used without parameters, tracert displays help.

This diagnostic tool determines the path taken to a destination by sending ICMP Echo Request messages with varying Time to Live (TTL) values to the destination. Each router along the path is required to decrement the TTL in an IP packet by at least 1 before forwarding it.

Effectively, the TTL is a maximum link counter. When the TTL on a packet reaches 0, the router is expected to return an ICMP Time Exceeded message to the source computer. Tracert determines the path by sending the first Echo Request message with a TTL of 1 and incrementing the TTL by 1 on each subsequent transmission until the target responds or the maximum number of hops is reached. The maximum number of hops is 30 by default and can be specified using the -h parameter.

The path is determined by examining the ICMP Time Exceeded messages returned by intermediate routers and the Echo Reply message returned by the destination. However, some routers do not return Time Exceeded messages for packets with expired TTL values and are invisible to the tracert command. In this case, a row of asterisks (*) is displayed for that hop.

Examples:

To trace the path to the host named www.google.co.in type:
tracert www.google.co.in



To trace the path to the host named www.google.com and prevent the resolution of each IP address to its name, type:
tracert -d www.google.com

To trace the path to the host named www.google.com and use the loose source route 10.12.0.1-10.29.3.1-10.1.44.1, type:
tracert -j 10.12.0.1 10.29.3.1 10.1.44.1 www.google.com

Syntax

```
tracert [-d] [-h MaximumHops] [-j HostList] [-w Timeout] [TargetName]
```

Parameters

-d Prevents tracert from attempting to resolve the IP addresses of intermediate routers to their names. This can speed up the display of tracert results.

-h MaximumHops Specifies the maximum number of hops in the path to search for the target (destination). The default is 30 hops.

-j HostList Specifies that Echo Request messages use the Loose Source Route option in the IP header with the set of intermediate destinations specified in HostList. With loose source routing, successive intermediate destinations can be separated by one or multiple routers. The maximum number of addresses or names in the host list is 9. The HostList is a series of IP addresses (in dotted decimal notation) separated by spaces.

-w Timeout Specifies the amount of time in milliseconds to wait for the ICMP Time Exceeded or Echo Reply message corresponding to a given Echo Request message to be received. If not received within the time-out, an asterisk (*) is displayed. The default time-out is 4000 (4 seconds).

Arp

Displays and modifies entries in the Address Resolution Protocol (ARP) cache, which contains one or more tables that are used to store IP addresses and their resolved Ethernet or Token Ring physical addresses. There is a separate table for each Ethernet or Token Ring network adapter installed on your computer.

Syntax

```
arp [-a [InetAddr] [-N IfaceAddr]] [-g [InetAddr] [-N IfaceAddr]] [-d InetAddr [IfaceAddr]] [-s InetAddr EtherAddr [IfaceAddr]]
```

Parameters

Used without parameters, ping displays help

-a [InetAddr] [-N IfaceAddr] Displays current ARP cache tables for all interfaces. To display the ARP cache entry for a specific IP address, use arp -a with the InetAddr parameter, where InetAddr is an IP address. To display the ARP cache table for a specific interface, use the -N IfaceAddr parameter where IfaceAddr is the IP address assigned to the interface. The -N parameter is case-sensitive.

-g [InetAddr] [-N IfaceAddr] Identical to -a.

-d InetAddr [IfaceAddr] Deletes an entry with a specific IP address, where InetAddr is the IP address. To delete an entry in a table for a specific interface, use the IfaceAddr parameter where IfaceAddr is the IP address assigned to the interface. To delete all entries, use the asterisk (*) wildcard character in place of InetAddr.

-s InetAddr EtherAddr [IfaceAddr] Adds a static entry to the ARP cache that resolves the IP address InetAddr to the physical address EtherAddr. To add a static ARP cache entry to the table for a specific interface, use the IfaceAddr parameter where IfaceAddr is an IP address assigned to the interface.

Examples:

To display the ARP cache tables for all interfaces, type:
arp -a

To display the ARP cache table for the interface that is assigned the IP address 10.0.0.99, type:

 The image cannot be displayed. Your computer may not have enough memory to open the image, or the image may have been corrupted. Restart your computer, and then open the file again. If the red x still appears, you may have to delete the image and then insert it again.

arp -a -N 10.0.0.99

To add a static ARP cache entry that resolves the IP address 10.0.0.80 to the physical address 00-AA-00-4F-2A-9C,
type:
arp -s 10.0.0.80 00-AA-00-4F-2A-9C

 The image cannot be displayed. Your computer may not have enough memory to open the image, or the image may have been corrupted. Restart your computer, and then open the file again. If the red x still appears, you may have to delete the image and then insert it again.

nslookup

Nslookup (Name Server lookup) is a UNIX shell command to query Internet domain name servers.



Definitions

- **Nameserver:** These are the servers that the internet uses to find out more about the domain. Usually they are an ISP's computer.
- **Mailserver:** Where email is sent to.
- **Webserver:** The domains website.
- **FTPserver:** FTP is file transfer protocol, this server is where files may be stored.
- **Hostname:** The name of the host as given by the domain.
- **Real Hostname:** This is hostname that you get by reverse resolving the IP address, may be different to the given hostname.
- **IP Address:** Unique four numbered identifier that is obtained by resolving the hostname.

finger

finger looks up and displays information about system users.

finger syntax

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

Options

-s	<p>Displays the user's login name, real name, terminal name and write status (as a "*" after the terminal name if write permission is denied), idle time, login time, office location and office phone number.</p> <p>Login time is displayed as month, day, hours and minutes, unless more than six months ago, in which case the year is displayed rather than the hours and minutes.</p> <p>Unknown devices as well as nonexistent idle and login times are displayed as single asterisks.</p>
-l	<p>Produces a multi-line format displaying all of the information described for the -s option as well as the user's home directory, home phone number, login shell, mail status, and the contents of the files ".plan", ".project", ".pgpkey" and ".forward" from the user's home directory.</p> <p>Phone numbers specified as eleven digits are printed as "+N-NNN-NNN-NNNN". Numbers specified as ten or seven digits are printed as the appropriate subset of that string. Numbers specified as five digits are printed as "xN-NNNN". Numbers specified as four digits are printed as "xNNNN".</p> <p>If write permission is denied to the device, the phrase "(messages off)" is appended to the line containing the device name. One entry per user is displayed with the -l option; if a user is logged on multiple times, terminal information is repeated once per login.</p> <p>Mail status is shown as "No Mail." if there is no mail at all, "Mail last read DDD MMMM ## HH:MM YYYY (TZ)" if the person has looked at their mailbox since new mail arriving, or "New mail received ...", "Unread since ..." if they have new mail.</p>

-p	Prevents the -l option of finger from displaying the contents of the ".plan", ".project" and ".pgpkey" files.
-m	Prevent matching of usernames. The <i>user</i> is usually a login name; however, matching will also be done on the users' real names, unless the -m option is supplied. All name matching performed by finger is case insensitive.

If no options are specified, **finger** defaults to the **-l** style output if operands are provided, otherwise to the **-s** style. Note that some fields may be missing, in either format, if information is not available for them.

If no arguments are specified, **finger** will print an entry for each user currently logged into the system.

Finger may be used to look up users on a remote machine. The format is to specify a user as "**user@host**", or "@**host**", where the default output format for the former is the **-l** style, and the default output format for the latter is the **-s** style. The **-l** option is the only option that may be passed to a remote machine.

If standard output is a socket, **finger** will emit a carriage return (^M) before every linefeed (^J). This format is for processing remote finger requests when invoked by **fingerd**, the finger daemon.

Files

~/.nofinger	If finger finds this file in a user's home directory, it will, for finger requests originating outside the local host, firmly deny the existence of that user. For this to work, the finger program, as started by fingerd , must be able to see the .nofinger file. This generally means that the home directory containing the file must have the other-users-execute bit set (o+x). (See chmod). If you use this feature for privacy, please test it with " finger @localhost " before relying on it, just in case.
~/.plan	These files are printed as part of a long-format request. The .plan file may be of any length.
~/.project	
~/.pgpkey	

finger Examples

```
finger -p ch
```

Display information about the user **ch**. Output will appear similar to the following:

```
Login name: admin  
In real life: Computer Hope  
On since Feb 11 23:37:16 on pts/7 from domain.computerhope.com  
28 seconds Idle Time  
Unread mail since Mon Feb 12 00:22:52 2001
```

Nmap / Port Scan

The **Nmap** aka Network Mapper is an open source and a very versatile tool for **Linux** system/network administrators. **Nmap** is used for exploring networks, perform security scans, network audit and finding open ports on remote machine

Example

```
$ nmap 207.218.248.50
```

Sample outputs:

Output

```
Starting Nmap 5.00 ( http://nmap.org ) at 2012-11-18 14:41 IST
```

Interesting ports on 207.218.248.50:

Not shown: 997 closed ports

PORt	STATE	SERVICE
------	-------	---------

```
23/tcp open telnet
```

```
53/tcp open domain
```

```
80/tcp open http
```

```
MAC Address: 55:87:06:25:65:FC (Unknown)
```

```
Nmap done: 1 IP address (1 host up) scanned in 1.28 seconds
```

Other Examples

3) Scan an IP

```
nmap 207.218.248.50
```

4) Scan a range of IP address

```
nmap 207.218.248.5-45
```

5) Scan entire subnet

```
nmap 192.168.2.0/24
```

6) Ping only scan

```
nmap -sP 207.218.248.50
```

7) Scan and do traceroute

```
nmap -traceroute IP-ADDRESS
```

```
nmap -traceroute DOMAIN-NAME-HERE
```

8) TCP SYN Scan

```
nmap -sS 207.218.248.50
```

9) UDP Scan

```
nmap -sU 207.218.248.50
```

10) IP protocol scan

```
nmap -sO 207.218.248.50
```

11) Scan port 80, 25, 443

```
nmap -p 80 207.218.248.50
```

```
nmap -p http 207.218.248.50
```

```
nmap -p 25 207.218.248.50
```

```
nmap -p smtp 207.218.248.50
```

```
nmap -p 443 207.218.248.50
```

```
nmap -p 80,24,443 207.218.248.50
```

12) Scan port ranges

```
nmap -p 512-1024 207.218.248.50
```

13) Scan for Operating System Detection

```
nmap -O 207.218.248.50
```

```
nmap -O --osscan-guess 207.218.248.50
```

14) Scan for application server version

```
nmap -sV 207.218.248.50
```

15) Scan a host name

```
nmap google.com
```

16) Scan a host name with more info

```
nmap -v google.com
```

17) Scan a host when protected by the firewall

```
nmap -PN 207.218.248.50
```

```
nmap -PN google.com
```

18) Perform a fast scan

```
nmap -F 207.218.248.50
```

19) Show host interfaces and routes

```
nmap -iflist
```

20) Scan for IP protocol

This type of scan allows you to determine which IP protocols (TCP, ICMP, IGMP, etc.) are supported by target machines:

```
nmap -sO 207.218.248.50
```

SUBNETTING

This assignment explains Subnetting in easy language with examples. This assignment is divided in three sections. First section provides a basic overview of Subnetting tools. Second section explains Subnetting process in easy steps. Last section includes Subnetting examples for beginners.

Subnetting

Subnetting is a process of dividing large network into the smaller networks based on layer 3 IP address. Every computer on network has an IP address that represents its location on network. Two version of IP addresses are available IPv4 and IPv6. In this article we will perform Subnetting on IPv4.

IPv4

IP addresses are displayed in dotted decimal notation, and appear as four numbers separated by dots. Each number of an IP address is made from eight individual bits known as octet. Each octet can create number value from 0 to 255. An IP address would be 32 bits long in binary divided into the two components, network component and host component. Network component is used to identify the network that the packet is intended for, and host component is used to identify the individual host on network.

IP addresses are broken into the two components:

Network component :- Defines network segment of device.

Host component :- Defines the specific device on a particular network segment

IP Classes in decimal notation

Class A addresses range from 1-126
Class B addresses range from 128-191
Class C addresses range from 192-223
Class D addresses range from 224-239
Class E addresses range from 240-254

- 0 [Zero] is reserved and represents all IP addresses.
- 127 is a reserved address and is used for testing, like a loop back on an interface.
- 255 is a reserved address and is used for broadcasting purposes.

This assignment is the second part of our article “Network Addressing Explained with Subnetting and VLSM”. You can read other parts of this article here.

Basic of Network Addressing

Subnet mask

Subnet mask is a 32 bits long address used to distinguish between network address and host address in IP address. Subnet mask is always used with IP address. Subnet mask has only one purpose, to identify which part of an IP address is network address and which part is host address.

For example how will we figure out network partition and host partition from IP address 192.168.1.10 ? Here we need subnet mask to get details about network address and host address.

- In decimal notation subnet mask value 1 to 255 represent network address and value 0 [Zero] represent host address.
- In binary notation subnet mask **ON** bit [1] represent network address while **OFF** bit[0] represent host address.

In decimal notation

IP address	192.168.1.10
Subnet mask	255.255.255.0

Network address is **192.168.1** and host address is **10**.

In binary notation

IP address	11000000.10101000.00000001.00001010
Subnet mask	11111111.11111111.11111111.00000000

Network address is 11000000.10101000.00000001 and host address is 00001010



IP Class	Default Subnet	Network bits	Host bits	Total hosts	Valid hosts
A	255.0.0.0	First 8 bits	Last 24 bits	16, 777, 216	16, 777, 214
B	255.255.0.0	First 16 bits	Last 16 bits	65,536	65,534
C	255.255.255.0	First 24 bits	Last 8 bits	256	254

Network ID

First address of subnet is called network ID. This address is used to identify one segment or broadcast domain from all the other segments in the network.

Block Size

Block size is the size of subnet including network address, hosts addresses and broadcast address.

Broadcast ID

There are two types of broadcast, direct broadcast and full broadcast.

Direct broadcast or local broadcast is the last address of subnet and can be heard by all hosts in subnet.

Full broadcast is the last address of IP classes and can be heard by all IP hosts in network. Full broadcast address is 255.255.255.255

The main difference between direct broadcast and full broadcast is that routers will not propagate local broadcasts between segments, but they will propagate directed broadcasts.

Host Addresses

All address between the network address and the directed broadcast address is called host address for the subnet. You can assign host addresses to any IP devices such as PCs, servers, routers, and switches.

Subnetting

Subnetting is a process of breaking large network in small networks known as subnets. Subnetting happens when we extend default boundary of subnet mask. Basically we borrow host bits to create networks. Let's take a example

Being a network administrator you are asked to create two networks, each will host 30 systems.

Single class C IP range can fulfill this requirement, still you have to purchase 2 class C IP range, one for each network. Single class C range provides 256 total addresses and we need only 30 addresses, this will waste 226 addresses. These unused addresses would make additional route advertisements slowing down the network.

With subnetting you only need to purchase single range of class C. You can configure router to take first 26 bits instead of default 24 bits as network bits. In this case we would extend default boundary of subnet mask and borrow 2 host bits to create networks. By taking two bits from the host range and counting them as network bits, we can create two new subnets, and assign hosts them. As long as the two new network bits match in the address, they belong to the same network. You can change either of the two bits, and you would be in a new subnet.

Advantage of Subnetting

- Subnetting breaks large network in smaller networks and smaller networks are easier to manage.
- Subnetting reduces network traffic by removing collision and broadcast traffic, that overall improve performance.
- Subnetting allows you to apply network security policies at the interconnection between subnets.
- Subnetting allows you to save money by reducing requirement for IP range.

Subnetting math

Subnetting process involves binary math calculation. Computers communicate with each other's in binary language. To succeed in any kind of networking career, you might be fluent in binary math calculation. Subnetting needs two type of calculation, convert decimal to binary and convert binary to decimal.

Binary system works exactly same as decimal system, except the base number. Base number is 2 in binary system and 10 in decimal system. To calculate decimal equivalent value of a binary number, you have to replace base value 10 with 2. Binary numbers are displayed in columns and each position in binary system has double value than the position in right. From earlier section of this article you know that each number of an IP address is made from eight individual bits known as octet. So you should remember at least eight decimal equivalent value from binary position.

Base position	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Decimal value	128	64	32	16	8	4	2	1

Convert decimal to binary

To convert a decimal number in binary we would use addition till number method. In this method we start adding from left to get target value. If after adding right position value, sum is lower than target number, keep adding, or if sum is greater than target number skip the position value. Only the value of on bit [1] will be added in sum. Off bit [0] has zero value. For example, convert decimal number 117 in binary.

Target decimal number 117

Move direction From Left =====> to Right

Base position	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Decimal value	128	64	32	16	8	4	2	1
Bit status	0	1	1	1	0	1	0	1
Decimal value in addition	0	64	32	16	0	4	0	1

Binary value of 117 is 01110101.

Decimal calculation	Bit in binary
128 is greater than 117	off the bit
$0+64 = 64$ is less than 117	on the bit
$0+64+32 = 96$ is less than 117	on the bit
$0+64+32+16 = 112$ is less than 117	on the bit
$0+64+32+16+8 = 120$ is greater than 117	off the bit
$0+64+32+16+0+4 = 116$ is less than 117	on the bit
$0+64+32+16+0+4+2 = 118$ is greater than 117	off the bit
$0+64+32+16+0+4+0+1 = 117$ is equivalent to 117	on the bit

Convert binary in decimal

To convert a binary in decimal we will follow above method in reverse mode. We will find the decimal value of on binary bit position and add them. For example convert 10101010 binary number in decimal.

Target binary number 10101010

Move direction From Left =====> to Right

Base position	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Decimal value	128	64	32	16	8	4	2	1
Bit status	1	0	1	0	1	0	1	0
Decimal value in addition	128	0	32	0	8	0	2	0

Decimal value of 10101010 is 170 [$128+0+32+0+8+0+2+0$]

Binary bit	Decimal value
1 On bit	128
0 Off bit	0
1 On bit	64
0 Off bit	0
1 On bit	32
0 Off bit	0
1 On bit	8
0 Off bit	0
1 On bit	2
0 Off bit	0

Review above method and do as much practice of following as you can.

- Pick any number from 0 - 255 and convert it in binary.
- Pick any combination from 00000000 - 11111111 and convert it in decimal.

Better you are with this conversion, the better you will do on the subnetting.

Combination provided by binary position

Now you know the process of converting binary in decimal and decimal in binary. Our next step is to figure out the combination provided by a binary position.

First bit provide two combination 0 or 1. If we take two bits than we have four combinations 00,01,10,11. IP address has 32 bits, so you should be able to find the number of combinations provided by each bit position until position number 32.

Number of bit[s]	Number of combinations	Number of bits	Number of combinations
1	2	17	131072
2	4	18	262144
3	8	19	524288
4	16	20	1048576
5	32	21	2097152
6	64	22	4194304
7	128	23	8388608
8	256	24	16777216
9	512	25	33554432
10	1024	26	67108864
11	2048	27	134217728
12	4096	28	268435456
13	8192	29	536870912
14	16384	30	1073741824
15	32768	31	2147483648
16	65536	32	4294967296

Default subnet mask

Class	Subnet Mask	Format
A	255.0.0.0	Network.Host.Host.Host
B	255.255.0.0	Network.Network.Host.Host
C	255.255.255.0	Network.Network.Network.Host

Key terms to remember

- A subnet is a smaller portion of large network treated as its own separate network. To create subnet we borrow bits from host portion and assign them as network bits. This means more networks, fewer hosts.
- If the network bits on two addresses do not match, then the two packets are intended for two separate networks.
- On a 32 bits IP address at least eight bits must belong to the network portion and at least 2 bits must belong to the host portion.
- Each IP address has a predefined IP class and that cannot be changed.
- Each class has a predefined default subnet mask that tells us the octets, which are already part of the network portion, as well as how many bits we have available to work with.
- Whatever network class it is, we cannot change those bits that are already assigned.
- We cannot assign the network ID and the broadcast address to a host.
- Regardless how many bits are left in the host field, network ID and the broadcast address must be reserved.
- Subnet bits start at the left and go to the right, without skipping bits.

CIDR [Classless Inter Domain Routing]

CIDR is a slash notation of subnet mask. CIDR tells us number of one bits in a network address.

- Class A has default subnet mask 255.0.0.0. that means first octet of the subnet mask has all one bits. In slash notation it would be written as /8, means address has 8 bits one.
- Class B has default subnet mask 255.255.0.0. that means first two octets of the subnet mask have all one bits. In slash notation it would be written as /16, means address has 16 bits one.
- Class C has default subnet mask 255.255.255.0. that means first three octets of the subnet mask have all one bits. In slash notation it would be written as /24, means address has 24 bits one.

Method of subnetting

In subnetting we find the answer of following questions.

- What is subnet mask for given address?
- How many subnets does given subnet mask provide ?
- What is block size for given subnet mask?
- What are the valid subnets?

- What are the total hosts?
- How many valid hosts are available per subnet?
- What is broadcast address of each subnet?
- What is network address of each subnet?

To answer above questions we use following method of subnetting.

What is subnet mask for given address?

Subnetting takes place when we extend the default subnet mask. We cannot perform subnetting with default subnet mask and every class has default subnet mask. To figure out subnetted subnet mask, we first need to write down the default subnet mask. Now find the host bits borrowed to create subnets and convert them in decimal. For example find the subnet mask of address 188.25.45.48/20? This address belongs to class B and class B has default subnet mask 255.255.0.0 [/16 in CIDR]. We borrowed 4 bits from hosts portion. As you know subnetting moves from left to right and it cannot skip any network bit. So this subnet mask in binary would be 11111111.11111111.11110000.00000000. First two octets have default value so its decimal value would be 255.255. We will convert third octet in decimal value. To convert a binary number in decimal we add its decimal equivalent value. In our example it would be $128+64+32+16+0+0+0+0 = 240$. Our fourth octet has all bits off so its decimal value would be $0+0+0+0+0+0+0+0 = 0$. Our answer subnet mask would be 255.255.240.0

How many subnets does given subnet mask provide ?

To calculate the number of subnets provided by given subnet mask we use 2^N , where N = number of bits borrowed from host bits to create subnets. For example in 192.168.1.0/27, N is 3. By looking at address we can determine that this address belongs to class C and class C has default subnet mask 255.255.255.0 [/24 in CIDR]. In given address we borrowed $27 - 24 = 3$ host bits to create subnets. Now $2^3 = 8$, so our answer is 8.

What is block size for subnet mask?

Block size or increment number is used to calculate the valid subnets. Once you figure out the block size, calculation of valid subnets become piece of cake. To figure out the block size, use this formula
 $256 - \text{Subnet mask} = \text{block size}$. For example block size for subnet mask 255.255.255.240 is $256 - 240 = 16$.

What are the valid subnets?

Calculating valid subnet is two steps process. First calculate total subnet by using formula 2^N . In second step find the block size and count from zero in block until you reach the subnet mask value. For example calculate the valid subnets for 192.168.1.0/26.

Borrowed host bits are 2 [26-24].

Total subnets are $2^2 = 4$.

Subnet mask would be 255.255.255.192.

Block size would be $256 - 192 = 64$.

Start counting from zero at blocks of 64, so our valid subnets would be 0,64,128,192.

What are the total hosts?

Total hosts are the hosts available per subnet. To calculate total hosts use formula $2^H = \text{Total hosts}$. H is the number of host bits. For example in address 192.168.1.0/26 we have $32 - 26$ [Total bits in IP address - Bits consumed by network address] = 6. Total hosts per subnet would be $2^6 = 64$.

How many valid hosts are available per subnet?

Valid hosts are the number of hosts those can be assigned to devices. As we know, we need to reduce two address per subnet, one for network ID and another for broadcast ID. So our formula, to calculate valid hosts would be Total hosts - 2 = Valid hosts. In above example we have 64 hosts per subnet, so valid hosts in each subnet would be $64 - 2 = 62$.

What is broadcast address of each subnet?

Broadcast address is the last address of subnet. This address is reserve for network broadcast, and cannot be assigned to any host. In above example

0 Subnet has broadcast address 63

64 Subnet has broadcast address 127

128 Subnet has broadcast address 191

192 Subnet has broadcast address 255

What is the network address of each subnet?

Network address is the first address of subnet. This address is used to locate the network, and cannot be assigned to any host. In above example address 0,64,128,192 are the network address.

- Network address is always the first IP address of subnet.

- Broadcast address is always the last IP address of subnet (IP address before the next subnet).
- Valid hosts are the IP addresses between network address and broadcast address.

At this point you have powered with all essential tools for subnetting. In last section of this article we will practically implement what we have learned so far. Due to length of this article I will include examples only from class C.

Class C Subnetting

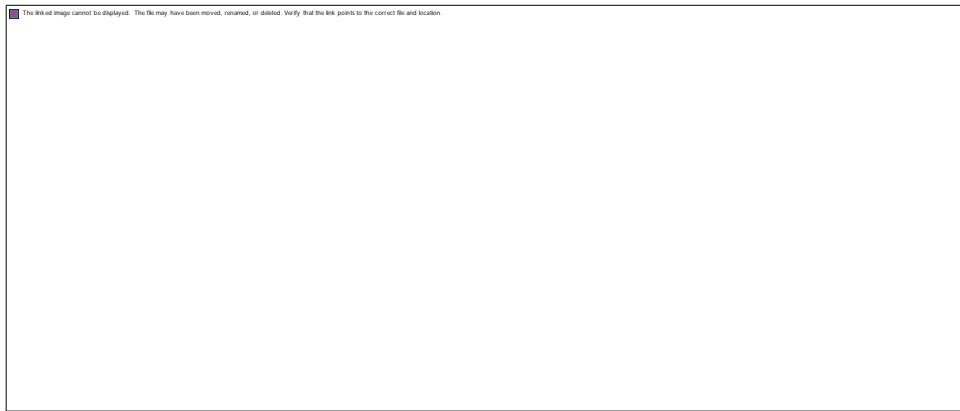
Default subnet mask of class C is 255.255.255.0. CIDR notation of class C is /24, which means 24 bits from IP address are already consumed by network portion and we have 8 host bits to work with. We cannot skip network bit, when we turn them on. Subnetting moves from left to right. So Class C subnet masks can only be the following:

CIDR	Decimal	Binary
/25	128	10000000
/26	192	11000000
/27	224	11100000
/28	240	11110000
/29	248	11111000
/30	252	11111100

As we have already discussed earlier in this article that we have to have at least 2 host bits for assigning IP addresses to hosts, that means we can't use /31 and /32 for subnetting.

/25

CIDR /25 has subnet mask 255.255.255.128 and 128 is 10000000 in binary. We used one host bit in network address.



N = 1 [Number of host bit used in network]

H = 7 [Remaining host bits]

Total subnets (2^N) :- $2^1 = 2$

Block size (256 - subnet mask) :- $256 - 128 = 128$

Valid subnets (Count blocks from 0) :- 0,128

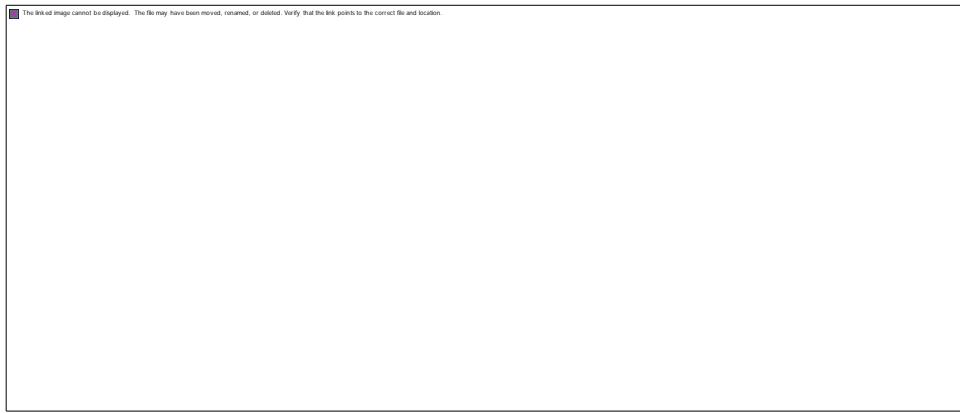
Total hosts (2^H) :- $2^7 = 128$

Valid hosts per subnet (Total host - 2) :- $128 - 2 = 126$

Subnets	Subnet 1	Subnet 2
Network ID	0	128
First host	1	129
Last host	126	254
Broadcast ID	127	255

/26

CIDR /26 has subnet mask 255.255.255.192 and 192 is 11000000 in binary. We used two host bits in network address.



$$N = 2$$

$$H = 6$$

$$\text{Total subnets } (2^N) : - 2^2 = 4$$

$$\text{Block size (256 - subnet mask)} : - 256 - 192 = 64$$

$$\text{Valid subnets (Count blocks from 0)} : - 0, 64, 128, 192$$

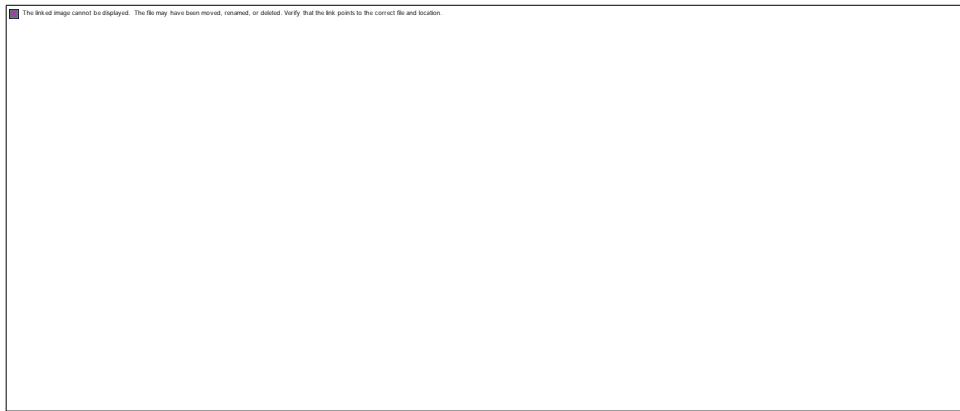
$$\text{Total hosts } (2^H) : - 2^6 = 64$$

$$\text{Valid hosts per subnet (Total host - 2)} : - 64 - 2 = 62$$

Subnets	Subnet 1	Subnet 2	Subnet 3	Subnet 4
Network ID	0	64	128	192
First host	1	65	129	193
Last host	62	126	190	254
Broadcast ID	63	127	191	255

/27

CIDR /27 has subnet mask 255.255.255.224 and 224 is 11100000 in binary. We used three host bits in network address.



$$N = 3$$

$$H = 5$$

$$\text{Total subnets } (2^N) : - 2^3 = 8$$

$$\text{Block size (256 - subnet mask)} : - 256 - 224 = 32$$

$$\text{Valid subnets (Count blocks from 0)} : - 0, 32, 64, 96, 128, 160, 192, 224$$

$$\text{Total hosts } (2^H) : - 2^5 = 32$$

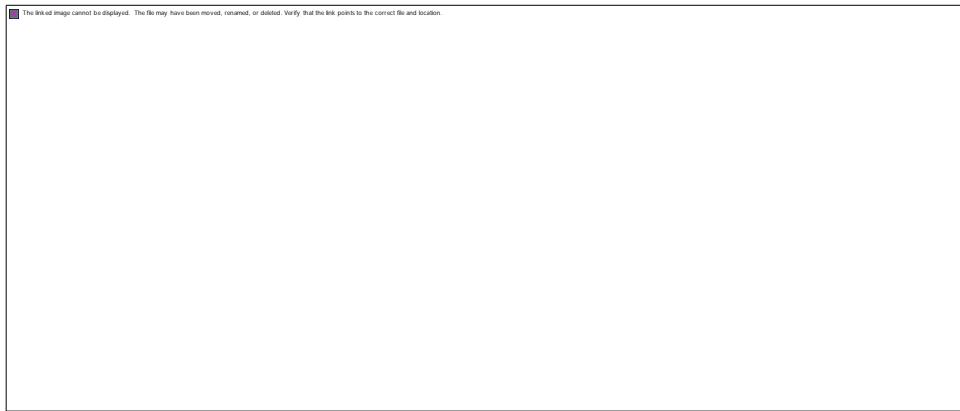
$$\text{Valid hosts per subnet (Total host - 2)} : - 32 - 2 = 30$$

Subnets	Sub 1	Sub 2	Sub 3	Sub 4	Sub 5	Sub 6	Sub 7	Sub 8
Network ID	0	32	64	96	128	160	192	224
First host	1	33	65	97	129	161	193	225
Last host	30	62	94	126	158	190	222	254
Broadcast ID	31	63	95	127	159	191	223	255

Sub = Subnet

/28

CIDR /28 has subnet mask 255.255.255.240 and 240 is 11110000 in binary. We used four host bits in network address.



$$N = 4$$

$$H = 4$$

$$\text{Total subnets } (2^N) : - 2^4 = 16$$

$$\text{Block size (256 - subnet mask)} : - 256 - 240 = 16$$

$$\text{Valid subnets (Count blocks from 0)} : - 0, 16, 32, 48, 64, 80, 96, 112, 128, 144, 160, 176, 192, 208, 224, 240$$

$$\text{Total hosts } (2^H) : - 2^4 = 16$$

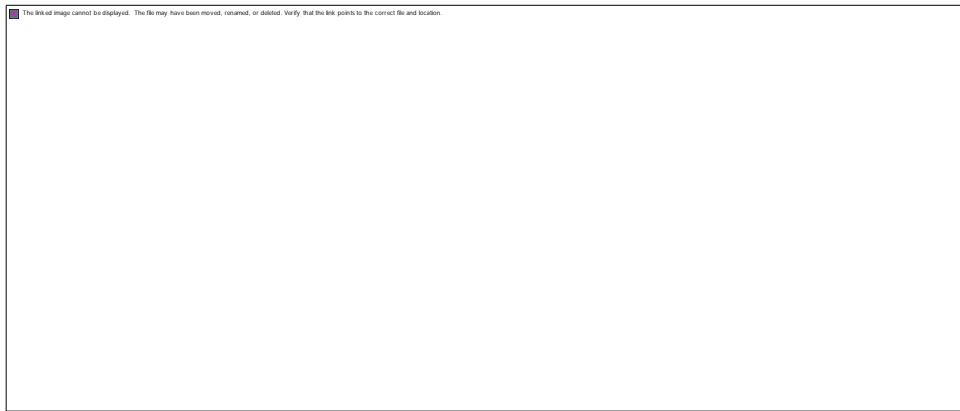
$$\text{Valid hosts per subnet (Total host - 2)} : - 16 - 2 = 14$$

I hope you have understand the pattern of making the subnet chart with above information's. Let's join me in this practice. For this example and next two examples I will fill only two subnets [first and last one], it's your turn to fill the middle subnets.

Subnets	Subnet 1	Subnet 2 To Subnet 15(Filled by you)	Subnet 16
Network ID	0	240
First host	1	241
Last host	14	254
Broadcast ID	15	255

/29

CIDR /29 has subnet mask 255.255.255.248 and 248 is 11111000 in binary. We used five host bits in network address.



$$N = 5$$

$$H = 3$$

$$\text{Total subnets } (2^N) : - 2^5 = 32$$

$$\text{Block size (256 - subnet mask)} : - 256 - 248 = 8$$

Valid subnets (Count blocks from 0) :-

0,8,16,24,32,40,48,56,64,72,80,88,96,104,112,120,128,136,144,152,160,168,176,184,192,200,208,
216,224,232,240,248

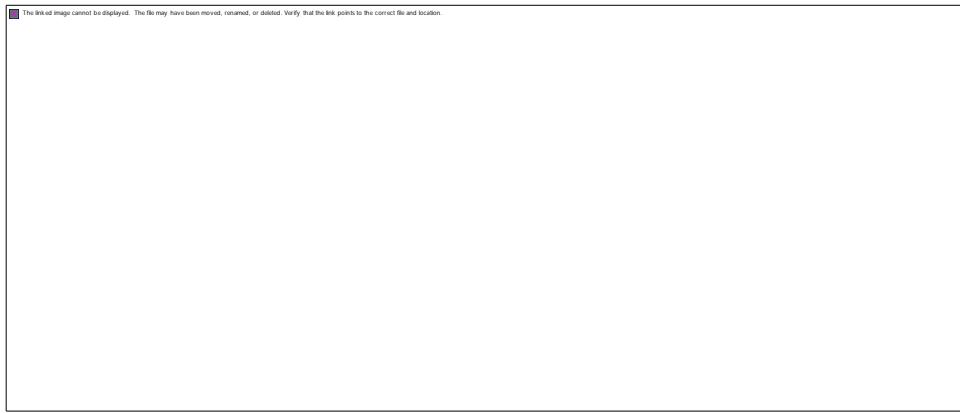
$$\text{Total hosts } (2^H) : - 2^3 = 8$$

$$\text{Valid hosts per subnet (Total host - 2) : - } 8 - 2 = 6$$

Subnets	Subnet 1	Subnet 2 To Subnet 31(Filled by you)	Subnet 32
Network ID	0	248
First host	1	249
Last host	6	254
Broadcast ID	7	255

/30

CIDR /30 has subnet mask 255.255.255.252 and 252 is 11111100 in binary. We used six host bits in network address.



$$N = 6$$

$$H = 2$$

$$\text{Total subnets } (2^N) : - 2^6 = 64$$

$$\text{Block size (256 - subnet mask)} : - 256 - 252 = 4$$

Valid subnets (Count blocks from 0) :-

0,4,8,12,16,20,24,28,32,36,40,44,48,52,56,60,64,68,72,76,80,84,88,92,96,100,104,108,112,116,120
,124,128,132,136,140,144,148,152,156,160,164,168,172,176,180,184,188,192,196,200,204,208,21
2,216,220,224,228,232,236,240,244,248,252

$$\text{Total hosts } (2^H) : - 2^2 = 4$$

$$\text{Valid hosts per subnet (Total host - 2)} : - 4 - 2 = 2$$

Subnets	Subnet 1	Subnet 2 To Subnet 63 (Filled by you)	Subnet 64
Network ID	0	252
First host	1	253
Last host	2	254
Broadcast ID	3	255

At the end of this long article I have a small word for you, that is practice.

Access control list

ACLs are basically a set of commands, grouped together by a number or name that is used to filter traffic entering or leaving an interface.

When activating an ACL on an interface, you must specify in which direction the traffic should be filtered:

- **Inbound (as the traffic comes into an interface)**
- **Outbound (before the traffic exits an interface)**

Inbound ACLs: Incoming packets are processed before they are routed to an outbound interface. An inbound ACL is efficient because it saves the overhead of routing lookups if the packet will be discarded after it is denied by the filtering tests. If the packet is permitted by the tests, it is processed for routing.

Outbound ACLs: Incoming packets are routed to the outbound interface and then processed through the outbound ACL.

Universal fact about Access control list

1. ACLs come in two varieties : **Numbered and named**
2. Each of these references to ACLs supports two types of filtering: **standard and extended.**
3. Standard IP ACLs can filter only on the **source IP address** inside a packet.
4. Whereas an extended IP ACLs can filter on the **source and destination IP addresses** in the packet.
5. There are two actions an ACL can take: **permit or deny.**
6. Statements are processed top-down.
7. Once a match is found, no further statements are processed—therefore, order is important.
8. If no match is found, the imaginary **implicit deny statement at the end of the ACL** drops the packet.
9. An ACL should have at least one permit statement; otherwise, all traffic will be dropped because of the hidden implicit deny statement at the end of every ACL.

No matter what type of ACL you use, though, you can have only one ACL per protocol, per interface, per direction. For example, you can have one IP ACL inbound on an interface and another IP ACL outbound on an interface, but you cannot have two inbound IP ACLs on the same interface.

Access List Ranges

Type	Range
IP Standard	1–99
IP Extended	100–199
IP Standard Expanded Range	1300–1999
IP Extended Expanded Range	2000–2699

Standard ACLs

A standard IP ACL is simple; it filters based on source address only. You can filter a source network or a source host, but you cannot filter based on the destination of a packet, the particular protocol being used such as the Transmission Control Protocol (TCP) or the User Datagram Protocol (UDP), or on the port number. You can permit or deny only source traffic.

Extended ACLs:

An extended ACL gives you much more power than just a standard ACL. Extended IP ACLs check both the source and destination packet addresses. They can also check for specific protocols, port numbers, and other parameters, which allow administrators more flexibility and control.

Named ACLs

One of the disadvantages of using IP standard and IP extended ACLs is that you reference them by number, which is not too descriptive of its use. With a named ACL, this is not the case because you can name your ACL with a descriptive name. The ACL named Deny Mike is a lot more meaningful than an ACL simply numbered 1. There are both IP standard and IP extended named ACLs.

Another advantage to named ACLs is that they allow you to remove individual lines out of an ACL. With numbered ACLs, you cannot delete individual statements. Instead, you will need to delete your existing access list and re-create the entire list.

Configuration Guidelines

- Order of statements is important: put the most restrictive statements at the top of the list and the least restrictive at the bottom.
- ACL statements are **processed top-down until a match is found**, and then no more statements in the list are processed.
- If no match is found in the ACL, the packet is dropped (implicit deny).
- Each ACL needs either a unique number or a unique name.
- The router cannot filter traffic that it, itself, originates.
- You can have only one IP ACL applied to an interface in each direction (inbound and outbound)—you can't have two or more inbound or outbound ACLs applied to the same interface. (Actually, you can have one ACL for each protocol, like IP and IPX, applied to an interface in each direction.)
- Applying an empty ACL to an interface permits all traffic by default: in order for an ACL to have an implicit deny statement, you need at least one actual permit or deny statement.
- Remember the numbers you can use for IP ACLs. Standard ACLs can use numbers ranging **1–99 and 1300–1999**, and extended ACLs can use **100–199 and 2000–2699**.
- Wildcard mask is not a subnet mask. Like an IP address or a subnet mask, a wildcard mask is composed of 32 bits when doing the conversion; subtract each byte in the subnet mask from 255.

There are two special types of wildcard masks:

0.0.0.0 and 255.255.255.255

A 0.0.0.0 wildcard mask is called a host mask

255.255.255.255. If you enter this, the router will cover the address and mask to the keyword any.

Placement of ACLs

Standard ACLs should be placed as close to the destination devices as possible.

Extended ACLs should be placed as close to the source devices as possible.

Standard access lists

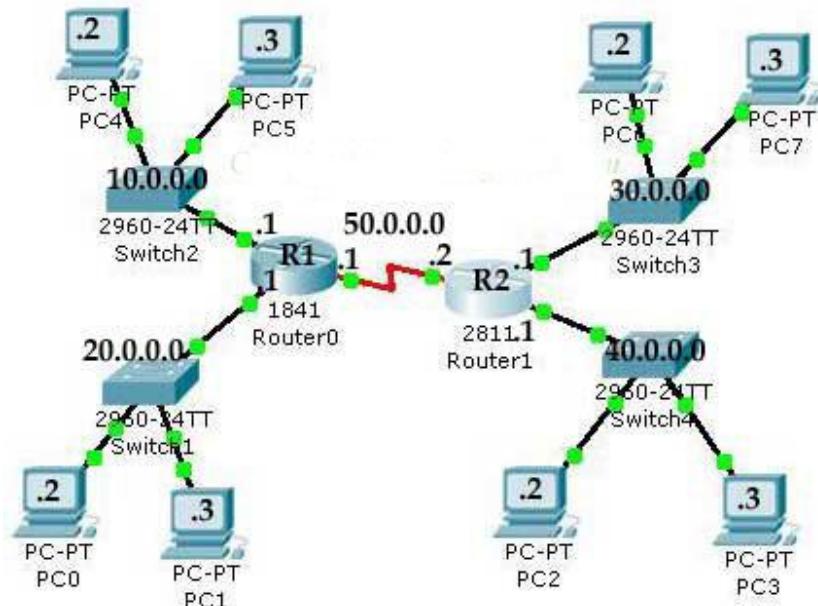
Because a standard access list filters only traffic based on source traffic, all you need is the IP address of the host or subnet you want to permit or deny. ACLs are created in global configuration mode and then applied on an interface. The syntax for creating a standard ACL is

```
access-list {1-99 | 1300-1999} {permit | deny} source-address  
[wildcard mask]
```

In this article we will configure standard access list. If you want read the feature and characteristic of access list reads this previous article.

Access control list

In this article we will use a RIP running topology. Which we created in RIP routing practical.



Three basic steps to configure Standard Access List

- Use the access-list global configuration command to create an entry in a standard ACL.
- Use the interface configuration command to select an interface to which to apply the ACL.
- Use the ip access-group interface configuration command to activate the existing ACL on an interface.

With Access Lists you will have a variety of uses for the wild card masks, but typically For CCNA exam prospective you should be able to do following:

1. Match a specific host,
2. Match an entire subnet,
3. Match an IP range, or
4. Match Everyone and anyone

Match specific hosts

Task

You have given a task to block 10.0.0.3 from gaining access on 40.0.0.0. While 10.0.0.3 must be able to communicate with networks. Other computer from the network of 10.0.0.0 must be able to connect with the network of 40.0.0.0.

Decide where to apply ACL and in which directions.

Our host must be able to communicate with other host except 40.0.0.0 so we will place this access list on FastEthernet 0/1 of R2 (2811) connected to the network of 40.0.0.0. Direction will be outside as packet will be filter while its leaving the interface. If you place this list on R1(1841) then host 10.0.0.3 will not be able to communicate with any other hosts including 40.0.0.0.

To configure R2 double click on it and select CLI (Choose only one method result will be same)

```
R2>enable  
R2#configure terminal  
Enter configuration commands, one per line. End with CNTL/Z.  
R2(config)#access-list 1 deny host 10.0.0.3  
R2(config)#access-list 1 permit any  
R2(config)#interface fastEthernet 0/1  
R2(config-if)#ip access-group 1 out
```

OR

```
R2>enable  
R2#configure terminal  
Enter configuration commands, one per line. End with CNTL/Z.  
R2(config)#access-list 1 deny 10.0.0.3 0.0.0.0  
R2(config)#access-list 1 permit any  
R2(config)#interface fastEthernet 0/1  
R2(config-if)#ip access-group 1 out
```

To test first do ping from 10.0.0.3 to 40.0.0.3 it should be request time out as this packet will filter by ACL. Then ping 30.0.0.3 it should be successfully replay.

```
PC>ping 40.0.0.3
```

*Pinging 40.0.0.3 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.*

*Ping statistics for 40.0.0.3:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),*

```
PC>ping 30.0.0.3
```

Pinging 30.0.0.3 with 32 bytes of data:

Request timed out.

Reply from 30.0.0.3: bytes=32 time=140ms TTL=126

Reply from 30.0.0.3: bytes=32 time=156ms TTL=126

Reply from 30.0.0.3: bytes=32 time=112ms TTL=126

Ping statistics for 30.0.0.3:

Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),

Approximate round trip times in milli-seconds:

Minimum = 112ms, Maximum = 156ms, Average = 136ms

As we applied access list only on specific host so other computer from the network of 10.0.0.0 must be able to connect with the network of 40.0.0.0. To test do ping from 10.0.0.2 to 40.0.0.3

PC>ipconfig

IP Address.....: 10.0.0.2

Subnet Mask.....: 255.0.0.0

Default Gateway.....: 10.0.0.1

PC>ping 40.0.0.3

Pinging 40.0.0.3 with 32 bytes of data:

Request timed out.

Reply from 40.0.0.3: bytes=32 time=141ms TTL=126

Reply from 40.0.0.3: bytes=32 time=140ms TTL=126

Reply from 40.0.0.3: bytes=32 time=125ms TTL=126

Ping statistics for 40.0.0.3:

Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),

Approximate round trip times in milli-seconds:

Minimum = 125ms, Maximum = 141ms, Average = 135ms

Match an entire subnet

Task

You have given a task to the network of 10.0.0.0 from gaining access on 40.0.0.0. While 10.0.0.0 must be able to communicate with networks .

Wildcards

Wildcards are used with access lists to specify an individual host, a network, or a certain range of a network or networks.

Formula to calculate wild card mask for access list

The key to matching an entire subnet is to use the following formula for the wildcard mask. It goes as follows:

Wildcard mask = 255.255.255.255 - subnet

So for example if my current subnet was 255.0.0.0, the mask would be 0.255.255.255.

```
255.255.255.255
255.0.0.0 -
-----
0.255.255.255
-----
```

Once you have calculated the wild card mask rest is same as we did in previous example

R2>enable

Enter configuration commands, one per line. End with CNTL/Z.

```
R2(config)#access-list 2 deny 10.0.0.0 0.255.255.255
R2(config)#access-list 2 permit any
R2(config)#interface fastethernet 0/1
R2(config-if)#ip access-group 2 out
R2(config-if)#
```

To test first do ping from 10.0.0.3 to 40.0.0.3 it should be request time out as this packet will filter by ACL. Then ping 30.0.0.3 it should be successfully replay.

Now do ping from 10.0.0.2 to 40.0.0.3 and further 30.0.0.2 result should be same as the packet is filtering on **network based**

Match an IP range

You are a network administrator at ComputerNetworkingNotes.com. Your task is to block an ip range of 10.3.16.0 – 10.3.31.255 from gaining access to the network of 40.0.0.0

Solutions

Our range is 10.3.16.0 – 10.3.31.255. In order to find the mask, take the higher IP and subtract from it the lower IP.

```
10.3.31.255
10.3.16.0 -
-----
0.0.15.255
-----
```

In this case the wildcard mask for this range is 0.0.15.255.
To permit access to this range, you would use the following:

R2>enable

Enter configuration commands, one per line. End with CNTL/Z.

```
R2(config)#access-list 2 deny 10.3.16.0 0.0.15.255
R2(config)#access-list 2 permit any
```

```
R2(config)#interface fastethernet 0/1  
R2(config-if)#ip access-group 2 out  
R2(config-if)#
```

One thing to note is that each non-zero value in the mask must be one less than a power of 2, i.e. 0, 1, 3, 7, 15, 31, 63, 127, 255.

Match Everyone and Anyone

This is the easiest of Access-Lists to create, just use the following:
access-list 1 permit any
or
access-list 1 permit 0.0.0.0 255.255.255.255

Secure telnet session via standard ACL

This is among the highly tested topic in CCNA exam. We could use extended ACL to secure telnet session but if you did that, you'd have to apply it inbound on every interface, and that really wouldn't scale well to a large router with dozens, even hundreds, of interfaces. Here's a much better solution:

Use a standard IP access list to control access to the VTY lines themselves.

To perform this function, follow these steps:

1. Create a standard IP access list that permits only the host or hosts you want to be able to telnet into the routers.
2. Apply the access list to the VTY line with the **access-class** command

Secure R2 in a way that only 20.0.0.2 can telnet it beside it all other telnet session should be denied

```
R2>enable  
R2#configure terminal  
Enter configuration commands, one per line. End with CNTL/Z.  
R2(config)#access-list 3 permit host 20.0.0.2  
R2(config)#line vty 0 4  
R2(config-line)#password vinita  
R2(config-line)#login  
R2(config-line)#access-class 3 in
```

To test do telnet from 20.0.0.2 first is should be successful.

PC>ipconfig

IP Address.....: 20.0.0.2
Subnet Mask.....: 255.0.0.0

Default Gateway.....: 20.0.0.1

PC>telnet 50.0.0.2
Trying 50.0.0.2 ...

User Access Verification

Password:
R2>

Now telnet it from any other pc apart from 20.0.0.2. it must be filter and denied

PC>ipconfig

IP Address.....: 20.0.0.3
Subnet Mask.....: 255.0.0.0
Default Gateway.....: 20.0.0.1

PC>telnet 50.0.0.2
Trying 50.0.0.2 ...

% Connection refused by remote host
PC>

Configure Extended Access Lists

An extended ACL gives you much more power than just a standard ACL. Extended IP ACLs check both the source and destination packet addresses. They can also check for specific protocols, port numbers, and other parameters, which allow administrators more flexibility and control.

```
access-list access-list-number {permit | deny}
protocol source source-wildcard [operator port]
destination destination-wildcard [operator port]
[established] [log]
```

Command Parameters	Descriptions
access-list	Main command
access-list-number	Identifies the list using a number in the ranges of 100–199 or 2000–2699.
permit / deny	Indicates whether this entry allows or blocks the specified address.
protocol	IP, TCP, UDP, ICMP, GRE, or IGRP.
source and destination	Identifies source and destination IP addresses.
source-wildcard and destination-wildcard	The operator can be lt (less than), gt (greater than), eq (equal to), or neq (not equal to). The port number referenced can be either the source port or the destination port, depending on where in the ACL the port number is configured. As an alternative to the port number, well-known application names can be used, such as Telnet, FTP, and SMTP.
established	For inbound TCP only. Allows TCP traffic to pass if the packet is a response to an outbound-initiated session. This type of traffic has the acknowledgement (ACK) bits set. (See the Extended ACL with the Established Parameter example.)
log	Sends a logging message to the console.

Before we configure Extended Access list you should cram up some important port number

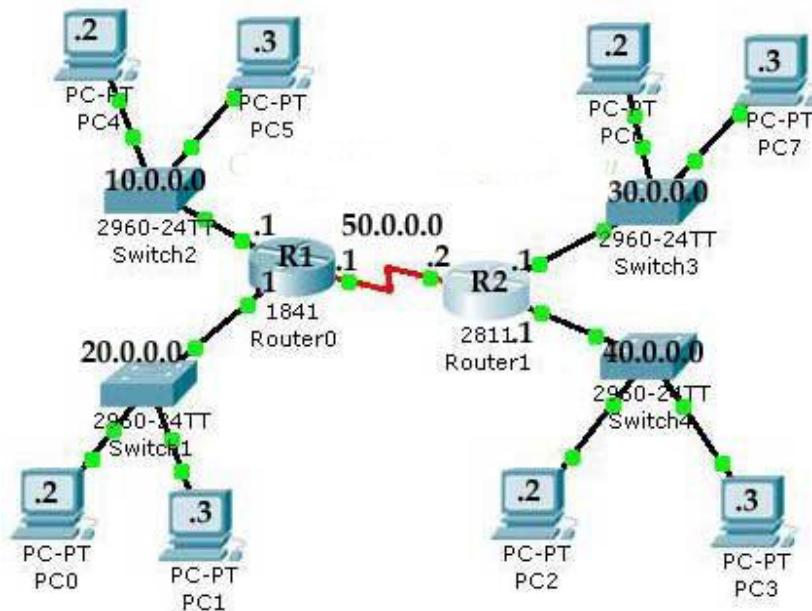
Well-Known Port Numbers and IP Protocols

Port Number	IP Protocol
20 (TCP)	FTP data
21 (TCP)	FTP control
23 (TCP)	Telnet
25 (TCP)	Simple Mail Transfer Protocol (SMTP)
53 (TCP/UDP)	Domain Name System (DNS)
69 (UDP)	TFTP
80 (TCP)	HTTP

In this article we will configure Extended access list. If you want to read the feature and characteristic of access list reads this previous article.

Access control list

In this article we will use a RIP running topology. Which we created in RIP routing practical.



Three basic steps to configure Extended Access List

- Use the access-list global configuration command to create an entry in a Extended ACL.
- Use the interface configuration command to select an interface to which to apply the ACL.
- Use the ip access-group interface configuration command to activate the existing ACL on an interface.

With Access Lists you will have a variety of uses for the wild card masks, but typically For CCNA exam prospective you should be able to do following:

1. ***Block host to host***
2. ***Block host to network***
3. ***Block Network to network***
4. ***Block telnet access for critical resources of company***
5. ***Limited ftp access for user***
6. ***Stop exploring of private network form ping***
7. ***Limited web access***
8. ***Configure established keyword***

Block host to host

Task

You are the network administrator at **ComputerNetworkingNotes.com**. Your company hire a new employee and give him a pc 10.0.0.3. your company's critical record remain in 40.0.0.3. so you are asked to block the access of 40.0.0.3 from 10.0.0.3. while 10.0.0.3 must be able connect with other computers of network to perform his task.

Decide where to apply ACL and in which directions.

As we are configuring Extended access list. With extended access list we can filter the packets as soon as it generates. So we will place our access list on F0/0 of Router1841 the nearest port of 10.0.0.3

To configure Router1841 (Hostname R1) double click on it and select CLI

```
R1>enable  
R1#configure terminal  
Enter configuration commands, one per line. End with CNTL/Z.  
R1(config)#access-list 101 deny ip host 10.0.0.3 40.0.0.3 0.0.0.0  
R1(config)#access-list 101 permit ip any any  
R1(config)#interface fastEthernet 0/0  
R1(config-if)#ip access-group 101 in  
R1(config-if)#exit  
R1(config)#
```

Verify by doing ping from 10.0.0.3 to 40.0.0.3. It should be request time out. Also ping other computers of network including 40.0.0.2. ping should be successfully.

Block host to network

Task

Now we will block the 10.0.0.3 from gaining access on the network 40.0.0.0. (if you are doing this practical after configuring previous example don't forget to remove the last access list 101. With no access-list command. Or just close the packet tracer without saving and reopen it to be continue with this example.)

```
R1(config)#access-list 102 deny ip host 10.0.0.3 40.0.0.0 0.255.255.255  
R1(config)#access-list 102 permit ip any any  
R1(config)#interface fastEthernet 0/0  
R1(config-if)#ip access-group 102 in  
R1(config-if)#exit  
R1(config)#
```

Verify by doing ping from 10.0.0.3 to 40.0.0.3. and 40.0.0.2. It should be request time out. Also ping computers of other network. ping should be successfully.

Once you have calculated the wild card mask rest is same as we did in previous example

```
R2>enable  
Enter configuration commands, one per line. End with CNTL/Z.  
R2(config)#access-list 2 deny 10.0.0.0 0.255.255.255  
R2(config)#access-list 2 permit any  
R2(config)#interface fastethernet 0/1  
R2(config-if)#ip access-group 2 out  
R2(config-if)#
```

To test first do ping from 10.0.0.3 to 40.0.0.3 it should be request time out as this packet will filter by ACL. Then ping 30.0.0.3 it should be successfully replay.

Network to Network Access List

Task

Student's lab is configured on the network of 10.0.0.0. While management's system remain in the network of 40.0.0.0. You are asked to stop the lab system from gaining access in management systems

Now we will block the network of 10.0.0.0 from gaining access on the network 40.0.0.0. (if you are doing this practical after configuring previous example don't forget to remove the last access list 101. With no access-list command. Or just close the packet tracer without saving and reopen it to be continue with this example.)

```
R1(config)#access-list 103 deny ip 10.0.0.0 0.255.255.255 40.0.0.0 0.255.255.255  
R1(config)#access-list 103 permit ip any any  
R1(config)#interface fastethernet 0/0  
R1(config-if)#ip access-group 103 in  
R1(config-if)#exit  
R1(config)#
```

Verify by doing ping from 10.0.0.3 and 10.0.0.2 to 40.0.0.3. and 40.0.0.2. It should be request time out. Also ping computers of other network. ping shuld be sucessfully.

Network to host

Task

For the final scenario you will block all traffic to 40.0.0.3 from the Network of 10.0.0.0 To accomplish this write an extended access list. The access list should look something like the following.

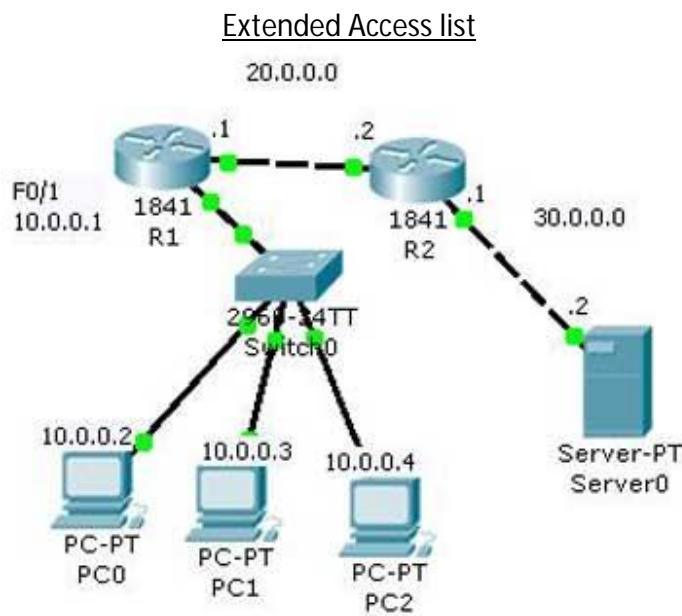
```
R1(config)#interface fastethernet 0/0  
R1(config-if)#no ip access-group 103 in  
R1(config-if)#exit  
R1(config)#no access-list 103 deny ip 10.0.0.0 0.255.255.255 40.0.0.0 0.255.255.255  
R1(config)#access-list 104 deny ip 10.0.0.0 0.255.255.255 40.0.0.3 0.0.0.0  
R1(config)#access-list 104 permit ip any any  
R1(config)#interface fastethernet 0/0  
R1(config-if)#ip access-group 104 in
```

```
R1(config-if)#exit  
R1(config)#
```

Verify by doing ping from 10.0.0.3 and 10.0.0.2 to 40.0.0.3. It should be request time out. Also ping computers of other network. ping shuld be sucessfully.

Application based Extended Access list

In pervoious example we filter ip base traffic. Now we will filter applicaion base traffic. To do this practical either create a topology as shown in figure and enable telnet and http and ftp service on server or download this pre configured topology and load it in packet tracer.



The established keyword

The **established** keyword is a advanced feature that will allow traffic through only if it sees that a TCP session is already established. A TCP session is considered established if the three-way handshake is initiated first. This keyword is added only to the end of extended ACLs that are filtering TCP traffic.

You can use TCP established to deny all traffic into your network except for incoming traffic that was first initiated from inside your network. This is commonly used to block all originating traffic from the Internet into a company's network except for Internet traffic that was first initiated from users inside the company. The following configuration would accomplish this for all TCP-based traffic coming in to interface serial 0/0/0 on the router:

```
R1(config)#access-list 101 permit tcp any any established  
R1(config)#interface serial 0/0/0  
R1(config-if)#ip access-group 101 in  
R1(config-if)#exit
```

Although the access list is using a permit statement, all traffic is denied unless it is first established from the inside network. If the router sees that the three-way TCP handshake is successful, it will then begin to allow traffic through.

To test this access list double click on any pc from the network 10.0.0.0 and select web brower. Now give the ip of 30.0.0.2 web server. It should get sucessfully access the web page. Now go 30.0.0.2 and open command prompt. And do ping to 10.0.0.2 or any pc from the network the 10.0.0.0. it will request time out.

Stop ping but can access web server

We host our web server on 30.0.0.2. But we do not want to allow external user to ping our server as it could be used as denial of services. Create an access list that will filter all ping requests inbound on the serial 0/0/0 interface of router2.

```
R2(config)#access-list 102 deny icmp any any echo  
R2(config)#access-list 102 permit ip any any  
R2(config)#interface serial 0/0/0  
R2(config-if)#ip access-group 102 in
```

To test this access list ping from 10.0.0.2 to 30.0.0.2 it should be request time out. Now open the web browser and access 30.0.0.2 it should be successfully retrieve

Grant FTP access to limited user

You want to grant ftp access only to 10.0.0.2. no other user need to provide ftp access on server. So you want to create a list to prevent FTP traffic that originates from the subnet 10.0.0.0/8, going to the 30.0.0.2 server, from traveling in on Ethernet interface E0/1 on R1.

```
R1(config)#access-list 103 permit tcp host 10.0.0.2 30.0.0.2 0.0.0.0 eq 20  
R1(config)#access-list 103 permit tcp host 10.0.0.2 30.0.0.2 0.0.0.0 eq 21  
R1(config)#access-list 103 deny tcp any any eq 20  
R1(config)#access-list 103 deny tcp any any eq 21  
R1(config)#access-list 103 permit ip any any  
R1(config)#interface fastethernet 0/1  
R1(config-if)#ip access-group 103 in  
R1(config-if)#exit
```

Grant Telnet access to limited user

For security purpose you don't want to provide telnet access on server despite your own system. Your system is 10.0.0.4. create a extended access list to prevent telnet traffic that originates from the subnet of 10.0.0.0 to server.

```
R1(config)#access-list 104 permit tcp host 10.0.0.4 30.0.0.2 0.0.0.0 eq 23  
R1(config)#access-list 104 deny tcp 10.0.0.0 0.255.255.255 30.0.0.2 0.0.0.0 eq 23  
R1(config)#access-list 104 permit ip any any  
R1(config)#interface fast 0/1  
R1(config-if)#ip access-group 104 in  
R1(config-if)#exit
```

Features and characteristics of EIGRP

- It is a Cisco Proprietary routing protocol.
- It is based on IGRP Routing protocol.
- It is an enhanced version of IGRP (Interior Gateway Routing Protocol) protocol.
- In comparison of IGRP it provides faster convergence times, superior handling of routing loops and improved scalability.
- It was released in 1994.
- It is a hybrid routing protocol.
- It has characteristics of both distance vector and link state protocols.
- It uses DUAL (Diffusing Update Algorithm) algorithm to select the best path.
- It uses RTP (Reliable Transport Protocol) to communicate with neighbors.
- It uses multicast for routing updates.
- It supports IP [Both IPv4 and IPV6], Apple Talk and IPX routed protocols.
- It includes subnet mask information in routing updates.
- It supports route summarization and discontiguous networks.
- It supports VLSM/CIDR.
- It supports load balancing across the six routes for a single destination.
- It supports trigger updates.

From introduction to till the preparation of this tutorial, EIGRP is ruling the world of routing protocols. The only negative about EIGRP was Cisco kept this protocol as proprietary protocol. In order to run this protocol, we had to buy all routers from Cisco. This thing was changed a little in 2013 when partial functionality of EIGRP was converted in open standard. Now we can also buy routers from other vendors along with Cisco, still running EIGRP on all routers.

Since EIGRP is hybrid protocol, it has advantages of both link state and distance vector protocol. It uses composite metric calculation formula to select the best route for destination. It sends partial or full update only when something is change in network. It maintains three tables for ultra-fast convergence.

1. Neighbor Table
2. Topology Table
3. Routing Table

Neighbor Table

EIGRP shares routing information only with neighbors. To know who the neighbors are, it uses neighbor table. When a new neighbor is discovered, EIGRP would add its address and interface on which neighbor is connected in neighbor table. EIGRP uses separate neighbor table for each routed protocol.

Topology Table

EIGRP uses this table to store all routes which it learned from neighbors. It contains a list of all destinations and routes advertised by neighboring routers. EIGRP selects single best route for each destination from this list. That route goes in routing table. Remaining routes are marked

as backup routes. EIGRP refers selected route as Successor and backup route as Feasible Successor. EIGRP uses separate topology table for each routed protocol.

Routing Table

EIGRP stores single best (Successor) route for each destination in this table. Router uses this table to forward the packet. There is a separate routing table for each routed protocol.

Protocol Dependent Modules

PDMs are the special feature of EIGRP. Through these modules EIGRP supports multiple network layer protocols. It maintains separate tables for separate routed (Network Layer) protocols. For example if you are using both (IPv4 and IPv6) versions of IP protocol, it will maintain separate IPv4/EIGRP and IPv6/EIGRP tables.

Metric

EIGRP uses metric to select the best route from all available routes for destination. Metric has five components.

- Bandwidth
- Load
- Delay
- Reliability
- MTU

From these only bandwidth and delay are by default enabled.

RTP

EIGRP uses RTP to communicate with other EIGRP speaking routers. RTP (Reliable Transport Protocol) uses multicast and unicast to exchange the data with neighbors. It uses class D address 224.0.0.10 for multicast. It keeps track of each multicast it sends out. EIGRP maintains a list of the neighbors who have replied. If it doesn't receive a reply from any neighbor, RTP will resend the same data using unicast. It will make 16 unicast attempts before declaring neighbor is dead.

DUAL

EIGRP uses DUAL (Diffusing Update Algorithm) to provide the fastest route convergence among all protocols. Route convergence includes:-

- Selecting best route from all available routes
- Supporting VLSMs
- Dynamically recovering from route failure
- Finding an alternative route if primary route goes down

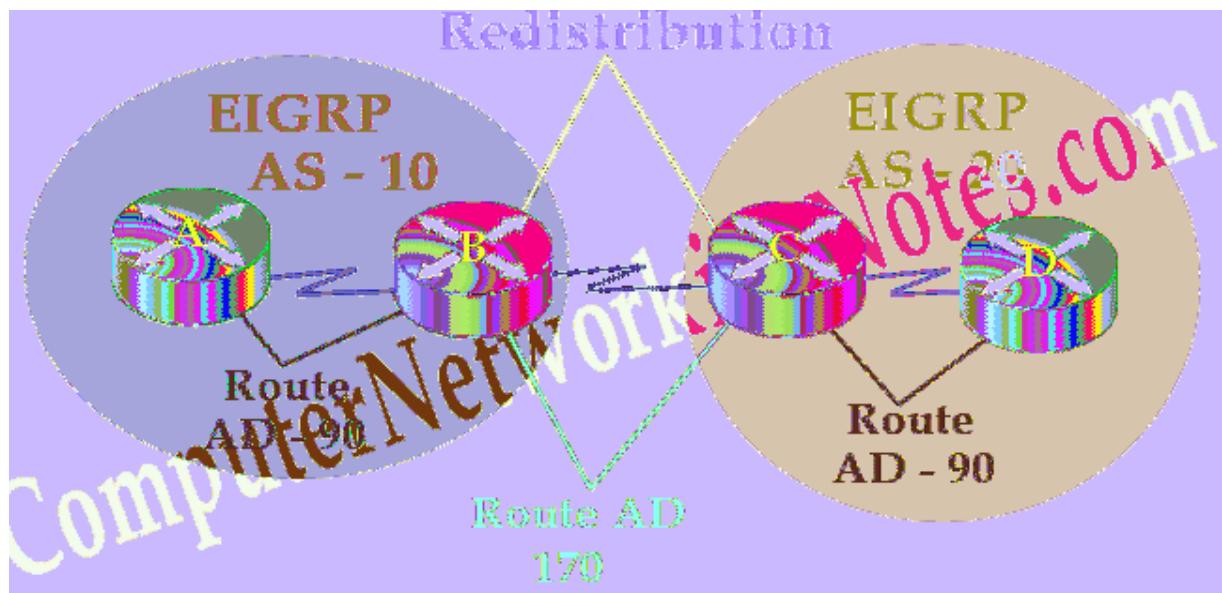
DUAL uses topology table along with RTP to accomplish above tasks in minimal time. As we know EIGRP maintain a copy of all routes including neighbors in topology table, so it would be the first place to look for an alternative route in a route failure situation. If EIGRP does not find an alternative here, it will ask neighbors for help. If neighbors have any updates about asked route, they will reply back with that information. This strong mechanism allows DUAL to find and maintain the best routes for destination speedily.

Autonomous System

EIGRP shares routing information only with neighbors. In order to become a neighbor AS number must be matched. AS create a logical boundary for route information. By default router will not propagate route information outside the AS. For example a router which belongs to AS number 10 will not share routing information with the router that belongs to AS number 20 or any other AS numbers except AS number 10. For easy administration a large network may have multiple ASes.

Not all routing protocols understand the concept of AS. Luckily EIGRP not only understand the concept of AS but also supports multiple ASes. We can easily configure multiple AS instance with EIGRP to divide a large network in smaller segments. By default EIGRP routers will not share routing information between different AS.

Redistribution is used to exchange the route information between different ASes. When a route is learned through the redistribution, it has higher AD value than its original source. For example EIGRP has two AD values 90 for interior EIGRP and 170 for exterior EIGRP. Exterior EIGRP means EIGRP instance which has different AS number.



Administrative Distance

In a complex network, we may have multiple routing protocols running simultaneously. Different routing protocols use different metrics to calculate the best path for destination. In

this situation router may receive different routes information for a single destination network. Routers use AD value to select the best path among these routes. Lower ad value has more trustworthiness.

AD value Protocol / Source

0	Directly connected interface
0 or 1	Static route
90	EIGRP (Interior)
110	OSPF
120	RIP
170	EIGRP (Exterior)
255	Unknown source

Let's understand it with a simple example; a router learned two different paths for 20.0.0.0/8 network from EIGRP Interior and EIGRP Exterior. Which one should it select?

Answer of this question is hidden in above table. Check the AD value of both protocols. Administrative distance is the believability of routing protocols. Routers measure each route source on a scale of 0 to 255. 0 is the best route. 255 is the worst, router will never use the route learned by this source. In our question we have two protocols EIGRP Interior and EIGRP Exterior. EIGRP Interior has lower AD value than EIGRP Exterior. So its route will be selected for routing table.

That's all for this part. In this part we covered basic terminology used in EIGRP routing protocol.

Essential configuration values

EIGRP Router doesn't trust anyone blindly. It checks following configuration values to insure that requesting router is eligible to become his neighbor or not.

1. Active Hello packets
2. AS Number
3. K-Values

Active Hello packets

EIGRP uses hello packets to maintain the neighborship between routers. It uses them for neighbor discovery and recovery process. Hello packets are periodically sent from all active interfaces.

By default when we enable EIGRP routing, all interfaces (that meet network command criteria) become participate of it. EIGRP allows us to exclude any interface from it.

Passive interface

passive-interface command is used to exclude an interface from EIGRP. Passive interface command is a double edged sword. If used carelessly, it could bring entire network down. Once you marked an interface as passive, EIGRP will never send a hello packet from it. And we know that hello packet is first condition of EIGRP neighborship. In this situation EIGRP neighborship will not take place on this interface. This could be critical if this interface is the only way to connect with other routers. Making this interface as passive will close all possible doors to communicate with those networks.

So our first condition that needs to be fulfilled in order to become an EIGRP neighbor is an active interface generating hello packets. Two routers will become neighbors only when they see each other's hello packets on a common network.

EIGRP sends hello packets from all active interfaces in hello interval. Hello interval is a time duration that EIGRP takes between two hello packets. Default hello interval for high bandwidth link is 5 seconds. For low bandwidth links, hello interval is 60 seconds.

- Ethernet, Token Ring, Point to Point serial links, HDLC leased lines are the examples of high bandwidth link.
- Multipoint circuits, Multipoint ATM, Multipoint Frame Relay, ISDN and BRIs are the example of low bandwidth links.

An EIGRP router must receive hello packets continuously from its neighbors. If it does not receive hello packets from any neighbor in hold down time, it will mark that neighbor as dead.

Hold time is the time duration that an EIGRP router waits before marking a router dead without receiving a hello packet from it. Typically hold down time is three times of hello interval. So for high bandwidth link it would be 15 seconds and 180 seconds for slow bandwidth link. We can adjust hold down time with *ip hold-time eigrp* command.

EIGRP uses multicast and unicast for hello packets delivery. It uses 224.0.0.10 IP address for multicast. Since hello packets do not have any important routing information, they need not be acknowledged.

Basically Hello packets perform two essential functions of EIGRP.

- Find another EIGRP router in network and help in building neighborship.
- Once neighborship is built, check continuously whether neighbor is alive or not.

Adjacency

Neighborship is referred as adjacency in EIGRP. So when you see New Adjacency in log, take it for new neighborship. It indicates that a new neighbor is found and neighborship with it has been established.

AS Number

An AS is a group of networks running under a single administrative control. This could be our company or a branch of company. Just like Subnetting AS is also used to break a large network in smaller networks.

AS creates a boundary for routing protocol which allow us to control how far routing information should be propagated. Beside this we can also filter the routing information before sharing it with other AS systems. These features enhance security and scalability of overall network.

Basically AS concept was developed for large networks. Routing protocols which were developed for small networks such as RIP do not understand the concept of AS systems.

There are two types of routing protocols IGP and EGP.

- **IGP** (Interior Gateway Protocol) is a routing protocol that runs in a single AS such as RIP, IGRP, EIGRP, OSPF and IS-IS.
- **EGP** (Exterior Gateway Protocol) is a routing protocol that performs routing between different AS systems. Nowadays only BGP (Border Gateway Protocol) is an active EGP protocol.

To keep distinguish between different autonomous systems, AS numbers are used. An AS number start from 1 and goes up to 65535. Same as IP addresses, AS numbers are divided in two types; Private and public.

- **Public AS Numbers:** - We only need to use public numbers if we connect our AS with Internet backbone through the BGP routes. IANA (Numbers Authority) controls the public AS numbers.
- **Private AS Numbers:** - Private AS numbers are used to break our internal network into the smaller networks.

EIGRP routers that belong to different ASs don't become neighbors therefore they don't share any routing information.

So our second condition that needs to be fulfilled in order to become EIGRP neighbor is the same AS number. Two routers will become neighbors only when they see same AS number in each other's hello packets.

K Values

EIGRP may use five metric components to select the best route for routing table. These are Bandwidth, Load, Delay, Reliability and MTU. By default EIGRP uses only two components; Bandwidth and delay. With K-Values we can control which components should be used in route metric calculation. For five metric components we have five K values.

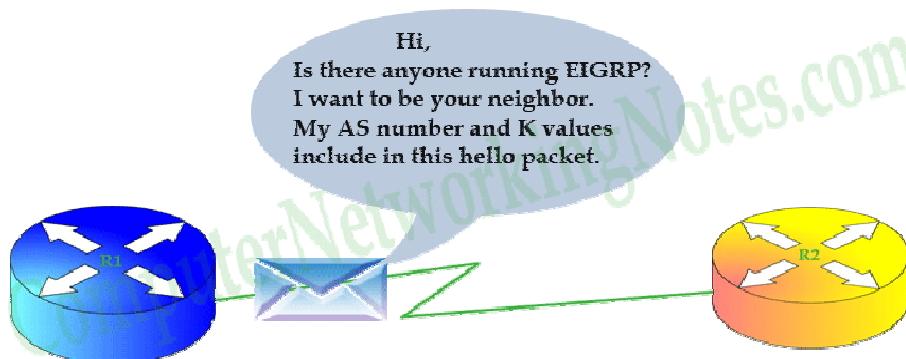
K Values	Metric components
K1	Bandwidth
K2	Load
K3	Delay
K4	Reliability
K5	MTU

Two routers must use same K Values in order to become the EIGRP neighbor. For example if one router is using three K- Values (K1, K2 and K3) while second router is using default K values (K1 and K3) then these two routers will never become neighbor.

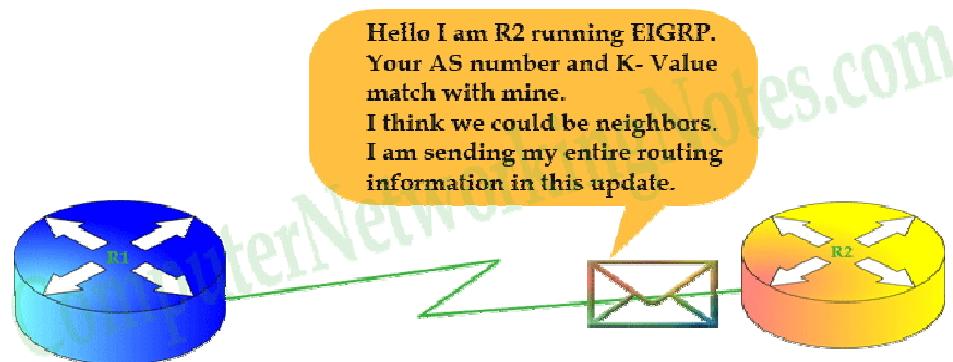
In order to become EIGRP neighbor two routers must use same K values.

EIGRP Neighbor Discovery process

Step 1:- First router R1 sends a hello packet from all active interfaces. This packet contains essential configuration values which are required to be a neighbor.



Step 2:- Receiving router R2 will compare these values with its own configuration values. If both necessary values match (AS number and K-values), it will reply with a routing update. This update includes all routes information from its routing table excluding one route. The route which it learned from the same interface that bring hello packet to it. This mechanism is known as split horizon. It states that if a router receives an update for route on any interface, it will not propagate same route information back to the sender router on same port. Split horizon is used to avoid routing loops.



Step 3:- First router will receive R2's routing update and sends an acknowledgement message back to R2.



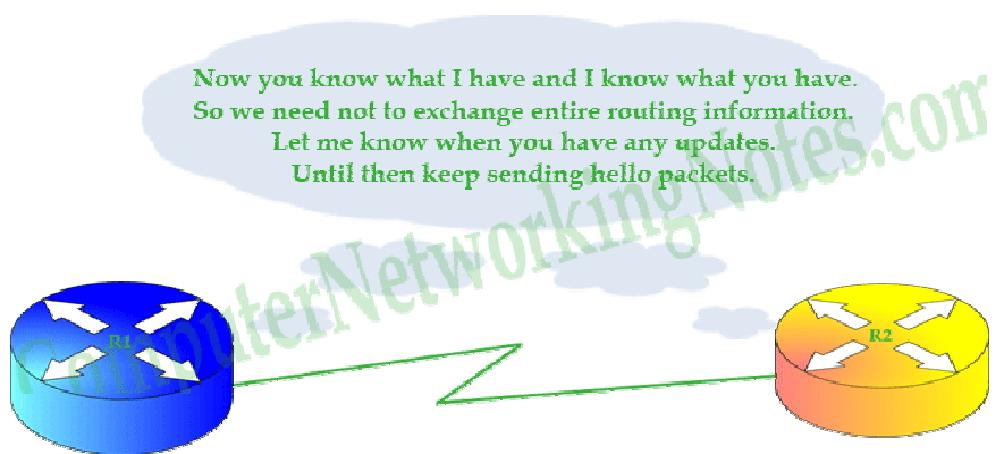
Step 4:- R1 will sync its EIGRP topology table with routing information that it received in routing update. It will also send a routing update containing all route information from its routing topology to R2.



Step 5:- R2 will respond with an acknowledgement message. It will also sync its EIGRP topology table with routing information that it received in routing update.



At this point, the two routers have become neighbors. Now they will maintain this neighborship with ongoing hello packets. If they see any change in network, they will update each other with partial updates.



Partial update contains information only about the recent change.

That's all for this part. In this part we explained how two routers become EIGRP neighbors.

K-Values and EIGRP Metrics

K-Values are the most confusing part of EIGRP. Usually newbies take K Values as EIGRP metric components. K Values are not the metric components in themself. They are only the place holder or influencer for actual metric components in metric calculation formula. So when we enable or disable a K value, actually we enable or disable its associate metric component.

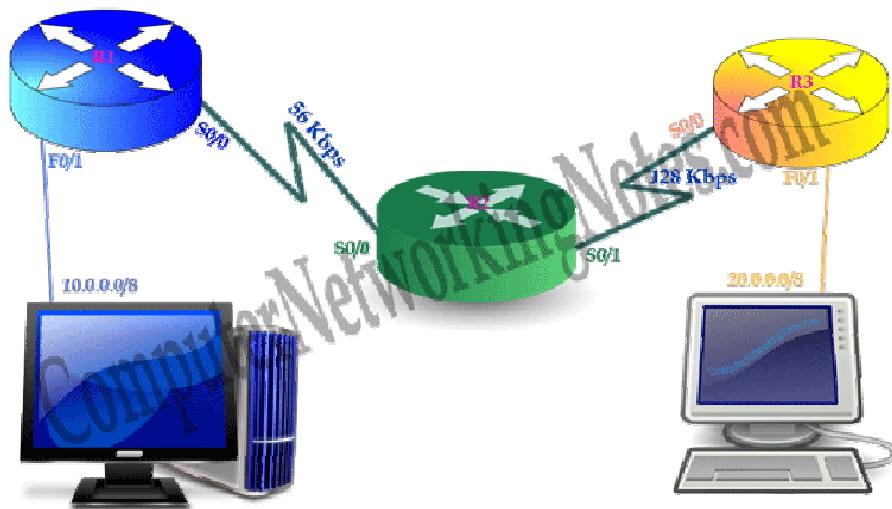
EIGRP uses four components out of five to calculate the routing metric.

K Value	Component	Description
K1	Bandwidth	Lowest bandwidth of route
K2	Load	Worst load on route based on packet rate
K3	Delay	Cumulative interface delay of route
K4	Reliability	Worst reliability of route based on keep alive
K5	MTU	Smallest MTU in path [Not used in route calculation]

Bandwidth (K1)

Bandwidth is a static value. It will change only when we make some physical (layer1) changes in route such as changing cable or upgrading link types. EIGRP picks lowest bandwidth from all outgoing interfaces of route to the destination network.

For example have a look on following figure.



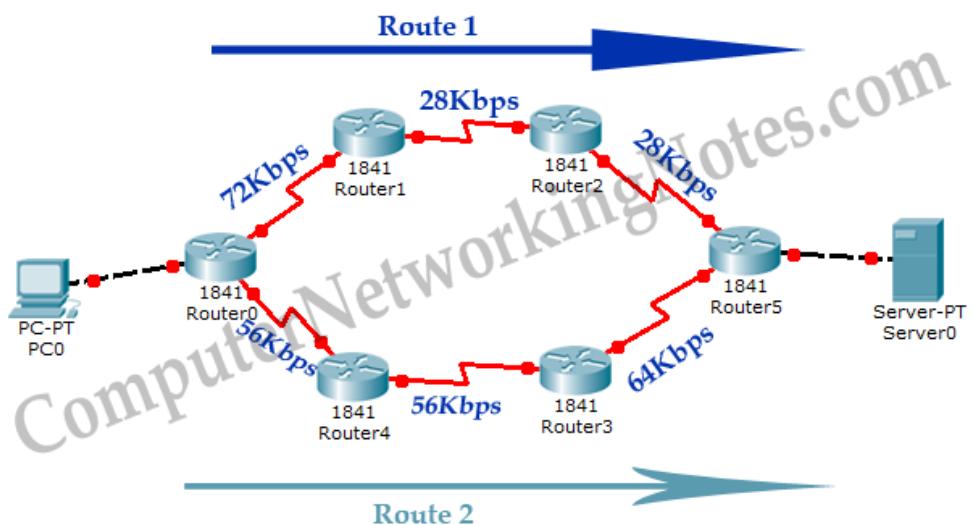
We have two serial links. One has 56Kbps bandwidth and other has 128Kbps. So which one will be selected?

Among these bandwidths EIGRP will pick 56Kbps for composite metric calculation formula.

You may surprise why it picks the lowest instead of the highest? Well picking the highest bandwidth doesn't give us a surety of equivalent bandwidth throughout the route. It's a maximum cap which means we will get its equivalent or lower bandwidth in this route.

While picking the lowest bandwidth gives us a guarantee of equivalent of higher bandwidth throughout the route. Since this is the bottleneck of route.

For example have a look on following network



With highest bandwidth comparison

Highest bandwidth of Route1 (72Kbps)

Highest bandwidth of Route2 (64Kbps)

Which route provides better bandwidth?

$72\text{Kbps (Route1)} > 64\text{Kbps (Route2)}$

With this comparison Route1 will be selected.

With lowest bandwidth comparison

Lowest bandwidth of Route1 (28Kbps)

Lowest bandwidth of Route2 (56Kbps)

Which route provides better bandwidth?

$56\text{Kbps (Route2)} > 28\text{Kbps (Route1)}$

With this comparison Route2 will be selected.

Looking at lowest bandwidth gives us the actual idea of route.

Next logical question is how EIGRP determine the bandwidth?

EIGRP first looks at **bandwidth** command. If bandwidth is set through this command, EIGRP will use it. If bandwidth is not set, it will use interface's default bandwidth.

When we enable an interface, router automatically assign a bandwidth value to it based on its type. For example serial interface has a default bandwidth value of 1544Kbps. Until we change this value with **bandwidth** command, it will be used where it is required.

Let me clear one more thing about bandwidth. Changing default bandwidth with **bandwidth** command does not change the actual bandwidth of interface. Neither default bandwidth nor bandwidth set by **bandwidth** command has anything to do with actual layer one link bandwidth.

Then what purpose does this command solve?

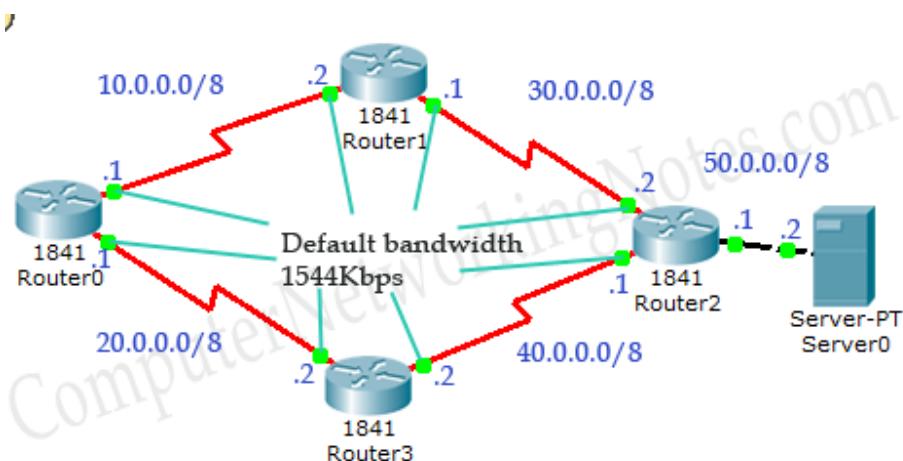
This command is only used to influence the routing protocol which uses bandwidth in route selection process such as EIGRP and OSPF.

Suppose we have two routes for single destination; Route1 and Route2. For some reason we want to take Route1 instead of Route2. How will we influence default metric calculation to select the Route1?

In starting of this article we talked about K-Values. K-Values allow us to influence the metric calculation. K1 is associated with bandwidth. K1 gets its weight from interface's default bandwidth or bandwidth set through the **bandwidth** command. Changing default bandwidth with **bandwidth** command will change the K1's value in metric calculation formula.

So to take Route1, we will have to make its lowest bandwidth higher than Route2. This can be done in two ways; either raise the lowest bandwidth of Route1 higher than Route2 or reduce the lowest bandwidth of Route2 lower than Route1. Both can be done easily with **bandwidth** command.

Let's understand this with a simple example. Following figure illustrate a simple EIGRP network.



In this network R0 has two routes to reach at 50.0.0.0/8 network.

1. Route1 (Via R0 – R1 – R2)
2. Route2 (Via R0 – R3 – R2)

EIGRP is configured on all routers and all links have default bandwidth.

```
Router#sh run
interface Serial0/0/0
 ip address 10.0.0.1 255.0.0.0
 clock rate 64000
!
interface Serial0/0/1
 ip address 20.0.0.1 255.0.0.0
router eigrp 1
 network 10.0.0.0
 network 20.0.0.0
 auto-summary

Router#sh run
interface Serial0/0/0
 ip address 10.0.0.2 255.0.0.0
!
interface Serial0/0/1
 ip address 30.0.0.1 255.0.0.0
 clock rate 64000
router eigrp 1
 network 30.0.0.0
 network 10.0.0.0
 auto-summary

Router#sh run
interface Serial0/0/0
 bandwidth 2800
 ip address 40.0.0.2 255.0.0.0
!
interface Serial0/0/1
 ip address 20.0.0.2 255.0.0.0
 clock rate 64000
router eigrp 1
 network 20.0.0.0
 network 40.0.0.0
 auto-summary
!
router eigrp 1
 network 20.0.0.0
 network 40.0.0.0
 auto-summary

Router#sh run
!
interface FastEthernet0/0
 ip address 50.0.0.1 255.0.0.0
 duplex auto
 speed auto
interface Serial0/0/0
 ip address 40.0.0.1 255.0.0.0
 clock rate 64000
!
interface Serial0/0/1
 ip address 30.0.0.2 255.0.0.0
!
interface Serial0/0/0
 ip address 40.0.0.1 255.0.0.0
 clock rate 64000
!
router eigrp 1
 network 30.0.0.0
 network 40.0.0.0
 network 50.0.0.0
 auto-summary
```

Serial link has default bandwidth of 1544Kbps. Until we change bandwidth of any route, both routes have equal lowest bandwidth.

Route1's lowest bandwidth (1544Kbps) = Route2's lowest bandwidth (1544Kbps)

Both routes are load balanced with equal cost value 2684416.

Router0

Physical Config CLI

IOS Command Line Interface

```

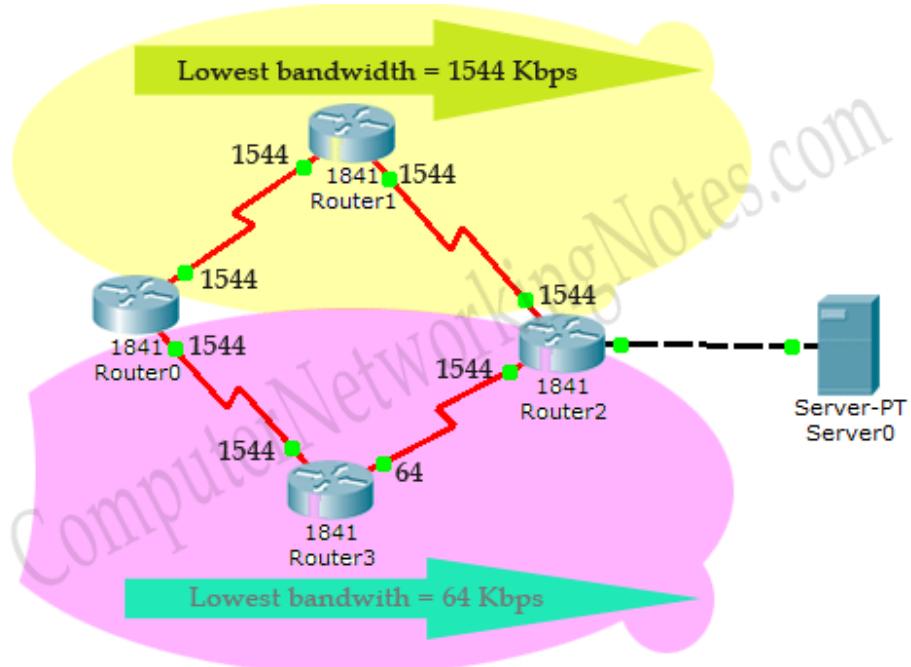
Router>enable
Router#show ip route eigrp
D 30.0.0.0/8 [90/2681856] via 10.0.0.2, 01:47:37, Serial0/0/0
D 40.0.0.0/8 [90/2681856] via 20.0.0.2, 01:38:39, Serial0/0/1
D 50.0.0.0/8 [90/2684416] via 10.0.0.2, 01:47:37, Serial0/0/0
[90/2684416] via 20.0.0.2, 01:38:35, Serial0/0/1
Router#

```

Ok, let's change default bandwidth to see how bandwidth component influence the route metric.

Set bandwidth to 64Kbps (lower than default 1544Kbps) on R3's serial 0/0/0 interface.

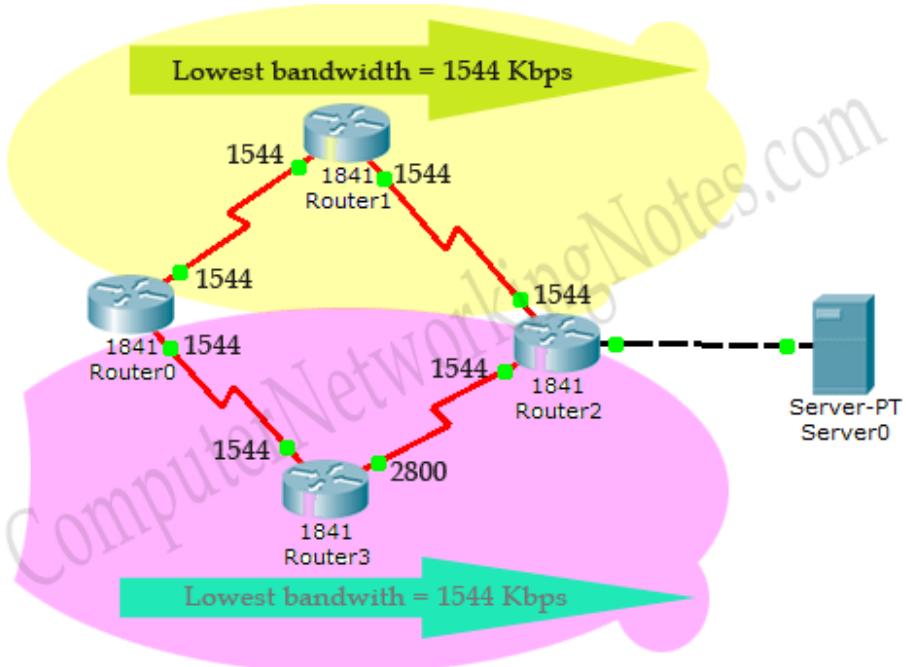
Route1's lowest bandwidth (1544Kbps) > Route2's lowest bandwidth (64Kbps)



Now Route1 has the higher lowest bandwidth so it would be selected.

Ok let's change bandwidth at R3 again this time increase default bandwidth to 2800Kbps.

Route1's lowest bandwidth (1544Kbps) = Route2's lowest bandwidth (1544Kbps)



Both routes have equal lowest bandwidth. They will be load balanced.

<p>Router0</p> <p>Physical Config CLI</p> <p>IOS Command Line Interface</p> <pre> Router>enable Router#show ip route eigrp Default D 30.0.0.0/8 [90/2681856] via 10.0.0.2, 02:15:22, Serial0/0/0 D 40.0.0.0/8 [90/2681856] via 20.0.0.2, 02:06:23, Serial0/0/1 D 50.0.0.0/8 [90/2684416] via 10.0.0.2, 02:15:22, Serial0/0/0 [90/2684416] via 20.0.0.2, 02:06:19, Serial0/0/1 Router#show ip route eigrp D 30.0.0.0/8 [90/2681856] via 10.0.0.2, 02:17:43, Serial0/0/0 D 40.0.0.0/8 [90/3193856] via 10.0.0.2, 00:00:16, Serial0/0/0 D 50.0.0.0/8 [90/2684416] via 10.0.0.2, 02:17:43, Serial0/0/0 Router#show ip route eigrp D 30.0.0.0/8 [90/2681856] via 10.0.0.2, 02:15:22, Serial0/0/0 D 40.0.0.0/8 [90/2681856] via 20.0.0.2, 02:06:23, Serial0/0/1 D 50.0.0.0/8 [90/2684416] via 10.0.0.2, 02:15:22, Serial0/0/0 [90/2684416] via 20.0.0.2, 02:06:19, Serial0/0/1 Router# </pre>	<p>Router3</p> <p>Physical Config CLI</p> <p>IOS Comm</p> <pre> Router>enable Router#configure terminal Enter configuration commands, one per Router(config)#interface serial 0/0/0 Router(config-if)#bandwidth 64 Router(config-if)# %DUAL-5-NBRCHANGE: IP-EIGRP 1: Neighbor e down %DUAL-5-NBRCHANGE: IP-EIGRP 1: Neighbor ency Router(config-if)#bandwidth 2800 Router(config-if)# %DUAL-5-NBRCHANGE: IP-EIGRP 1: Neighbor e down </pre>
---	--

Here I have question for you.

Why EIGRP load balanced between Route1 and Route2 while now Route2 has better bandwidth?

Because EIGRP uses the lowest bandwidth of route to calculate the path cost and that is still 1544Kbps.

Load (K2)

Load is a dynamic value that changes frequently. It is based on packet rate and bandwidth of interface. It calculates the volume of traffic passing through the interface in comparison of maximum capacity. It is expressed on a scale of 255 where 1 represent that an interface is empty and 255 represent that an interface is fully utilized.

Since data flows from both directions, router maintains two separate metric counters:

- **Txload** for outgoing traffic
- **Rxload** for incoming traffic

If K2 is enabled, maximum **Txload** value will be used in composite metric calculation formula.

Delay (k3)

Delay reflects the time taken by a packet in crossing the interface. It is measured in fractions of seconds. Like as bandwidth Cisco has implicit delay values for all interfaces based on the type of interface hardware. For example a FastEthernet has default delay of 100 microseconds. Since it is a static value, we can override it with **delay** command.

Delay can be set anywhere from 10 to 167,772,140 microseconds.

Default delay value or value set by **delay** command has nothing to do with the actual delay caused by interface. Just like bandwidth, this value is also an influencer.

It is expressed in terms of tens of microseconds. To define a delay of 1000 microseconds, we need to configure 100(1000/10) on interface. Output of **show interface** command will automatically multiply it with ten before displaying.

Total delay is used in metric calculation formula.

Total delay = delay received from neighboring router + its own interface delay

EIGRP is an enhanced distance vector routing protocol. It also uses route poisoning, withdrawing route, split horizon and poisoned reverse for loop free optimized network. For all these mentioned techniques EIGRP use the maximum delay as the indication of the unreachable route. To denote the unreachable route EIGRP uses the delay of 16,777,215 tens of the microseconds.

Reliability (K4)

Just like load, reliability is also a dynamic value. It compares all successfully received frames against all received frames. 100% reliability indicates that all the frames which we received were good. We don't have any issue with physical link. If we have any issue with physical link, this value will be decrease.

Reliability is expressed as the fraction of 255. 255 expresses 100% reliability while 0 represents 0% reliability. If K4 is enabled in metric calculation formula, it will use minimal reliability.

MTU (K5)

MTU stands for maximum transmission unit. It is advertised with routing update but it does not actively participate in metric calculation. EIGRP allows us to load balance between equal cost paths (6 maximum, default set to 4). It is used when equal cost paths for same destination exceed the number of allowed paths set from **maximum-paths** command. For example we set maximum allowed paths for load balancing to 5 and metric calculates 6 equal cost paths for a single destination. In this situation path with lowest MTU will be ignored.

EIGRP Metric Calculation Formula

EIGRP uses following formula to produce a single 32 bit metric:-

$$\left[\left(K_1 \cdot \text{Bandwidth}_E + \frac{K_2 \cdot \text{Bandwidth}_E}{256 - \text{Load}} - K_3 \cdot \text{Delay}_E \right) \cdot \frac{K_5}{K_4 + \text{Reliability}} \right] \cdot 256$$

At first glance this formula looks like a complicated equation. But it is not as difficult as it sounds. Let's make it easier.

As we know MTU (K5) is not actively participate in formula. So set its value to Zero. When K5 is equal to 0 then $[K_5 / (K_4 + \text{reliability})]$ is defined to be 1.

By default EIGRP does not use dynamic values in metric. This will disable two more components; load (K_2) and reliability (K_4).

Now only two static values remain in formula.

Use of default constants [K_1 (Enabled), K_2 (Disabled), K_3 (Enabled), K_4 (Disabled), K_5 (Disabled not used)] reduce our formula to:-

$$\text{Metric} = (\text{Bandwidth}_E + \text{Delay}_E) * 256.$$

Cisco uses following configuration values for Bandwidth and delay

$\text{Bandwidth}_E = 10^7 / \text{least bandwidth of route}$ [Lowest bandwidth from all interfaces between source and destination. Use interface default bandwidth wherever bandwidth is not set through the **bandwidth** command]

$\text{Value}_E = \text{cumulative delay of route}$ [Sum of all outgoing interface's delay. Use interface default delay, if not set through the **delay** command]

Putting these configuration values will make formula to look like this

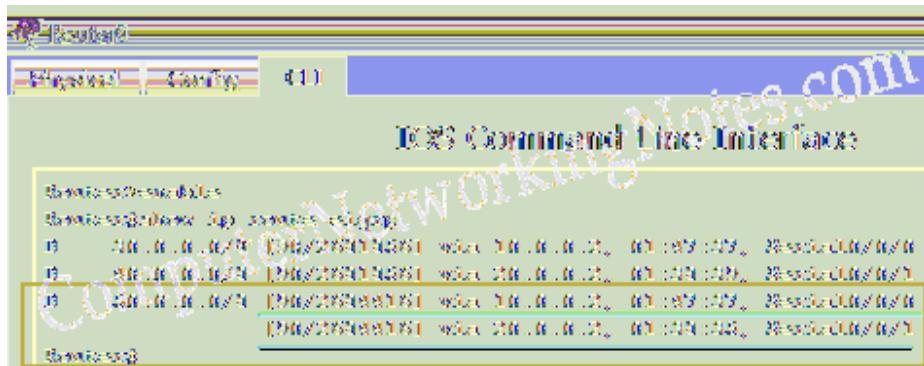
$$\text{Metric} = \left(\left(\frac{10^7}{\text{Least-bandwidth}} + \text{Cumulative-delay} \right) * 256 \right)$$

Before we move further, let me explain why EIGRP keeps dynamic values disable by default.

Dynamic values change over the time. Enabling dynamic values will force EIGRP routers to calculate metric all the time and send updates each other just because the load or reliability of an interface has changed. This will create serious performance issue. To avoid such a situation EIGRP only enables static values for metric calculation.

If we only enable static values for metric calculation, EIGRP will not recalculate the metric unless it changed. Static values change only when a physical change occurred in network such as an interface is down or router is dead. This will keep EIGRP nice and clean.

Let's see this formula in action. Earlier in this tutorial we used an example topology to explain the bandwidth component. Load that topology in packet tracer and run **show ip route eigrp** command from privilege mode. We have four routes for three destination networks. One destination network has two routes.



30.0.0.0/8

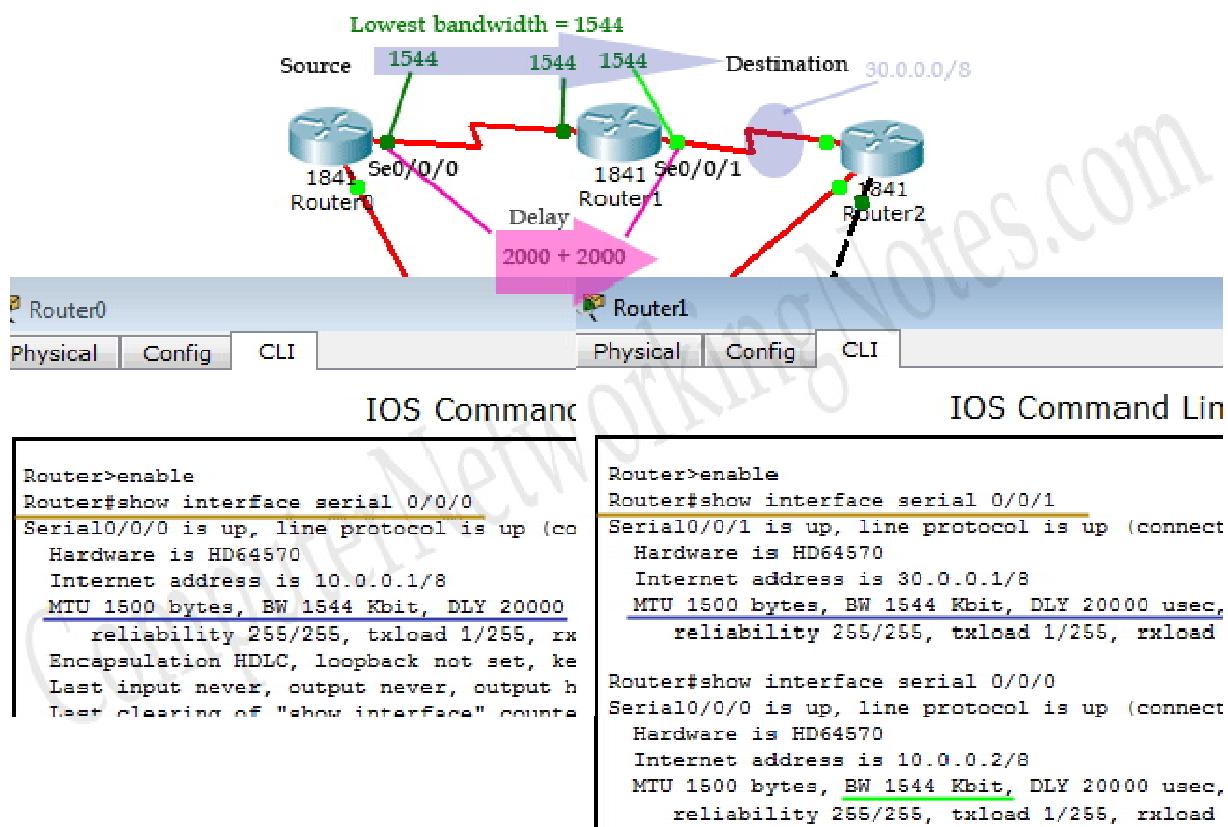
For this destination network metric cost is 2681856. Before we learn how this cost was calculated, we need to understand some key points associated with formula.

- EIGRP picks the lowest bandwidth from all interfaces in route.
- EIGRP picks delay from all outgoing interfaces in route.
- **show interface [interface]** command of privilege mode will display the configured value of metric components.
- While calculating the cost term least-bandwidth uses the unit of Kbps (Kilobits per second).
- **show interface [interface]** command list bandwidth in Kbps. So we can use listed bandwidth in formula as it is.
- While calculating the cost term cumulative-delay uses the unit of tens of microseconds.
- **show interface [interface]** command list delay in microseconds. So we need to divide it with 10 before using it in formula.

- Any decimal value will be rounded back to the nearest integer before performing the rest of the formula.

We have three serial interfaces between source and destination. So our first step is to find out the value of bandwidth and delay.

We can use show interface command to know the values.



All interfaces have equal bandwidth so our least bandwidth would be 1544Kbps.

We have two outgoing interfaces between source and destination. Both have a default delay of 20000 microseconds so total delay would be 40000 microseconds. As we know this delay is in microseconds and formula uses the unit of "tens of microseconds". We need to divide 40000 with 10. So our cumulative delay would be $40000/10 = 4000$.

Okay now we have least bandwidth (1544Kbps) and cumulative delay (4000) let's put them in formula

$$\text{Metric} = \left(\left(\frac{10^7}{\text{Least-bandwidth}} \right) + \text{Cumulative-delay} \right) * 256$$

$$\text{Metric} = ((10000000/1544) + 4000) * 256$$

$$\text{Metric} = ((6476.6839) + 4000) * 256$$

As I said "Any decimal value will be rounded back to the nearest integer before performing the rest of the formula."

Before solving rest of the formula, convert decimal value back in positive integer.

$$\text{Metric} = ((6476) + 4000) * 256$$

$$\text{Metric} = (10476) * 256$$

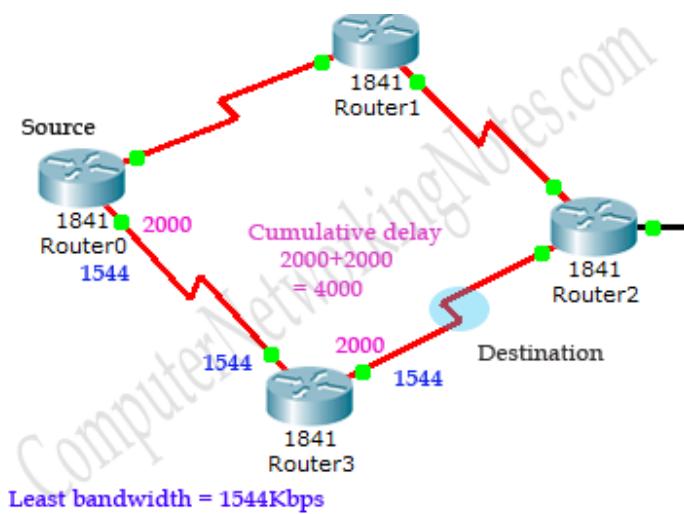
$$\text{Metric} = 10476 * 256$$

$$\text{Metric} = 2681856$$

Great! We have revealed the cost calculation method. Let's do this calculation again for next route.

40.0.0.0/8

For this route we have lowest bandwidth 1544Kbps and cumulative delay of 4000(ten of microseconds).



Let's put these values in our formula

$$\text{Metric} = ((10000000/1544)+4000)*256$$

$$\text{Metric} = (6476 + 4000) * 256$$

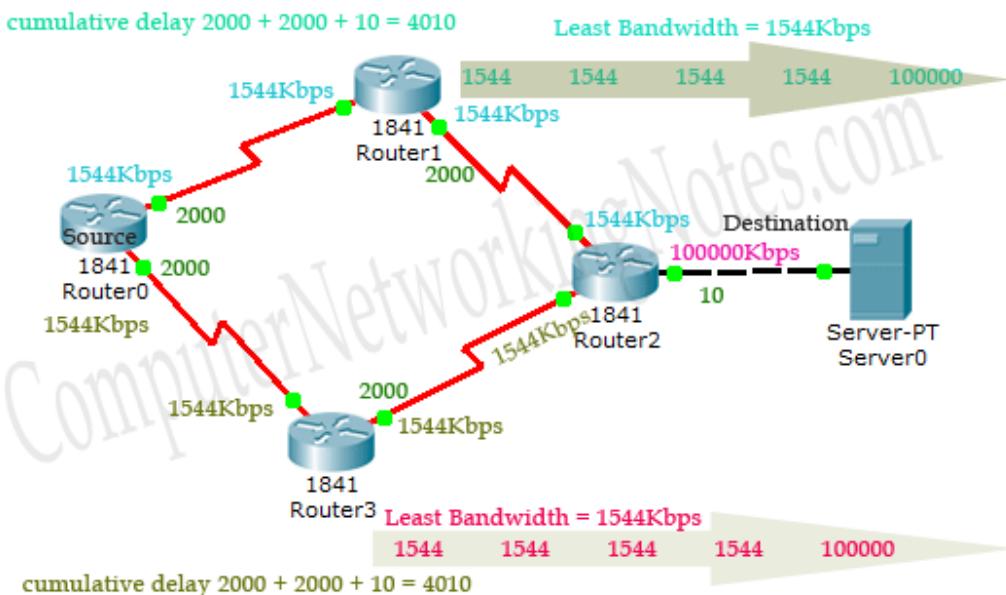
$$\text{Metric} = 10476 * 256$$

$$\text{Metric} = 2681856$$

Fine, now we have only route left. Let's figure out its cost also.

50.0.0.0/8

For this destination we have two routes. Both routes have equal least bandwidth and cumulative delay. So naturally their cost will also be same. As we know EIGRP automatically load balance equal cost routes and these routes have equal cost. So they both make their way to routing table.



$$\text{Metric} = ((10000000/1544) + 4010) * 256$$

$$\text{Metric} = (6476 + 4010) * 256$$

$$\text{Metric} = 10486 * 256$$

$$\text{Metric} = 2684416$$

This is how EIGRP calculates the route cost. In job life you will rarely need to calculate the route cost manually

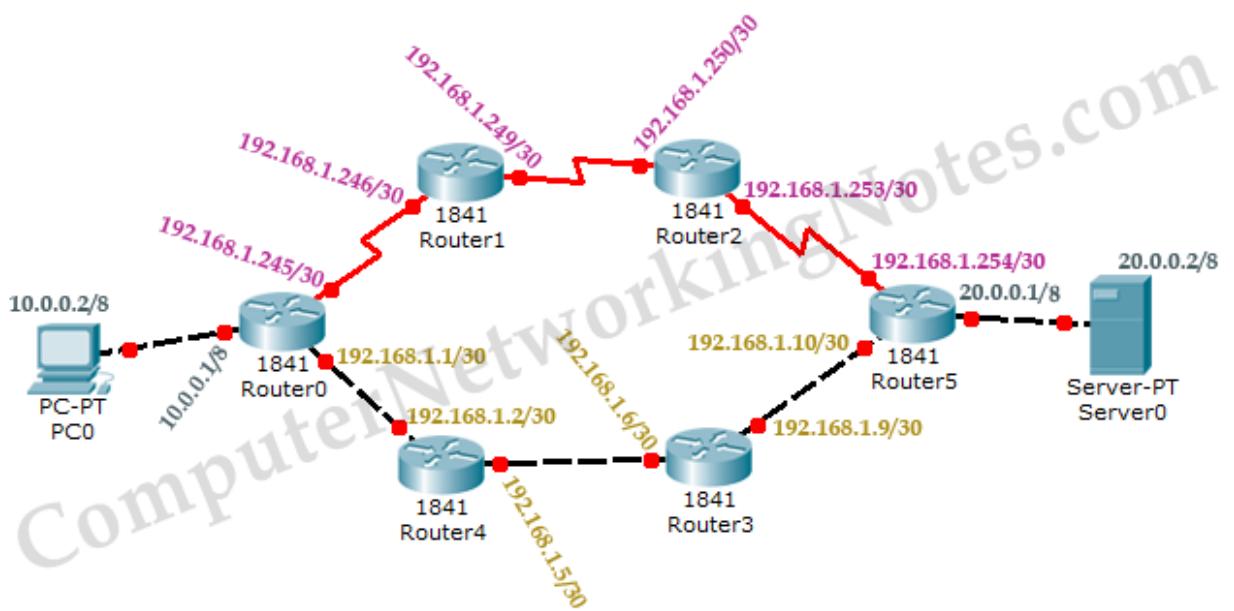
Configuration for EIGRP Routing Protocol

In this assignment we will see basic concepts of EIGRP such as Features and characteristics of EIGRP, Neighbor Table, Topology Table, Routing Table, Protocol Dependent Modules, Metric, RTP, DUAL, Autonomous System and Administrative Distance.

Also we will see how two routers become EIGRP neighbor and maintain this neighborship. In order to become an EIGRP neighbor, three essential configuration values must be matched.

EIGRP uses composite metric calculation formula to calculate the best path. Bandwidth, reliability, delay, load and MTU are the components of formula. In this we explained these components with formula in easy language with examples.

Create a topology as illustrated in following figure or download this pre-created topology.

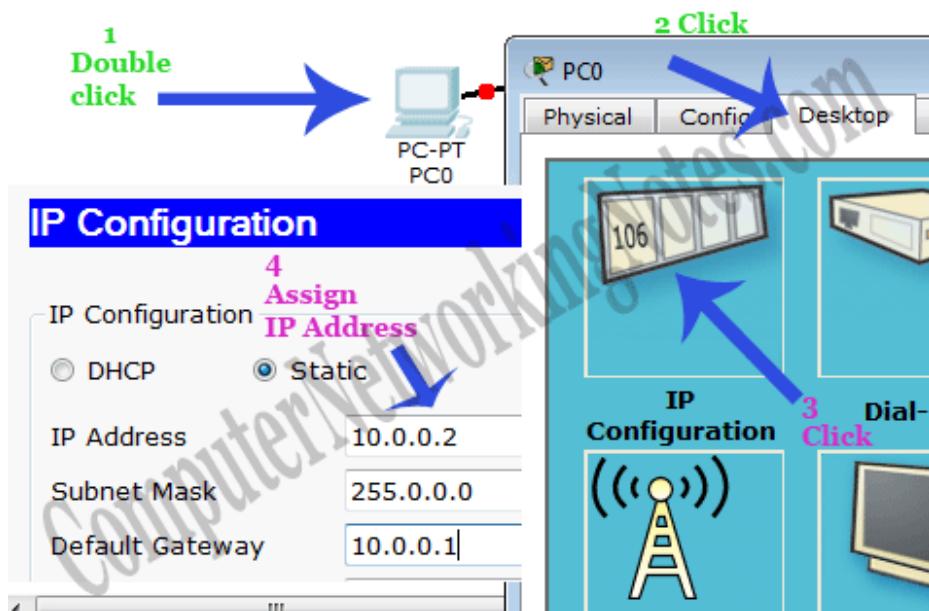


Device	Interface	IP Configuration	Connected with
PC0	Fa0/0	10.0.0.2/8	Router0's Fa0/0
Router0	Fa0/0	10.0.0.1/8	PC0's Fa0/0
Router0	Fa0/1	192.168.1.1/30	Router4's Fa0/1
Router4	Fa0/1	192.168.1.2/30	Router0's Fa0/1
Router4	Fa0/0	192.168.1.5/30	Router3's F0/0
Router3	Fa0/0	192.168.1.6/30	Router4's Fa0/0
Router3	Fa0/1	192.168.1.9/30	Router5's Fa0/1

Router5	Fa0/1	192.168.1.10/30	Router3's Fa0/1
Router5	Fa0/0	20.0.0.1/8	Serve0's Fa0/0
Server	Fa0/0	20.0.0.2/8	Router5's Fa0/0
Router5	Se0/0/0	192.168.1.254/30	Router2's Se0/0/0
Router2	Se0/0/0	192.168.1.253/30	Router5's Se0/0/0
Router2	Se0/0/1	192.168.1.250/30	Router1's Se0/0/1
Router1	Se0/0/1	192.168.1.249/30	Router2's Se0/0/1
Router1	Se0/0/0	192.168.1.246/30	Router0's Se0/0/0
Router0	Se0/0/0	192.168.1.245/30	Router1's Se0/0/0

Assign IP address to PCs

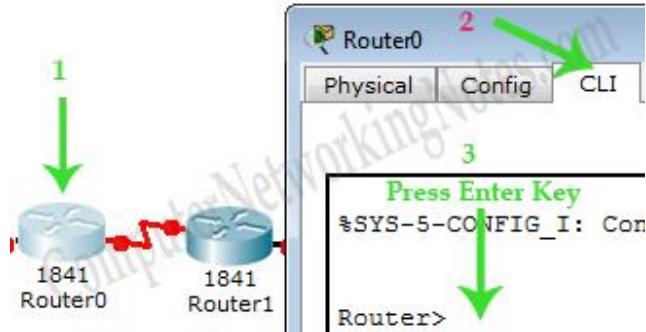
Double click **PC0** and click **Desktop** menu item and click **IP Configuration**. Assign IP address **10.0.0.2/8** to **PC0**.



Repeat same process for Server0 and assign IP address 20.0.0.2/8.

Assign IP address to interfaces of routers

Double click **Router0** and click **CLI** and press Enter key to access the command prompt of **Router0**.



Three interfaces **FastEthernet0/0**, **FastEthernet0/1** and **Serial0/0/0** of **Router0** are used in this topology. 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)#
```

From global configuration mode we can enter in interface mode. From there we can configure the interface. Following commands will assign IP address on FastEthernet0/0 and FastEthernet0/1.

```
Router(config)#interface fastEthernet 0/0
Router(config-if)#ip address 10.0.0.1 255.0.0.0
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#interface fastEthernet 0/1
Router(config-if)#ip address 192.168.1.1 255.255.255.252
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#
```

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

ip address 10.0.0.1 255.0.0.0 command will assign IP address to interface.

no shutdown command will 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.

```
Router#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.

```
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 192.168.1.245 255.255.255.252
Router(config-if)#clock rate 64000
Router(config-if)#bandwidth 64
Router(config-if)#no shutdown
Router(config-if)#exit
```

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 192.168.1.245 255.255.255.252 Command assigns IP address to interface. For serial link we usually use IP address from /30 subnet.

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

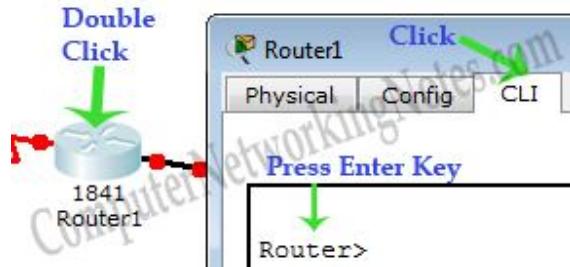
Bandwidth works as an influencer. It is used to influence the metric calculation of EIGRP or any other routing protocol which uses bandwidth parameter in route selection process.

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 remaining routers. We need to provided clock rate and bandwidth only on DCE side of serial interface. Following command will assign IP addresses on interface of Router1.

Router1



```
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 192.168.1.246 255.255.255.252
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#interface serial 0/0/1
Router(config-if)#ip address 192.168.1.249 255.255.255.252
Router(config-if)#clock rate 64000
Router(config-if)#bandwidth 64
Router(config-if)#no shutdown
Router(config-if)#exit
```

We will use same commands to assign IP addresses on interfaces of remaining routers.

Router2

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface serial 0/0/1
Router(config-if)#ip address 192.168.1.250 255.255.255.252
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#interface serial 0/0/0
Router(config-if)#ip address 192.168.1.253 255.255.255.252
Router(config-if)#clock rate 64000
Router(config-if)#bandwidth 64
Router(config-if)#no shutdown
Router(config-if)#exit
```

Router5

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

```
Router(config)#interface fastEthernet 0/0
Router(config-if)#ip address 20.0.0.1 255.0.0.0
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#interface fastEthernet 0/1
Router(config-if)#ip address 192.168.1.10 255.255.255.252
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#
Router(config)#interface serial 0/0/0
Router(config-if)#ip address 192.168.1.254 255.255.255.252
Router(config-if)#no shutdown
Router(config-if)#exit
```

Router3

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface fastEthernet 0/0
Router(config-if)#ip address 192.168.1.6 255.255.255.252
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#interface fastEthernet 0/1
Router(config-if)# ip address 192.168.1.9 255.255.255.252
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#

```

Router4

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface fastEthernet 0/0
Router(config-if)#ip address 192.168.1.5 255.255.255.252
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#interface fastEthernet 0/1
Router(config-if)# ip address 192.168.1.2 255.255.255.252
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#

```

Great job we have finished our half journey. Now routers have information about the networks that they have on their own interfaces. Routers will not exchange this information between

them on their own. We need to implement EIGRP routing protocol that will insist them to share this information.

To be on same track I have uploaded my practice topology on our server. Use this if you want to skip the IP configuration part.

Configure EIGRP routing protocol

Enabling EIGRP is a two steps process:-

1. Enable EIGRP routing protocol from global configuration mode.
2. Tell EIGRP which interfaces we want to include.

For these steps following commands are used respectively.

```
Router(config)# router eigrp autonomous_system_#
Router(config-router)# network IP_network_# [subnet_mask]
```

Router(config)# router eigrp autonomous_system_#

This command will enable EIGRP routing protocol in router. We can use any ASN (Autonomous System Number) from 1 to 65,535. In order to become EIGRP neighbors this number must be same on all participates.

Router(config-router)# network IP_network_# [subnet_mask]

This command allows us to specify the local interfaces which we want to include in EIGRP. Basically we define a range of addresses and router search for these addresses in local interfaces. If match found EIGRP will be enabled on that interface. Once enabled, EIGRP will starts advertising about the connected subnets with that interface.

We have two options while defining the range of addresses with **network** command

1. Without wildcard mask
2. With wildcard

Without wildcard

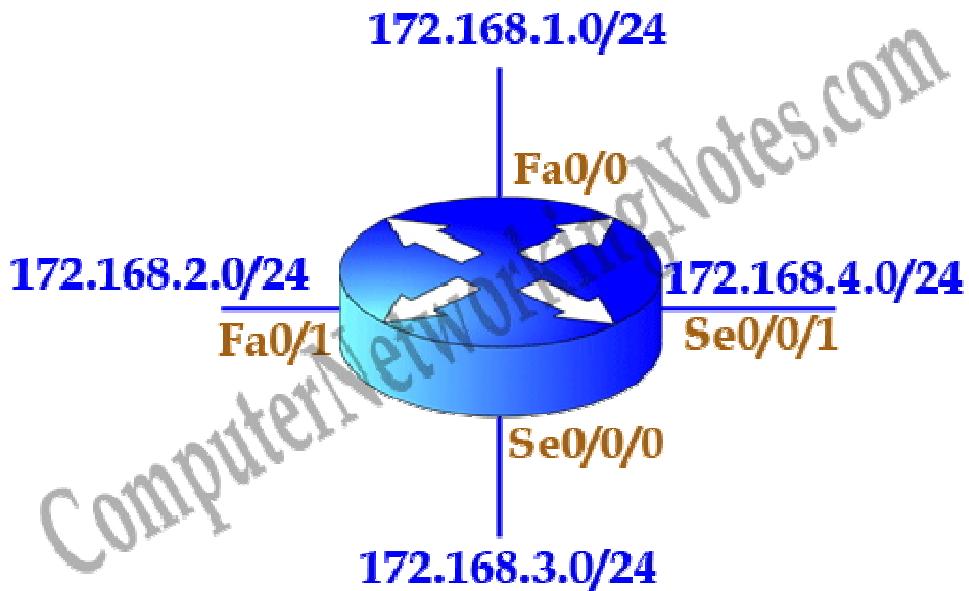
Choosing this option allows us to configure the classful network. This option is very straightforward. All we need to do is, type the network ID with network command. For example network 172.168.0.0 command will enable EIGRP on all interfaces which belong to network 172.168.0.0.

What if I type network number instead of network ID?

Well in this situation EIGRP will automatically convert it back to network ID in which this network number is resides. For example 172.168.1.1 will be converted back in 172.168.0.0.

This creates another query. Why it will be converted in 172.168.0.0 instead of 172.168.1.0?

Answer of this question is hidden in classful configuration. In classful configuration EIGRP will match network addresses within default boundary. Consider following figure



We have four networks 172.168.1.0/24, 172.168.2.0/24, 172.168.3.0/24 and 172.168.4.0/24 Subnetted from single class B network 172.168.0.0/16. Classful configuration does not understand the concept of Subnetting. In classful configuration all these networks belong to a single network. Classful configuration works only within default boundary of mask. Default boundary of this address is 16 bits. So it will match only first 16 bits (172.168.x.y) of network address.

If we want to exclude serial interfaces from EIGRP, we need to configure network command with more specific information.

With wildcard

In this option we provide wildcard mask along with network ID. Wildcard mask allows us to match exact networks. With wildcard we are no longer limited with default boundaries. We can match Subnetted networks as well as default networks.

For example we were tasked to exclude serial interfaces in above configuration. We can use a wildcard mask of 0.0.0.255 to match the subnet mask of /24.

```
Router(config-router)# network 172.168.1.0 0.0.0.255
Router(config-router)# network 172.168.2.0 0.0.0.255
```

Above commands will ask router to match /24 bits of address instead of default /16 bits. Now router will look for 172.168.1.x and 172.168.2.x network. Our serial interfaces have 172.168.3.0/24 and 172.168.4.0/24 networks which do not fall in these search criteria.

If you are unfamiliar with wildcard mask, I suggest you to read our tutorials on ACL where we explained wildcard mask in detail with examples.

Until you learn wildcard mask, use subnet mask in the place of wildcard mask. Following commands are also valid and do the same job by matching /24 bits of address.

```
Router(config-router)# network 172.168.1.0 255.255.255.0
Router(config-router)# network 172.168.2.0 255.255.255.0
```

Subnet mask is a substitute, not a replacement of wildcard mask. When we use Subnet mask, router converts them in wildcard mask before searching for associated interfaces. We can look in running configuration to know what exactly being used by router.

IOS Command Line Interface

```
Router>enable
Router#config term
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#router eigrp 20
Router(config-router)#network 172.168.1.0 255.255.255.0
Router(config-router) #network 172.168.2.0 0.0.0.255
Router(config-router) #exit
Router(config)#exit
Router#
Router#show run
Building configuration...

Current configuration : 751 bytes
:
interface Serial0/0/0
 bandwidth 64
 ip address 192.168.1.245 255.255.255.252
 clock rate 64000
!
interface Serial0/0/1
 no ip address
 shutdown
!
interface Vlan1
 no ip address
 shutdown
!
router eigrp 20
 network 172.168.1.0 0.0.0.255
 network 172.168.2.0 0.0.0.255
 auto-summary
```

EIGRP configuration

Now we know the essential commands for configuration. Let's implement them in our network.

Router0

```
Router(config)#router eigrp 20
Router(config-router)#network 10.0.0.0 0.0.0.255
Router(config-router)#network 192.168.1.244 0.0.0.3
```

```
Router(config-router)#network 192.168.1.0 0.0.0.3  
Router(config-router)#{}
```

Router1

```
Router(config)#router eigrp 20  
Router(config-router)#network 192.168.1.244 0.0.0.3  
Router(config-router)#{  
%DUAL-5-NBRCHANGE: IP-EIGRP 20: Neighbor 192.168.1.245 (Serial0/0/0) is up: new adjacency  
Router(config-router)#network 192.168.1.248 0.0.0.3  
Router(config-router)#{}
```

Router2

```
Router(config)#router eigrp 20  
Router(config-router)#network 192.168.1.248 0.0.0.3  
Router(config-router)#{  
%DUAL-5-NBRCHANGE: IP-EIGRP 20: Neighbor 192.168.1.249 (Serial0/0/1) is up: new adjacency  
Router(config-router)#network 192.168.1.252 0.0.0.3  
Router(config-router)#{}
```

As I mentioned earlier, we can use both wildcard mask and subnet mask with network command. We have used wildcard mask for above routers. In remaining routers we will use subnet mask.

Router5

```
Router(config)#router eigrp 20  
Router(config-router)#network 20.0.0.0 255.0.0.0  
Router(config-router)#network 192.168.1.252 255.255.255.252  
Router(config-router)#{  
%DUAL-5-NBRCHANGE: IP-EIGRP 20: Neighbor 192.168.1.253 (Serial0/0/0) is up: new adjacency  
Router(config-router)#network 192.168.1.8 255.255.255.252  
Router(config-router)#{}
```

Router3

```
Router(config)#router eigrp 20  
Router(config-router)#network 192.168.1.8 255.255.255.252  
Router(config-router)#{  
%DUAL-5-NBRCHANGE: IP-EIGRP 20: Neighbor 192.168.1.10 (FastEthernet0/1) is up: new adjacency  
Router(config-router)#network 192.168.1.4 255.255.255.252  
Router(config-router)#{}
```

Router4

```
Router(config)#router eigrp 20
Router(config-router)#network 192.168.1.4 255.255.255.252
Router(config-router)#
%DUAL-5-NBRCHANGE: IP-EIGRP 20: Neighbor 192.168.1.6 (FastEthernet0/0) is up: new
adjacency
Router(config-router)#network 192.168.1.0 255.255.255.252
Router(config-router)#
%DUAL-5-NBRCHANGE: IP-EIGRP 20: Neighbor 192.168.1.1 (FastEthernet0/1) is up: new
adjacency
Router(config-router)#

```

That's it. Our network is ready to take the advantage of EIGRP routing. To verify the setup we will use ping command. **ping** command is used to test the connectivity between two devices. We have two routes between source and destination. **tracert** command is used to know the route which is used to get the destination.

Access the command prompt of PC1 and use ping command to test the connectivity from Server0. After that use **tracert** command to print the taken path.

The screenshot shows a 'Command Prompt' window within the Packet Tracer interface. The window title is 'Command Prompt'. The content of the window is as follows:

```
Packet Tracer PC Command Line 1.0
PC>ping 20.0.0.2

Pinging 20.0.0.2 with 32 bytes of data:

Request timed out.
Reply from 20.0.0.2: bytes=32 time=11ms TTL=124
Reply from 20.0.0.2: bytes=32 time=12ms TTL=124
Reply from 20.0.0.2: bytes=32 time=11ms TTL=124

Ping statistics for 20.0.0.2:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
        Minimum = 11ms, Maximum = 12ms, Average = 11ms

PC>tracert 20.0.0.2

Tracing route to 20.0.0.2 over a maximum of 30 hops:

  1  356 ms      0 ms      0 ms      10.0.0.1
  2  0 ms       0 ms      0 ms      192.168.1.2
  3  0 ms       0 ms      0 ms      192.168.1.6
  4  0 ms       0 ms      1 ms      192.168.1.10
  5  12 ms     11 ms     11 ms      20.0.0.2

Trace complete.
```

Good going we have successfully implemented EIGRP routing protocol in our network. For cross check we have uploaded a configured topology on our server. You can use this if not getting same output.

EIGRP protocol automatically manages all routes for us. If one route goes down, it will automatically switch to another available route. To explain this process more clearly we have added one additional route in our network.

Currently there are two routes between PC0 and Server.

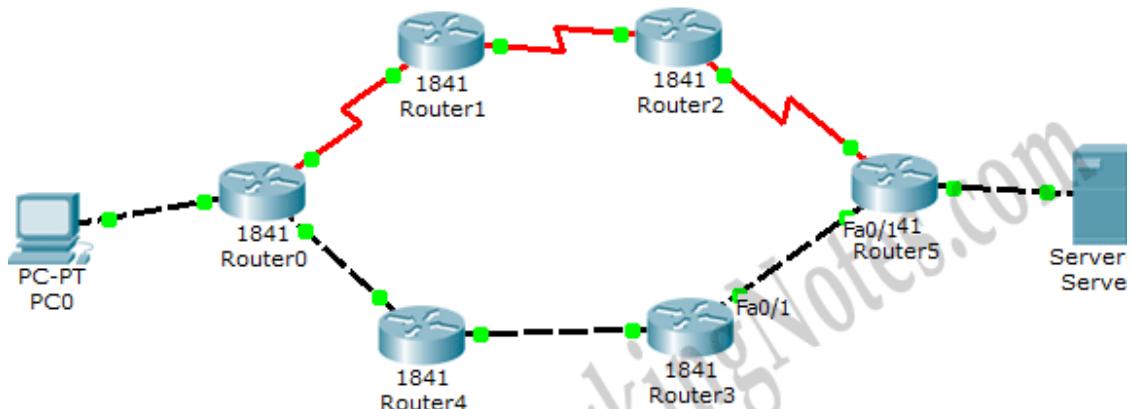
Route 1

PC0 <==> Router0 <==> Router4 <==> Router3 <==> Router5 <==> Server0

Route 2

PC0 <==> Router0 <==> Router1 <==> Router2 <==> Router5 <==> Server0

By default EIGRP uses the route that has low metric value. Our path separates from Router0, so let's see which route it takes to deliver the packet of 20.0.0.0 network. **show ip route eigrp** command will list all available routes.



Router# show ip route eigrp

```
D 20.0.0.0/8 [90/35840] via 192.168.1.2, 00:01:01, FastEthernet0/1
  192.168.1.0/24 is variably subnetted, 7 subnets, 2 masks
  D 192.168.1.0/24 is a summary, 00:01:02, Null0
  D 192.168.1.4/30 [90/30720] via 192.168.1.2, 00:01:01, FastEthernet0/1
  D 192.168.1.8/30 [90/33280] via 192.168.1.2, 00:01:01, FastEthernet0/1
  D 192.168.1.248/30 [90/2689536] via 192.168.1.2, 00:00:55, FastEthernet0/1
D 192.168.1.252/30 [90/2177536] via 192.168.1.2, 00:01:01, FastEthernet0/1
```

Output of show ip route eigrp Explained

D: - It indicates that route is learned by EIGRP. Cisco chose letter D for EIGRP, because letter E was already taken by Exterior Gateway Protocol (EGP).

20.0.0.0/8: - It is our destination network.

90: - Administrative distance of EIGRP.

35840: - Is the metric value of this route calculated by EIGRP

Via 192.168.1.2: - IP address of the next hop.

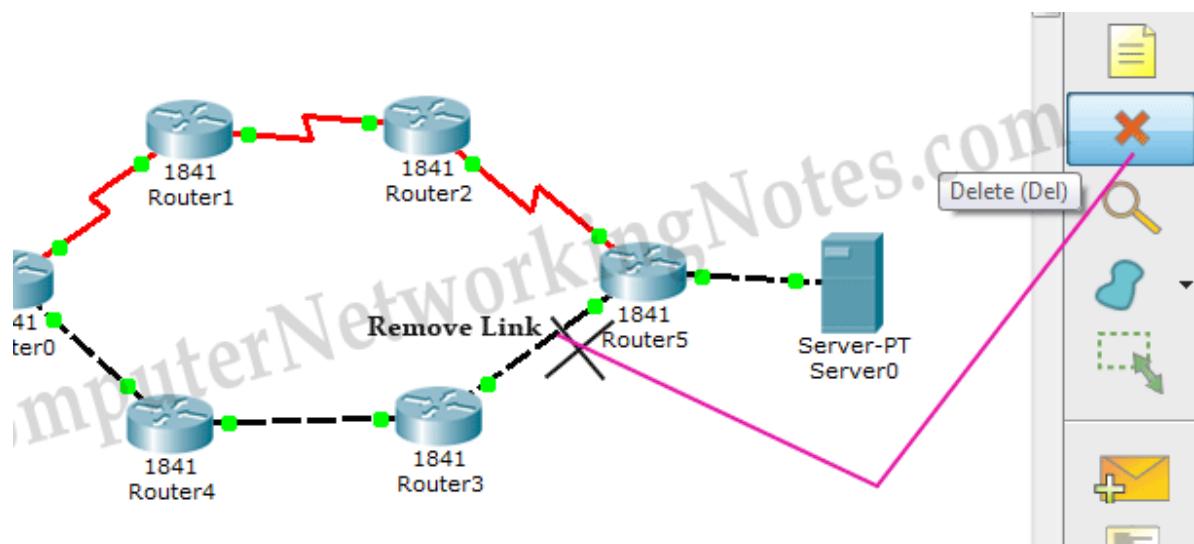
00:01:01: - How long this route was learned (Age of route)

FastEthernet1: - Exit interface of this router to get the next hop.

You may wonder where Route2 is in this output. Well EIGRP puts only the best route in routing table. Route2's metric value is higher than Route1. Till route1 is available, it will not insert route2 in routing table. When route1 is down, it will look for next possible route. If other routes are available, it will replace current route with new route which has the lowest metric value. We can watch this process live with **debug eigrp fsm** command. On debug process on Router0.

```
Router# debug eigrp fsm
```

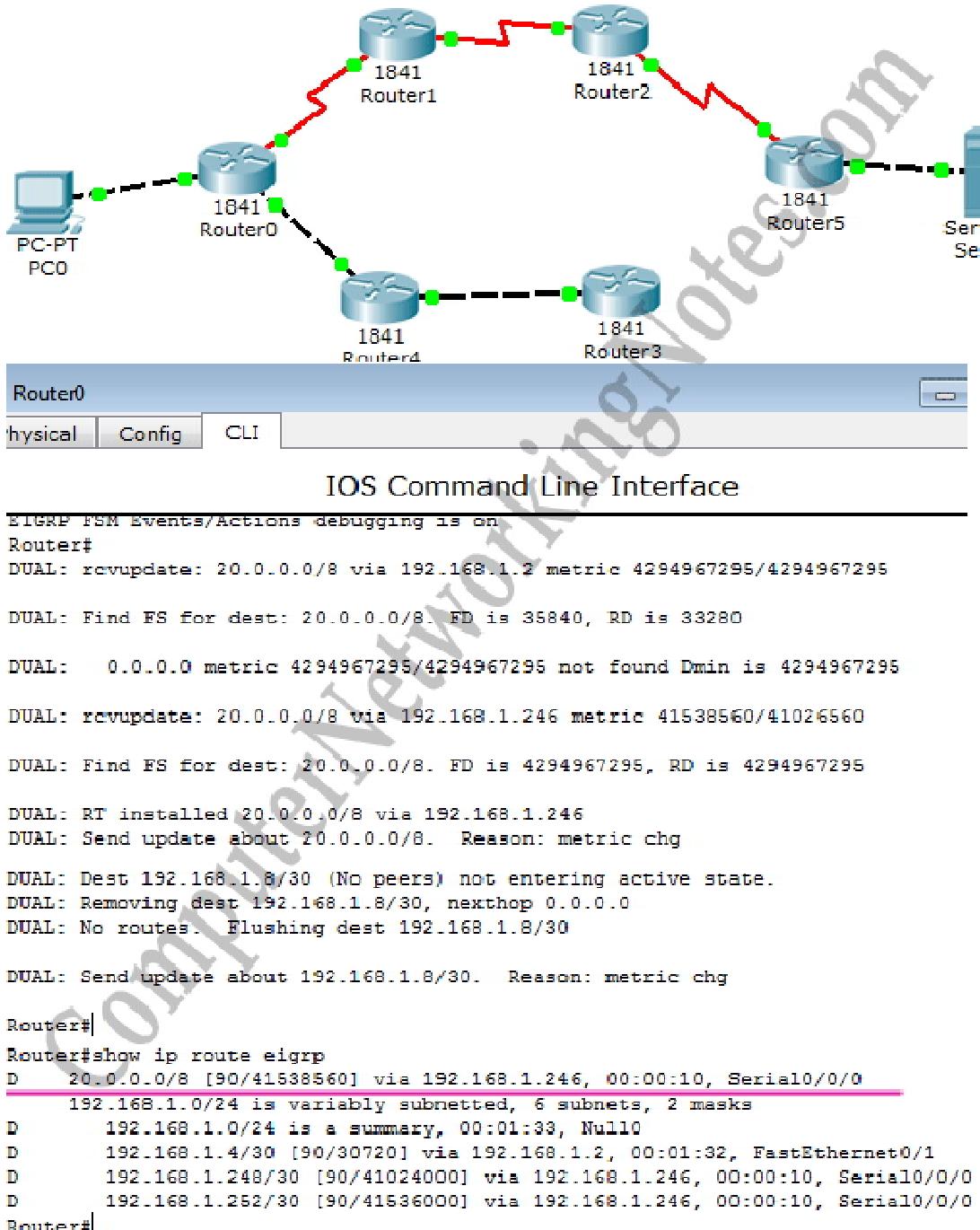
Now suppose route1 is down. We can simulate this situation by removing the cable attached between Router3 [Fa0/1] and Router5 [Fa0/1].



Okay our primary route went down. What will happen now?

EIGRP will look in topology table for next available routes. If single alternative is available, it will be selected. If multiple routes are available, it will select the route with the lowest metric value.

We can use **show ip route eigrp** command again to see the selected route.



Run **tracert** command again from PC0 to verify the change.

```

PC0
Physical Config Desktop Custom Interface

Command Prompt
Minimum = 11ms, Maximum = 12ms, Average = 11ms

PC>tracert 20.0.0.2

Tracing route to 20.0.0.2 over a maximum of 30 hops:

 1  356 ms    0 ms    0 ms    10.0.0.1
 2  0 ms      0 ms    0 ms    192.168.1.2
 3  0 ms      0 ms    0 ms    192.168.1.6
 4  0 ms      0 ms    1 ms    192.168.1.10
 5  12 ms     11 ms   11 ms   20.0.0.2

Trace complete.

PC>tracert 20.0.0.2

Tracing route to 20.0.0.2 over a maximum of 30 hops:

 1  1 ms      0 ms    0 ms    10.0.0.1
 2  0 ms      3 ms    1 ms    192.168.1.246
 3  4 ms      0 ms    3 ms    192.168.1.250
 4  0 ms      1 ms    2 ms    192.168.1.254
 5  2 ms      3 ms    10 ms   20.0.0.2

Trace complete.

PC>

```

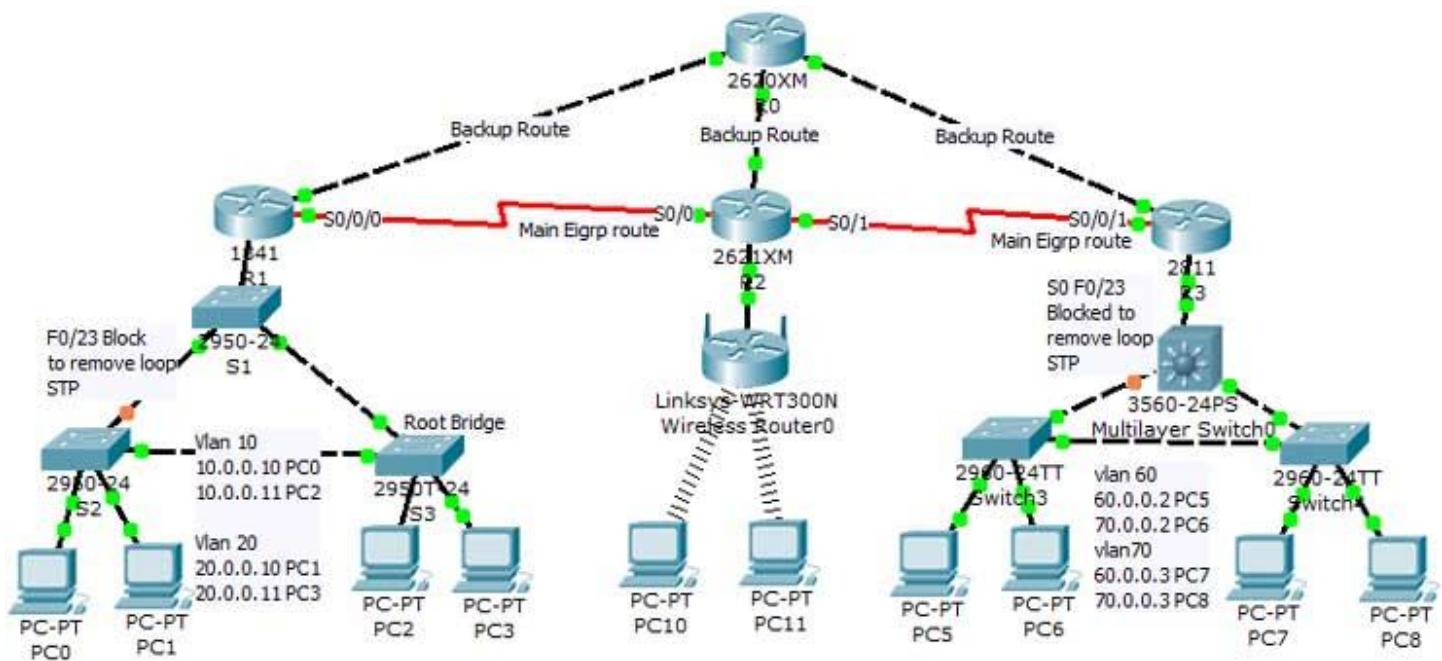
That's all for this article. Before closing just do a quick recap of important commands.

EIGRP configuration commands cheat sheet

Command	Description
Router(config)#router eigrp 20	Enable EIGRP with AS number 20. AS number must be same on all routers to become EIGRP neighbor.
Router(config-router)#network 10.10.0.0	Enable EIGRP on interfaces which belongs to network 10.0.0.0/8. [Classful implementation].
Router(config-router)#network 10.10.0.0 0.0.255.255	Enable EIGRP on interfaces which belongs to network 10.10.0.0/16. [Classless implementation – Wildcard mask method].
Router(config-router)#network 10.10.0.0 255.255.0.0	Enable EIGRP on interfaces which belongs to network 10.10.0.0/16. [Classless implementation – Subnet mask method].
Router(config-router)#no network 10.10.0.0	Disable EIGRP on interfaces which belongs to network 10.0.0.0/8.

Router(config-router)#no network 10.10.0.0 0.0.255.255	Disable EIGRP on interfaces which belongs to network 10.10.0.0/16.
Router(config-router)#no network 10.10.0.0 255.255.0.0	Disable EIGRP on interfaces which belongs to network 10.10.0.0/16.
Router(config-router) #metric weights tos k1 k2 k3 k4 k5	Enable/Disable K values used in metric calculation formula. Default values are tos=0, k1=1, k2=0, k3=1, k4=0, k5=0 Tos(type of service), K1(bandwidth), K2(load), K3(delay), K4(reliability), K5(MTU). By default only K1 and K3 are enabled.
Router(config-router)#auto-summary	Enable auto summarization feature of EIGRP. (Default – disable)
Router(config-router)#no auto-summary	Disable auto summarization feature of EIGRP.
Router(config)#no router eigrp 20	Disable EIGRP routing process 20.
Router(config-if)#bandwidth 64	Set bandwidth to 64Kbps. Used to influence the metric calculation.
Router#show ip eigrp neighbors	Display the neighbor table in brief.
Router#show ip eigrp neighbors detail	Display the neighbor table in detail. Used to verify whether a neighbor is configured as stub router or not.
Router#show ip eigrp interfaces	Display information about all EIGRP interfaces.
Router#show ip eigrp interfaces serial 0/0	Display information about a particular EIGRP interface.
Router#show ip eigrp interfaces 20	Display information about EIGRP interfaces running AS process 20.
Router#show ip eigrp topology	Displays the topology table.
Router#show ip eigrp traffic	Displays the number and type of packets sent and received.
Router#show ip route eigrp	Display EIGRP route from routing table.
Router#debug eigrp fsm	Displays the events or actions related to feasible successor metrics (FSM).
Router#debug eigrp packet	Displays the events or actions related to EIGRP packets.
Router#no debug eigrp fsm	Turn off debug message related to feasible successor metrics (FSM).
Router#no debug eigrp packet	Turn off debug message related to EIGRP packets.

Configure EIGRP with RIP on same network



R0		
Port	IP address	Connected to
F0/0	80.0.0.1	R1 F0/1
F1/0	90.0.0.1	R2 F0/1
F1/1	100.0.0.1	R3 F0/1

R1		
Port	IP address	Connected to
F0/0.10	10.0.0.1	S1 F0/24
F0/0.20	20.0.0.1	S1 F0/24
F0/1	80.0.0.2	R0 F0/0
S0/0/0	30.0.0.1	R2 S0/0

R2		
Port	IP address	Connected to
F0/1	90.0.0.2	R0 F1/0
S0/0	30.0.0.2	R1 S0/0/0
F0/0	40.0.0.1	WR1 0/1
S0/1	50.0.0.1	R3 S0/0/1

R3		
Port	IP address	Connected to
F0/1	100.0.0.2	R0 F1/1
S0/0/1	50.0.0.2	R2 S0/1
F0/0.60	60.0.0.1	S1 G0/1
F0/0.70	70.0.0.1	S1 G0/1

Configuration of R0

First we will configure R0. To configure double click on R0 select CLI and configure it as given below

To configure and enable RIP as backup routing on R0 follow these commands exactly.

R0>enable

R0#sh ip interface brief

Interface	IP-Address	OK?	Method	Status	Protocol
-----------	------------	-----	--------	--------	----------

FastEthernet0/0	80.0.0.1	YES	manual	up	
-----------------	----------	-----	--------	----	--

FastEthernet1/0	90.0.0.1	YES	manual	up	
-----------------	----------	-----	--------	----	--

FastEthernet1/1	100.0.0.1	YES	manual	up	
-----------------	-----------	-----	--------	----	--

R0#configure terminal

Enter configuration commands, one per line. End with CNTL/Z.

R0(config)#router rip

R0(config-router)#network 80.0.0.0

R0(config-router)#network 90.0.0.0

R0(config-router)#network 100.0.0.0

R0(config-router)#exit

R0(config)#exit

%SYS-5-CONFIG_I: Configured from console by console

R0#copy run start

Destination filename [startup-config]?

Building configuration...

[OK]

R0#

We need not to configure EIGRP on it as its only going to be a backup route

Configuration of R1

Now configure R1. On R1 we need to configure both RIP and EIGRP. RIP for backup and EIGRP for main route.

R1>enable

R1#show ip interface brief

Interface	IP-Address	OK?	Method	Status	Protocol
-----------	------------	-----	--------	--------	----------

FastEthernet0/0	unassigned	YES manual	up
FastEthernet0/0.10	10.0.0.1	YES manual	up
FastEthernet0/0.20	20.0.0.1	YES manual	up
FastEthernet0/1	80.0.0.2	YES manual	up
Serial0/0/0	30.0.0.1	YES manual	up
Serial0/0/1	unassigned	YES manual	administratively down
Vlan1	unassigned	YES manual	administratively down

```

R1#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#router rip
R1(config-router)#network 10.0.0.0
R1(config-router)#network 20.0.0.0
R1(config-router)#network 30.0.0.0
R1(config-router)#network 80.0.0.0
R1(config-router)#exit
R1(config)#router eigrp 1
R1(config-router)#network 10.0.0.0
R1(config-router)#network 20.0.0.0
R1(config-router)#network 30.0.0.0
R1(config-router)#exit
R1(config)#exit
%SYS-5-CONFIG_I: Configured from console by console
R1#copy run start
Destination filename [startup-config]?
Building configuration...
[OK]

```

R1#

Why did we not include network 80.0.0.0 in EIGRP routing?

Network 80.0.0.0 is connecting R1 to R0 which is backup route. And only going to be used in the failure of main EIGRP route.

If we include network 80.0.0.0 in EIGRP routing, still R1 will take main EIGRP (via R2) route to reach R3 why?

Because by default Matrix of EIGRP is bandwidth and delay, and in this scenario backup route (R1 – R0 – R3) is using Ethernet cable link and main EIGRP route (R1 – R2 – R3) is using serial cable link. Serial link have better matrix than Ethernet link that's why R1 will take main EIGRP route to reach R3.

Configuration of R2

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

Router>enable

R2#show ip interface brief

Interface	IP-Address	OK? Method Status	Protocol
FastEthernet0/0	40.0.0.1	YES manual up	up
FastEthernet0/1	90.0.0.2	YES manual up	up
Serial0/0	30.0.0.2	YES manual up	up
Serial0/1	50.0.0.1	YES manual up	up

R2#configure terminal

Enter configuration commands, one per line. End with CNTL/Z.

R2(config)#router rip

R2(config-router)#network 30.0.0.0

R2(config-router)#network 40.0.0.0

R2(config-router)#network 50.0.0.0

R2(config-router)#network 90.0.0.0

R2(config-router)#exit

R2(config)#router eigrp 1

R2(config-router)#network 30.0.0.0

```

R2(config-router)#
R2(config-router)#network 40.0.0.0
R2(config-router)#network 50.0.0.0
R2(config-router)#exit
R2(config)#exit
%SYS-5-CONFIG_I: Configured from console by console
R2#

```

Configuration of R3

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

Router>enable

R3#show ip interface brief

Interface	IP-Address	OK?	Method	Status	Protocol
FastEthernet0/0	unassigned	YES	manual	up	
FastEthernet0/0.60	60.0.0.1	YES	manual	up	
FastEthernet0/0.70	70.0.0.1	YES	manual	up	
FastEthernet0/1	100.0.0.2	YES	manual	up	
Serial0/0/0	unassigned	YES	manual	administratively down	down
Serial0/0/1	50.0.0.2	YES	manual	up	
Vlan1	unassigned	YES	manual	administratively down	down

R3#configure terminal

Enter configuration commands, one per line. End with CNTL/Z.

R3(config)#router rip

R3(config-router)#network 50.0.0.0

R3(config-router)#network 60.0.0.0

R3(config-router)#network 70.0.0.0

R3(config-router)#network 100.0.0.0

R3(config-router)#exit

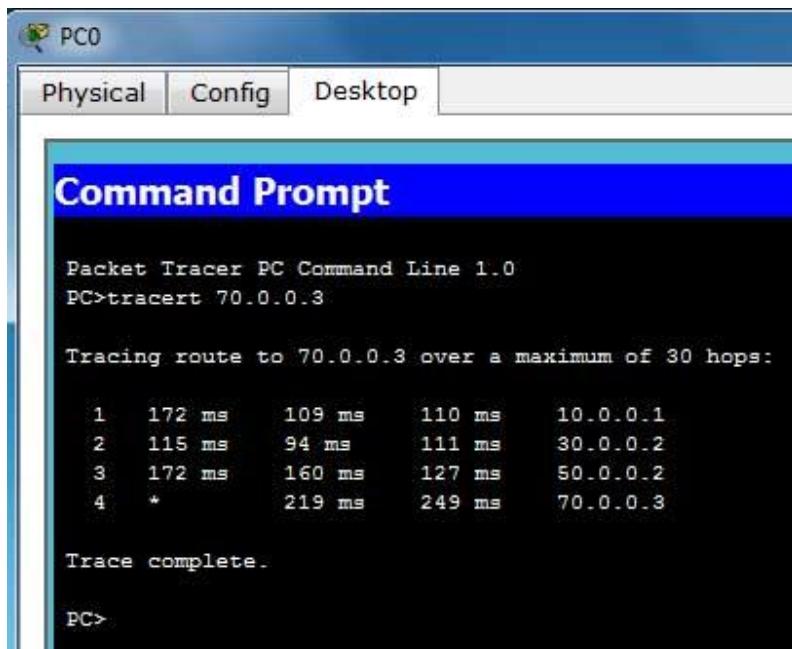
```

R3(config)#router eigrp 1
R3(config-router)#network 50.0.0.0
R3(config-router)#network
R3(config-router)#network 60.0.0.0
R3(config-router)#network 70.0.0.0
R3(config-router)#exit
R3(config)#exit
%SYS-5-CONFIG_I: Configured from console by console
R3#

```

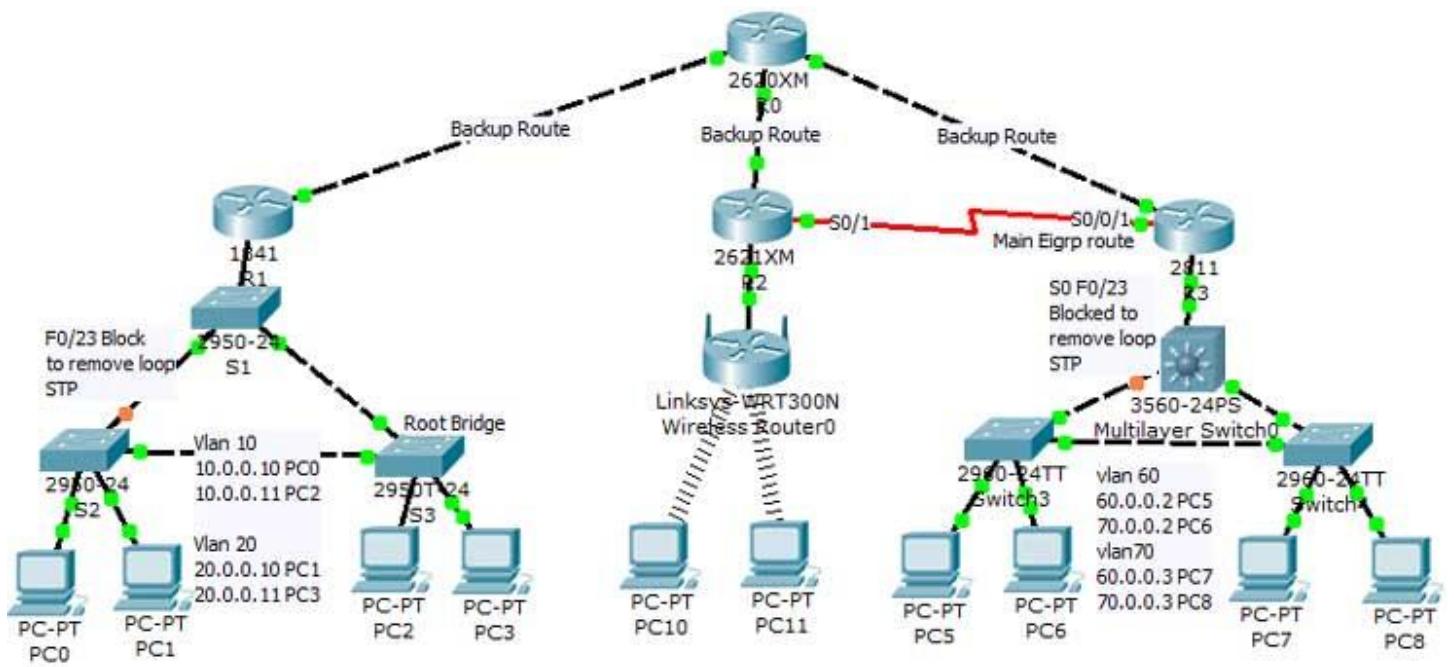
Testing of EIGRP with RIP

Now we have configured both RIP and EIGRP in this network. To test this network double click on PC-PT PC0 and select command prompt tracert 70.0.0.3



As you can see image R1 is taking Main EIGRP route to rich at R3

Now remove the serial cable by using Red Cross sign shown in right control panel



Now again tracert 70.0.0.3 from PC0

```

PC>tracert 70.0.0.3

Tracing route to 70.0.0.3 over a maximum of 30 hops:
  1  109 ms    64 ms    124 ms   10.0.0.1
  2  124 ms    109 ms    140 ms   30.0.0.2
  3  140 ms    187 ms    187 ms   50.0.0.2
  4  220 ms    265 ms    203 ms   70.0.0.3

Trace complete.

PC>tracert 70.0.0.3

Tracing route to 70.0.0.3 over a maximum of 30 hops:
  1  125 ms    94 ms    78 ms   10.0.0.1
  2  125 ms    *        156 ms   80.0.0.1
  3  156 ms    156 ms    187 ms   100.0.0.2
  4  234 ms    155 ms    265 ms   70.0.0.3

Trace complete.

PC>

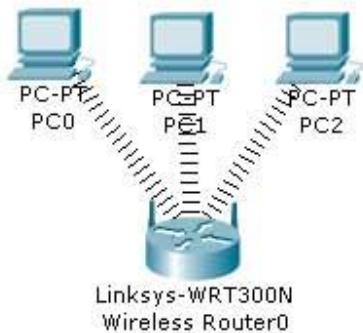
```

As you can see in image this time R1 takes R0 to reach R3 because Main EIGRP route is down.

How to Configure Wireless Network

This assignment explains how to configure wireless network in packet tracer step by step including how to enable static routing in Linksys router.

We have covered all CCNA exam topics in our wireless networking section. As its new topic in CCNA exam so you have very little chance to get simulated base question on wireless networking. Still we recommended you to get familiar with some basic wireless configuration.



In this topology we have three pc connected with Linksys Wireless routers.

- DHCP is configured and enabled on Wireless router
- IP pool for DHCP is 192.168.0.100 to 192.168.0.150
- PC are configured to receive IP from DHCP Server
- No security is configured
- Default SSID is configured to Default
- Topology is working on infrastructure mode
- Default user name and password is admin
- IP of wireless is set to 192.168.0.1

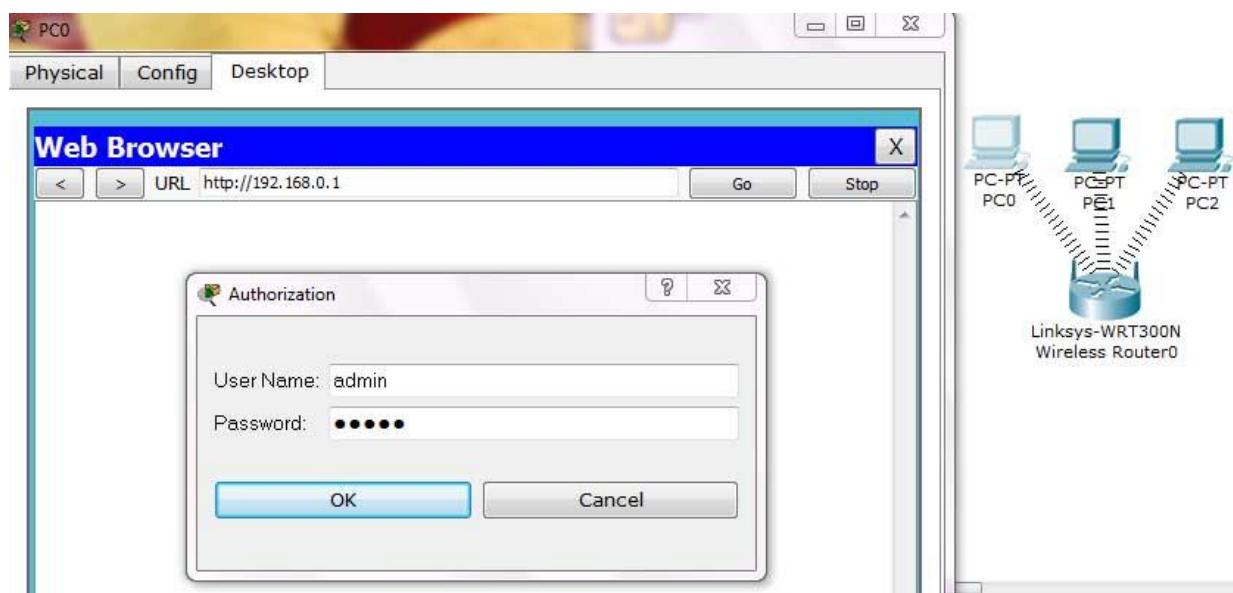
Now your task is to:-

- Configure Static IP on PC and Wireless Router
- Change SSID to MotherNetwork
- Change IP address of router to 10.0.0.1 and 10.0.0.2 of PC0 10.0.0.3 of PC1 10.0.0.4 of PC2
- Secure your network by configuring WAP key on Router
- Connect PC by using WAP key

To complete these tasks follow this step by step guide of how to configure wireless network

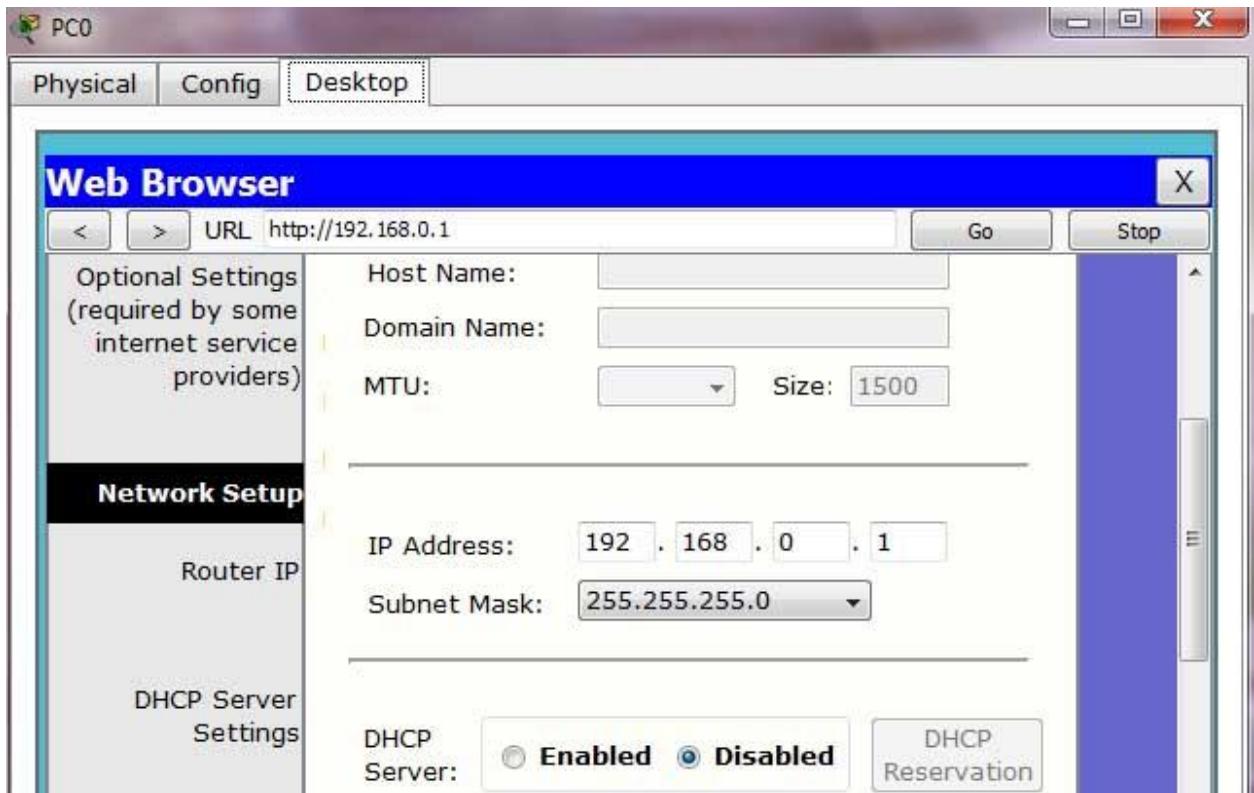
As given in question our network is running on 192.168.0.0 network and all PC's are DHCP clients and functioning properly. So we will first connect to Wireless router to off DHCP.

Double click on PC and select Web Browser. As given in question IP of Wireless router is 192.168.0.1 so give it in Web browser and press enter, now it will ask for authentication which is also given in question. Give user name admin and Password to admin



This will bring GUI mode of Wireless router. Scroll down screen to Network Step and Select Disable

DHCP



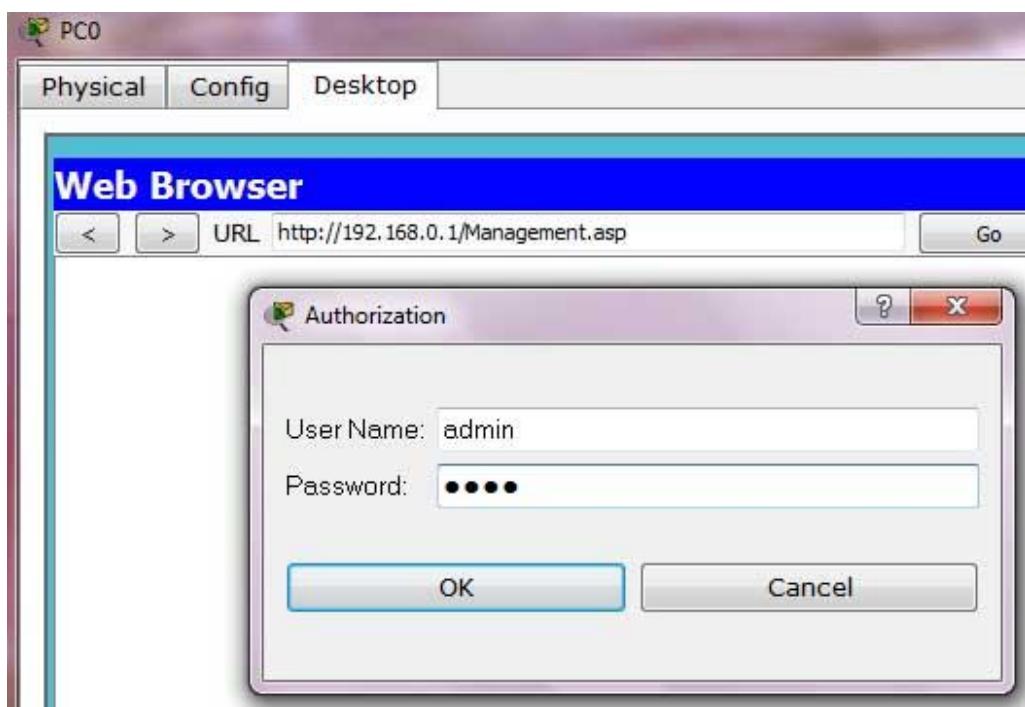
Go in end of page and click on Save setting this will save setting click on continue for further setting



Now select Administration from top Manu and change password to test and go in the end of page and Click on Save Setting



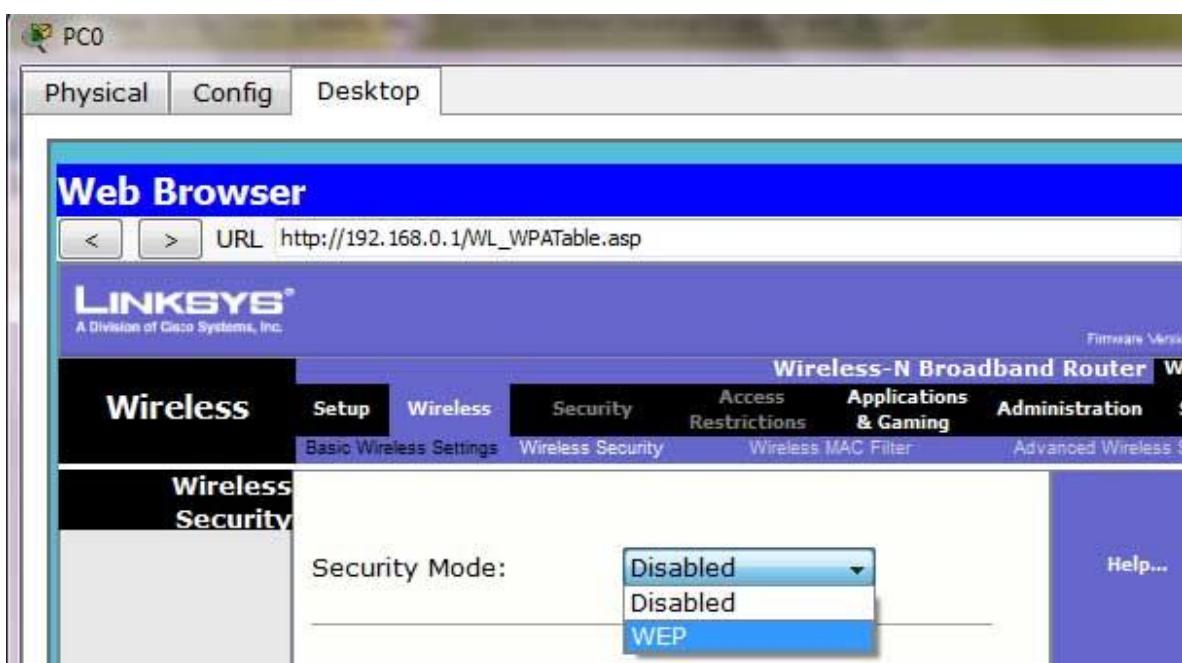
Click on continue for further setting. This time it will ask you to authenticate again give new password test this time



Now click on wireless tab and set default SSID to MotherNetwork



Now Select wireless security and change Security Mode to WEP



Set Key1 to 0123456789



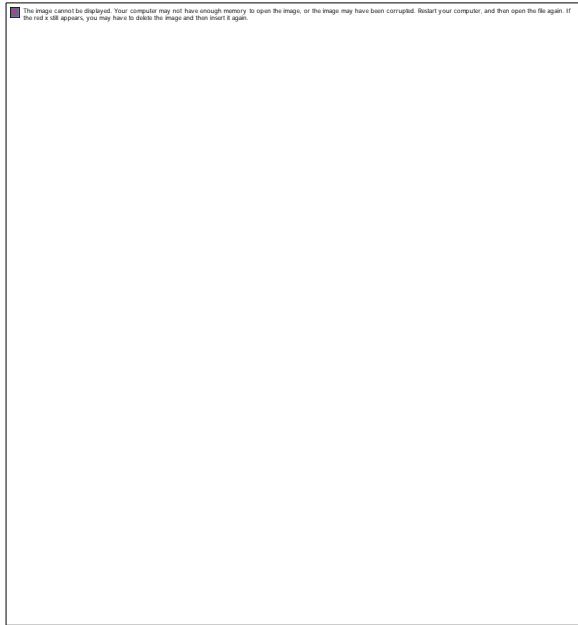
Again go in the end of page and Click on Save Setting

Now we have completed all given task on Wireless router. Now configure the static IP on all three PC's

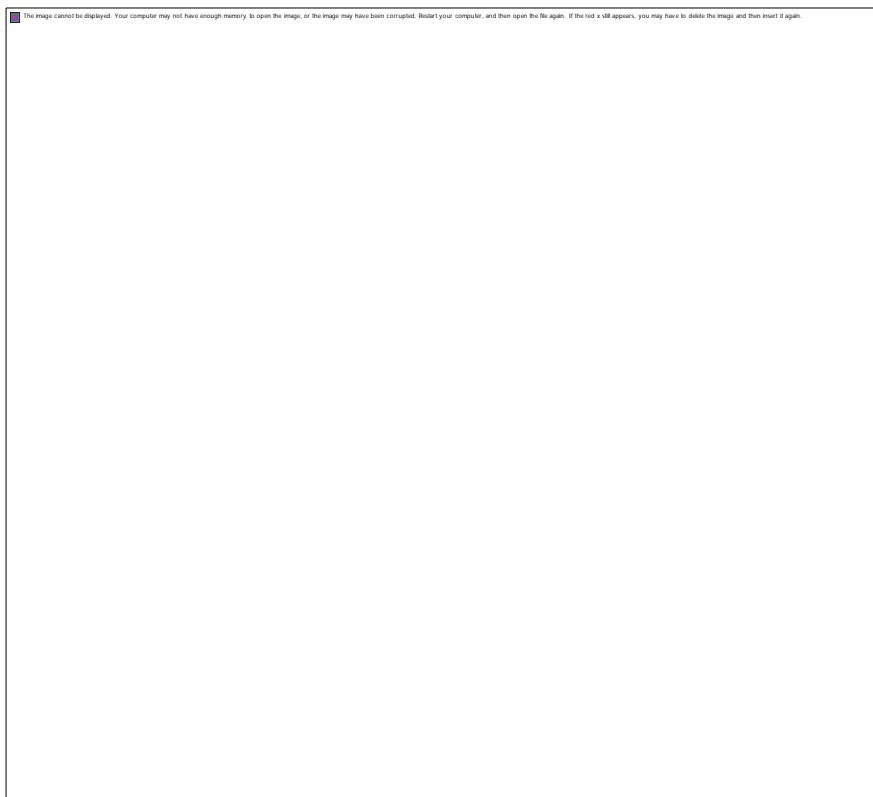
Double click on pc select Desktop tab click on IP configuration select Static IP and set IP as given below

PC	IP	Subnet Mask	Default Gateway
PC0	192.168.0.2	255.255.255.0	192.168.0.1
PC1	192.168.0.3	255.255.255.0	192.168.0.1
PC2	192.168.0.4	255.255.255.0	192.168.0.1

Now it's time to connect PC's from Wireless router. To do so click PC select Desktop click on PC Wireless



Click on connect tab and click on Refresh button



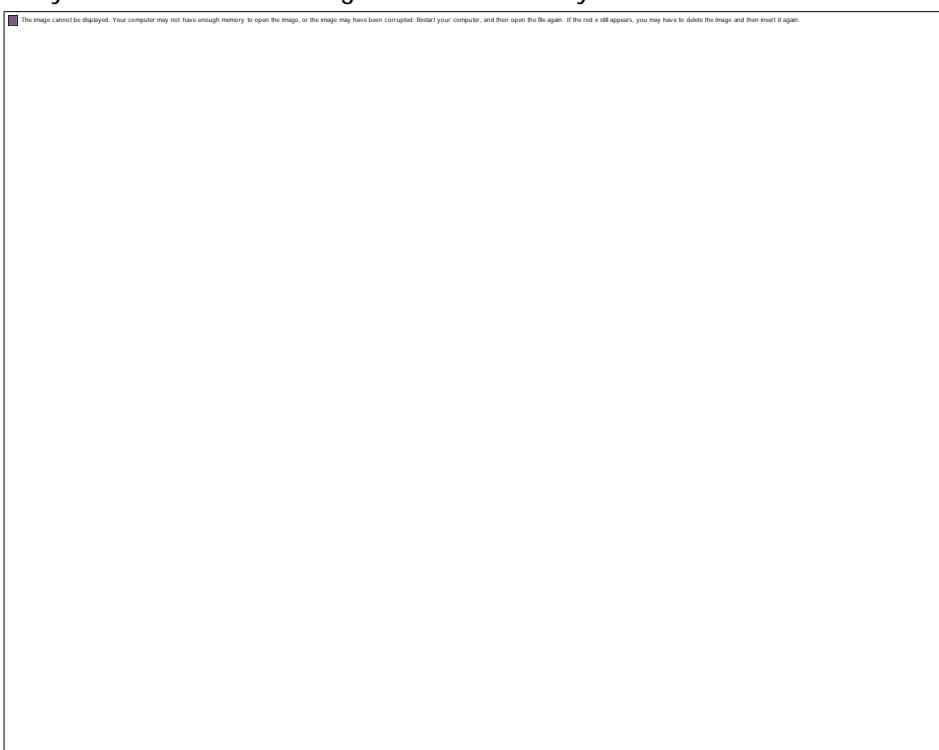
As you can see in image that Wireless device is accessing Mother Network on CH 6 and signal strength is 100%. In left side you can see that WEP security is configured in network. Click on connect button to connect Mother Network

It will ask for WAP key insert 0123456789 and click connect



It will connect you with wireless router.

As you can see in image below that system is connected. And PCI card is active.



Repeat same process on PC1 and PC2.

VLAN

VLAN Basic Concepts

This part explains VLAN basic concepts such as what is VLAN, advantage of VLAN, VLAN membership static and dynamic, VLAN connections and trunk tagging in detail with examples.

What is VLAN

VLAN is a logical grouping of networking devices. When we create VLAN, we actually break large broadcast domain in smaller broadcast domains. Consider VLAN as a subnet. Same as two different subnets cannot communicate with each other without router, different VLANs also requires router to communicate.

Advantage of VLAN

VLAN provides following advantages:-

- Solve broadcast problem
- Reduce the size of broadcast domains
- Allow us to add additional layer of security
- Make device management easier
- Allow us to implement the logical grouping of devices by function instead of location

Solve broadcast problem

When we connect devices into the switch ports, switch creates separate collision domain for each port and single broadcast domain for all ports. Switch forwards a broadcast frame from all possible ports. In a large network having hundreds of computers, it could create performance issue. Of course we could use routers to solve broadcast problem, but that would be costly solution since each broadcast domain requires its own port on router. Switch has a unique solution to broadcast issue known as VLAN. In practical environment we use VLAN to solve broadcast issue instead of router.

Each VLAN has a separate broadcast domain. Logically VLANs are also subnets. Each VLAN requires a unique network number known as VLAN ID. Devices with same VLAN ID are the members of same broadcast domain and receive all broadcasts. These broadcasts are filtered from all ports on a switch that aren't members of the same VLAN.

Reduce the size of broadcast domains

VLAN increase the numbers of broadcast domain while reducing their size. For example we have a network of 100 devices. Without any VLAN implementation we have single broadcast domain that contain 100 devices. We create 2 VLANs and assign 50 devices in each VLAN. Now we have two broadcast domains with fifty devices in each. Thus more VLAN means more broadcast domain with less devices.

Allow us to add additional layer of security

VLANs enhance the network security. In a typical layer 2 network, all users can see all devices by default. Any user can see network broadcast and responds to it. Users can access any network resources located on that specific network. Users could join a workgroup by just attaching their system in existing switch. This could create real trouble on security platform. Properly configured VLANs gives us total control over each port and users. With VLANs, you can control the users from gaining unwanted access over the resources. We can put the group of users that need high level security into their own VLAN so that users outside from VLAN can't communicate with them.

Make device management easier

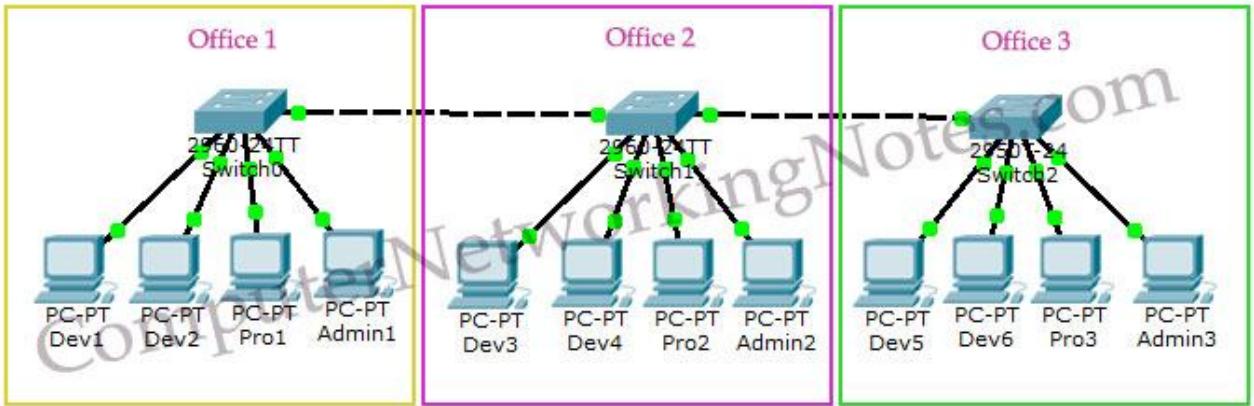
Device management is easier with VLANs. Since VLANs are a logical approach, a device can be located anywhere in the switched network and still belong to the same broadcast domain. We can move a user from one switch to another switch in same network while keeping his original VLAN. For example our company has a five story building and a single layer two network. In this scenario, VLAN allows us to move the users from one floor to another floor while keeping his original VLAN ID. The only limitation we have is that device when moved, must still be connected to the same layer 2 network.

Allow us to implement the logical grouping of devices by function instead of location

VLANs allow us to group the users by their function instead of their geographic locations. Switches maintain the integrity of your VLANs. Users will see only what they are supposed to see regardless what their physical locations are.

VLAN Examples

To understand VLAN more clearly let's take an example.



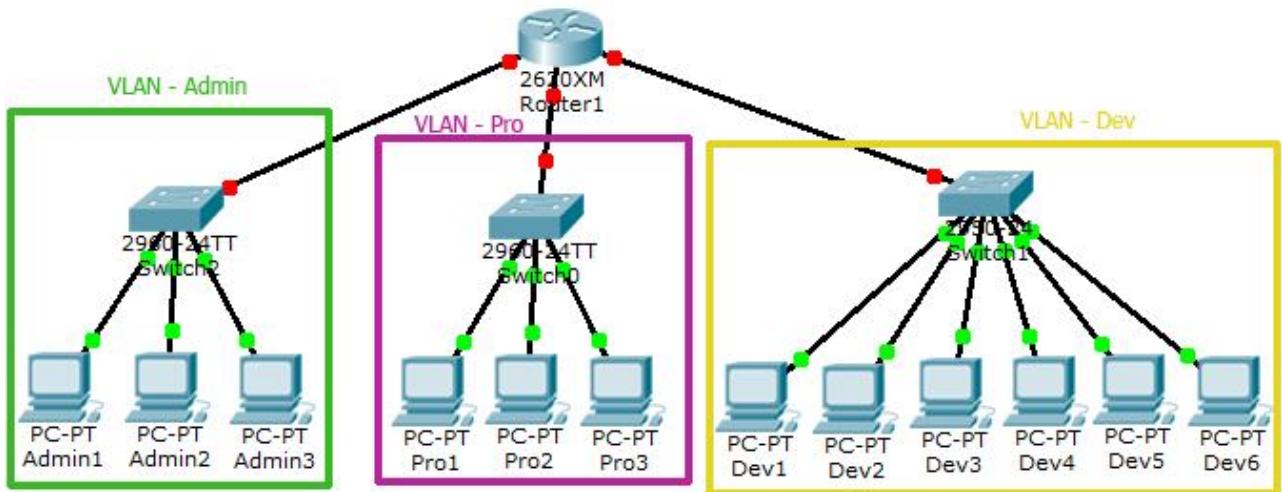
- Our company has three offices.
- All offices are connected with back links.
- Company has three departments Development, Production and Administration.
- Development department has six computers.
- Production department has three computers.
- Administration department also has three computers.
- Each office has two PCs from development department and one from both production and administration department.
- Administration and production department have sensitive information and need to be separate from development department.

With default configuration, all computers share same broadcast domain. Development department can access the administration or production department resources.

With VLAN we could create logical boundaries over the physical network. Assume that we created three VLANs for our network and assigned them to the related computers.

- **VLAN Admin** for Administration department
- **VLAN Dev** for Development department
- **VLAN Pro** for Production department

Physically we changed nothing but logically we grouped devices according to their function. These groups [VLANs] need router to communicate with each other. Logically our network look likes following diagram.



With the help of VLAN, we have separated our single network in three small networks. These networks do not share broadcast with each other improving network performance. VLAN also enhances the security. Now Development department cannot access the Administration and Production department directly. Different VLAN can communicate only via Router where we can configure wild range of security options.

So far in this article we have explained VLAN, in following section we will explain VLAN terms in more details.

VLAN Membership

VLAN membership can be assigned to a device by one of two methods

1. Static
2. Dynamic

These methods decide how a switch will associate its ports with VLANs.

Static

Assigning VLANs statically is the most common and secure method. It is pretty easy to set up and supervise. In this method we manually assign VLAN to switch port. VLANs configured in this way are usually known as port-based VLANs.

Static method is the most secure method also. As any switch port that we have assigned a VLAN will keep this association always unless we manually change it. It works really well in a networking environment where any user movement within the network needs to be controlled.

Dynamic

In dynamic method, VLANs are assigned to port automatically depending on the connected device. In this method we have configure one switch from network as a server. Server contains device specific information like MAC address, IP address etc. This information is mapped with VLAN. Switch acting as server is known as VMPS (VLAN Membership Policy Server). Only high end switch can configured as VMPS. Low end switch works as client and retrieve VLAN information from VMPS.

Dynamic VLANs supports plug and play movability. For example if we move a PC from one port to another port, new switch port will automatically be configured to the VLAN which the user belongs. In static method we have to do this process manually.

VLAN Connections

During the configuration of VLAN on port, we need to know what type of connection it has.

Switch supports two types of VLAN connection

1. Access link
2. Trunk link

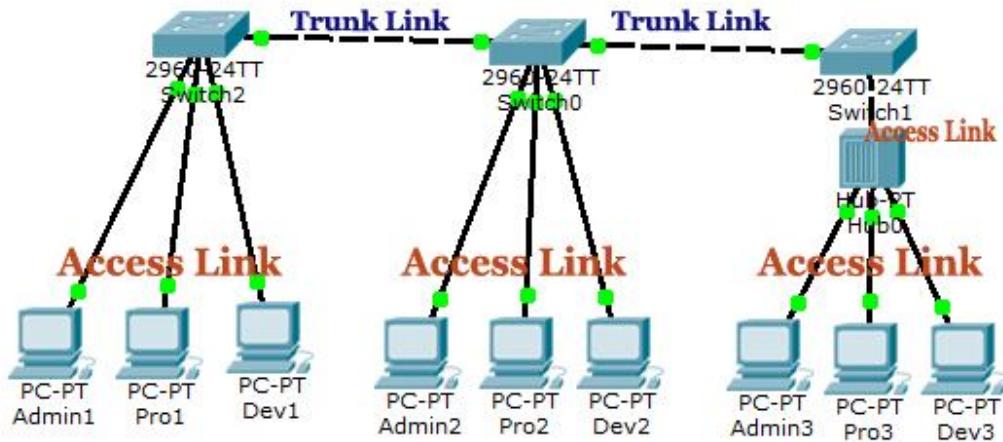
Access link

Access link connection is the connection where switch port is connected with a device that has a standardized Ethernet NIC. Standard NIC only understand IEEE 802.3 or Ethernet II frames. Access link connection can only be assigned with single VLAN. That means all devices connected to this port will be in same broadcast domain.

For example twenty users are connected to a hub, and we connect that hub with an access link port on switch, then all of these users belong to same VLAN. If we want to keep ten users in another VLAN, then we have to purchase another hub. We need to plug in those ten users in that hub and then connect it with another access link port on switch.

Trunk link

Trunk link connection is the connection where switch port is connected with a device that is capable to understand multiple VLANs. Usually trunk link connection is used to connect two switches or switch to router. Remember earlier in this article I said that VLAN can span anywhere in network, that is happen due to trunk link connection. Trunking allows us to send or receive VLAN information across the network. To support trunking, original Ethernet frame is modified to carry VLAN information.



Trunk Tagging

In trunking a separate logical connection is created for each VLAN instead of a single physical connection. In tagging switch adds the source port's VLAN identifier to the frame so that other end device can understand what VLAN originated this frame. Based on this information destination switch can make intelligent forwarding decisions on not just the destination MAC address, but also the source VLAN identifier.

Since original Ethernet frame is modified to add information, standard NICs will not understand this information and will typically drop the frame. Therefore, we need to ensure that when we set up a trunk connection on a switch's port, the device at the other end also supports the same trunking protocol and has it configured. If the device at the other end doesn't understand these modified frames it will drop them. The modification of these frames, commonly called tagging. Tagging is done in hardware by application-specific integrated circuits (ASICs).

Switch supports two types of Ethernet trunking methods:

1. ISL [Inter Switch Link, Cisco's proprietary protocol for Ethernet]
2. Dot1q [IEEE's 802.1Q, protocol for Ethernet]

Vlan Practice Lab Setup on Packet Tracer

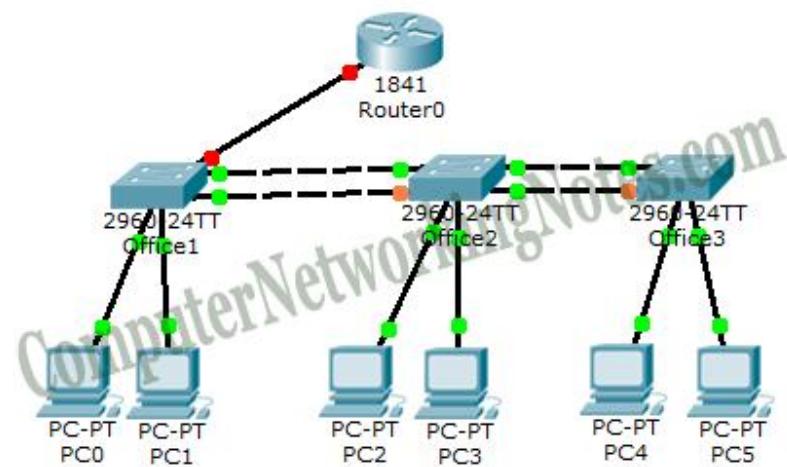
Scenario

You are a network administrator at *ComputerNetworkingNotes.com*. Company has three offices. Offices are connected with each other via layer 2 links. For redundancy purpose each office has one more layer 2 link. Company has two department sales and management. In each office we have one PC from each department. Company has one router. You can use router's Ethernet port for inter VLAN communication.

In this we will create a practical lab for the practice of VLAN, VTP, DTP, and Router on Stick.

LAB Setup

To replicate given scenario create a topology in packet tracer, as shown in following image.



Configurations used in this topology are following

PCs Configuration

Device	IP Address	Subnet Mask	Gateway	VLAN	Connected With
PC0	10.0.0.2	255.0.0.0	10.0.0.1	VLAN 10	Office 1 Switch on F0/1
PC1	20.0.0.2	255.0.0.0	20.0.0.1	VLAN 20	Office 1 Switch on F0/2
PC2	10.0.0.3	255.0.0.0	10.0.0.1	VLAN 10	Office 2 Switch on F0/1
PC3	20.0.0.3	255.0.0.0	20.0.0.1	VLAN 20	Office 2 Switch on F0/2
PC4	10.0.0.4	255.0.0.0	10.0.0.1	VLAN 10	Office 3 Switch on F0/1
PC5	20.0.0.4	255.0.0.0	20.0.0.1	VLAN 20	Office 3 Switch on F0/2

Office 1 Switch Configuration

Port	Connected To	VLAN	Link	Status
F0/1	With PC0	VLAN 10	Access	OK
F0/2	With PC1	VLAN 20	Access	OK
Gig1/1	With Router	VLAN 10,20	Trunk	OK
Gig 1/2	With Switch2	VLAN 10,20	Trunk	OK
F0/24	Witch Switch2	VLAN 10,20	Trunk	STP - Blocked

Office 2 Switch Configuration

Port	Connected To	VLAN	Link	Status
F0/1	With PC0	VLAN 10	Access	OK
F0/2	With PC1	VLAN 20	Access	OK
Gig 1/2	With Switch1	VLAN 10,20	Trunk	OK
Gig 1/1	With Switch3	VLAN 10,20	Trunk	OK
F0/24	Witch Switch1	VLAN 10,20	Trunk	STP - Blocked
F0/23	Witch Switch3	VLAN 10,20	Trunk	STP - Blocked

Office 3 Switch Configuration

Port	Connected To	VLAN	Link	Status
F0/1	With PC0	VLAN 10	Access	OK
F0/2	With PC1	VLAN 20	Access	OK
Gig 1/1	With Switch2	VLAN 10,20	Trunk	OK
F0/24	Witch Switch1	VLAN 10,20	Trunk	STP - Blocked

Router Configuration

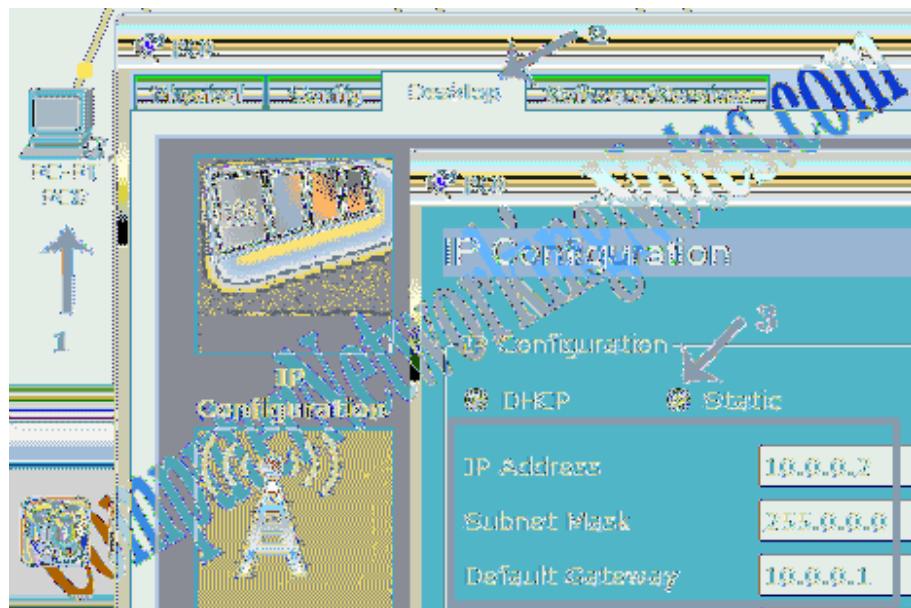
Port	Connected To	VLAN	Link	Status
Fa0/0	with Office 1 Switch Gig 1/2	VLAN 10, 20	Trunk	Ok

VLAN Configuration

VLAN Number	VLAN Name	Gateway IP	PCs
10	Sales	10.0.0.1	PC0,PC2,PC4
20	Management	20.0.0.1	PC1,PC3,PC5

Assign IP Addresses to PCs

Assigning IP addresses is bit easy task in packet tracer. Just double Click on **PC-PT** and Click **Desktop** menu item and Click **IP Configuration** Select **Static** from radio option and fill IP address, subnet mask and default gateway IP in given input boxes. Use PC Configuration table to assign correct IP address.



That's all information we need to complete this exercise. In next part of this article we will configure VLAN, VTP, STP, DTP and Router on Stick in this topology. Before you jump in next part make sure you have above topology with IP addresses configured on all PCs. You can download this initial topology with IP addresses configured on all PCs from following link.

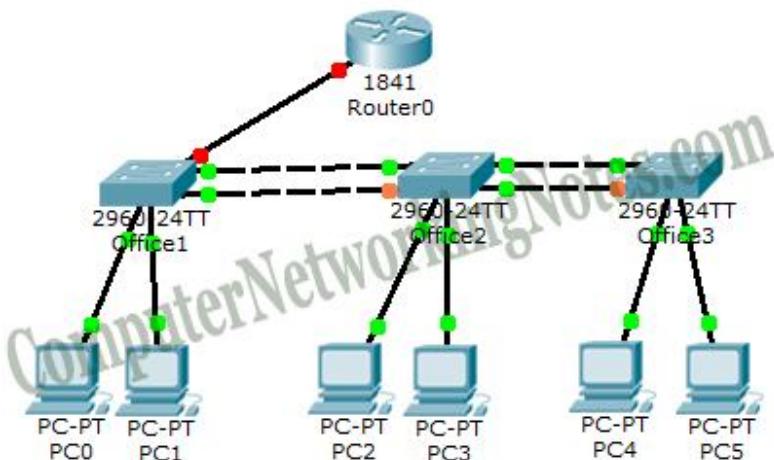
Configure VTP Server and Client In Switch

This explains how to configure VTP Server and Client in Cisco switches including basic concepts of VTP and VTP modes (Server, Transparent and Client).

VLAN Trunk Protocol (VTP) is a Cisco proprietary protocol used to share VLAN configuration across the network. Cisco created this protocol to share and synchronize their VLAN information throughout the network. Main goal of VTP is to manage all configured VLANs across the network.

Basic concepts of VTP Protocol

For this tutorial we assume that you have following topology running in packet tracer. You can create this topology by following the instruction given in second part of this article or alternatively download the pre created topology from there.



In our network we only have three switches. We can easily add or remove VLAN manually on all three switches. However this process could be more tedious and difficult if we have 50 switches. In a large network, we might make a mistake in VLAN configuration. We might forget to add VLAN on one of the switch, or we may assign wrong VLAN number. Vice versa we may forget to remove VLAN on one of the switch, while removing VLANs.

VTP is a life saver protocol in this situation. With VTP we can add or remove VLANs on one switch and this switch will propagate VLAN information to all other switches in network.

VTP Messages

VTP share VLANs information via VTP messages. VTP messages can only be propagate through the trunk connections. So we need to set up trunk connection between switches. VTP messages are propagated as layer 2 multicast frames.

VTP Domain

VTP domain is a group of switches that share same VLAN information. A switch can have a single domain. VTP messages include domain name. Switch only update VLAN information if it receive VTP message from same domain.

VTP Mode

VTP can be configured in three different modes.

1. Server
2. Transparent
3. Client

VTP Server Mode

VTP Server can add, modify, and delete VLANs. It will propagate a VTP message containing all the changes from all of its trunk ports. If server receives a VTP message, it will incorporate the change and forward the message from all remaining trunk ports.

VTP Transparent Mode

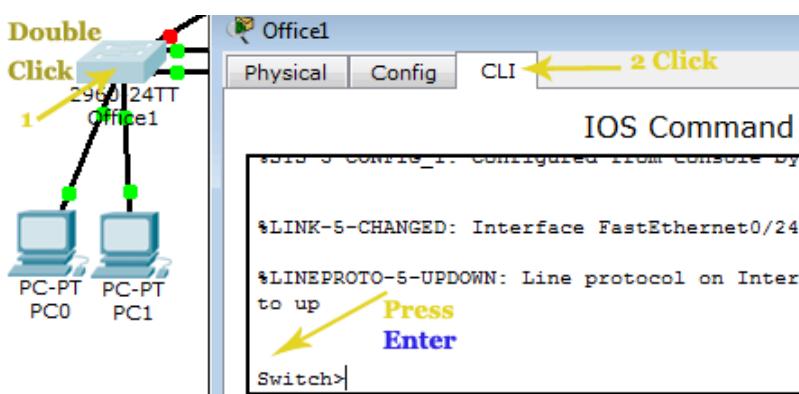
VTP Transparent switch can also make change in VLANs but it will not propagate these changes to other switches. If transparent switch receives a VTP message, it will not incorporate the change and forward the message as it receives, from all remaining trunk ports.

VTP Client Mode

VTP client switch cannot change the VLAN configurations itself. It can only update its VLAN configuration through the VTP messages that it receive from VTP server. When it receives a VTP message, it incorporates with the change and then forwards it from remaining trunk ports.

Configure VTP Server

We will configure **Office 1 Switch** as VTP Server. Double click on **Office 1 Switch** and Click **CLI** menu item and press **Enter key** to start CLI session.



By default all switches work as VTP server so we only need few commands to configure it. In following commands we will

- Set hostname to **S1**
- Set domain name to *example*
- Set password to *vinita*. (Password is case sensitive)

```
Switch>enable
Switch#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#hostname S1
S1(config)#vtp mode server
Device mode already VTP SERVER.
S1(config)#vtp domain example
Changing VTP domain name from NULL to example
S1(config)#vtp password vinita
Setting device VLAN database password to vinita
```

Configure VTP Client

We will configure Office 2 Switch and Office 3 Switch as VTP client switch. Access **CLI** prompts of **Office 2 Switch** and execute following commands

```
Switch>enable
Switch#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#hostname S2
S2(config)#vtp mode client
Setting device to VTP CLIENT mode.
S2(config)#vtp domain example
Changing VTP domain name from NULL to example
S2(config)#vtp password vinita
Setting device VLAN database password to vinita
S2(config)#

```

Now access **CLI** prompts of **Office 3 Switch** and enter following commands

```
Switch>enable
Switch#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#hostname S3
```

```
S3(config)#vtp mode client  
Setting device to VTP CLIENT mode.  
S3(config)#vtp domain example  
Changing VTP domain name from NULL to example  
S3(config)#vtp password vinita  
Setting device VLAN database password to vinita  
S3(config)#
```

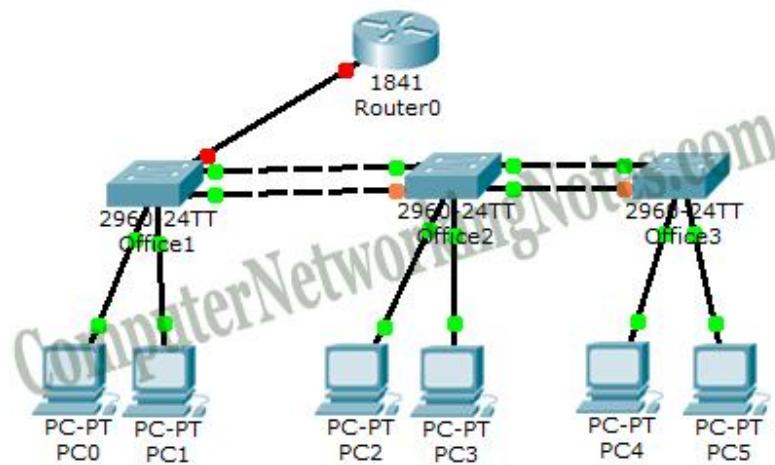
We have configured VTP server and VTP client. At this moment VTP client will not receive VTP messages from server. We need to configure DTP between switches.

Dynamic Trunking Protocol Mode And Configuration

This part explains DTP mode (ON, OFF, Auto, desirable and No –negotiate), DTP configuration in Cisco switches, VLAN tagging process, Switch port mode access and Switch port mode trunk in detail.

In VLAN configuration a switch port can operate in two mode; access and trunk. In access mode it can carry only single VLAN information while in trunk mode it can carry multiple VLANs information. Access mode is used to connect the port with end devices while trunk mode is used to connect two switching devices.

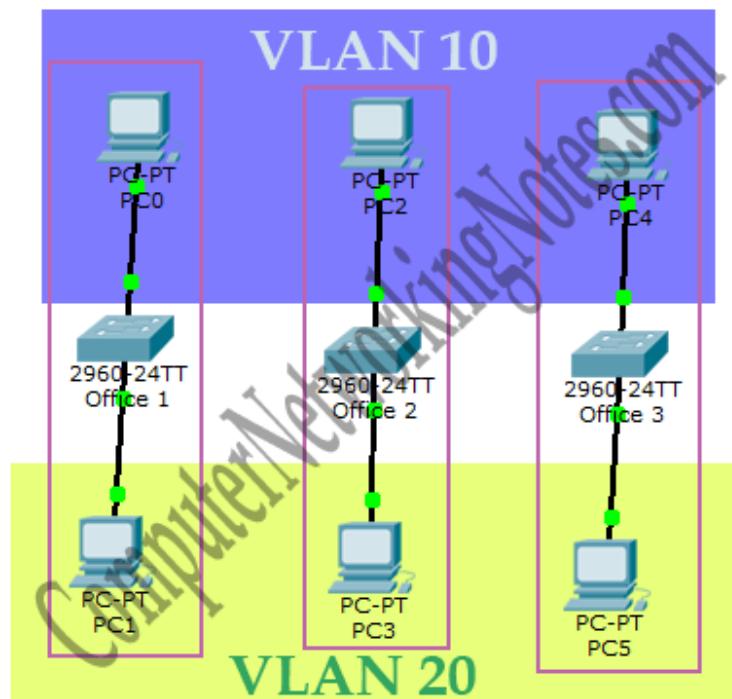
For this tutorial we assume that you have topology running in packet tracer.



Access Link and Trunk Link

An access link can carry single VLAN information while trunk link can carry multiple VLANs information. Configuring VLANs on single switch does not require trunk link. It is required only when you configure VLANs across the multiple switches.

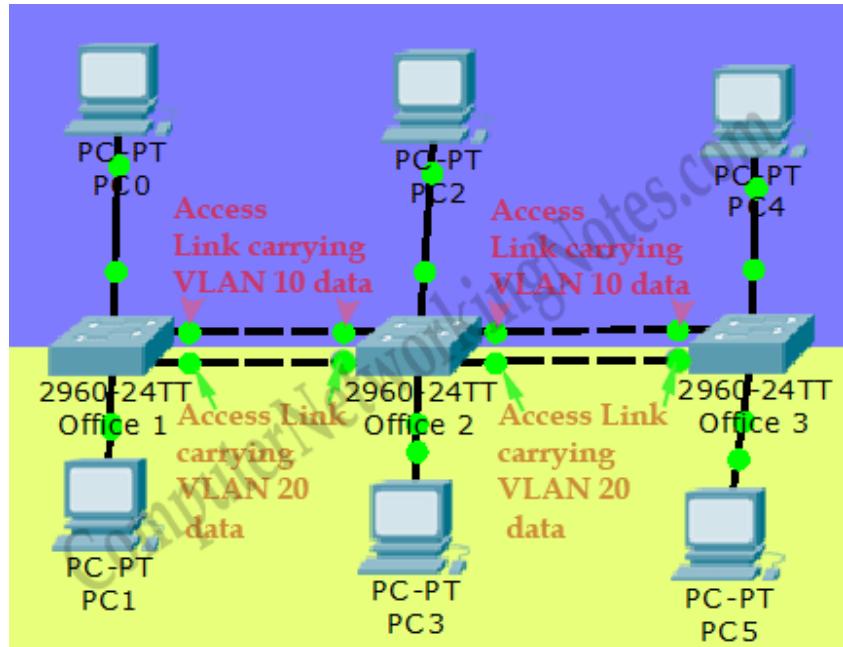
For example if we do not connect all switches in our network, we do not require to configure the trunk link. In this case PC0, PC2 and PC4 cannot communicate with each other. Although they all belongs to same VLAN group but they have no link to share this information.



Trunk link connections are used to connect multiple switches sharing same VLANs information.

You may think why we cannot use access link to connect these switches. Of course we can use access link to connect these switches but in that case we need to use a separate link for each VLAN. If we have two VLANs we need two links.

With this implementation we need links equal to VLANs that does not scale very well. For example if our design require 30 VLANs, we will have to use 30 links to connect switches.

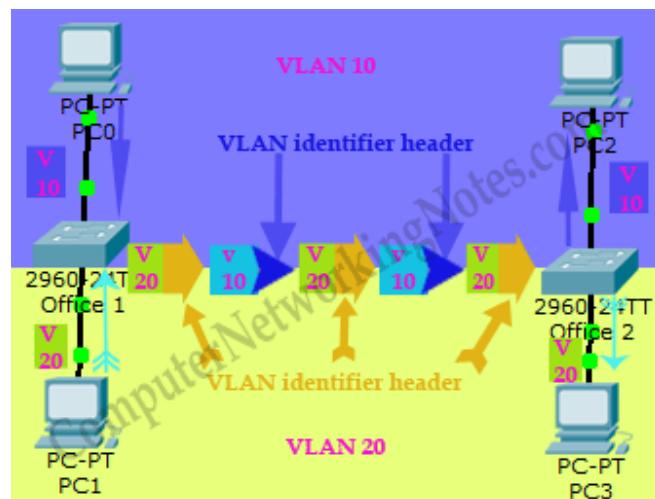


Inshort

- An access link can carry single VLAN information.
- Theoretically we can use access link to connect switches.
- If we use access link to connect switches, we have to use links equal to VLANs.
- Due to scalability we do not use access link to connect the switches.
- A trunk link can carry multiple VLAN information.
- Practically we use trunk links to connect switches.

VLAN Tagging

Trunk links use VLAN tagging to carry the multiple VLANs traffic separately.



In VLAN tagging process sender switch add a VLAN identifier header to the original Ethernet frame. Receiver switch read VLAN information from this header and remove it before forwarding to the associate ports. Thus original Ethernet frame remains unchanged. Destination PC receives it in its original shape.

VLAN Tagging process with example

- PC1 generates a broadcast frame.
- Office1 switch receives it and know that it is a broadcast frame for VLAN20.
- It will forward this frame from all of its port associated with VLAN20 including trunk links.
- While forwarding frame from access links, switch does not make any change in original frame. So any other port having same VLAN ID in switch will receive this frame in original shape.
- While forwarding frame from trunk links, switch adds a VLAN identifier header to the original frame. In our case switch will add a header indicating that this frame belongs to VLAN20 before forwarding it from trunk link.
- Office2 switch will receive this frame from trunk link.
- It will read VLAN identifier header to know the VLAN information.
- From header it will learn that this is a broadcast frame and belong to VLAN20.
- It will remove header after learning the VLAN information.
- Once header is removed, switch will have original broadcast frame.
- Now office2 switch has original broadcast frame with necessary VLAN information.
- Office2 Switch will forward this frame from all of its ports associated with VLAN20 including trunk links. For trunk link same process will be repeated.
- Any device connected in ports having VLAN20 ID in Office2 switch will receive original frame.

Now we know that in VLAN tagging process sender switch adds VLAN identifier header to the original frame while receive switch removes it after getting necessary VLAN information. Switches use VLAN trunking protocol for VLAN tagging process.

VLAN Trunking Protocol

Cisco switches supports two types of trunking protocols ISL and 802.1Q.

ISL

ISL (Inter-Switch Link) is a Cisco proprietary protocol. It was developed a long time before the 802.1Q. It adds a 26-byte header (containing a 15-bit VLAN identifier) and a 4-byte CRC trailer to the frame.

802.1Q

It is an open standard protocol developed by IEEE. It inserts 4 byte tag in original Ethernet frame. Over the time 802.1Q becomes more popular trunking protocols.

Key difference between ISL and 802.1Q

- ISL was developed Cisco while 802.1Q was developed by IEEE.
- ISL is a proprietary protocol. It will work only in Cisco switches. 802.1Q is an open standard based protocol. It will work on all switches.
- ISL adds 26 bytes header and 4 byte trailer to the frame.
- 802.1Q inserts 4 byte tag in original frame.

802.1Q is a lightweight and advanced protocol with several enhanced security features. Even Cisco has adopted it as a standard protocol for tagging in newer switches. 2960 Switch supports only 802.1Q tagging protocol.

VLAN Trunk Configuration

We can configure trunking in Cisco switches by two ways statically or dynamically. In static method we need to configure trunking in interface statically while in dynamic mode it is automatically done by a DTP trunking protocol.

Dynamic Trunking Protocol

DTP [Dynamic Trunking Protocol] is a Cisco proprietary protocol. It automatically configures trunking on necessary ports. It operates in five modes.

DTP Modes

DTP Mode ON

In ON mode interface is set to trunk, regardless remote end supports trunking or not. On mode cause interface to generate DTP messages and tag frames based on trunk type.

DTP Mode Desirable

In Desirable mode interface will generate the DTP messages and send them to other end. Interface will work as access link until it gets replies from remote end. If reply messages indicate

that remote device is trunking capable, DTP will change connection link in trunk from access link. If other end does not respond to DTP message, the interface will work as access link connection.

DTP Mode Auto

In auto mode interface works as access link and passively listen for DTP messages. Interface will change connection link to trunk, if it receives a DTP message from remote end.

DTP Mode No-Negotiate

In No-Negotiate mode interface is set as trunk connection. Interface will tag frames but it will not generate DTP messages. DTP is a Cisco's proprietary protocol, thus a non Cisco device will not understand it. This mode is used to trunk connection between Cisco device and a non Cisco device.

DTP Mode OFF

In off mode interface is configured as access-link. No DTP message will be generated nor frames will be tagged.

In our topology we need to configure trunk on following interfaces

Switch	Interfaces
Office 1	Gig1/1, Gig1/2, F0/24
Office 2	Gig1/1, Gig1/2, F0/23, F0/24
Office 3	Gig1/1, Gig1/2

By default all interface on switch starts as access link. *switchport mode trunk* command is used to change connection link in trunk. Run this command from interface mode. In next section we will change all necessary interfaces [given in above table] connection link in trunk.

Office 1 Switch

```
S1(config)#interface fastEthernet 0/24
S1(config-if)#switchport mode trunk
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/24,
changed state to down
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/24,
changed state to up
```

```
S1(config-if)#exit
S1(config)#interface gigabitEthernet 1/1
S1(config-if)#switchport mode trunk
S1(config-if)#exit
S1(config)#interface gigabitEthernet 1/2
S1(config-if)#switchport mode trunk
%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet1/2,
changed state to down
%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet1/2,
changed state to up
S1(config-if)#exit
S1(config)#+
```

Office 2 Switch

```
S2(config)#interface gigabitEthernet 1/1
S2(config-if)#switchport mode trunk
%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet1/1,
changed state to down
%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet1/1,
changed state to up
S2(config-if)#exit
S2(config)#interface gigabitEthernet 1/2
S2(config-if)#switchport mode trunk
S2(config-if)#exit
S2(config)#interface fastEthernet 0/23
S2(config-if)#switchport mode trunk
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/23,
changed state to down
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/23,
changed state to up
S2(config-if)#exit
S2(config)#interface fastEthernet 0/24
S2(config-if)#switchport mode trunk
S2(config-if)#exit
```

Office 3 Switch

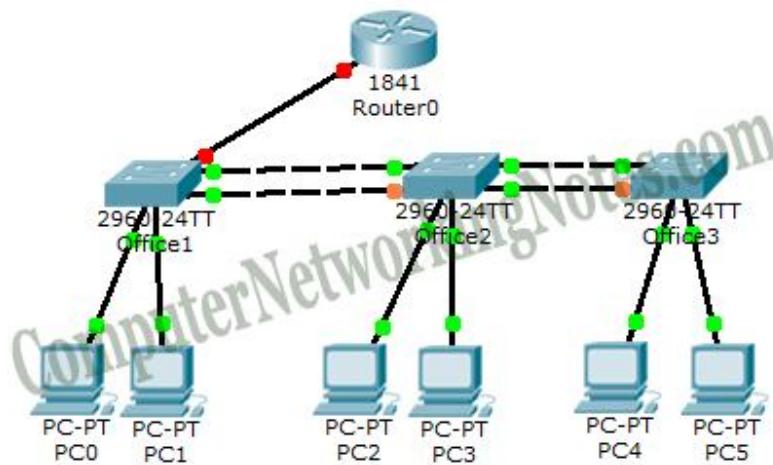
```
S3(config)#interface fastEthernet 0/24
S3(config-if)#switchport mode trunk
S3(config-if)#exit
```

```
S3(config)#interface gigabitEthernet 1/1
S3(config-if)#switchport mode trunk
S3(config-if)#exit
```

That's all configurations we needs. Now our trunk links are ready to move multiple VLANs traffic.

VLAN Configuration Commands Step By Step Explained

This Assignment explains how to create VLAN, how to assign VLAN Membership static and dynamic, how to configure router on stick and how to configure VLAN in Cisco Switches and router step by step.



How to create VLAN

In our network Office1 Switch is configured as VTP Server. Office2 and Office3 switches are configured as VTP clients. We only need to create VLANs in VTP Server. VTP Server will propagate this information to all VTP clients.

vlan *vlan number* command is used to create the VLAN.

Office 1 Switch

```
S1(config)#vlan 10
S1(config-vlan)#exit
S1(config)#vlan 20
S1(config-vlan)#exit
S1(config)#+
```

How to assign VLAN Membership

VLAN can be assigned statically or dynamically. CCNA exam only includes static method; therefore we will also use static method to assign VLAN membership. **switchport access vlan [vian number]** command is used to assign VLAN to the interface. Following commands will assign VLANs to the interfaces.

Office 1 Switch

```
S1(config)#interface fastEthernet 0/1
S1(config-if)#switchport access vlan 10
S1(config-if)#interface fastEthernet 0/2
S1(config-if)#switchport access vlan 20
```

Office 2 Switch

```
S2(config)#interface fastEthernet 0/1
S2(config-if)#switchport access vlan 10
S2(config-if)#interface fastEthernet 0/2
S2(config-if)#switchport access vlan 20
```

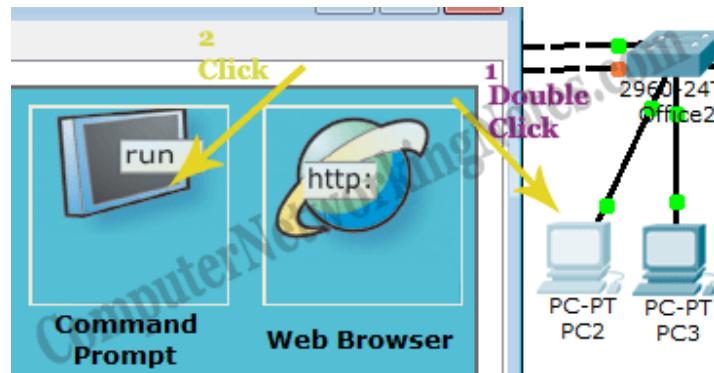
Office 3 Switch

```
S3(config)#interface fastEthernet 0/1
S3(config-if)#switchport access vlan 10
S3(config-if)#interface fastEthernet 0/2
S3(config-if)#switchport access vlan 20
```

We have successfully assigned VLAN membership. It's time to test our configuration. To test this configuration, we will use *ping* command. *ping* command is used to test connectivity between two devices. As per our configuration, devices from same VLAN can communicate. Devices from different VLANs must not be able to communicate with each other without router.

Test VLAN configuration

Access PC's command prompt to test VLAN configuration. Double click **PC-PT** and click **Command Prompt**



We have two VLAN configurations VLAN 10 and VLAN 20. Let's test VLAN 10 first. In VLAN 10 we have three PCs with IP addresses 10.0.0.2, 10.0.0.3 and 10.0.0.4. These PCs must be able to communicate with each other's. At this point PCs from VLAN 10 should not be allowed to access PCs from VLAN 20. VLAN 20 also has three PCs 20.0.0.2, 20.0.0.3 and 20.0.0.4.

```
PC>ipconfig
IP Address.....: 10.0.0.3
Subnet Mask....: 255.0.0.0
Default Gateway.: 10.0.0.1
PC>ping 10.0.0.2
Reply from 10.0.0.2: bytes=32 time=1ms TTL=128
Reply from 10.0.0.2: bytes=32 time=0ms TTL=128
PC>ping 10.0.0.4
Reply from 10.0.0.4: bytes=32 time=1ms TTL=128
Reply from 10.0.0.4: bytes=32 time=1ms TTL=128
PC>ping 20.0.0.4
Request timed out.
Request timed out.
PC>ping 20.0.0.3
Request timed out.
Request timed out.
PC>ping 20.0.0.2
Request timed out.
Request timed out.
```

We have successfully implemented VLAN 10 now test VLAN 20.

Same as VLAN 10, PCs from VLAN 20 must be able to communicate with other PCs of same VLAN while they should not be able to access VLAN 10.

```
PC>ipconfig
IP Address.....: 20.0.0.3
Subnet Mask.....: 255.0.0.0
Default Gateway.: 20.0.0.1

PC>ping 20.0.0.2
Reply from 20.0.0.2: bytes=32 time=34ms TTL=128
Reply from 20.0.0.2: bytes=32 time=0ms TTL=128

PC>ping 20.0.0.4
Reply from 20.0.0.4: bytes=32 time=12ms TTL=128
Reply from 20.0.0.4: bytes=32 time=0ms TTL=128

PC>ping 10.0.0.2
Request timed out.
Request timed out.

PC>ping 10.0.0.3
Request timed out.
Request timed out.

PC>ping 10.0.0.4
Request timed out.
Request timed out.
```

Congratulations we have successfully achieved one more mile stones of this article.

Configure Router on Stick

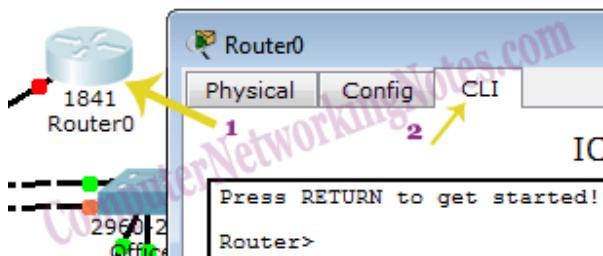
Typically routers are configured to receive data on one physical interface and forward that data from another physical interface based on its configuration. Each VLAN has a layer 3 address that should be configured as default gateway address on all its devices. In our scenario we reserved IP address 10.0.0.1 for VLAN 10 and 20.0.0.1 for VLAN 20.

With default configuration we need two physical interfaces on router to make this intra VLAN communication. Due to price of router, it's not a cost effective solution to use a physical interface of router for each VLAN. Usually a router has one or two Ethernet interface. For example if we have 50 VLANs, we would need nearly 25 routers in order to make intra VLANs communications. To deal with situation we use Router on Stick.

Router on Stick is router that supports trunk connection and has an ability to switch frames between the VLANs on this trunk connection. On this router, single physical interface is sufficient to make communication between our both VLANs.

Access command prompt of Router

To configure Router on Stick we have to access CLI prompt of Router. Click **Router** and Click **CLI** from menu items and Press **Enter key** to access the CLI



Run following commands in same sequence to configure Router on Stick

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface fastEthernet 0/0
Router(config-if)#no ip address
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#interface fastEthernet 0/0.10
Router(config-subif)#encapsulation dot1Q 10
Router(config-subif)#ip address 10.0.0.1 255.0.0.0
Router(config-subif)#exit
Router(config)#interface fastEthernet 0/0.20
Router(config-subif)#encapsulation dot1Q 20
Router(config-subif)#ip address 20.0.0.1 255.0.0.0
Router(config-subif)#exit
```

- In above configuration we broke up single physical interface [FastEthernet 0/0] into two logical interfaces, known as sub-interfaces. Router supports up to 1000 interfaces including both physical and logical.
- By default interface link works as access link. We need to change it into trunk link. encapsulation commands specify the trunk type and associate VLAN with sub-interface.
- In next step we assigned IP address to our sub-interface.

That's all configuration we need to switch VLANs. Now we can test different VLAN communications. To test intra VLANs communication open command prompt of PC and ping the PC of other VLAN.

Command Prompt

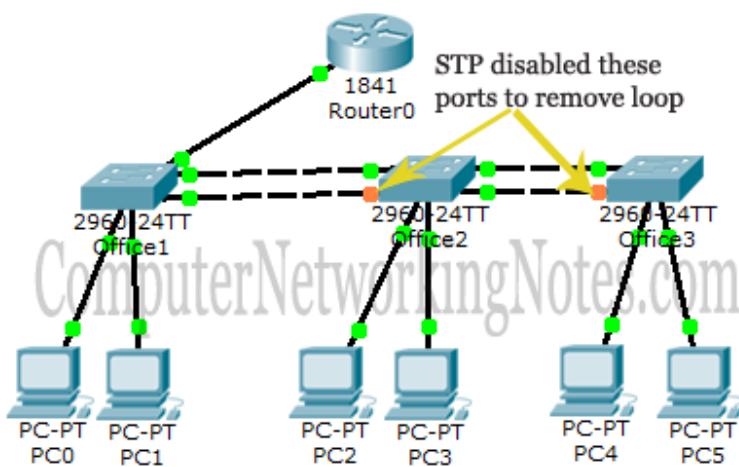
```
Packet Tracer PC Command Line 1.0
PC>ipconfig
IP Address.....: 10.0.0.3
Subnet Mask....: 255.0.0.0
Default Gateway.: 10.0.0.1

PC>ping 20.0.0.2
Reply from 20.0.0.2: bytes=32 time=0ms TTL=127
Reply from 20.0.0.2: bytes=32 time=0ms TTL=127
PC>
```

PC [10.0.0.3] from VLAN 10 can now access PC [20.0.0.2] from VLAN 20.

Spanning Tree Protocol (STP)

STP is a layer 2 protocol, used for removing loops. For backup purpose we typically create backup links for important resources. In our scenario, all offices have backup links that create loops in topology. STP automatically removes layer 2 loops. STP multicasts frame that contain information about switch interfaces. These frames are called BPDU (Bridge Protocol Data Units). Switch use BPDUs to learn network topology. If it found any loop, it will automatically remove that. To remove loop, STP disables port or ports that are causing it.



How to configure VLAN VTP DTP cheat sheet

Command	Descriptions
Switch(config)#vtp mode server	Configure Switch as VTP Server
Switch(config)#vtp mode client	Configure Switch as VTP Client
Switch(config)#vtp mode transparent	Configure Switch as VTP Transparent
Switch(config)#no vtp mode Configure	Switch to default VTP Server Mode
Switch(config)#vtp domain domain-name	Set VTP Domain name.
Switch(config)#vtp password password	Set VTP password. Password is case sensitive
Switch#show vtp status	Display VTP status including general information
Switch#show vtp counters	Show VTP counters of switch
Switch(config-if) #switchport mode trunk	Change interface mode in Trunk
Switch(config)#vlan 10	Create VLAN and associate number ID 10 with it
Switch(config-vlan)#name Sales	Assign name to VLAN
Switch(config-vlan)#exit	Return in Global configuration mode from VLAN configuration mode
Switch(config)#interface fastethernet 0/1	Enter in interface configuration mode
Switch(config-if)#switchport mode access	Set interface link type to access link
Switch(config-if)#switchport access vlan 10	Assign this interface to VLAN 10
Switch#show vlan	Displays VLAN information
Switch#show vlan brief	Displays VLAN information in short
Switch#show vlan id 10	Displays information VLAN ID 10 only
Switch#show vlan name sales	Displays information about VLAN named sales only
Switch(config)#interface fastethernet 0/8	Enter in Interface configuration mode
Switch(config-if)#no switchport access vlan 10	Removes interface from VLAN 10 and reassigns it to the default VLAN - VLAN 1
Switch(config-if)#exit	Move back to Global configuration mode
Switch(config)#no vlan 10	Delete VLAN 10 from VLAN database
Switch#copy running-config startup-config	Saves the running configuration in NVRAM

OSPF Neighborship Condition and Requirement

This explains OSPF Neighborship Requirements Area ID, Authentication, Hello and Dead intervals, Stub Flag and MTU size in detail with examples.

OSPF routers share routing information only with neighbors. OSPF uses hello packets to discover neighbors in segments. A hello packet contains some essential configuration values that must be same on both routers who want to build an OSPF neighborship. In this tutorial we will explain these configuration values in detail with example.

OSPF Neighborship Requirement

In order to become OSPF neighbor following values must be match on both routers.

- Area ID
- Authentication
- Hello and Dead Intervals
- Stub Flag
- MTU Size

Area ID

OSPF uses area concept to scale an enterprise size network. I have explained OSPF Areas in first part of this article. Just for reference, OSPF areas create a logical boundary for routing information. By default routers do not share routing information beyond the area. So in order to become neighbor, two routers must belong to same area. Here one confusing fact needs to clear. Area is associated with specific interface, not with entire router. This allows us to configure the router in multiple areas. For example a router that has two interfaces; Serial interface and FastEthernet interface, can run Serial interface in one area and FastEthernet in another area. It means link which connects two routers need be in same area including its both ends interface. Beside this interfaces should have same network ID and subnet mask.

Following figure illustrate a simple OSPF network. In this network **R1** is eligible to form neighborship with **R4** and **R2** respectively on **S0/0** and **F0/0**.



I have question for you. Why neighborship cannot be built between **R1** and **R3**?

Let's find out the answer step by step.

Both interfaces should be in same area.

Yes both interfaces (R1's Fo/1 and R3's F0/1) are in same area.

Both interfaces should be in same segment.

Yes both interfaces (R1's Fo/1 and R3's F0/1) are connected with direct link.

Both interfaces should have same subnet mask.

Yes both interfaces have same subnet mask /30.

Both interfaces should have same network ID.

No both interfaces have different network ID. **R1's F0/1** has network ID **192.168.0.4/30** while **R3's F0/1** has network ID **192.168.0.8/30**. This condition does not match. Thus these two routers on these interfaces cannot build neighborship.

Authentication

To enhance the security of network, OSPF allows us to configure the password for specific areas. Routers who have same password will be eligible for neighborship. If you want to use this facility, you need to configure password on all routers which you want to include in network. If you skip any router, that will not be able to form an OSPF neighborship.

Suppose that our network has two routers R1 and R2. Both routers are connected with direct link and meet all criteria mentioned in first requirement. What if I configure password in R1 and leave R2 as it is? Will it form neighborship with R2?

Well in this situation neighborship will not take place. Because when both routers see each other's hello packet in segment, they try to match all configuration values including password field. One packet has a value in password field while other has nothing in it. In this case routers will simply ignore each other's packet.

Hello packets and hello interval

Hello packets are the special type of LSAs (Link State Advertisements) which are used to discover the neighbors in same segment. And once neighborship is built same hello packets are used to maintain the neighborship. Hello packets contain all necessary information that is required to form a neighborship. Hello packets are generated and distributed in hello interval via multicast. Hello interval is the length of time in seconds between the hello packets. Default hello interval is **10** seconds.

Dead Intervals

As we already know once neighborship is built, hello packets are used to maintain the neighborship.

So a router must see hello packets from neighbor in particular time interval. This time interval is known as dead interval. Dead interval is the number of seconds that a router waits for hello packet from neighbor, before declaring it as dead.

Default dead interval is **40** seconds. If a router does not receive hello packet in **40** seconds from neighbor it will declare that as dead. When this happens, router will propagate this information to other OSPF neighboring router via LSA message.

Hello and dead interval must be same between two neighbors. If any of these intervals are different, neighborship will not form.

Stub Area Flag

This value indicates that whether sending router belongs to stub area or not. Routers who want to build OSPF neighborship must have same stub area flag.



For example we have two routers R1 and R2:-

- Both routers belong to same stub area, neighborship can be built
- Both routers belong to different stub area, neighborship cannot be built
- Both routers do not belong to any stub area, neighborship can be built
- Only one router belongs to a stub area, neighborship cannot be built

Just like another areas, Stub area also has some specific meanings in OSPF hierachal design.

A stub area has following requirements:-

- A stub area can have only single exit point from that area.
- Stub area cannot be used as a transit area for virtual links.
- Routing from stub area to outside of the area should not have to take an optimal path.
- Any external networks (redistributed from other protocols into OSPF) should not be flooded in stub area.

Configuring a stub area reduces the size of topology table inside that area. Thus routers running in this area require less memory.

MTU

Technically MTU (Maximum Transmission Unit) is not a part of compulsory matching conditions. Still we should match this value. If this value does not match routers may stuck in Exstart/Exchange exchange stage.

Consider a situation where MTU setting between two OSPF routers does not match. If the router with the higher MTU sends a packet larger than the MTU set on the neighboring router, the neighboring router will ignores this packet. This function creates serious problem for database updates. Database updates are heavier in nature. Once an update becomes larger than the configured MTU setting, it needs to be spilt. In a case of miss match MTU, database update may lost few bytes. Due to this, OSPF will ignore that update and cannot sync with database. It will be stuck in Exstart/Exchange stage.

It is always worth to spend a little extra time in matching optional values along with compulsory values. Matching configuration values will make troubleshooting easier.

OSPF Neighbor States Explained With Example

OSPF routers go through the seven states, called Down, Attempt/Init, Two ways, Exstart, Exchange, Loading and full while building adjacency with other OSPF speaking routers. In this we will see these states in easy language with examples. Along with these states, also explanation of few other terminologies used in this process.

OSPF Neighborship states

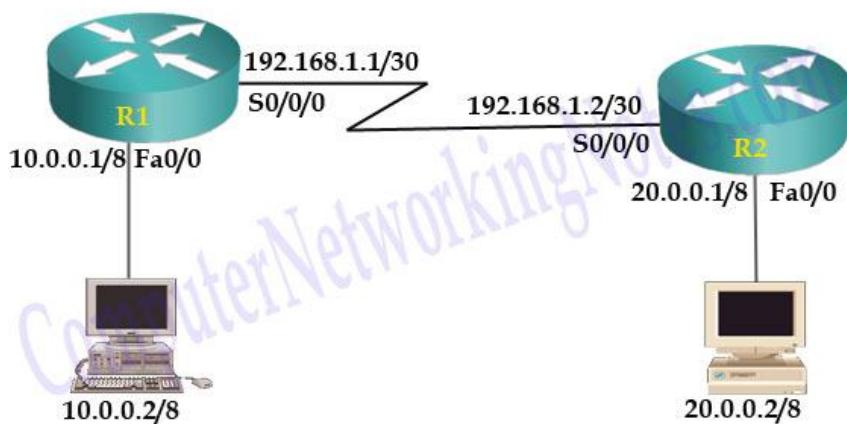
OSPF routers go through the seven states while building neighborship with other routers.

- Down state
- Attempt/Init state
- Two ways state
- Exstart state
- Exchange state
- Loading state
- Full state

Let's understand these states with a simple example. Assume that our network has two routers running OSPF routing protocol. Routers are connected with each other via serial link. We just turned on both routers simultaneously.

Down state

At this point both routers have no information about each other. R1 does not know which protocol is running on R2. Vice versa R2 have no clue about R1. In this stage OSPF learns about the local interfaces which are configured to run the OSPF instance.



In down state routers prepares themselves for neighborship process. In this state routers choose RID (Router ID). RID plays a big role in OSPF process. Before we move in next state let's understand what is RID.

RID

RID is a unique identifier of Router in OSPF network. It must be unique within the autonomous system. Routers identify each other through the RID in AS.

How do routers choose RID?

An OSPF router looks in three places for RID:-

1. Manual configuration
2. Loopback interface IP configuration
3. Active interfaces IP configuration

Manual configuration

Because RID plays a significant role in network, OSPF allows us to configure it manually. RID is 32 bit long. IP address is also 32 bit in length. We can use IP address as a RID. This gives us more flexibility over RID. For example we can use a simple and sequential IP scheme such as 1.1.1.1 for R1, 1.1.1.2 for R2, 1.1.1.3 for R3, 1.1.1.4 for R4, 1.1.1.5 for R5 and so on.

We can assign RID from OSPF sub command mode.

```
Router(config)#router ospf 1  
Router(config-router)#router-id ip_address
```

If we have assigned RID manually, OSPF will not look in next two options. Suppose we did not assign it through the command. In this situation OSPF will look in next option to find the RID.

Loopback interface IP configuration

If loopback interface is configured, OSPF will choose its IP address as RID. If multiple loopback interfaces are configured, highest IP address will be chosen from all loopback interfaces configuration.

If loopback interface is not configured, OSPF will look in next and last possible place to choose the RID.

Active interface IP configuration

OSPF will choose the highest IP address from all operational IP interfaces. We should not let the OSPF to use this option. This option does not provide a fix RID which is very necessary for network stability.

This option has several reasons which may force OSPF to recalculate the RID such as Interface which IP address is chosen may go down or for troubleshooting we may enable / disable the interfaces.

Key points

- OSPF will follow the sequence (Manual configuration => Loopback interface => Active interface) of options while selecting RID. If RID is found, it will not look in next option.
- OSPF will choose IP address only from operational IP interface. Operational means interface should be listed as line is up and line protocol is up in the output of show ip interface brief command.
- When multiple IP addresses are available, OSPF will always pick highest IP address for RID.
- For network stability we should always set RID from either **router-id** command or by using loopback interfaces.
- By default Router chooses OSPF RID when it initialized. Once RID is selected it will use that RID until next reboot.
- OSPF will not consider any change in RID which we make after initialization. We have two options to implement new RID. Either reboots the router or clear the OPSF process with clear ip ospf process command.
- If OSPF fails to select the RID, it will halt the OSPF process. We cannot use OSPF process without RID.

 The image cannot be displayed. Your computer may not have enough memory to open the image, or the image may have been corrupted. Restart your computer, and then open the file again. If the red x still appears, you may have to delete the image and then insert it again.

In down state router do following

- Choose RID and initialize the OSPF process
- Run OSPF instance on local interfaces which are configured through the network command such as **R1(config-router)#network 10.0.0.0 0.0.255.255 area 0**.
- Collect necessary information for Hello packet such RID and configuration values which are required to build the neighborship.

Attempt/Init state

Neighborship building process starts from this state. R1 multicasts first hello packet so other routers in network can learn about the existence of R1 as an OSPF router. This hello packet contains Router ID and some essential configuration values such as area ID, hello interval, hold down timer, stub flag and MTU. Essential configuration values must be same on routers who want to build an OSPF neighborship.

In previous part of this article I explained essential configuration values in detail with example. For this tutorial I assume that these values match on both routers. If essential configuration values match, R2 will add R1 in his neighbor Table.



In Init state routers do following

- R1 will generate a hello packet with RID and essential configuration values and send it out from all active interfaces.
- The hello packets are sent to the **multicast address 224.0.0.5**.
- R2 will receive this packet.
- R2 will read RID from packet and look in neighbor table for existing entry.
- If match found, R2 would skip neighborship building process and reset the dead interval timer for that entry.
- If OSPF does not find a match in neighbor table, it will consider R1 (sender router) as a possible OSPF neighbor and start neighborship building process.
- R2 will match its essential configuration values with values listed in packet.
- If all necessary configuration values match, R2 will add R1 in its neighbor table.

At this moment R1 has no idea about R2. R1 will learn about R2 when it will respond.

Before we enter in third state, let's have a quick look on attempt state.

Attempt

In Non-broadcast multi-access environment such as Frame Relay and X.25, OSPF uses Attempt state instead of Init state. OSPF uses this state only if neighbors are statically configured with **neighbor** command. In this situation, it does not have to discover them dynamically. As it already knows the neighbors, it will use unicast instead of multicast in this state.

Once neighborship is built, OSPF uses hello packets as keep alive. If a router does not receive a hello packet from any particular neighbor in dead interval, it will change its state to down from full. After changing the state it will make an effort to contact the neighbor by sending Hello packets. This effort is made in Attempt state.

Basically Both Init and Attempt states describe similar situation where one router has sent a hello packet and waiting for response.

Two ways state

If essential configuration values match, R2 will add R1 in neighbor table and reply with its hello packet. As R2 knows the exact address of R1, it will use unicast for reply. Beside RID and configuration values, this packet also contains the R2's neighbor table data. As we know R2 has already added R1 in its neighbor table. So when R1 will see R2's neighbor table data, R1 would also see its name in this data. This will assure R1 that R2 has accepted its neighborship request.

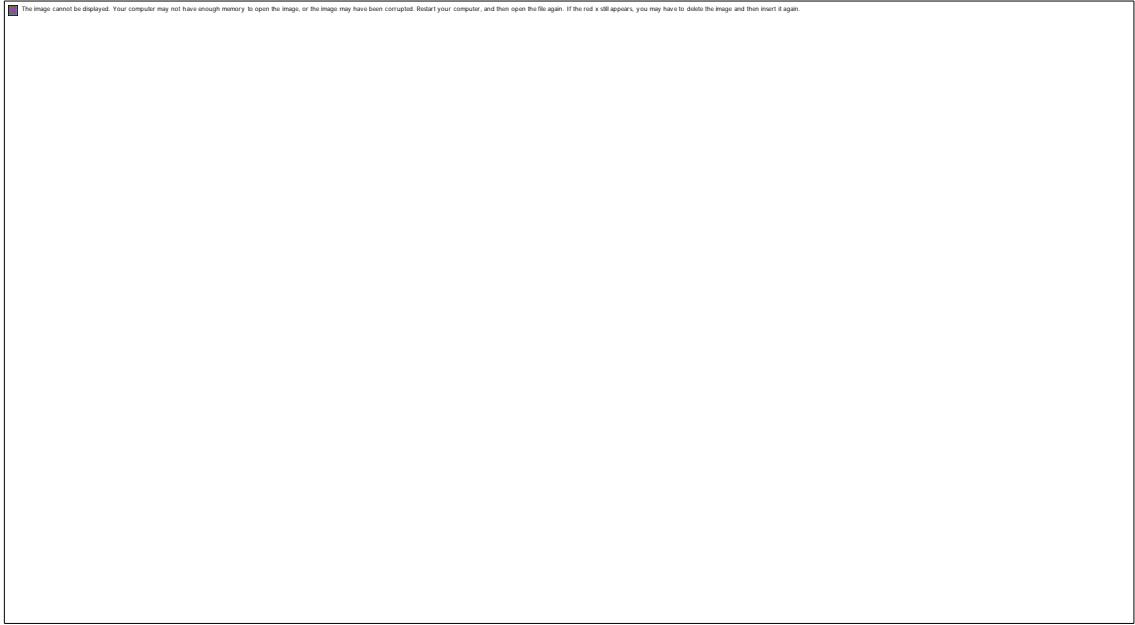
At this point:-

- R2 has checked all essential configuration values listed in hello packet which it received from R1.
- R2 is ready to build neighborship with these parameters.
- R2 has added R1 in its neighbor table.
- To continue the neighborship process, R2 has replied with its hello packet.
- R1 has received a reply from neighbor, with its own RID listed in R2's neighbor table.

 The image cannot be displayed. Your computer may not have enough memory to open the image, or the image may have been corrupted. Restart your computer, and then open the file again. If the red x still appears, you may have to delete the image and then insert it again.

Now it is R1's turn to take action on R2's reply. This reply would be based on hello packet which it received from R2. As we know that this hello packet contains one additional field; Neighbor table data field which indicates that this is not a regular neighbor discovery hello packet. This packet is a reply of its own request.

- R1 will take following actions:-
- It will read RID from hello packet and look in its neighbor table for existing entry.
- If a match for RID found in neighbor table, it would reset the dead interval timer for that entry.
- If a match is not found in neighbor table, it would read the essential configuration values from packet.
- It will match configuration values with its own values. If values match, it will add R2's RID in neighbor table.
- If packet contains neighbor table data with its own RID, it will consider that as request to enter in two way state.
- R1 will reply with a hello packet which contains its neighbor table data.
- This packet is a confirmation of two ways state.



The image cannot be displayed. Your computer may not have enough memory to open the image, or the image may have been corrupted. Restart your computer, and then open the file again. If the red x still appears, you may have to delete the image and then insert it again.

Fine, our routers are neighbor now. They are ready to exchange the routing information.

Before we understand how routers will exchange routing information, we need to understand the types of network. OSPF uses different types of exchange process for different types of network.

Point to point network

It is a Cisco specific network type. It connects a single pair of routers. HDLC and PPP are example of point to point network type. In this type of network:-

- All routers form full adjacencies with each other.
- Hello packets are sent using a multicast address 224.0.0.5
- No DR and BDR are required.
- All routers are considered as **AllSPFRouters**.

I will explain the terms adjacencies, DR, BDR and AllSPFRouters shortly.

Broadcast Networks

Broadcast networks are capable in connecting more than two devices. Ethernet and FDDI are the example of broadcast type network. In this type of network:-

- A single transmitted packet can be received by all attached devices.
- DR and BDR are required.
- All routers form full adjacencies only with DR and BDR.
- Routers use a multicast address 224.0.0.6 to update the DR.
- DR uses a multicast address 224.0.0.5 to update the all routers.

NMBA

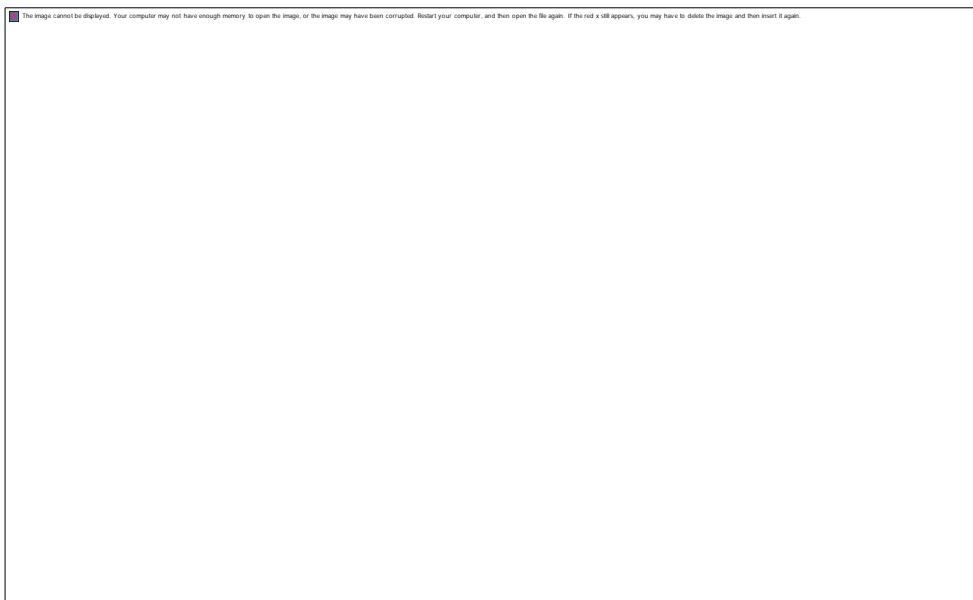
Non-broadcast Multi-access networks are also capable in connecting more than two devices. But they do not have broadcast capability. X.25 and Frame Relay are the example of NMBA type network. In this type of network:-

- As network does not have broadcast capability, dynamic network discovery will not be possible.
- OSPF neighbors must have to define statically.
- All OSPF packets are unicast.
- DR and BDR are required.

Point to multipoint

Point to multipoint is a special implementation of NMBA network where networks are configured as a collection of point to point links. In this type of network:-

- Network must be configured statically.
- No DR and BDR are selected in this type of network.
- OSPF packets are multicast.



We can divide these networks in two types;

1. Networks which need DR and BDR such as broadcast and NBMA
2. Networks which do not need DR and BDR such as point to point and point to multipoint

So what does DR and BDR actually do? Why do we need them in our network?

DR and BDR

OSPF routers in a network which need DR (Designated router) and BDR (Backup designated router) do not share routing information directly with all each other's. To minimize the routing information exchange, they select one router as designated router (DR) and one other router as backup designated router (BDR). Remaining routers are known as DROTHERs.

All DROTHERs share routing information with DR. DR will share this information back to all DROTHERs. BDR is a backup router. In case DR is down, BDR will immediately take place the DR and would elect new BDR for itself.

Main reason behind this mechanism is that routers have a central point for routing information exchange. Thus they need not to update each other's. A DROTHER only need to update the central point (DR) and other DROTHERs will receive this update from DR.

Practically this will cut the numbers of routing information exchange from $O(n*n)$ to $O(n)$ where n is the number of routers in a multi-access segment.

For example following figure illustrates a simple OSPF network. In this network R4 is selected as DR and R5 is selected as BDR. DROTHERs (R1, R2 and R3) will share routing information with R4 (DR) and R5 (BDR), but they will not share routing information with each other. Later DR will share this information back to all DROTHERs.



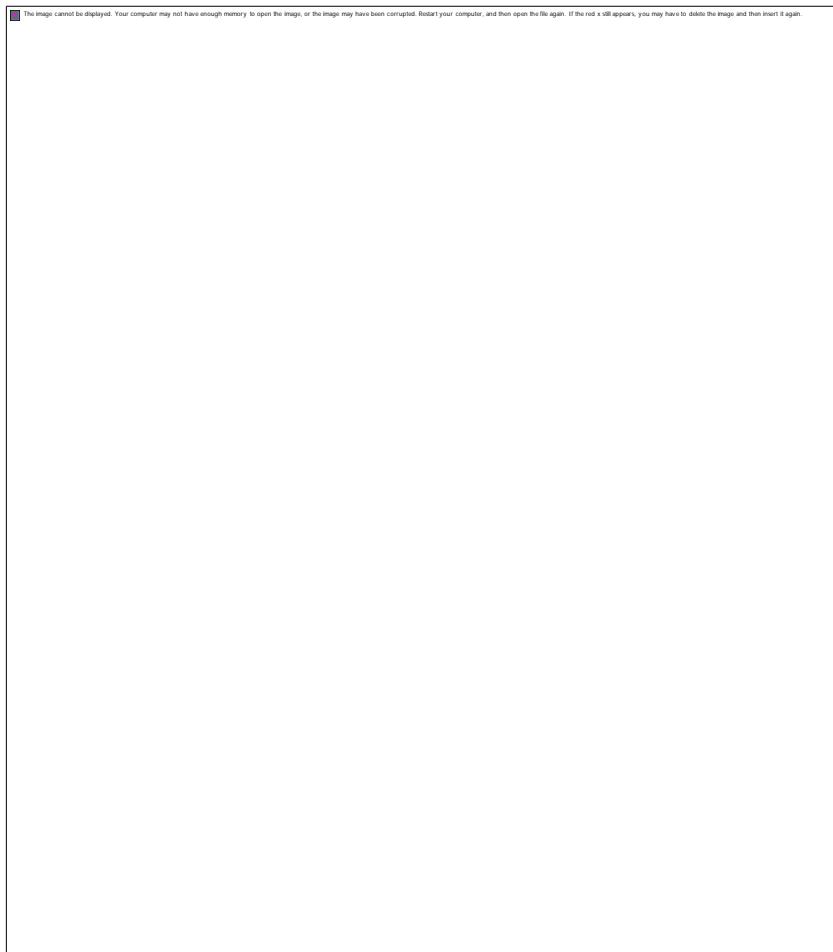
DR and BDR Election process

OSPF uses priority value to select DR and BDR. OSPF router with the highest priority becomes DR. Router with second highest priority becomes BDR. If there is a tie, router with the highest RID will be chosen.

Priority value is 8 bit in length. Default priority value is 1. We can set any value from range 0 to 255. We can change it from Interface Sub-configuration mode with *ip ospf priority* command.

We can force any router to become DR (Highest) or BDR (Second highest) by changing its priority value. If we set priority value to 0, it will never become DR or BDR.

For example following figure illustrates a simple OSPF network. In this network we have five routers. We do not want that R3 becomes DR or BDR. So we changed its default priority value to 0. Now let's see how these routers select DR and BDR.



Condition 1:- *Use the highest priority value*

This condition says "Arrange all routes in high to low order and pick the highest for DR and second highest for BDR". If we arrange our routers in high to lower order, R3 will stand at last. Remaining routers have equal priority value. So at the end of this condition we have a tie between four routers.

Condition 2:- *If there is a tie use the highest RID*

This condition says "If there is a tie, use RID value to choose". In our network we have a tie between four routers, so our routers will use RID to elect the DR and BDR. Arranging routers in high to low order will give us the DR and BDR.

As we know that there are two types of network; networks which do not require DR and BDR for exchange process and networks which require DR and BDR for exchange process.

In first type all routers will exchange routing information with each other's. In second type DROTHERs will exchange routing information with DR and BDR.

Routers which will exchange routing information are known as adjacent. Relationship between two adjacent is known as adjacency. This terminology is associated with interfaces.

A router which has two interfaces can be **adjacent** in one interface and **DROTHER** in other interface.

For example following figure illustrates an OSPF running NBMA network. In this network;

R3 will build adjacency with R1, so in this relationship they will be considered as **Adjacent**.

R3 will not build adjacency with R4, so in this relationship they will be considered only **DROTHER**.



In a network which doesn't require DR and BDR, all routers will be considered as **Adjacent** and relationship between them will be considered as **Adjacency**.

Only adjacent routers will enter in next states to build the adjacency.

Exstart state

Routers who decided to build adjacency will form a master / slave relationship. In each adjacency router who has higher RID will become master and other will become slave. Do not mix Master /Slave relationship with DR/ BDR/ DROTHER relationship. Both terms look similar but have different meaning. DR/ BDR/ DROTHER relationship is built in a segment and have a wider meaning while Master / Slave relationship is built between two interfaces which need to exchange routing information. Master / Slave relationship has limited purpose. It is used to decide the Router who will start exchange process. Always Master starts exchange process.



Once routers settle down on Master/Slave, they will establish the initial sequence numbers which will be used in routing information exchange process. Sequence numbers insure that routers get most accurate information.

Exchange state

In exchange state, Master and slave decide how much information needs to be exchanged. A router that has more than one interface may learn same network information from different sources. An OSPF router is smart enough to filter the updates before receiving it. It will ask only for the updates which it does not have. In this state, routers will filter the updates which need to be exchanged.

Before we learn how routes will filter this information, let's understand few relative terms.

LSA and LSDB are explained in the first part of this tutorial. To maintain the flow of this article I am including the summary of these terms here again.

LSA

Link state advertisement (LSA) is a data packet which contains link-state and routing information. OSPF uses it share and learn network information.

LSDB

Every OSPF router maintains a Link state database (LSDB). LSDB is collection of all LSAs received by a router. Every LSA has a sequence number. OSPF stores LSA in LADB with this sequence number.

DBDs

Database description packets (also referred as DDPs) contain the list of LSA. This list includes link state type, cost of link, ID of advertising router and sequence number of link. Make sure you understand this term correctly. It is only a list of all LSAs from its respective database. It does not include full LSAs.

In this state, routers exchanges DBDs. Through DBDs routers can learn which LSAs they already have. For example in following network R1 has A1, A2 and B2 LSAs in its LADB. So it will send a list of these LSAs to R2. This list is a DBDs. R2 will send an acknowledgment of receiving the list with LSACK signal. Same as R2 will send its DBDs to R1 and R1 would acknowledge that with its LSACK single.

LSR

Upon receiving DBDs, routers will compare it with their own LADB. Thus they will learn what they need to order. For example R1 received a check list (DBDs) of A1 and B1. When it will compare this list with its own LSA database (LADB), it will learn that it already has A1. So it does not need to order this LSA again. But it does not have B1, so it needs to order for this LSA. After a complete comparison, both routers will prepare a list of LSAs which they do not have in their own LADB. This list is known as LSR (Link State Request).

What other have (DBDs) – What I have (LADB) = What I need to order (LSR)



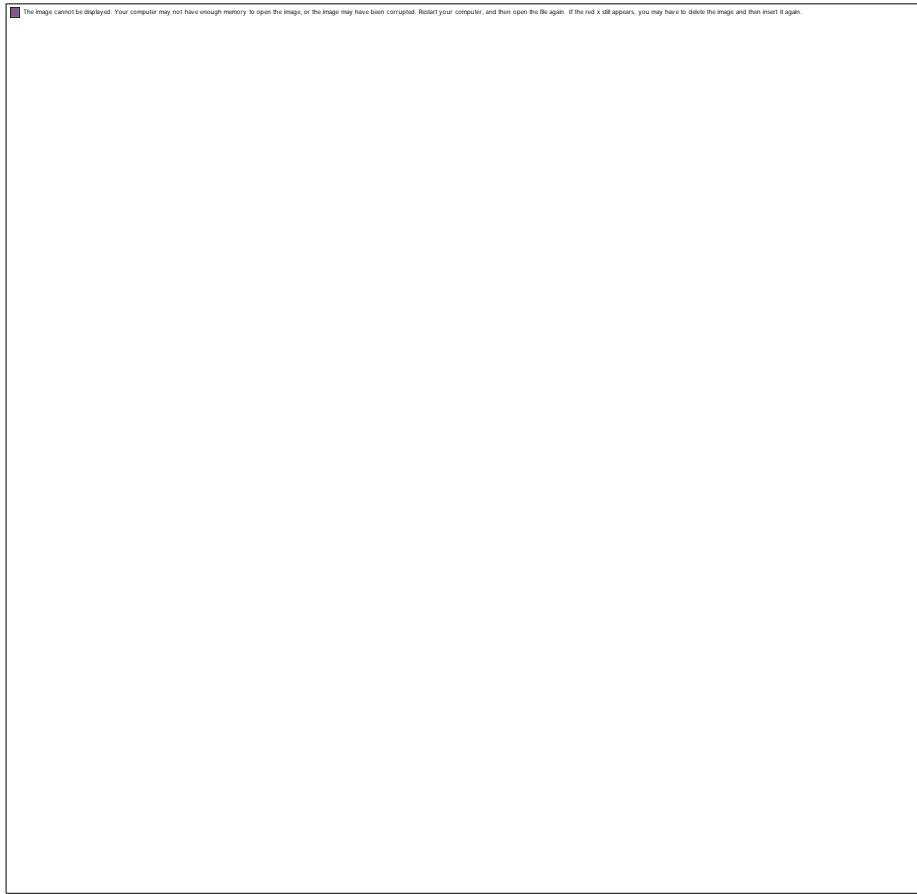
At the end of this state both routers have a list of LSAs which need to be exchanged.

Loading state

In this state actual routing information is exchanged. Routers exchange LSAs from LSR list.

Routers will use LSU (Link state update) to exchange the LSAs. Each LSA contains routing information about a particular link. Routers also maintain a retransmission list to make sure that every sent LSA is acknowledged.

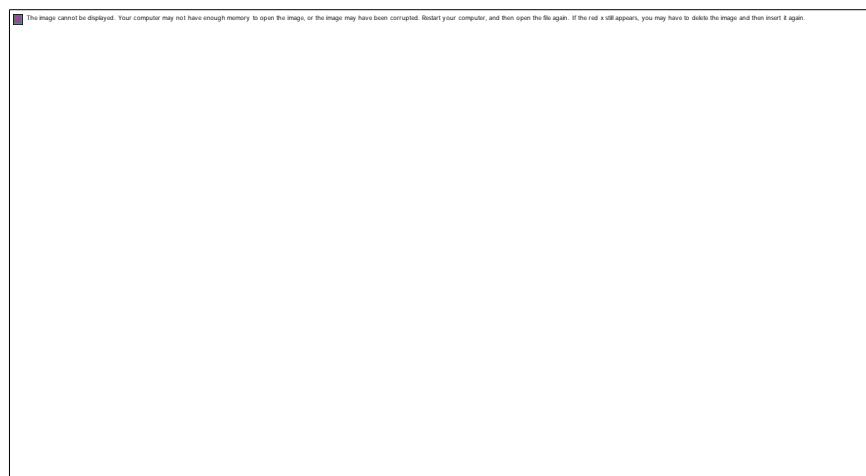
For example following figure illustrates loading state of above example. R1 sent a LSU which contain two LSAs but it received acknowledgement of only one, so it had to resend lost LSA again.



This exchange process will continue till router has any unsent LSA in LSR list.

Full state

Full state indicates that both routers have exchanged all LSAs from LSR list. Now they have identical LSDB.



Adjacent routers remain in this state for life time. This state also referred as adjacency. If any change occurs in network, routers will go through this process again.

Maintaining adjacency

- Routers will send hello messages in hello interval.
- If a router does not receive hello message from neighbor in dead interval, it will declare that neighbor as dead.
- Once a neighbor is dead, router will flood this change to other connected neighbors.
- Beside this if router detect any change in network or receive any update, it will flood that change.
- A LSA has a default lifetime of 30 minutes. Any unchanged LSAs must be reflooded in every 30 minutes.

That's all for this part. In next part, I will explain configuration part of OSPF.

To keep this tutorial simple, I used terms *neighbor* and *adjacencies* synonymously. Technically both terms are related but have different meanings especially in OSPF. Neighboring routers are defined in RFC 2328.

Neighboring routers are the routers that have interfaces in common network.

Adjacency is a relationship formed between neighboring routers for the purpose of exchanging routing information. Not every pair of neighboring routers becomes adjacent.

OSPF Metric Cost Calculation Formula Explained

OSPF uses SPF (Shortest Path First) algorithm to select the best route for routing table. SPF algorithm was invented in 1956 by Edsger W. Dijkstra. It is also referred as Dijkstra algorithm. SPF is a quite complex algorithm. In this tutorial we will explain a simplified overview of this algorithm.

Shortest Path First (SPF) Algorithm

As we know upon initialization or due to any change in routing information an OSPF router generates a LSA (Link State Advertisement) contains the collection of all link-states on that router. Router propagates this LSA in network. Each router that receives this LSA would store a copy of it in its LSA database then flood this LSA to other routers. After database is updated, router selects a single best route for each destination from all available routes. Router uses SPF algorithm to select the best route.

Just like other routing algorithm SPF also uses a metric component called cost to select the best route for routing table.

OSPF Metric cost

Logically a packet will face more overhead in crossing a 56Kbps serial link than crossing a 100Mbps Ethernet link. Respectively it will take less time in crossing a higher bandwidth link than a lower bandwidth link. OSPF uses this logic to calculate the cost. Cost is the inverse proportional of bandwidth. Higher bandwidth has a lower cost. Lower bandwidth has a higher cost.

OSPF uses following formula to calculate the cost

$$\text{Cost} = \frac{\text{Reference bandwidth}}{\text{Interface bandwidth in bps}}$$

Reference bandwidth was defined as arbitrary value in OSPF documentation (RFC 2338). Vendors need to use their own reference bandwidth. Cisco uses 100Mbps (10^8) bandwidth as reference bandwidth. With this bandwidth, our equation would be

$$\text{Cost} = \frac{10^8}{\text{Interface bandwidth in bps}}$$

Key points

- Cost is a positive integer value.
- Any decimal value would be rounded back in nearest positive integer.
- Any value below 1 would be considered as 1.

Now we know the equation, let's do some math and figure out the default cost of some essential interfaces.

Default cost of essential interfaces.

Interface Type	bandwidth	Metric Calculation	Cost
Ethernet Link	10Mbps	$100000000 / 10000000 = 10$	10
FastEthernet Link	100Mbps	$100000000 / 100000000 = 1$	1
Serial Link	1544Kbps(default)	$100000000 / 1544000 = 64.76$	64

Cost of common lines

Line	Bandwidth	Metric calculation	Cost
56 Kbps line	56Kbps	$100000000 / 56000 = 1785.71$	1785
64 Kbps line	64Kbps	$100000000 / 64000 = 1562.5$	1562
128 Kbps line	128Kbps	$100000000 / 128000 = 781.25$	781
512 Kbps line	512 Kbps	$100000000 / 512000 = 195.31$	195
1 Mbps line	1Mbps	$100000000 / 1000000 = 100$	100
10 Mbps line	10Mbps	$100000000 / 10000000 = 10$	10
100 Mbps line	100Mbps	$100000000 / 100000000 = 1$	1
1 Gbps line	1Gbps	$100000000 / 100000000 = 0.1$	1
10 Gbps line	10Gbps	$100000000 / 1000000000 = 0.01$	1

SPT (Shortest Path Tree)

OSPF router builds a Shortest Path Tree. SPT is just like a family tree where router is the root and destination networks are the leaves. SPF algorithm calculates the branch cost between leaves and root. Branch with lowest cost will be used to reach at leaf. In technical language route that has lowest cumulative cost value between source and destination will be selected for routing table.

Cumulative cost = Sum of all outgoing interfaces cost in route

Best route for routing table = Route which has the lowest cumulative cost

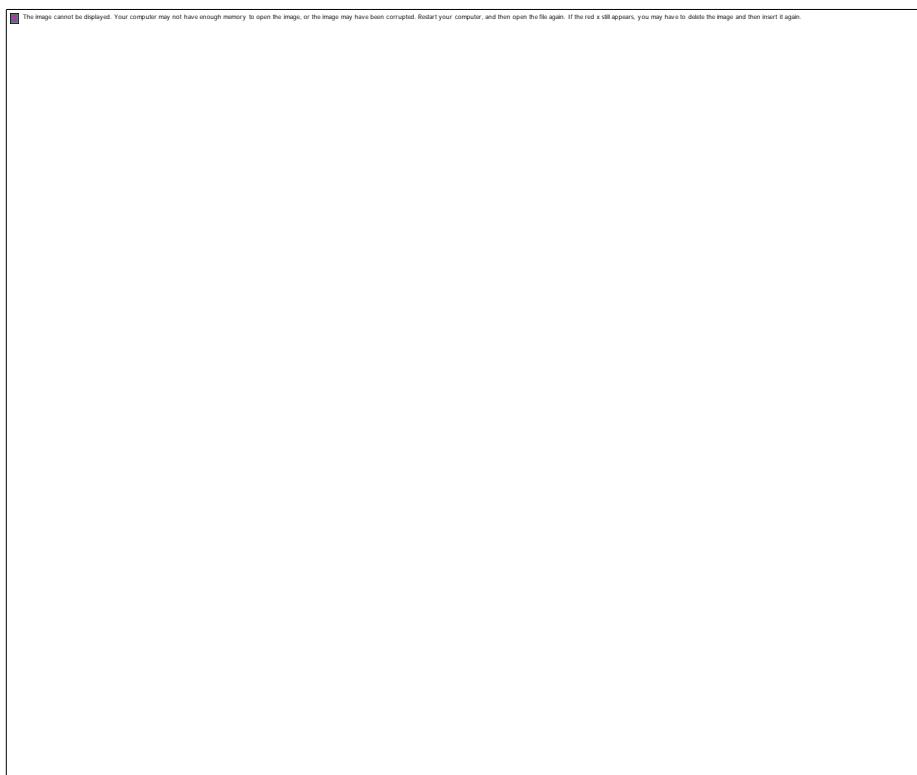
Summary

- OSPF uses SPT tree to calculate the best route for routing table.
- A SPT tree cannot grow beyond the area. So if a router has interfaces in multiple areas, it needs to build separate tree for each area.
- SPF algorithm calculates all possible routes from source router to destination network.
- Cumulative cost is the sum of the all costs of the outgoing OSPF interfaces in the path.
- While calculating cumulative cost, OSPF consider only outgoing interfaces in path. It does not add the cost of incoming interfaces in cumulative cost.
- If multiple routes exist, SPF compares the cumulative costs. Route which has the lowest cumulative cost will be chosen for routing table.

Now we have a basic understanding of SPF algorithm. In remaining part this tutorial we will explain how SPF algorithm selects the best route from available routes.

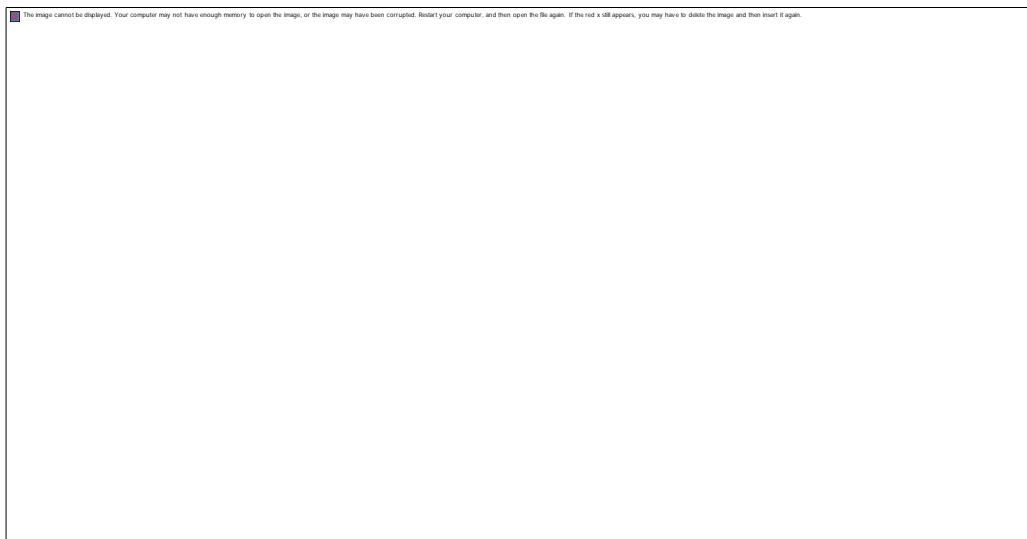
For demonstration purpose we will use same network topology which we have created in previous part of this article.

Following figure illustrates that network topology.



Open this topology in packet tracer and access CLI prompt of Router0.

Run **show ip route ospf** command from privilege mode to view all learned routes through the OSPF protocol.



As output shows, Router0 has six routes from OSPF in routing table. We will go through the each route and find out why it was chosen as the best route for routing table by OSPF.

Route 20.0.0.0

We have three routes to get 20.0.0.0/8 network. Let's calculate the cumulative cost of each route.

Via Route R0-R1-R2-R6

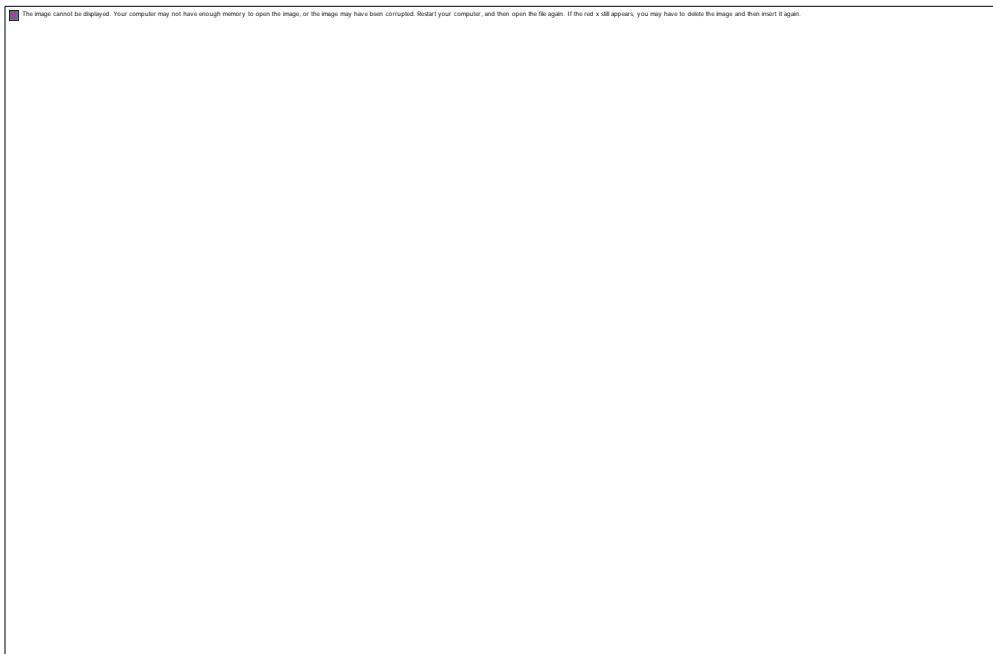
Router	Exit Interface	Bandwidth	Metric Calculation	Cost
R0	Se0/0/0	64Kbps (Manually Assigned)	$100000000 / 64000 = 1562.5$	1562
R1	Se0/0/1	64Kbps (Manually Assigned)	$100000000 / 64000 = 1562.5$	1562
R2	Se0/0/0	64Kbps (Manually Assigned)	$100000000 / 64000 = 1562.5$	1562
R6	Fa0/1	100Mbps	$100000000 / 100000000 = 1$	1
Cumulative cost of route (1562 + 1562 + 1562 + 1) = 4687				

Via route R0 – R3 – R4 – R6

Router	Exit Interface	Bandwidth	Metric Calculation	Cost
R0	Se0/0/1	1544Kbps (Default)	$100000000 / 1544000 = 64.76$	64
R3	Se0/0/0	1544Kbps (Default)	$100000000 / 1544000 = 64.76$	64
R2	Se0/0/1	1544Kbps (Default)	$100000000 / 1544000 = 64.76$	64
R6	Fa0/1	100Mbps	$100000000 / 100000000 = 1$	1
Cumulative cost of route (64 + 64 + 64 + 1) = 193				

Via route R0 – R5 – R6

Router	Exit Interface	Bandwidth	Metric Calculation	Cost
R0	Fa0/1	100Mbps	$100000000 / 100000000 = 1$	1
R5	Fa0/0	100Mbps	$100000000 / 100000000 = 1$	1
R0	Fa0/1	100Mbps	$100000000 / 100000000 = 1$	1
Cumulative cost of route (1+ 1 + 1) = 3				



Among these routes, route R0-R5-R6 has the lowest cumulative cost. So it was selected as the best route for routing table.

Route 192.168.0.4

Via Route R0 – R1

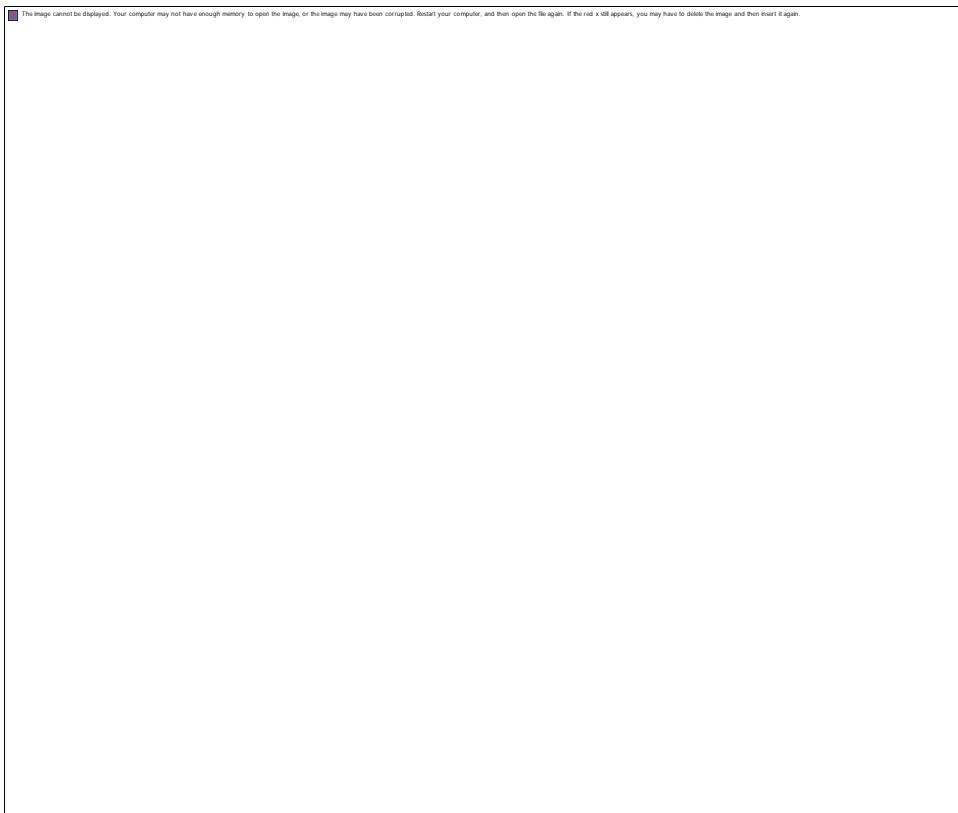
R0's Serial 0/0/0 cost (1562) + R1's Serial 0/0/1 cost (1562) = 3124 (Cumulative cost)

Via Route R0 – R3 – R4 – R6 – R2

R0's Serial 0/0/1 cost (64) + R3's Serial 0/0/0 cost (64) + R4's Serial 0/0/1 cost (64) + R6's Serial 0/0/0 cost (64) + R2's Serial 0/0/1 cost (64) = 320 (Cumulative cost)

Via Route R0 – R5 – R6 – R2

R0's FastEthernet 0/1 cost (1) + R5's FastEthernet 0/0 cost (1) + R6's Serial 0/0/0 cost (64) + R2's Serial 0/0/1 cost (64) = 130 (Cumulative cost)



Among these routes, Route R0 – R5 – R6 – R2 has the lowest cost so it was picked for routing table.

Route 192.168.0.8

Via Route R0 – R1

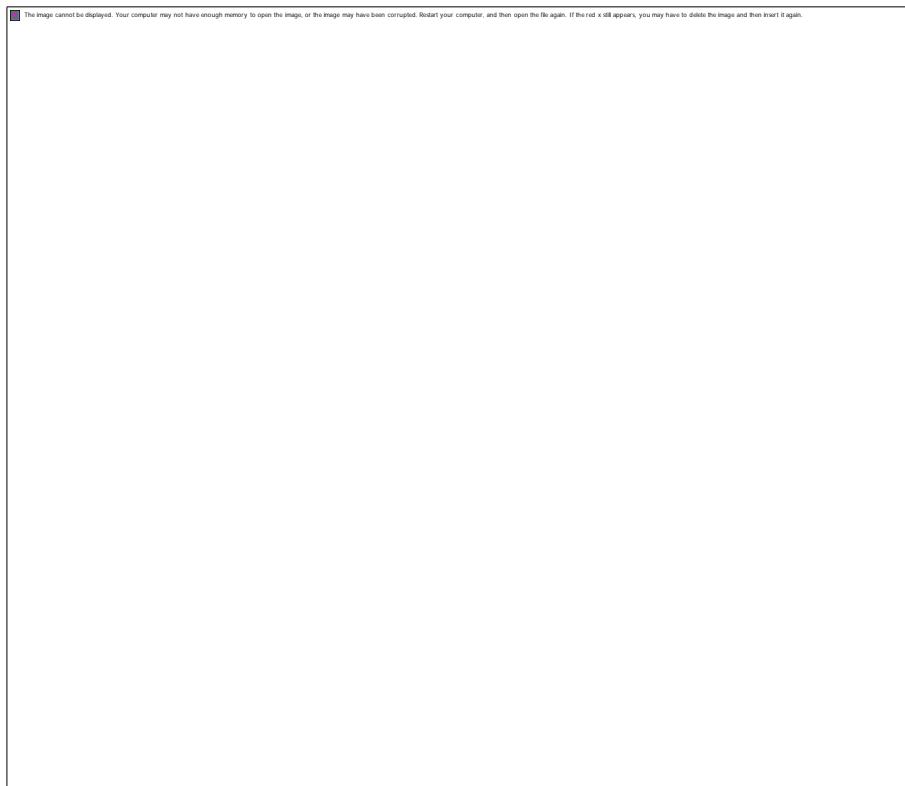
R0's Serial 0/0/0 cost (1562) + R1's Serial 0/0/1 cost (1562) + R2's Serial 0/0/0 (1562) = 4686
(Cumulative cost)

Via Route R0 – R3 – R4 – R6

R0's Serial 0/0/1 cost (64) + R3's Serial 0/0/0 cost (64) + R4's Serial 0/0/1 cost (64) + R6's Serial 0/0/0 cost (64) = 256 (Cumulative cost)

Via Route R0 – R5 – R6

R0's FastEthernet 0/1 cost (1) + R5's FastEthernet 0/0 cost (1) + R6's Serial 0/0/0 cost (64) = 66
(Cumulative cost)



Among these routes, Route R0 – R5 – R6 has the lowest cost so it was picked for routing table.

Route 192.168.1.4

Via Route R0 – R1 – R2 – R6

R0's Serial 0/0/0 cost (1562) + R1's Serial 0/0/1 (1562) + R2's Serial 0/0/0 (1562) + R6's FastEthernet 0/0 (1) = 4687 (Cumulative cost)

Via R0 – R3 – R4 – R6

R0's Serial 0/0/1 cost (64) + R3's Serial 0/0/0 cost (64) + R4's Serial 0/0/1 cost (64) + R6's FastEthernet 0/0 (1) = 193

Via R0 – R5

R0's FastEthernet 0/1 cost (1) + R5's FastEthernet 0/0 cost (1) = 2



Among these routes, Route R0 – R5 has the lowest cost so it was selected as the best route.

Route 192.168.2.4

Via Route R0 – R1 – R2 – R6 – R4

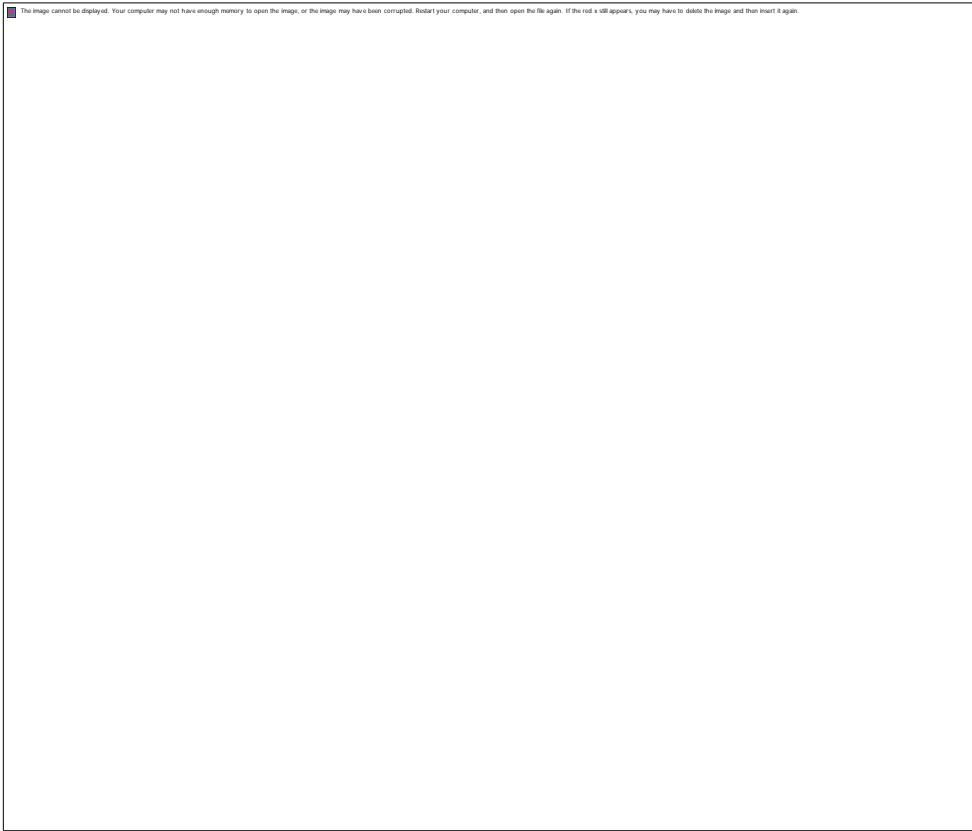
R0's Serial 0/0/0 cost (1562) + R1's Serial 0/0/1 cost (1562) + R2's Serial 0/0/0 cost (1562) + R6's Serial 0/0/1 cost (64) + R4's Serial 0/0/0 cost (64) = 4814

Via Route R0 – R5 – R6 – R4

R0's FastEthernet 0/1 cost (1) + R5's FastEthernet 0/0 cost (1) + R6's Serial 0/0/1 (64) + R4's Serial 0/0/0 cost (64) = 130

Via Route R0 – R3

R0's Serial 0/0/1 cost (64) + R3's serial 0/0/0 cost (64) = 128



Among these routes, Route R0 - R3 has the lowest cost for destination 192.168.2.4.

Route 192.168.2.8

Via Route R0 – R3 – R4

R0's Serial 0/0/1 cost (64) + R3's Serial 0/0/0 cost (64) + R4's Serial 0/0/1 cost (64) = 192

Via Route R0 – R1 – R2 – R6

R0's Serial 0/0/0 cost (1562) + R1's Serial 0/0/1 cost (1562) + R2's Serial 0/0/0 cost (1562) + R6's Serial 0/0/1 cost (64) = 4750

Via Route R0 – R5 – R6

R0's FastEthernet 0/1 cost (1) + R5's FastEthernet 0/0 cost (1) + R6's Serial 0/0/1 cost (64) = 66

 The image cannot be displayed. Your computer may not have enough memory to open the image, or the image may have been corrupted. Restart your computer, and then open the file again. If the red x still appears, you may have to delete the image and then insert it again.

Route R0 – R5 – R6 has the lowest cost value.

After selecting best route for each destination OSPF network look likes following figure.

 The image cannot be displayed. Your computer may not have enough memory to open the image, or the image may have been corrupted. Restart your computer, and then open the file again. If the red x still appears, you may have to delete the image and then insert it again.

OSPF Route cost Manipulation

We can manipulate OSPF route cost in two ways.

1. By changing bandwidth of interface
2. By changing reference bandwidth value

By changing bandwidth of interface

Sub interface mode command **Bandwidth** is used to set the bandwidth of supported interface.

If bandwidth is set through this command, OSPF will use it. If bandwidth is not set, it will use interface's default bandwidth.

When we enable an interface, router automatically assign a bandwidth value to it based on its type. For example serial interface has a default bandwidth value of 1544k. Until we change this value with bandwidth command, it will be used where it is required.

Let me clear one more thing about bandwidth. Changing default bandwidth with bandwidth command does not change actual bandwidth of interface. Neither default bandwidth nor bandwidth set by bandwidth command has anything to do with actual layer one link bandwidth.

Then what purpose does this command solve?

This command is only used to influence the routing protocol which uses bandwidth in route selection process such as OSPF and EIGRP.

We have already seen an example of this method in our example. We changed default bandwidth (1544Kbps) to custom (64kbps) bandwidth on R0's serial 0/0/0, R1's serial 0/0/1 and R2's serial 0/0/0. Due to this change R0 took another router for 192.168.0.4 network.

Let's understand this in more detail.

Current cost for destination 192.168.0.4 from R0

Via Route R0 – R1

R0's Serial 0/0/0 cost (1562) + R1's Serial 0/0/1 cost (1562) = 3124 (Cumulative cost)

Via Route R0 – R5 – R6 – R2

R0's FastEthernet 0/1 cost (1) + R5's FastEthernet 0/0 cost (1) + R6's Serial 0/0/0 cost (64) + R2's Serial 0/0/1 cost (64) = 130 (Cumulative cost)

Via Route R0 – R3 – R4 – R6 – R2

R0's Serial 0/0/1 cost (64) + R3's Serial 0/0/0 cost (64) + R4's Serial 0/0/1 cost (64) + R6's Serial 0/0/0 cost (64) + R2's Serial 0/0/1 cost (64) = 320 (Cumulative cost)

Among these routes, Route R0 – R5 – R6 – R2 has the lowest cost so it was picked for routing table.

Well ... Which route would have selected, if we had used default bandwidth?

Cost for destination 192.168.0.4 from R0 with default bandwidth.

Via Route R0 – R1

R0's Serial 0/0/0 cost (64) + R1's Serial 0/0/1 cost (64) = 128 (Cumulative cost)

Via Route R0 – R5 – R6 – R2

R0's FastEthernet 0/1 cost (1) + R5's FastEthernet 0/0 cost (1) + R6's Serial 0/0/0 cost (64) + R2's Serial 0/0/1 cost (64) = 130 (Cumulative cost)

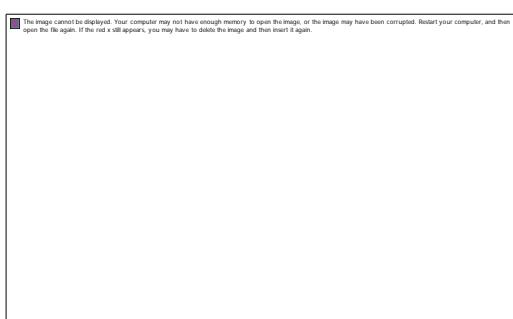
Via Route R0 – R3 – R4 – R6 – R2

R0's Serial 0/0/1 cost (64) + R3's Serial 0/0/0 cost (64) + R4's Serial 0/0/1 cost (64) + R6's Serial 0/0/0 cost (64) + R2's Serial 0/0/1 cost (64) = 320 (Cumulative cost)

Among these routes, Route R0 – R1 has the lowest cost value so it would be selected for routing table. Thus by changing interface bandwidth we actually influenced route selection process.

By changing reference bandwidth value

As I mention earlier, by default OSPF uses 100Mbps bandwidth as a reference bandwidth. Changing this value would also change the cost of route. If we use 1000Mbps as a reference bandwidth, cost of 100Mbps link would become 10. This sounds great, especially if we have higher bandwidth links in our network. For example have a look on following figure.



Which route will R2 take to get the network of 10.0.0.0/8?

Route R2 – R3

In this route we have two exit points. Both points have default 100 Mbps speed.

R2's FastEthernet cost ($100000000/100000000$) = 1

R3's FastEthernet cost ($100000000/100000000$) = 1

Cost of this route 1 + 1 = 2

Route R2 – R1 – R3

In this route we have three exit points. Two exit points (R2 and R1) have 1 Gbps link.

R2's FastEthernet cost ($100000000/100000000$) = .1 (Anything below 1 would be considered as 1)

R3's FastEthernet cost ($100000000/100000000$) = .1 (Anything below 1 would be considered as 1)

R3's FastEthernet cost ($100000000/100000000$) = 1

Cost of this route 1 + 1 + 1 = 3

With default reference bandwidth R2 will choose Route R2 – R3, which is not good.

We can adjust reference bandwidth with **auto-cost reference-bandwidth ref-band** command.

We need to adjust reference bandwidth on all routers of network. Mismatched reference bandwidth can cause routers to run the SPF algorithm continually, which could create a serious performance issue.

Reference bandwidth is assigned in Mbps. Valid range is 1 to 4294967. Default reference bandwidth is 100Mbps.

Sadly packet tracer does not include this command. For the practice of this command please use other simulator software which support this command or use real router.

Let's change reference bandwidth to 1000Mbps on all three routers using following commands

```
Router# configure terminal  
Enter configuration commands, one per line. End with CNTL/Z.  
Router (config)#router ospf 1  
Router (config-router)#auto-cost reference-bandwidth 1000  
% OSPF: Reference bandwidth is changed.  
    Please ensure reference bandwidth is consistent across all routers.  
Router (config-router)#exit  
Router #
```

Route cost with new reference bandwidth

Route R2 – R3

R2's FastEthernet cost ($1000000000/100000000$) = 10

R3's FastEthernet cost ($1000000000/100000000$) = 10

Cost of this route $10 + 10 = 20$

Route R2 – R1 – R3

R2's FastEthernet cost ($1000000000/1000000000$) = 1

R3's FastEthernet cost ($1000000000/1000000000$) = 1

R3's FastEthernet cost ($1000000000/1000000000$) = 10

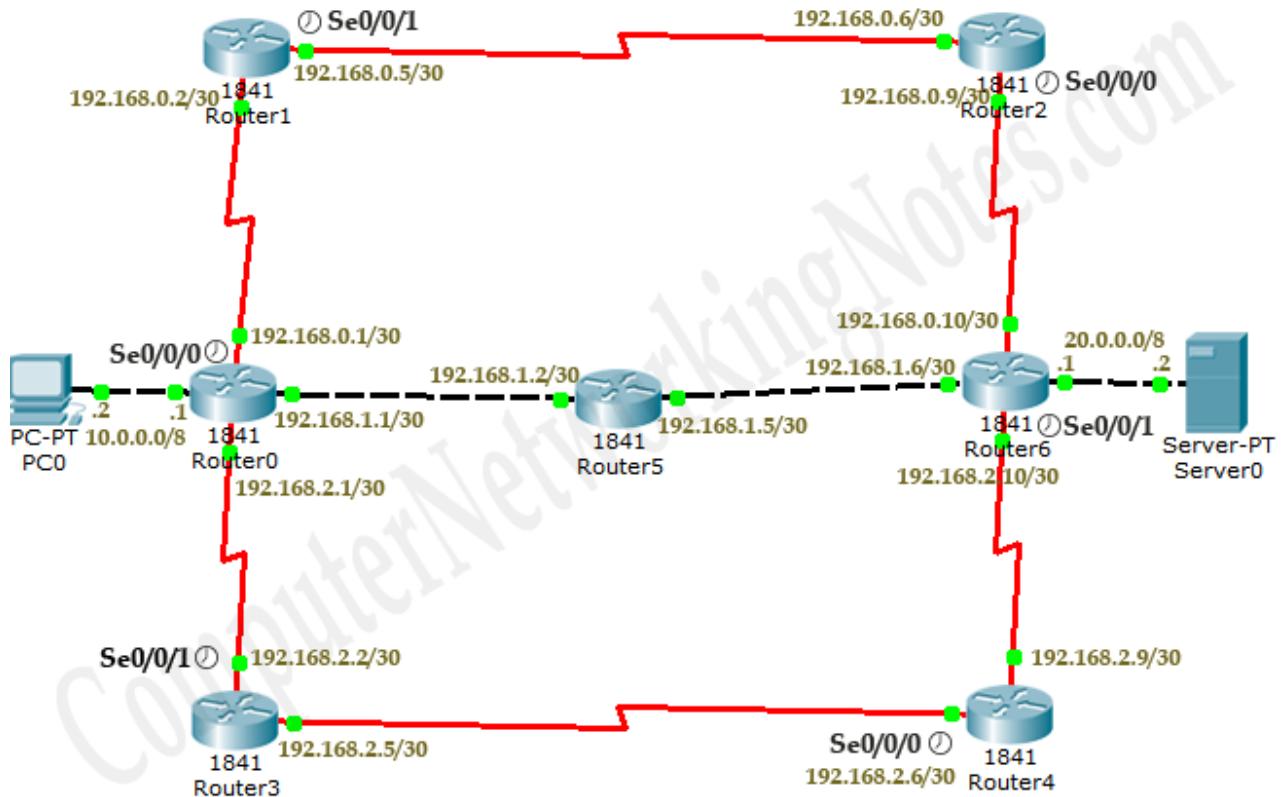
Cost of this route $1 + 1 + 10 = 12$

In this case Route R2-R1-R3 will be selected, which is the shortest route for destination.

That's all for this article. I hope now you have better understanding of OSPF Routing protocol

Configuration of OSPF Routing Protocol

Create a topology as illustrate in following figure.

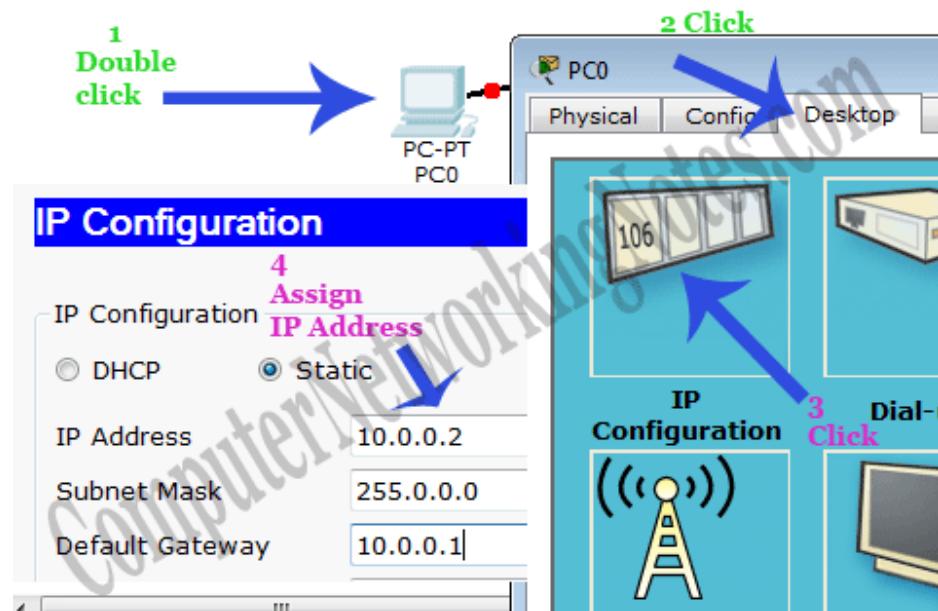


Device	Interface	IP Configuration	Connected with
PC0	Fa0/0	10.0.0.2/8	Router0's Fa0/0
Router0	Fa0/0	10.0.0.1/8	PC0's Fa0/0
Router0	Fa0/1	192.168.1.1/30	Router5's Fa0/1
Router5	Fa0/1	192.168.1.2/30	Router0's Fa0/1
Router5	Fa0/0	192.168.1.5/30	Router6's F0/0
Router6	Fa0/0	192.168.1.6/30	Router5's Fa0/0
Router6	Fa0/1	20.0.0.1/8	Server0's Fa0/0
Server0	Fa0/0	20.0.0.2/8	Router6's Fa0/1
Router0	Serial 0/0/0 (DCE)	192.168.0.1/30	Router1's Se0/0/0
Router1	Serial 0/0/0	192.168.0.2/30	Router0's Se0/0/0
Router1	Serial 0/0/1 (DCE)	192.168.0.5/30	Router2's Se0/0/1
Router2	Serial0/0/1	192.168.0.6/30	Router1's Se0/0/1
Router2	Serial 0/0/0 (DCE)	192.168.0.9/30	Router6's Se0/0/0
Router6	Serial 0/0/0	192.168.0.10/30	Router2's Se0/0/0
Router0	Serial 0/0/1	192.168.2.1/30	Router3's Se0/0/1
Router3	Serial 0/0/1 (DCE)	192.168.2.2/30	Router0's Se0/0/1

Router3	Serial 0/0/0	192.168.2.5/30	Router4's Se0/0/0
Router4	Serial 0/0/0 (DCE)	192.68.2.6/30	Router3's Se0/0/0
Router4	Serial 0/0/1	192.168.2.9/30	Router6's Se0/0/1
Router6	Serial0/0/1 (DCE)	192.168.2.10/30	Router4's Se0/0/1

Assign IP address to PC

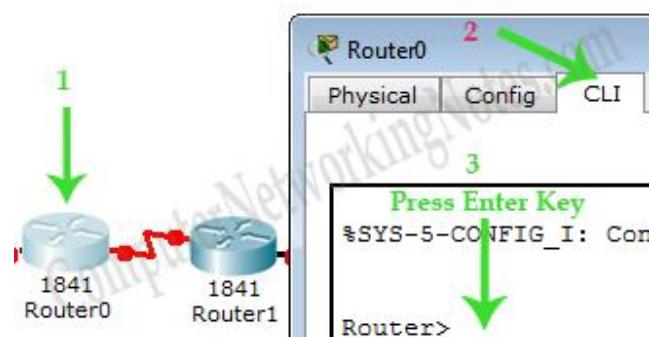
Double click **PC0** and click **Desktop** menu item and click **IP Configuration**. Assign IP address **10.0.0.2/8** to **PC0**.



Repeat same process for Server0 and assign IP address 20.0.0.2/8.

Assign IP address to interfaces of routers

Double click **Router0** and click **CLI** and press Enter key to access the command prompt of **Router0**.



Four interfaces FastEthernet0/0, FastEthernet0/1, Serial 0/0/0 and Serial0/0/1 of Router0 are used in this topology. 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)#
```

From global configuration mode we can enter in interface mode. From there we can configure the interface. Following commands will assign IP address on FastEthernet0/0 and FastEthernet0/1.

```
Router(config)#interface fastEthernet 0/0  
Router(config-if)#ip address 10.0.0.1 255.0.0.0  
Router(config-if)#no shutdown  
Router(config-if)#exit  
Router(config)#interface fastEthernet 0/1  
Router(config-if)#ip address 192.168.1.1 255.255.255.252  
Router(config-if)#no shutdown  
Router(config-if)#exit  
Router(config)#
```

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

ip address 10.0.0.1 255.0.0.0 command would assign IP address to interface.

no shutdown command would 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.

```
Router#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 interfaces.

```
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 192.168.0.1 255.255.255.252
Router(config-if)#clock rate 64000
Router(config-if)#bandwidth 64
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#interface serial 0/0/1
Router(config-if)#ip address 192.168.2.1 255.255.255.252
Router(config-if)#no shutdown
Router(config-if)#exit
```

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 192.168.0.1 255.255.255.252 Command assigns IP address to interface. For serial link we usually use IP address from /30 subnet.

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 clock rate here.

Router(config-if)#bandwidth 64 Bandwidth works as an influencer. It is used to influence the metric calculation of OSPF or any other routing protocol which uses bandwidth parameter in

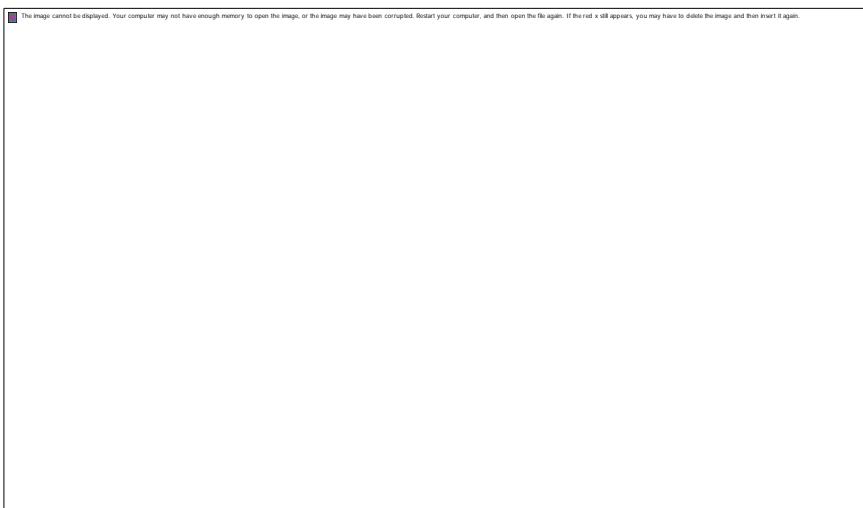
route selection process. Serial interface has default bandwidth of 1544Kbps. To explain, how bandwidth influence route selection process we will configure (64Kbps) bandwidth on three serial DCE interfaces of our network; R0's Se0/0/0, R1's Se0/0/1 and R2's Se0/0/0.

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 remaining routers.

Router1

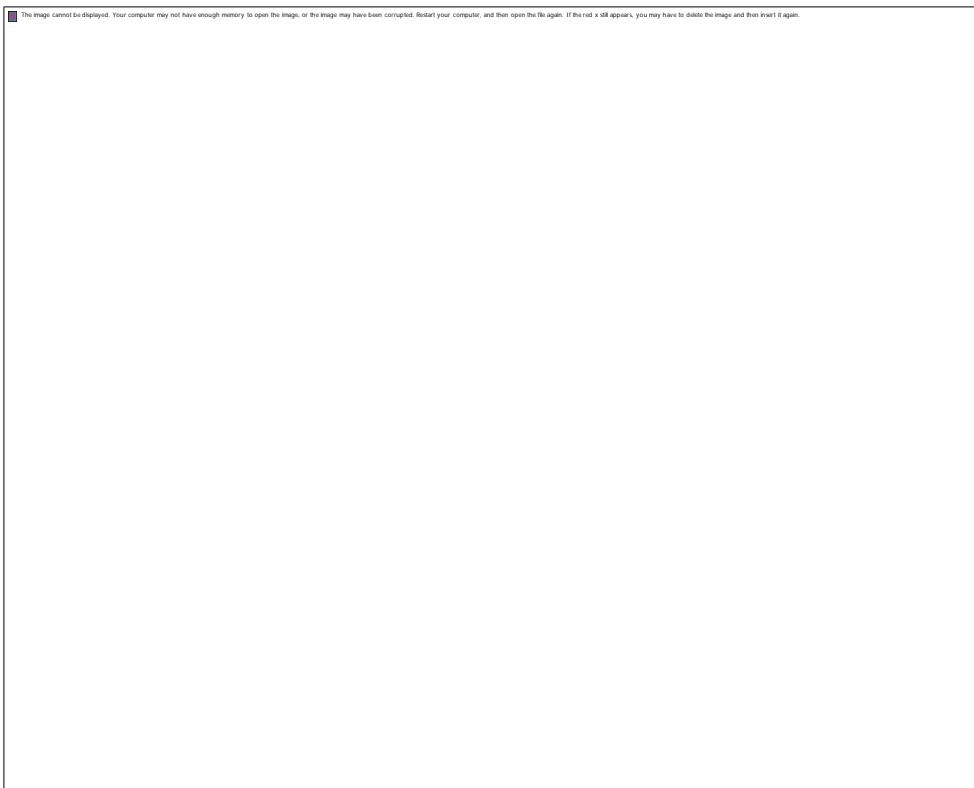


Router2

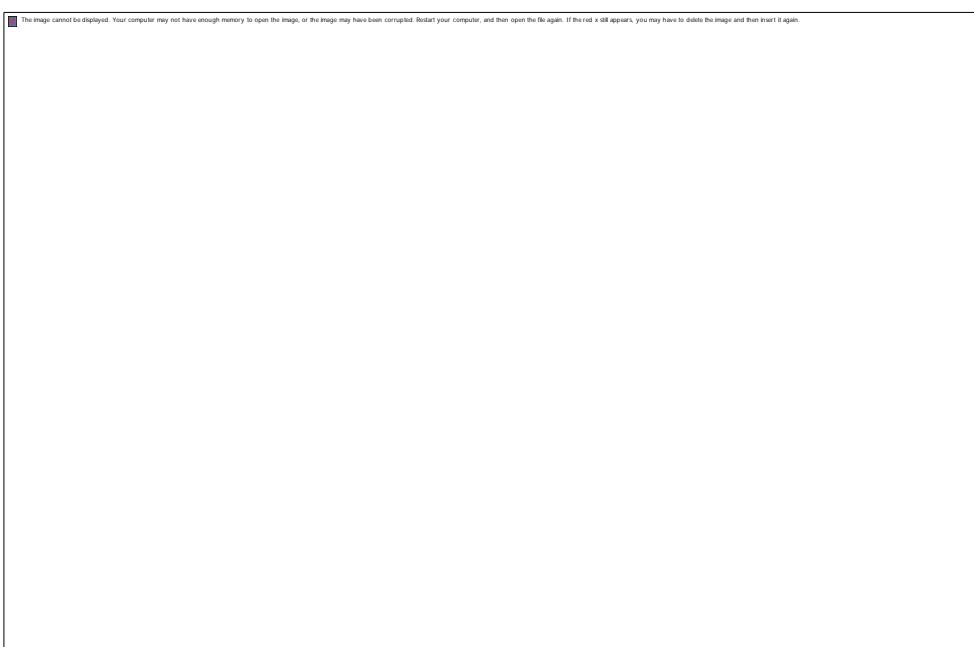


As I mention earlier, serial interface has a default bandwidth of 1544Kbps. If we don't assign any custom bandwidth, router would use default bandwidth. To see this feature in action we will not assign bandwidth on remaining routers.

Router6



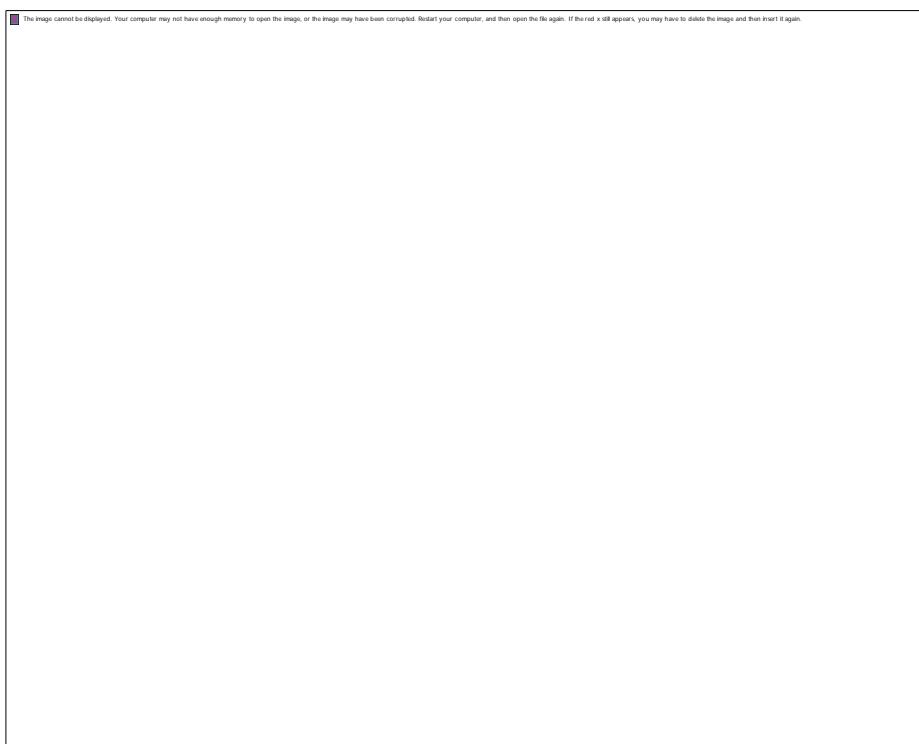
Router5



Router3



Router4



Great job we have finished our half journey. Now routers have information about the networks that they have on their own interfaces. Routers will not exchange this information between them on their own. We need to implement OSPF routing protocol that will insist them to share this information.

To be on same track I have uploaded my practice topology. Use this if you want to skip above IP configuration part.

Configure OSPF routing protocol

Enabling OSPF is a two steps process:-

- Enable OSPF routing protocol from global configuration mode.
- Tell OSPF which interfaces we want to include.

For these steps following commands are used respectively.

```
Router(config)# router ospf process_ID  
Router(config-router)# network IP_network_# [wild card mask] Area Number area number
```

Router(config)# router ospf process ID

This command will enable OSPF routing protocol in router. Process ID is a positive integer. We can use any number from 1 to 65,535. Process ID is locally significant. We can run multiple OSPF process on same router. Process ID is used to differentiate between them. Process ID need not to match on all routers.

Router(config-router)# network IP_network_# [wildcard_mask] area [area number]

Network command allows us to specify the interfaces which we want to include in OSPF process. This command accepts three arguments network number, wildcard mask and area number.

Network number

Network number is network ID. We can use any particular host IP address or network IP address. For example we can use 192.168.1.1 (host IP address) or we can use 192.168.1.0 (Network IP address). While targeting a specific interface usually we use host IP address (configured on that interface).

While targeting multiple interfaces, we use network IP address. So any interface that belongs to specified network ID will be selected.

Wildcard mask

Wildcard mask are used with network ID to filter the interfaces. Wildcard mask is different from subnet mask. Subnet mask is used to separate the network portion and host portion in IP address. While wildcard mask is used to match corresponding octet in network portion. Wildcard mask tells OSPF the part of network address that must be matched. Wildcard masks are explained with examples in access list tutorials of this category.

Key points

0 (*Decimal – octet format*) Wildcard mask indicates that corresponding octet in network address must be matched exactly.

255 (*Decimal – octet format*) Wildcard mask indicates that we don't care about corresponding octet in network address.

For example

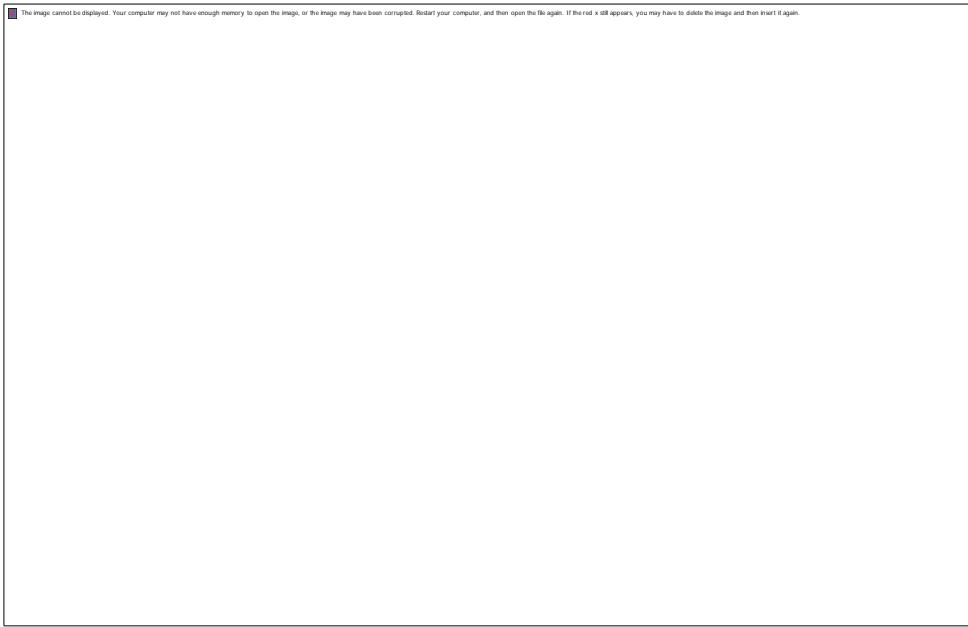


0 (*Binary – bit format*) Wildcard mask indicates that corresponding bit in network address must be matched exactly.

255 (*Binary – bit format*) Wildcard mask indicates that we don't care about corresponding bit in network address.



OSPF is a classless protocol. With wildcard we can also filter Subnetted networks. In classes implementation usually we use Subnetted networks. For example consider following figure



The image cannot be displayed. Your computer may not have enough memory to open the image, or the image may have been corrupted. Restart your computer, and then open the file again. If the red x still appears, you may have to delete the image and then insert it again.

We have four networks 172.168.1.0/24, 172.168.2.0/24, 172.168.3.0/24 and 172.168.4.0/24 subnetted from single class B network 172.168.0.0/16. Classful configuration does not understand the concept of subnetting. In classful configuration all these networks belong to a single network. Classful configuration works only with in default boundary of mask. Default boundary of this address is 16 bits. So a classful routing protocol will match only first 16 bits (172.168.x.y) of network address. A classful routing protocol such as RIP cannot distinguish between different Subnetted networks.

A classless routing protocol such as OSPF goes beyond the default boundary of mask and work well with Subnetted networks. With wildcard mask we can easily filter Subnetted networks.

With wildcard we are no longer limited with default boundaries. We can match Subnetted networks as well as default networks.

For example we want to exclude serial interfaces in above configuration. We can use a wildcard mask of 0.0.0.255 to match the subnet mask of /24.

```
Router(config-router)# network 172.168.1.0 0.0.0.255
Router(config-router)# network 172.168.2.0 0.0.0.255
```

Above commands will ask router to match /24 bits of address instead of default /16 bits. Now router will look for 172.168.1.x and 172.168.2.x network. Our serial interfaces have 172.168.3.0/24 and 172.168.4.0/24 networks which do not fall in these search criteria.

Let's take one more example, if we use following network command, which interfaces would be selected.

```
Router(config-router)# network 192.168.0.0 0.0.0.3
```



In this case valid host IP addresses are 192.168.0.1 and 192.168.0.2. So any interface that has these IP address would be selected. /30 network is usually used for serial link connection which need only two valid host IP addresses; one for each end.

If you are unfamiliar with wildcard mask, I suggest you to check our tutorials on access lists configuration in this category. In those tutorials wildcard masks are explained in detail with examples.

For this tutorial let's move on third argument. Third argument which network command accept is area number. This parameter say router to put matched interface in specified area. OSPF areas are explained in second part this article.

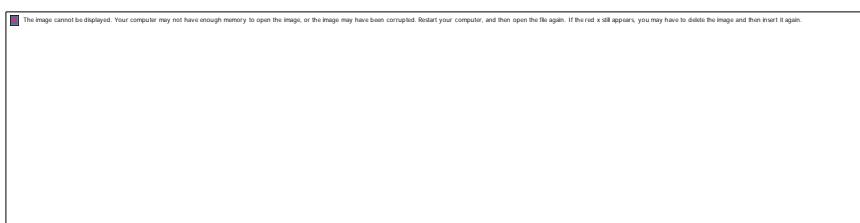
Now we know the essential commands for configuration. Let's implement them in our network.

OSPF configuration

Router0



Router1



Router2



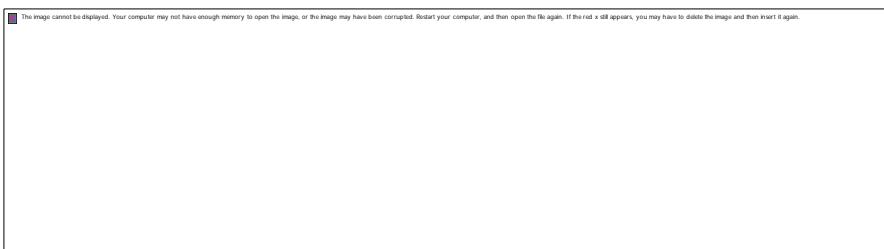
Router6



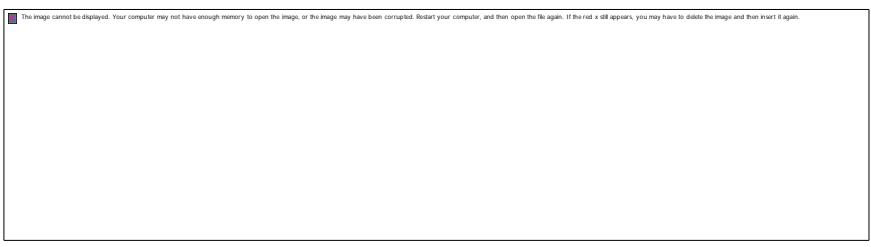
Router5



Router4



Router3



That's it. Our network is ready to take the advantage of OSPF routing. To verify the setup we will use **ping** command. ping command is used to test the connectivity between two devices. We have two routes between source and destination. tracert command is used to know the route which is used to get the destination.

Access the command prompt of PC1 and use ping command to test the connectivity from Server0. After that use **tracert** command to print the taken path.



Great! We have successfully implemented OSPF routing in our network. For cross check we have uploaded a configured topology on our server. You can use that if not getting same output.

Summary

Command	Description
Router(config)#router ospf 10	Enable OSPF routing protocol under process ID 10.
Router(config-router)#network 10.10.0.0 0.0.255.255 area 0	Enable OSPF with area 0 on matching interface.
Router(config)#interface loopback 0	Create a Loopback interface and move in sub interface configuration mode
Router(config-if)#ip address 192.168.250.250 255.255.255.0	Assign IP address to loopback interface.
Router(config-router)#router-id 1.1.1.1	Set 1.1.1.1 as router ID
Router(config)#interface serial 0/0	Enter in sub interface configuration mode

Router(config-if)#ip ospf priority 100	Used to influence DR/BDR selection process. Valid range is 0 to 255. 0 makes router ineligible for DR/BDR while 255 makes router guaranteed DR/BDR. Higher priority value means higher chance of becoming DR/BDR.
Router(config-if)#bandwidth 256	Used to influence route metric cost. Cost is the inverse of bandwidth. Higher bandwidth has lower cost. Bandwidth is defined in Kbps. 256 means 256 Kbps.
Router(config-if)#ip ospf hello-interval timer 15	Set hello interval timer to 15 seconds. Hello timer must be match on both routers in order become neighbors.
Router(config-if)#ip ospf dead-interval 60	Set dead interval timer to 60 seconds. Dead interval timer must be match on both routers in order to become neighbor
Router#show ip route	Display all routes from routing table
Router#show ip route ospf	Display all routers learned through OSPF from routing table
Router#show ip ospf	Display basic information about OSPF
Router#show ip ospf interface	Display information about all OSPF active interfaces
Router#show ip ospf interface serial 0/0/0	Display OSPF information about serial 0/0/0 interface
Router#show ip ospf neighbor List all	OSPF neighbors with basic info
Router#show ip ospf neighbor detail	List OSPF neighbors with detail info
Router#show ip ospf database	Display data for OSPF database
Router#clear ip route *	Clear all routes from routing table.
Router#clear ip route 10.0.0.0/8	Clear particular route from routing table
Router#clear ip ospf counters	Clear OSPF counters
Router#debug ip ospf events	Display all ospf events
Router#debug ip ospf packets	Display exchanged OSPF packets
Router#debug ip ospf adjacency	Display DR/BDR election process state

OSPF Fundamental Terminology

This explains basic concepts of OSPF including public AS number, private AS number, backbone area, ABR, IP, Link, state, LSA and LSDB in detail with examples.

OSPF stands for Open Shortest Path First. OSPF is a link state open standard based routing protocol. It was created in mid-1980. Since it is based on open standard, we can use it with any vendor's router.

Features and advantage of OSPF

- It supports both IPv4 and IPv6 routed protocols.
- It supports load balancing with equal cost routes for same destination.
- Since it is based on open standards, it will run on most routers.
- It provides a loop free topology using SPF algorithm.
- It is a classless protocol.
- It supports VLSM and route summarization.
- It supports unlimited hop counts.
- It scales enterprise size network easily with area concept.
- It supports trigger updates for fast convergence.

Just like other routing protocols, OSPF also has its negatives.

Disadvantage of OSPF

- It requires extra CPU process to run SPF algorithm.
- It requires more RAM to store adjacency topology.
- It is more complex to setup and hard to troubleshoot.

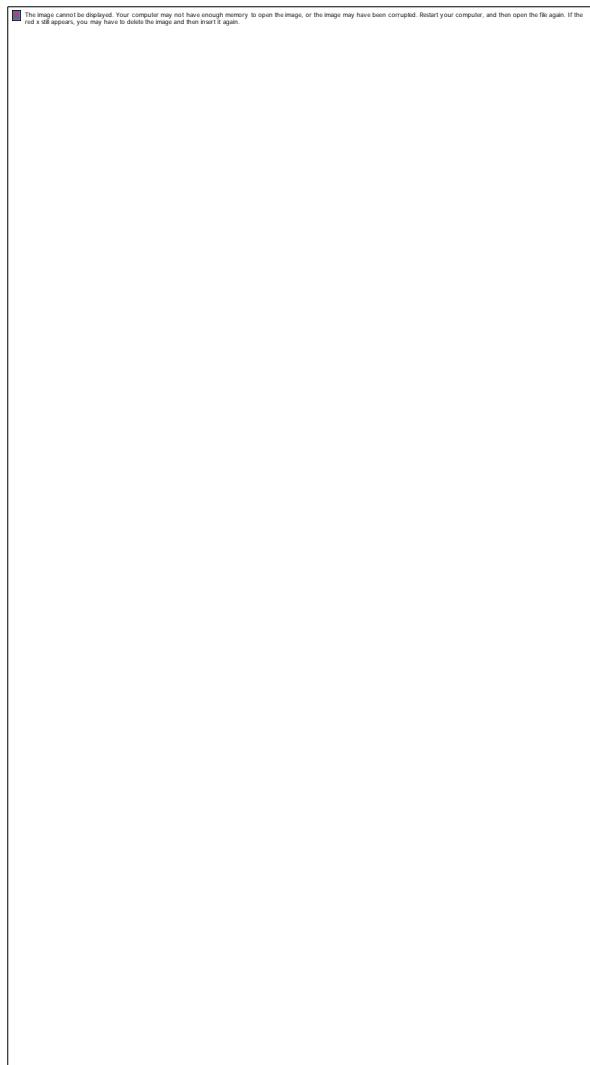
Basically OPSF was created to fulfill the requirement of enterprise size network. To scale a large size network it uses area concept. Area concept is similar to Subnetting. It allows us to separate the large internetwork into smaller networks known as areas.

Along with Area concept OSPF also supports Autonomous System (AS). Just like area, AS also divide a large network into smaller networks.

Difference between AS and Area concept

Area concept is a feature of OSPF. It is limited only with OSPF. We cannot use it with other routing protocol.

AS is an independent concept originally defined in RFC 1771. We can use it with any routing protocols which understand its concept.



Autonomous System

An AS is a group of networks running under a single administrative control. This could be our company or a branch of company. Just like Subnetting AS is also used to break a large network in smaller networks.

AS creates a boundary for routing protocol which allow us to control how far routing information should be propagated. Beside this we can also filter the routing information before sharing it with other AS system. These features enhance security and scalability of overall network.

Basically AS concept was developed for large networks. Routing protocols which were developed for small networks such as RIP do not understand the concept of AS systems.

There are two types of routing protocols IGP and EGP.

IGP (Interior Gateway Protocol) is a routing protocol that runs in a single AS such as RIP, IGRP, EIGRP, OSPF and IS-IS.

EGP (Exterior Gateway Protocol) is a routing protocol that performs routing between different AS systems. Nowadays only BGP (Border Gateway Protocol) is an active EGP protocol.

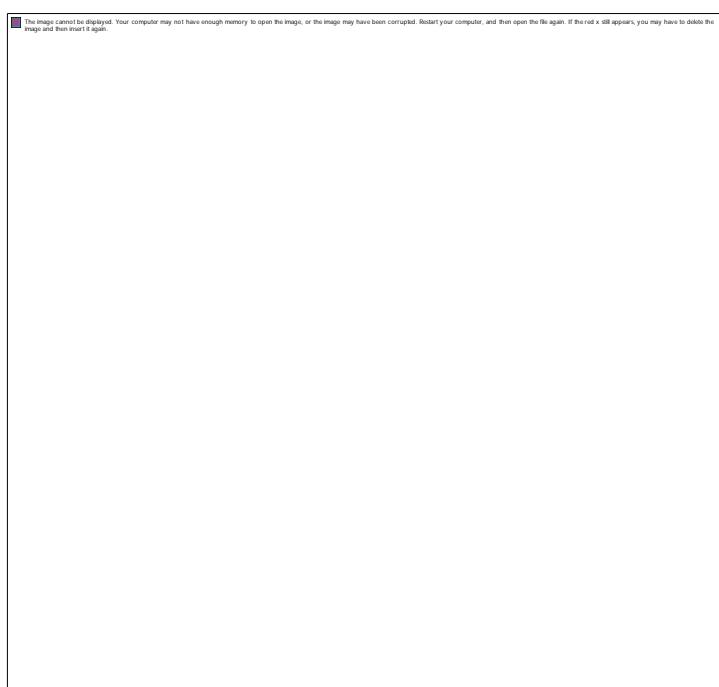
To keep distinguish between different autonomous systems, AS numbers are used. An AS number starts from 1 and goes up to 65535. Same as IP addresses, AS numbers are divided in two types; Private and public.

Public AS Numbers: - We only need to use public numbers if we are going to connect our AS with Internet backbone through the BGP routes from Internet. IANA (Numbers Authority) controls the public AS numbers.

Private AS Numbers: - Private AS numbers are used to break our internal network into the smaller networks. We can use any valid AS number in our network unless we connect it with public network. In above example we used two AS numbers 10 and 20 to divide our company network.

In OSPF implementation, routers which connect two different ASes are known as autonomous

Ok let's remove extra burden (ASes) from example network.



Area

OSPF implements two levels hierarchy with areas: backbone and area off backbone.

Backbone

Backbone is the central point of this implementation. Routers running in this area required to maintain a complete database of entire network. All areas need to connect with this area through a physical link or via a virtual link if physical link is not possible.

Area off backbone

Area off backbone is the extension of backbone. Routes running in this area required to maintain an area specific database instead of complete database. This is a cool feature. It will speed-up the convergence time.

ABR

Area Border Router (ABR) is a bridge between Backbone and Area off backbone. With correct IP addressing we can summarize routes information on this router.

IR

IR (Internal Router) is a router running in area off backbone. IR only needs to maintain an area centric local database.

Let's put all these together in a simple example. Assume that our company has 600 hosts. It decided to use three class C subnets; 192.168.0.0/24, 192.168.1.0/24 and 192.168.2.0/24. Each subnet has 200 hosts.

For easy administration these subnets are divided in smaller networks. With VLSM default subnet /24 is subnetted in /29. When we break a default class C Subnet /24 in Subnet /29, it produces 32 networks (8 hosts in each network).

In this situation if we use a classical flat network design, routers need to learn and advertise 96 ($32 + 32 + 32$) networks. With hierarchy design we can reduce this number to 34 ($32 + 1 + 1$). Well.... how could this be possible?

Create three areas (area 0, area 1 and area 2) one for each default subnet.

In a hierarchy design we always start from area 0. No matter how many areas you create, you should always start counting from 0. Area 0 has special privilege in OSPF implementation. OSPF treats area 0 as backbone area. Assign area 0 to our first subnet 192.168.0.0/24.

Create area 1 and assign it to subnet 192.168.1.0/24. Finally create area 2 and assign it to subnet 192.168.2.0/24.

So how this implementation reduce network broadcasts?

There are two types of router in an area; ABR and IR.

ABR is a special router which connects two areas. In a proper implementation it should share only summarized route information with remote area.

Through routes summarization on ABR, other areas would see only summarized networks for respective areas instead of full subnets. By doing this we are reducing the amount of information that ABR need to share.

Have a look on following figure. It illustrates our implementation. Area 0 has 32 sub networks (/29) created from one default network 192.168.0.0/24. ABR of area 0 is sharing only one route 192.168.0.0/24 with area 1 and area 2 instead of all internal routes connecting 32 networks.



Hierarchy design limits network instability in a single area. It also reduces routing overhead and speed up the convergence time. If properly implemented with VLSM, it can scale an enterprise size network. Vice versa a little mistake can make it a nightmare.

I have a good news for you, Cisco tests hierarchical design in CCNP. So you can feel relax until you prepare for CCNP exam. Till associate level exam Cisco limits designing part to single area. We will use area 0 to explain the remaining article. Of-course we can use any other valid area number, but it is good practice to take correct learning path from beginning. And we know that in a hierarchy design Area 0 stands on the top.

Okay let's remove other areas from network and make it simple. Now we have only single area to study. Let's explore it step by step. You learned that OSPF is a link state protocol. What does it mean? What is link? And what is state?

Link

Link is an interface running OSPF routing protocol. When we add an interface in OSPF process, it will be considered as a link.

State

State is the information associated with a link (interface). A link (interface) contains several information such as IP address, up/down status, subnet mask, type of interface, type of network , bandwidth and delay. OSPF consider this information as state.

LSA

Link state advertisement (LSA) is data packet. It contains link-state and routing information. OSPF uses it to share and learn network information.

LSDB

Every OSPF router maintains a Link state database (LSDB). LSDB is collection of all LSAs received by a router. Every LSA has a unique sequence number. OSPF stores LSA in LADB with this sequence number.

Upon initialization or due to any change in network information, an OSPF speaking router generates a LSA. This LSA includes the collection of all link-states or link state updates. All routers exchange LSA by flooding. Each router that receives a LSA will store a copy of it in its LSDB then propagate the LSA to other routers.

For example figure display a basic flooding process where R1 is generating LSA and flooding it to the other routers of network.

 The image cannot be displayed. Your computer may not have enough memory to open the image, or the image may have been corrupted. Restart your computer, and then open the file again. If the red x still appears, you may have to delete the image and then insert it again.

R2 and R5 are the first clients who receive this LSA. They will update their LSDB and then forward it to R3 and R6 respectively. R3 and R6 will update their database with this LSA and then forward it to R4. From here only one router either R3 or R6 will be able to forward this LSA to R4. Why does this happen?

Because flooding process has a mechanism to prevent the loops. Before sending a LSA to neighbors, it asks them "Do you have this LSA?" If neighbor reply with yes, it will avoid flooding that LSA to this neighbor. If neighbor reply with no, it will flood that LSA to this neighbor. Thus R4 will only receive this LSA only from one neighbor; either R3 or R6.

OSPF routers share LSA only with neighbors. To become an OSPF neighbor, certain conditions need to be matched.

NAT – PAT

Basic Concepts of NAT

This assignment explains basic concepts of static NAT, dynamic NAT, PAT, inside local, outside local, inside global and outside global in detail with examples.

Basic overview of NAT

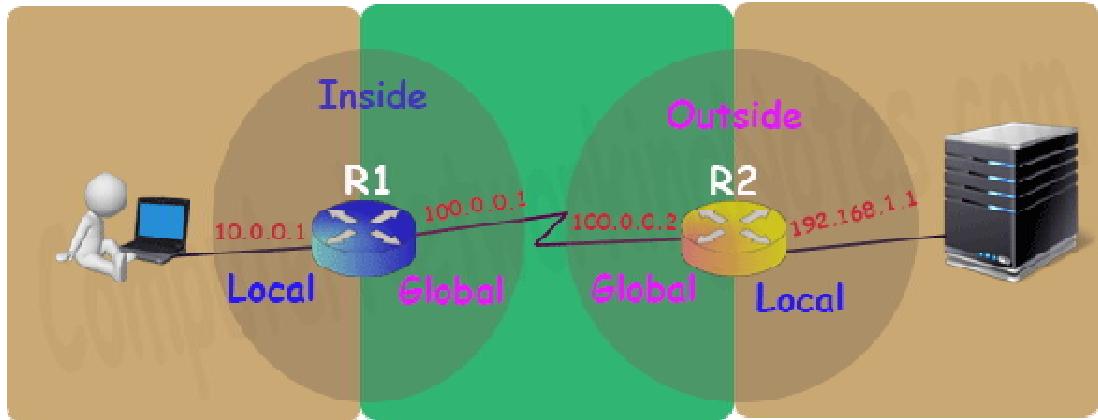
There are several situations where we need address translation such as, a network which do not have sufficient public IP addresses want to connect with the Internet, two networks which have same IP addresses want to merge or due to security reason a network want to hide its internal IP structure from the external world. NAT (Network Address Translation) is the process which translates IP address. NAT can be performed at firewall, server and router. In this assignment we will understand how it is performed at Cisco router.

NAT Terminology

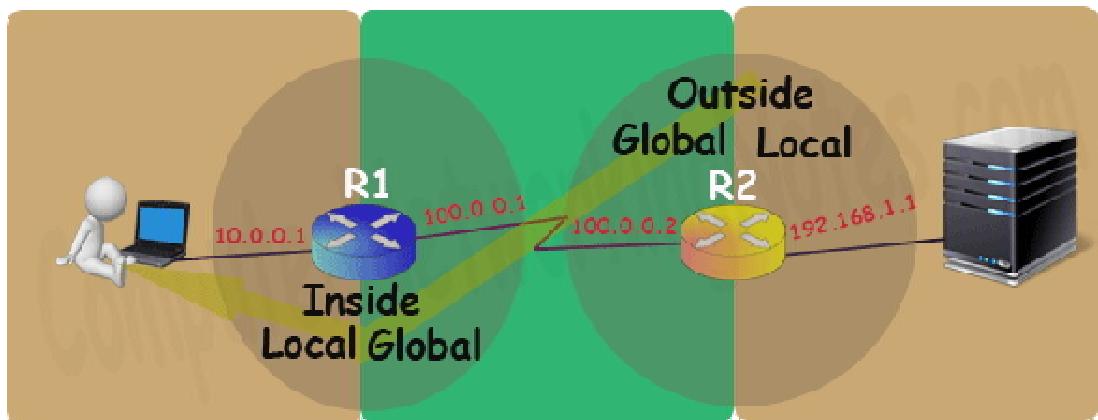
Before we understand NAT in details let's get familiar with four basic terms used in NAT.

Term	Description
Inside Local IP Address	Before translation source IP address located inside the local network.
Inside Global IP Address	After translation source IP address located outside the local network.
Outside Global IP Address	Before translation destination IP address located outside the remote network.
Outside Local IP Address	After translation destination IP address located inside the remote network.

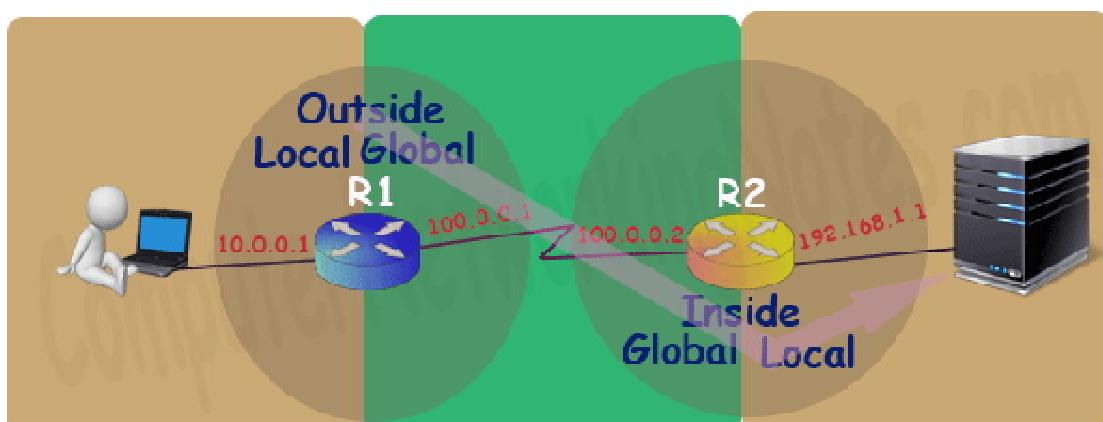
Let's understand these terms with an example. Suppose a user is browsing a website from his home computer. The network which connects his computer with internet is considered as a local network for him. Same as the network which connects the webserver where the website is located with internet is considered as a local network for webserver. The network which connects both networks on internet is considered as a global network.



On router the interface which is connected with local network will be configured with inside local IP address and the interface which is connected with global network will be configured with inside global IP address. Inside and outside depend on where we are standing right now. For example in above network for user router R1 is inside and router R2 is outside.



While for webserver router R2 is inside and router R1 is outside.



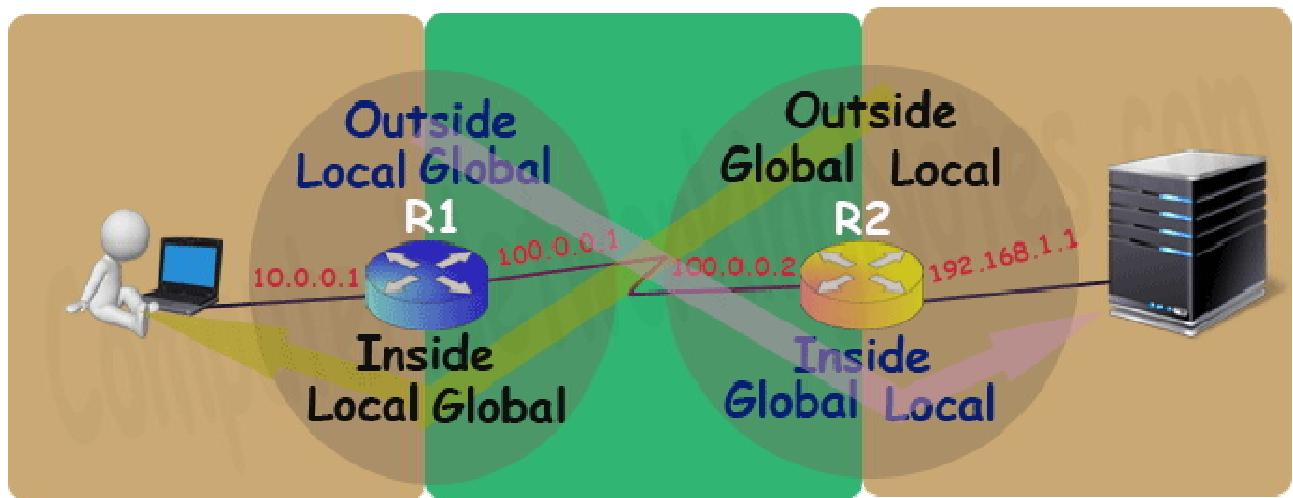
Basically on a NAT enabled router there are two types of interface inside local and inside global.

So, what about outside global and outside local? Well... these terms are used to explain the NAT process theoretically. Practically we never need to configure the outside local and outside global as they sound. For example let's discuss above example once again.

On R1 we will configure inside local address (10.0.0.1) and inside global address (100.0.0.1) which will become outside local address (10.0.0.1) and outside global address (100.0.0.1) for R2 respectively.

Same way on R2 we will configure inside local address (192.168.1.1) and inside global address (100.0.0.2) which will become outside local address (192.168.1.1) and outside global address (100.0.0.2) for R1 respectively.

So practically we only configure inside local and inside global. What is inside for one side is the outside for other side.



Types of NAT

There are three types of NAT; Static NAT, Dynamic NAT and PAT. These types define how inside local IP address will be mapped with inside global IP address.

Static NAT

In this type we manually map each inside local IP address with inside global IP address. Since this type uses one to one mapping we need exactly same number of IP address on both sides.

Dynamic NAT

In this type we create a pool of inside global IP addresses and let the NAT device to map inside local IP address with the available outside global IP address from the pool automatically.

PAT

In this type a single inside global IP address is mapped with multiple inside local IP addresses using the source port address. This is also known as PAT (Port Address Translation) or NAT over load.

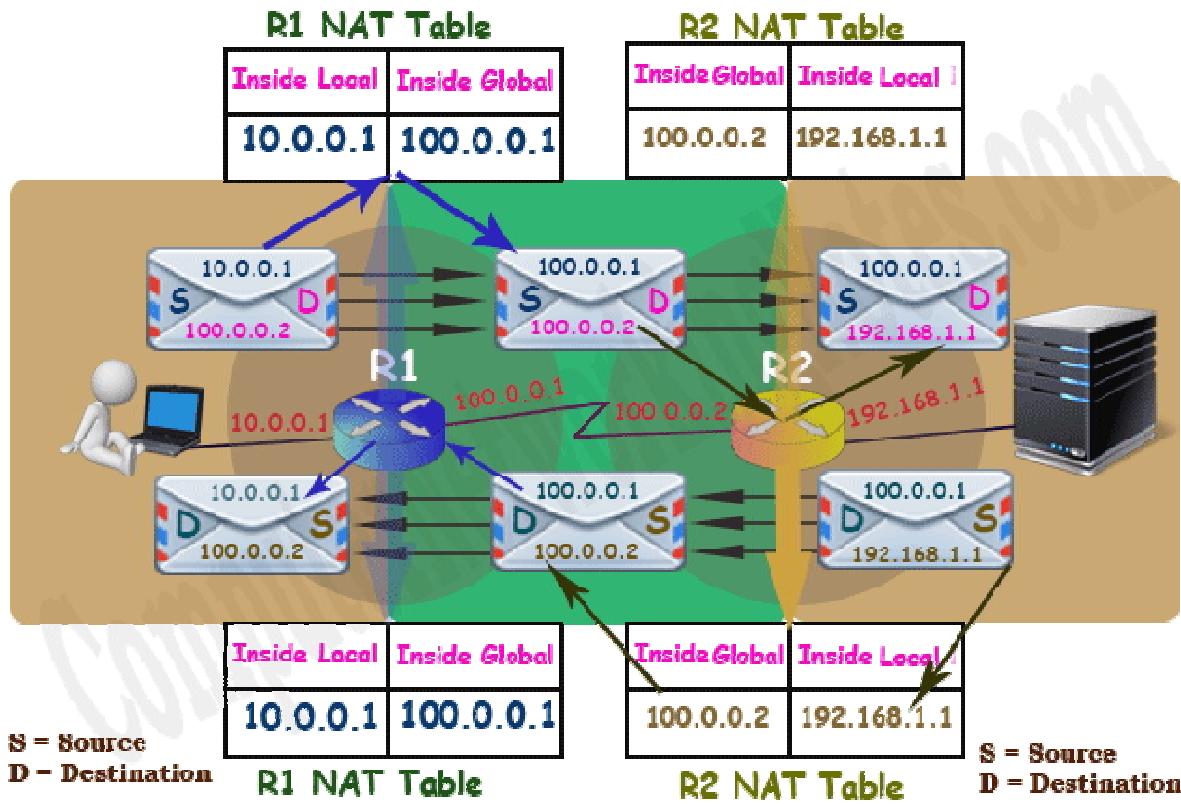
Situations where NAT is used

There are no hard and fast rules about where we should use NAT or where we should not use the NAT. Whether we should use the NAT or not is purely depends on network requirement for example NAT is the best solution in following situations: -

- Our network is built with private IP addresses and we want to connect it with internet. As we know to connect with internet we require public IP address. In this situation we can use NAT device which will map private IP address with public IP address.
- Two networks which are using same IP address scheme want to merge. In this situation NAT device is used to avoid IP overlapping issue.
- We want to connect multiple computers with internet through the single public IP address. In this situation NAT is used to map the multiple IP addresses with single IP address through the port number.

How NAT Works

To understand how NAT works, let's take one more example. In this example a user is accessing a web server. User and Webserver both are connected through the NAT devices. Both user and webserver are using private IP addresses which are not routable on the internet. Now let's understand how NAT makes this communication possible.



User generates a data packet for web server. This packet has source address 10.0.0.1 and destination address 100.0.0.2.

Here source address is the correct address but why the packet has destination address 100.0.0.2 instead of actual destination address 192.168.1.1?

When a system needs to connect with the website, it uses DNS server to resolve the IP address of the website. DNS server advertises the global IP address of the website. Outsider can connect with the website through the advertised IP address only. In our example the global IP address of web server is 100.0.0.2. For this reason the packet has the destination address 100.0.0.2 instead of 192.168.1.1.

This packet reaches at R1. Since this packet contains private IP address in source field which is not routable on internet, R1 has to update the private IP address with a routable public IP address before forwarding this packet.

R1 checks NAT table for available public IP addresses. Depending on what type of NAT (Static, Dynamic or PAT) is configured one routable public IP will be picked from NAT table for this packet.

In our example 100.0.0.1 is picked for this packet. Now R1 will replace 10.0.0.1 with 100.0.0.1 in the source field of the packet and forward it to the R2.

R2 receives this packet and reads the destination IP address. R2 looks in NAT table to find out the actual IP address of the destination. Since the NAT table of R2 has an entry for the address 100.0.0.2 which maps it with the address 192.168.1.1, R2 will replace the destination address 100.0.0.2 with the address 192.168.1.1 and forward it to the web server.

Webserver will process this packet and reply with its own packet. This packet has source address 192.168.1.1 and destination address 100.0.0.1.

Since webserver received this packet from 100.0.0.1 so it will reply to it instead of 10.0.0.1.

R2 receives this packet. Before forwarding this packet R2 will replace the source IP address with the mapped IP address in NAT table. In this example 192.168.1.1 will be replaced with 100.0.0.2.

R1 receives this packet and checks its destination address. R1 will perform a query in NAT table to figure out the IP address which is associated with this destination IP address. Since this destination IP address 100.0.0.1 is mapped with 10.0.0.1, R1 will replace this destination IP address 100.0.0.1 with 10.0.0.1 and forward it to the PC.

From user's point of view the IP address of the webserver is 100.0.0.2. While from web server's point of view the IP address of the user is 100.0.0.1. This way both user and webserver will never know to whom they are communicating actually.

Advantages and disadvantages of NAT

Nat provides following advantages: -

- NAT solves IP overlapping issue.
- NAT hides internal IP structure from external world.
- NAT allows us to connect with any network without changing IP address.
- NAT allows us to connect multiple computers with internet through the single the public IP address.

NAT has following disadvantages: -

- NAT adds additional delay in network.
- Several applications are not compatible with NAT.
- End to end IP traceability will not work with NAT.
- NAT hides actual end device.

That's all for this article. In next part of this assignment we will learn how to configure static NAT and dynamic NAT in Cisco router.

How to Configure Static NAT in Cisco Router

This assignment explains how to configure static NAT (Network Address Translation) in Cisco Router step by step with Packet Tracer examples.

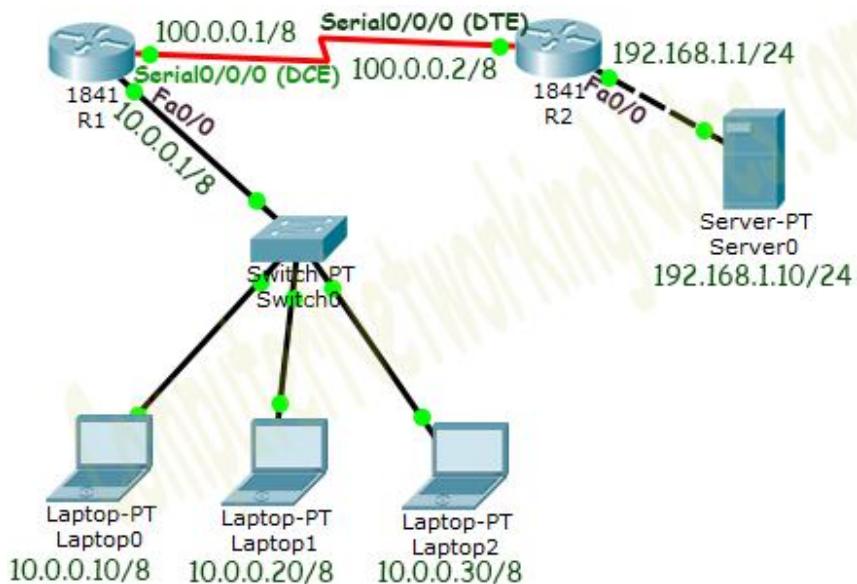
In order to configure NAT we have to understand four basic terms; inside local, inside global, outside local and outside global. These terms define which address will be mapped with which address.

Term	Description
Inside Local IP Address	Before translation source IP address located inside the local network.
Inside Global IP Address	After translation source IP address located outside the local network.
Outside Global IP Address	Before translation destination IP address located outside the remote network.
Outside Local IP Address	After translation destination IP address located inside the remote network.

Static NAT Practice LAB Setup

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

Create a lab as illustrates in following figure.



Initial IP Configuration

Device / Interface	IP Address	Connected With
Laptop0	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 of R2
Serial 0/0/0 of R2	100.0.0.2/8	Serial 0/0/0 of R2

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: -

- Define IP address mapping
- Define inside local interface
- 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.

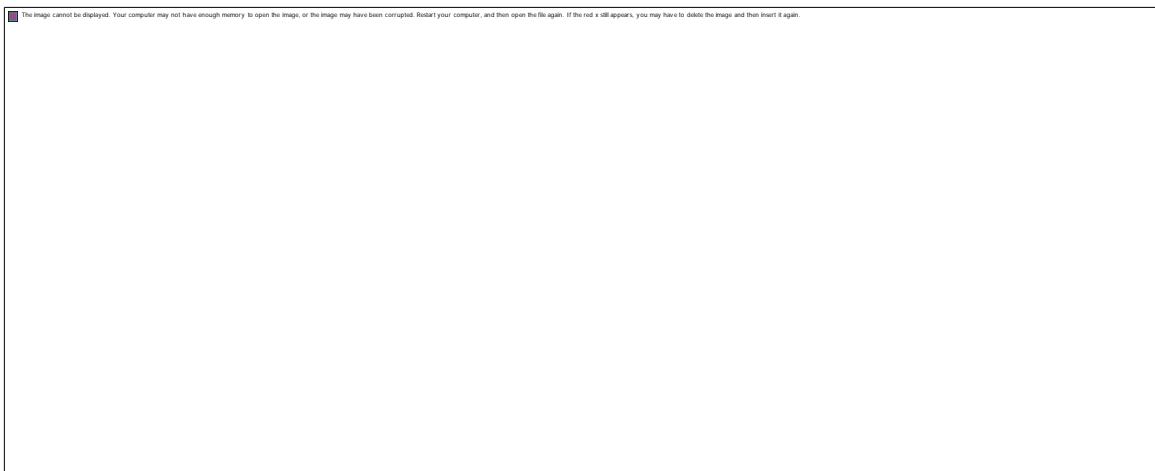
Following command will define interface Fa0/0 as inside local.

```
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 assignment 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 Address
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 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.

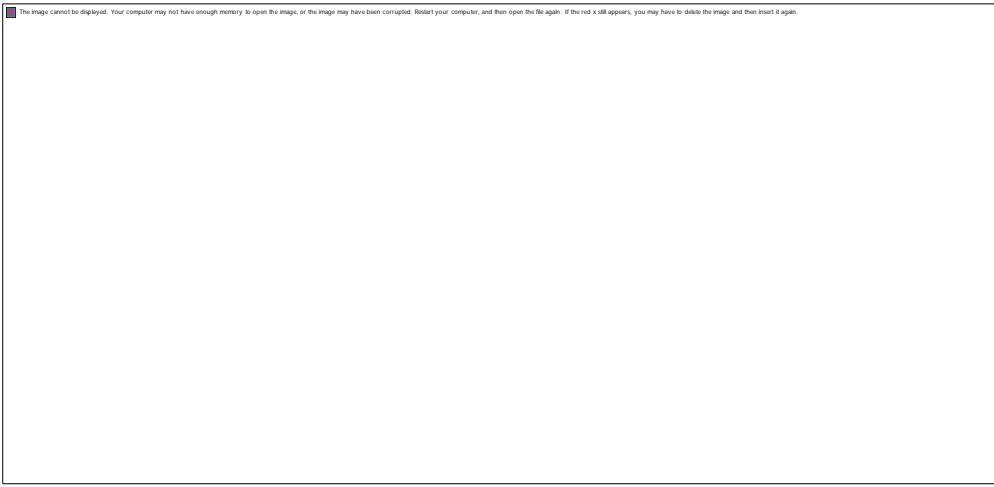
Now run **ping 200.0.0.10** command from Laptop1.



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 assignment 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

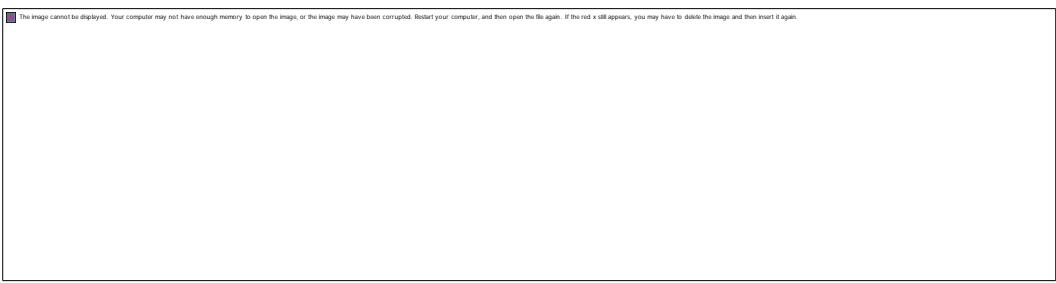
Download NAT Practice LAB with Static NAT configuration

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

Following figure illustrate this translation on router R1.



Following figure illustrate this translation on router R2



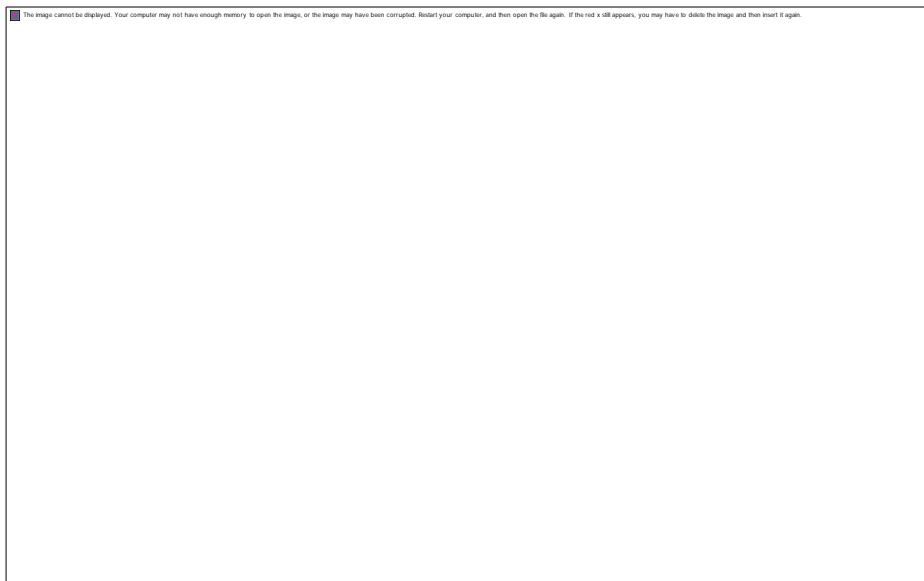
Pay a little bit extra attention on outside local address filed. Have you noticed one interesting feature of NAT in above output? Why actual outside local IP address is not listed in this filed?

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 assignment. In next part we will learn dynamic NAT configuration step by step with examples.

How to Configure Dynamic NAT in Cisco Router



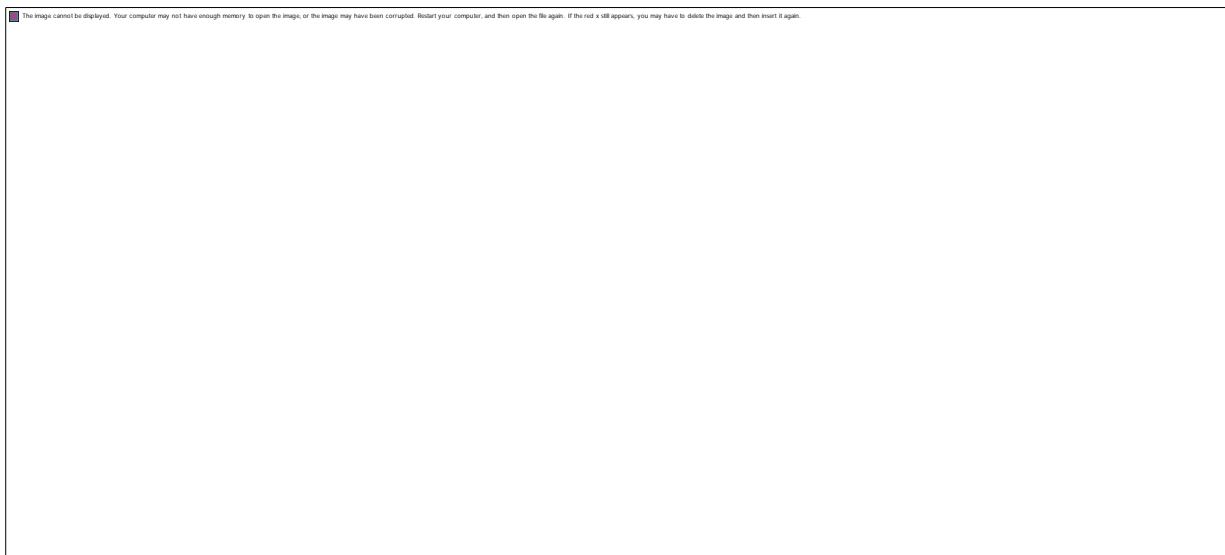
Initial IP Configuration

Device / Interface	IP Address	Connected With
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 of R2
Serial 0/0/0 of R2	100.0.0.2/8	Serial 0/0/0 of R2

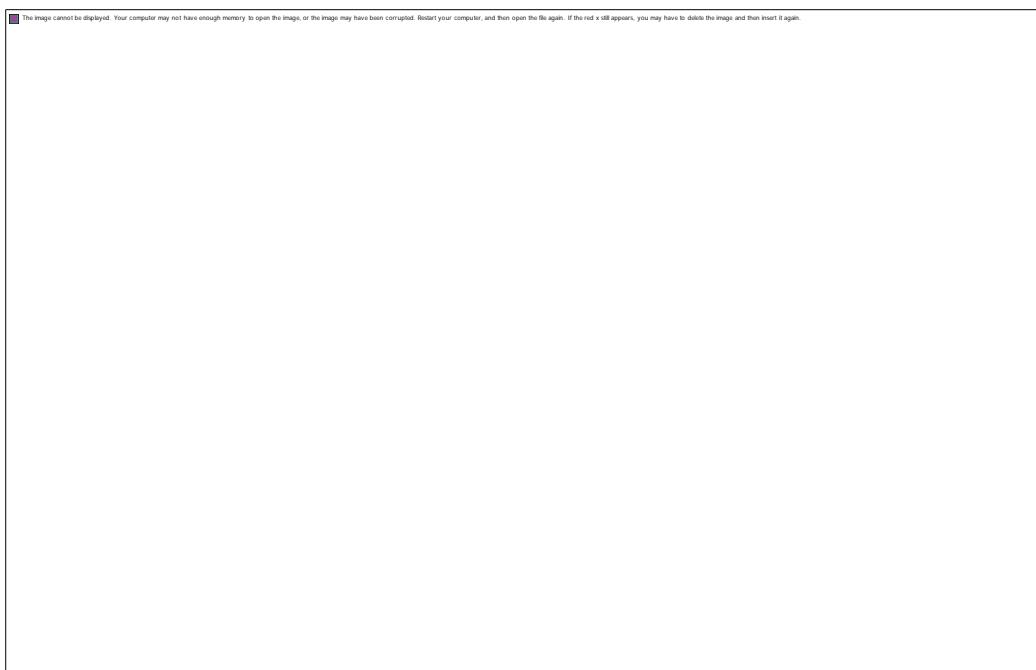
Alternatively you can download my practice topology which is configured with this initial IP configuration.

Download NAT Practice LAB with initial IP configuration

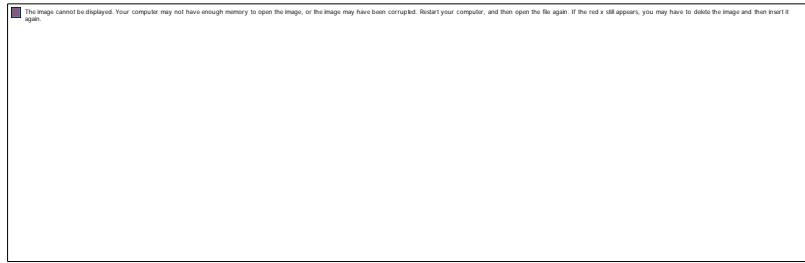
To assign IP address in Laptop click **Laptop** and click **Desktop** and click **IP configuration** and Select **Static** and set **IP address** as given in above table.



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: -

- Create an access list of IP addresses which need translation
- Create a pool of all IP address which are available for translation
- Map access list with pool
- 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 grouping 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
```

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
```

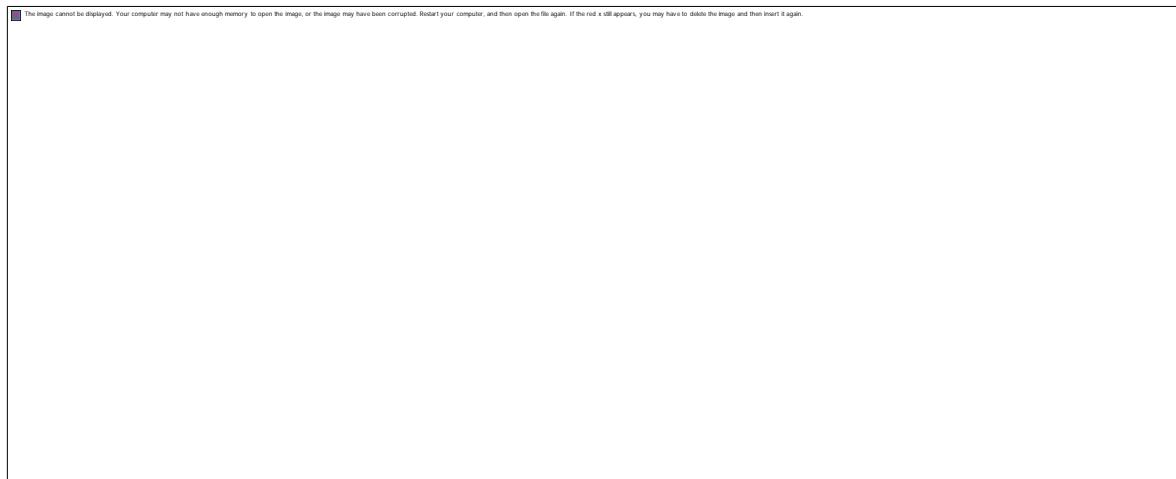
Finally we have to define which interface is connected with local network and which interface is connected with global network.

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 previous 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 assignment.

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 assignment 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
```

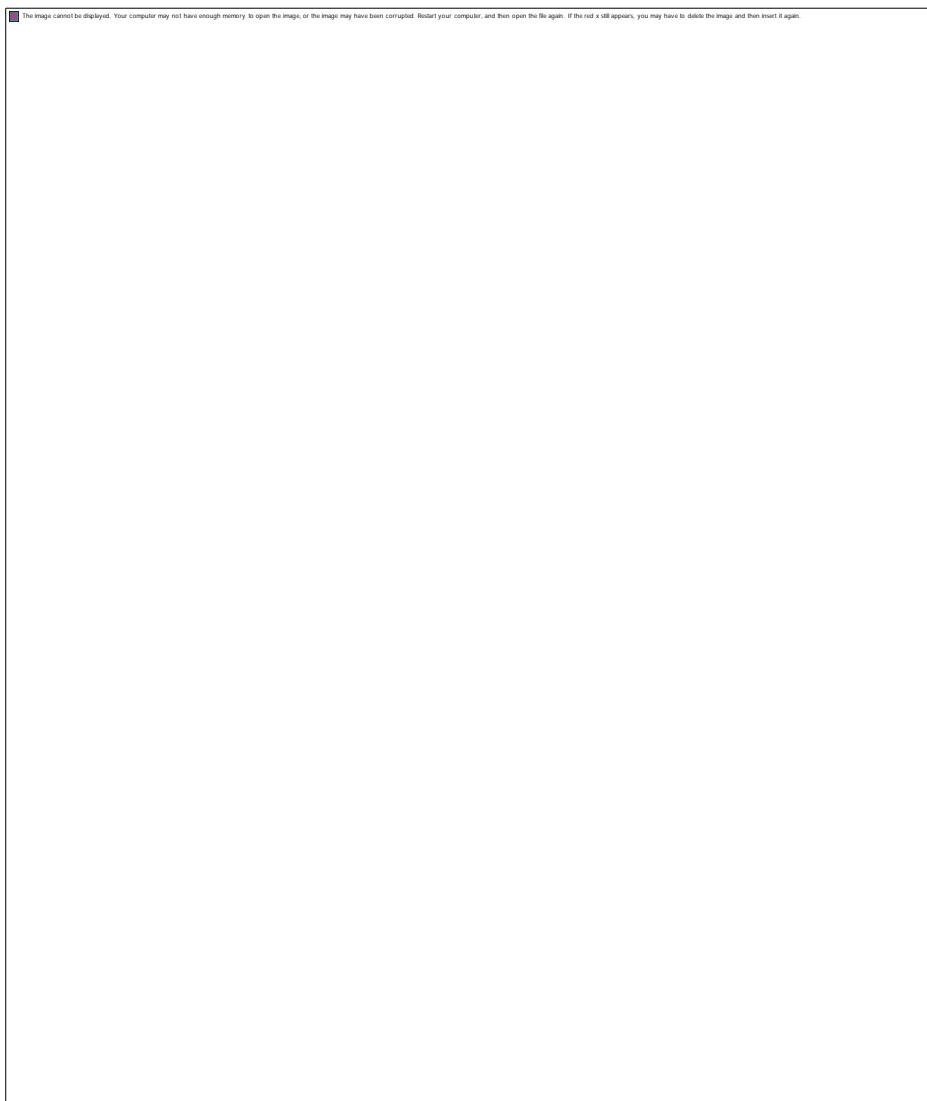
Testing Dynamic NAT Configuration

In this lab we configured dynamic NAT on R1 for 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.

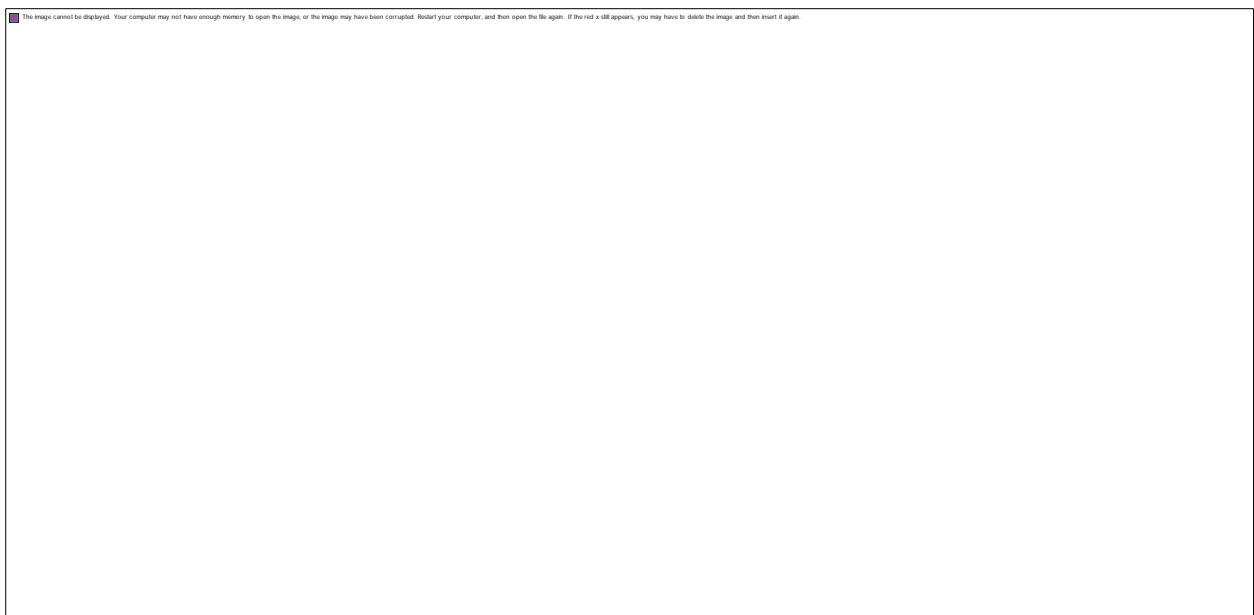


First command verifies that we are testing from correct NAT device.

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.

 The image cannot be displayed. Your computer may not have enough memory to open the image, or the image may have been corrupted. Restart your computer, and then open the file again. If the red x still appears, you may have to delete the image and then insert it again.

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

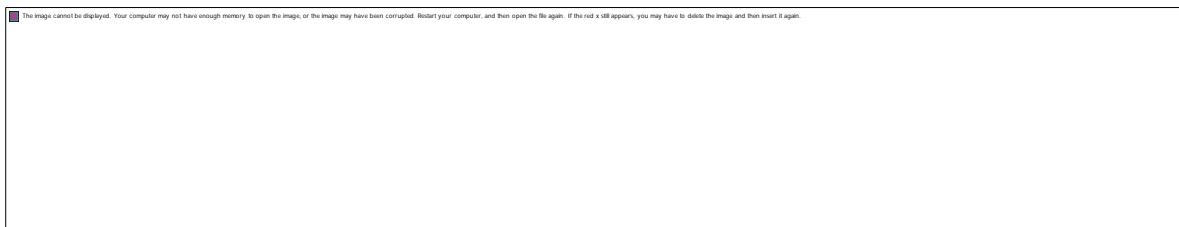
 The image cannot be displayed. Your computer may not have enough memory to open the image, or the image may have been corrupted. Restart your computer, and then open the file again. If the red x still appears, you may have to delete the image and then insert it again.

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.

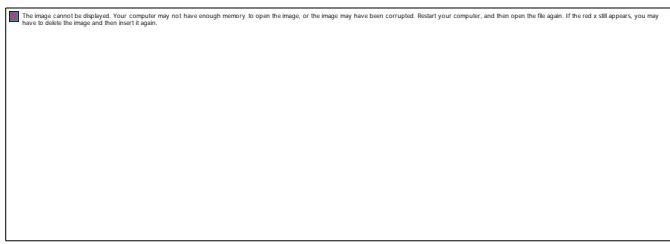
We can also verify this translation on router with *show ip nat translation* command.

Following figure illustrates this translation on router 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
```



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

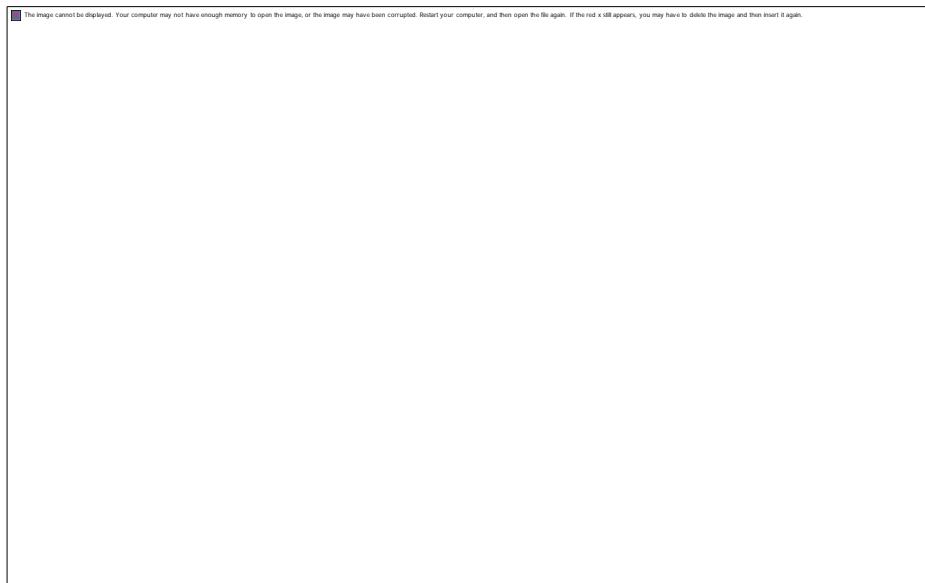
Following figure illustrate NAT translation on router R2



That's all for this assignment. In next part we will learn NAT overload (PAT) configuration step by step with examples.

Configure PAT in Cisco Router

This assignment explains how to configure PAT (Port Address Translation) also known NAT Overload in Cisco Router.



Initial IP Configuration

Device / Interface	IP Address	Connected With
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 of R2
Serial 0/0/0 of R2	100.0.0.2/8	Serial 0/0/0 of R2

Alternatively you can download my practice topology which is 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.

 The image cannot be displayed. Your computer may not have enough memory to open the image, or the image may have been corrupted. Restart your computer, and then open the file again. If the red x still appears, you may have to delete the image and then insert it again.

Following same way configure IP address in Server.

 The image cannot be displayed. Your computer may not have enough memory to open the image, or the image may have been corrupted. Restart your computer, and then open the file again. If the red x still appears, you may have to delete the image and then insert it again.

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

 The image cannot be displayed. Your computer may not have enough memory to open the image, or the image may have been corrupted. Restart your computer, and then open the file again. If the red x still appears, you may have to delete the image and then insert it again.

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: -

- Create an access list of IP addresses which need translation
- Create a pool of all IP address which are available for translation
- Map access list with pool
- 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 grouping 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 assignment.

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
```

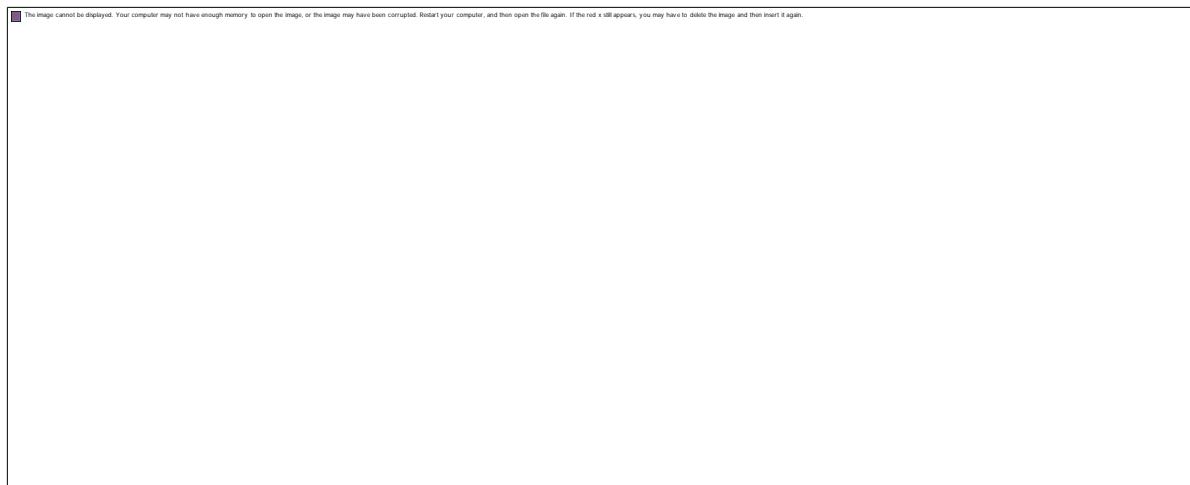
Finally we have to define which interface is connected with local network and which interface is connected with global network.

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 previous 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 assignment.

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 assignment 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
```

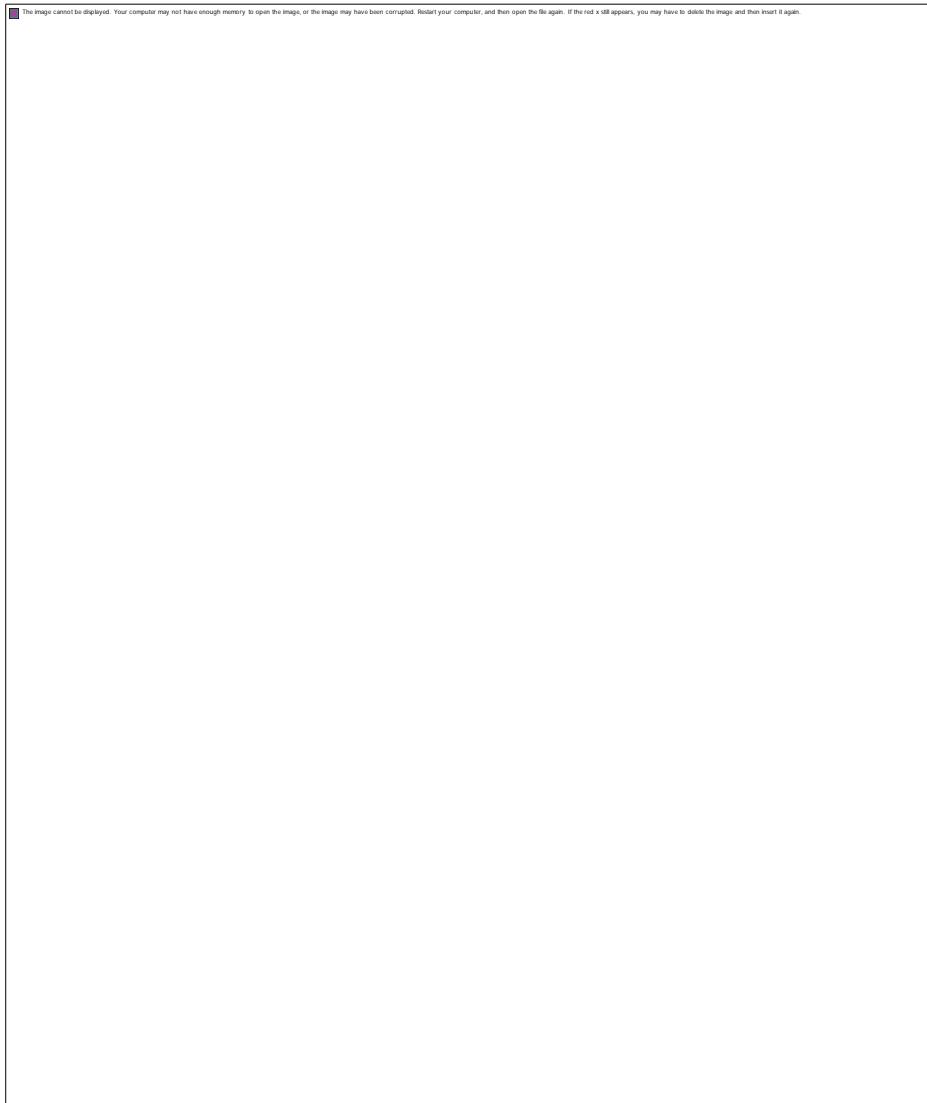
Testing PAT Configuration

In this lab we configured PAT on R1 for 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.

 The image cannot be displayed. Your computer may not have enough memory to open the image, or the image may have been corrupted. Restart your computer, and then open the file again. If the red x still appears, you may have to delete the image and then insert it again.

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

 The image cannot be displayed. Your computer may not have enough memory to open the image, or the image may have been corrupted. Restart your computer, and then open the file again. If the red x still appears, you may have to delete the image and then insert it again.

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 assignment 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.



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



In above output the Outside global field also confirms that all packets are coming from single IP address.

Assignment No: 06

Title: UNIX Sockets

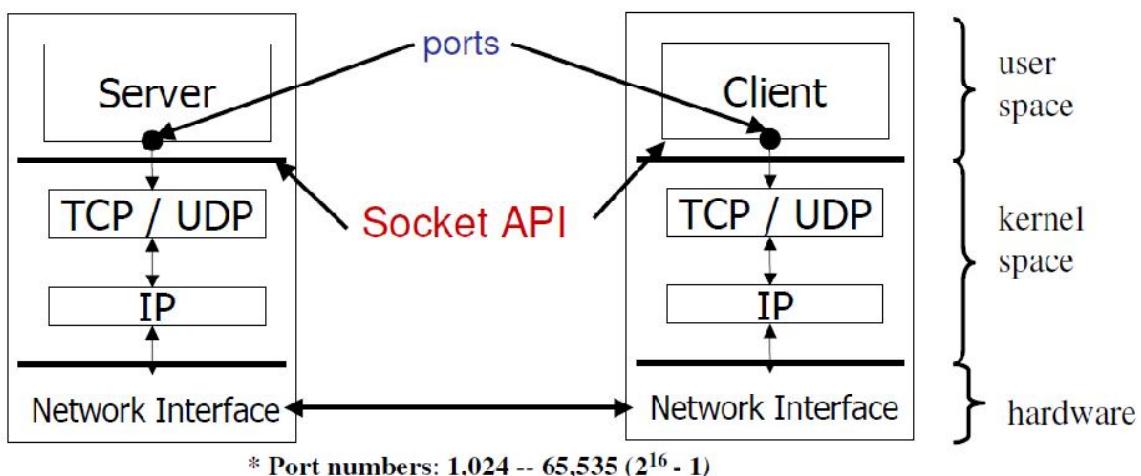
Problem Statement:.

UNIX Sockets: WAP program in C/C++ sockets API

- a. TCP sockets
- b. UDP sockets

Theory:

- Sockets are used for interprocess communication.
- Most of the interprocess communication follow a Client-Server model, where client and server are two separate processes in itself.
- Server and Client exchange messages over the network through a common Socket API



Server Examples

- Web server (port 80)
- FTP server (20, 21)
- Telnet server (23)
- Mail server (25)

Client Examples

- Examples of client programs
 - Web browsers, ftp, telnet, ssh

How does a client find the server?

- The IP address in the server socket address identifies the host
- The (well-known) port in the server socket address identifies the service, and thus implicitly identifies the server process that performs that service.

Examples of well known ports

- Port 7: Echo server
- Port 23: Telnet server
- Port 25: Mail server
- Port 80: Web server

What is an API ?

API expands as Application Programming Interface.

A set of routines that an application uses to request and carry out lower-level services performed by a computer's operating system.

What is a socket?

- An interface between application and network which is used for communication between processes
- Once configured the application can
 - pass data to the socket for network transmission
 - receive data from the socket (transmitted through the network by some other host)
- To the kernel, a socket is an endpoint of communication.
- To an application, a socket is a file descriptor that lets the application read/write from/to the network.
- Clients and servers communicate with each by reading from and writing to socket descriptors.
- Remember: All Unix I/O devices, including networks, are modeled as files.

Two essential types of sockets

SOCK_STREAM

- TCP
- connection-oriented
- reliable delivery
- in-order guaranteed
- bidirectional

SOCK_DGRAM

- UDP
- no notion of “connection” – app indicates dest. for each packet
- unreliable delivery
- no order guarantees
- can send or receive

Socket Primitives

Primitive	Meaning
SOCKET	Create a new communication end point
BIND	Attach a local address to a socket
LISTEN	Announce willingness to accept connections; give queue size
ACCEPT	Block the caller until a connection attempt arrives
CONNECT	Actively attempt to establish a connection
SEND	Send some data over the connection
RECEIVE	Receive some data from the connection
CLOSE	Release the connection

socket()

The function *socket()* creates an endpoint for communication and returns a file descriptor for the socket. *socket()* takes three arguments:

- *domain*, which specifies the protocol family of the created socket. For example:
 - *AF_INET* for network protocol IPv4 or
 - *AF_INET6* for IPv6.
 - *AF_UNIX* for local socket (using a file).
- *type*, one of:
 - *SOCK_STREAM* (reliable stream-oriented service or Stream Sockets)
 - *SOCK_DGRAM* (datagram service or Datagram Sockets)
 - *SOCK_SEQPACKET* (reliable sequenced packet service), or
 - *SOCK_RAW* (raw protocols atop the network layer).
- *protocol* specifying the actual transport protocol to use. The most common are *IPPROTO_TCP*, *IPPROTO_SCTP*, *IPPROTO_UDP*, *IPPROTO_DCCP*. These protocols are specified in file *netinet/in.h*. The value *0* may be used to select a default protocol from the selected domain and type.

The function returns -1 if an error occurred. Otherwise, it returns an integer representing the newly assigned descriptor.

Prototype:

- int socket(int domain, int type, int protocol)

bind()

bind() assigns a socket to an address. When a socket is created using *socket()*, it is only given a protocol family, but not assigned an address. This association with an address must be performed with the *bind()* system call before the socket can accept connections to other hosts. *bind()* takes three arguments:

- *sockfd*, a descriptor representing the socket to perform the bind on.
- *my_addr*, a pointer to a *sockaddr* structure representing the address to bind to.
- *addrlen*, a *socklen_t* field specifying the size of the *sockaddr* structure.

Bind() returns 0 on success and -1 if an error occurs.

Prototype:

- int bind(int sockfd, const struct sockaddr *my_addr, socklen_t addrlen);

listen()

After a socket has been associated with an address, *listen()* prepares it for incoming connections. However, this is only necessary for the stream-oriented (connection-oriented) data modes, i.e., for socket types (*SOCK_STREAM*, *SOCK_SEQPACKET*). *listen()* requires two arguments:

- *sockfd*, a valid socket descriptor.
- *backlog*, an integer representing the number of pending connections that can be queued up at any one time. The operating system usually places a cap on this value.

Once a connection is accepted, it is dequeued. On success, 0 is returned. If an error occurs, -1 is returned.

Prototype:

- int listen(int sockfd, int backlog);

accept()

When an application is listening for stream-oriented connections from other hosts, it is notified of such events (cf. *select()* function) and must initialize the connection using the *accept()* function. The *accept()* function creates a new socket for each connection and removes the connection from the listen queue. It takes the following arguments:

- *sockfd*, the descriptor of the listening socket that has the connection queued.
- *cliaddr*, a pointer to a *sockaddr* structure to receive the client's address information.
- *addrlen*, a pointer to a *socklen_t* location that specifies the size of the client address structure passed to *accept()*. When *accept()* returns, this location indicates how many bytes of the structure were actually used.

The `accept()` function returns the new socket descriptor for the accepted connection, or -1 if an error occurs. All further communication with the remote host now occurs via this new socket.

Datagram sockets do not require processing by `accept()` since the receiver may immediately respond to the request using the listening socket.

Prototype:

- `int accept(int sockfd, struct sockaddr *cliaddr, socklen_t *addrlen)`

connect()

The `connect()` system call *connects* a socket, identified by its file descriptor, to a remote host specified by that host's address in the argument list.

Certain types of sockets are *connectionless*, most commonly user datagram protocol sockets. For these sockets, `connect` takes on a special meaning: the default target for sending and receiving data gets set to the given address, allowing the use of functions such as `send()` and `recv()` on connectionless sockets.

`connect()` returns an integer representing the error code: 0 represents success, while -1 represents an error. Historically, in the BSD-derived systems, the state of a socket descriptor is undefined if the call to `connect()` fails (as it is specified in the Single Unix Specification), thus, portable applications should close the socket descriptor immediately and obtain a new descriptor with `socket()`, in the case the call to `connect()` fails.^[3]

Prototype:

- `int connect(int sockfd, const struct sockaddr *serv_addr, socklen_t addrlen)`

gethostbyname() and gethostbyaddr()

The `gethostbyname()` and `gethostbyaddr()` functions are used to resolve host names and addresses in the domain name system or the local host's other resolver mechanisms (e.g., `/etc/hosts` lookup). They return a pointer to an object of type *struct hostent*, which describes an Internet Protocol host. The functions take the following arguments:

- *name* specifies the name of the host. For example: `www.wikipedia.org`
- *addr* specifies a pointer to a *struct in_addr* containing the address of the host.
- *len* specifies the length, in bytes, of *addr*.
- *type* specifies the address family type (e.g., `AF_INET`) of the host address.

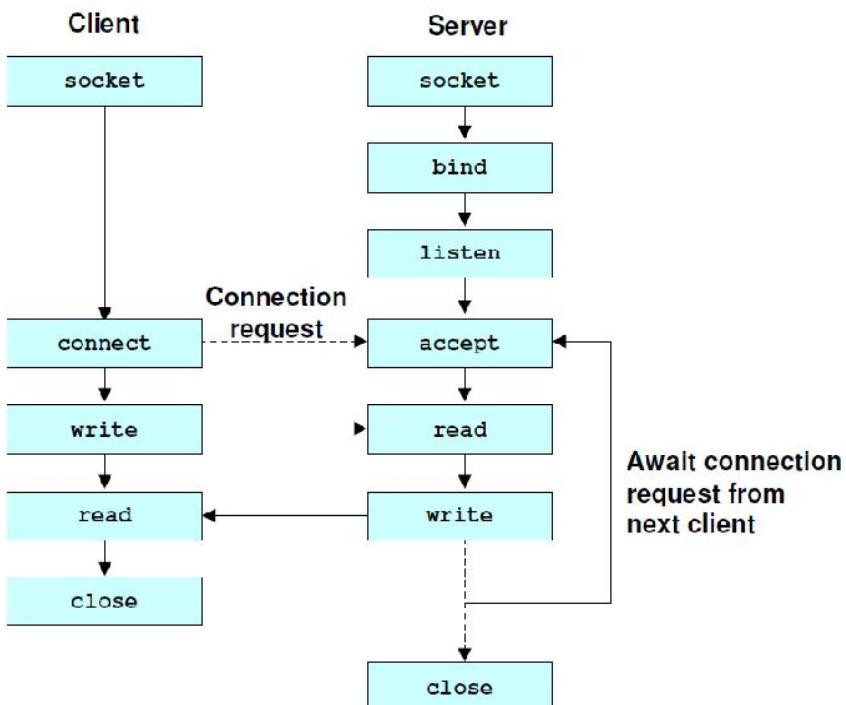
The functions return a NULL pointer in case of error, in which case the external integer *h_errno* may be checked to see whether this is a temporary failure or an invalid or unknown host. Otherwise a valid *struct hostent ** is returned.

These functions are not strictly a component of the BSD socket API, but are often used in conjunction with the API functions. Furthermore, these functions are now considered legacy interfaces for querying the domain name system. New functions that are completely protocol-agnostic (supporting IPv6) have been defined. These new function are *getaddrinfo()* and *getnameinfo()*, and are based on a new *addrinfo* data structure.

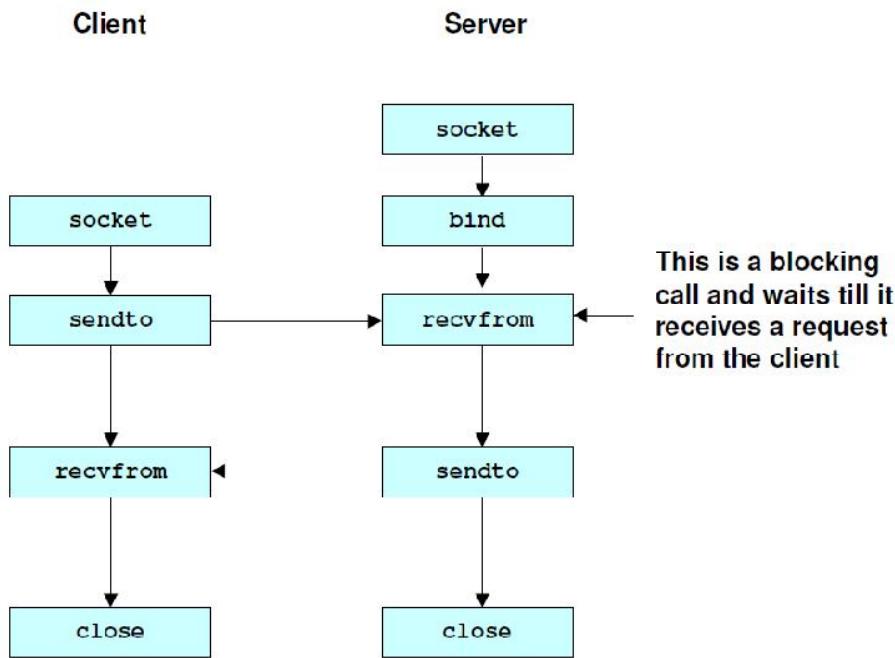
Prototypes:

- *struct hostent *gethostbyname(const char *name)*
- *struct hostent *gethostbyaddr(const void *addr, int len, int type)*

Socket programming with TCP



Socket programming with UDP



Conclusion: TCP & UDP socket programs are studied and executed.

Server Program

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <unistd.h>
#include <errno.h>

#define PORT 5561
#define BUF_SIZE 2000
#define CLADDR_LEN 100

char *itoaa(int val, int base);

int main()
{
    struct sockaddr_in addr, cl_addr;
    int sockfd, len, ret, newsockfd;
    char buffer[BUF_SIZE];
    pid_t childpid;
    char clientAddr[CLADDR_LEN];
    int num, rem, sum;
    char *str;

    sockfd = socket(AF_INET, SOCK_STREAM, 0);

    if (sockfd < 0)
    {
        printf("Error creating socket!\n");
        exit(1);
    }

    printf("Socket created...\n");

    memset(&addr, 0, sizeof(addr));
    addr.sin_family = AF_INET;
    addr.sin_addr.s_addr = INADDR_ANY;
    addr.sin_port = PORT;

    ret = bind(sockfd, (struct sockaddr *) &addr, sizeof(addr));
    if (ret < 0)
    {
        printf("Error binding!\n");
        exit(1);
    }
    else
        printf("Binding done...\n");

    printf("Waiting for a connection...@ port no : %d\n", PORT );
```

```

listen(sockfd, 5);

for (;;) //infinite loop
{
    len = sizeof(struct sockaddr_in);
    newsockfd = accept(sockfd, (struct sockaddr
*)&cl_addr, (socklen_t *)&len);
    if (newsockfd < 0)
    {
        printf("Error accepting connection!\n");
        exit(1);
    }
    else
        printf("Connection accepted from ");

    inet_ntop(AF_INET, &(cl_addr.sin_addr), clientAddr,
CLADDR_LEN);

    printf("Port %d of %s
Client\n", ntohs(cl_addr.sin_port), inet_ntoa(cl_addr.sin_addr));

    if ((childpid = fork()) == 0) //creating a child process
    {
        close(sockfd);
        //stop listening for new connections by the main
process.
        //the child will continue to listen.
        //the main process now handles the connected client.

        for (;;)
        {
            memset(buffer, 0, BUF_SIZE);

            ret = recvfrom(newsockfd, buffer, BUF_SIZE, 0,
(struct sockaddr *) &cl_addr, (socklen_t *)&len);

            if(ret < 0)
            {
                printf("Error receiving data!\n");
                exit(1);
            }
            else
                printf("Received data from Port No %d of
Client %s : %s\n ", ntohs(cl_addr.sin_port),clientAddr, buffer);

            num=atoi(buffer);
            sum=0;
            while(num>0)
            {
                sum = sum + (num % 10);
                num = num / 10;
            }
        }
    }
}

```

```

        strcat(buffer,"          = sum of digits = ");
        str=itoaa(sum,10);
        strcat(buffer,str);

        ret = sendto(newsockfd, buffer, BUF_SIZE, 0,
(struct sockaddr *) &cl_addr, len);

        if (ret < 0)
        {
            printf("Error sending data!\n");
            exit(1);
        }
        else
            printf("\tSent data to %s on Port No %d :
%s\n", clientAddr, ntohs(cl_addr.sin_port), buffer);

        printf("-----\n");
    }
}
close(newsockfd);
}
return(0);
}

char *itoaa(int val, int base)
{
    static char buf[32] = {0};

    int i = 30;

    for( ; val && i ; --i, val /= base)
        buf[i] = "0123456789abcdef"[val % base];

    return &buf[i+1];
}

```

Client Program

```
#include "stdio.h"
#include "stdlib.h"
#include "sys/types.h"
#include "sys/socket.h"
#include "string.h"
#include "netinet/in.h"
#include "netdb.h"

#define PORT 5561
#define BUF_SIZE 2000

int main(int argc, char**argv) {
    struct sockaddr_in addr, cl_addr;
    int sockfd, ret;
    char buffer[BUF_SIZE];
    struct hostent * server;
    char * serverAddr;

    if (argc < 2) {
        printf("usage: client < ip address >\n");
        exit(1);
    }

    serverAddr = argv[1];

    sockfd = socket(AF_INET, SOCK_STREAM, 0);
    if (sockfd < 0) {
        printf("Error creating socket!\n");
        exit(1);
    }
    printf("Socket created...\n");

    memset(&addr, 0, sizeof(addr));
    addr.sin_family = AF_INET;
    addr.sin_addr.s_addr = inet_addr(serverAddr);
    addr.sin_port = PORT;

    ret = connect(sockfd, (struct sockaddr *) &addr, sizeof(addr));
    if (ret < 0) {
        printf("Error connecting to the server! : : %d\n",ret);
        exit(1);
    }
    printf("Connected to the server @ %s\n",serverAddr);

    memset(buffer, 0, BUF_SIZE);
    printf("Enter your message(s): ");

    while (fgets(buffer, BUF_SIZE, stdin) != NULL) {
        ret = sendto(sockfd, buffer, BUF_SIZE, 0, (struct sockaddr *) &addr,
        sizeof(addr));
        if (ret < 0) {
            printf("Error sending data!\n\t-%s", buffer);
```

```
    }

    ret = recvfrom(sockfd, buffer, BUF_SIZE, 0, NULL, NULL);
    if (ret < 0) {
        printf("Error receiving data!\n");
    } else {
        printf("Received: ");
        fputs(buffer, stdout);
        printf("\n");
    }
}

return 0;
}
```

A. Telnet

Telnet is a network protocol used on the Internet or local area networks to provide a bidirectional interactive text-oriented communication facility using a virtual terminal connection. User data is interspersed in-band with Telnet control information in an 8-bit byte oriented data connection over the Transmission Control Protocol (TCP). The telnet commands allow you to communicate with a remote computer that is using the Telnet protocol. You can run telnet without parameters in order to enter the telnet context, indicated by the Telnet prompt (telnet>). From the Telnet prompt, use the following commands to manage a computer running Telnet Client.

B. FTP Server

Introduction

The File Transfer Protocol (FTP) is used as one of the most common means of copying files between servers over the Internet. Most web based download sites use the built in FTP capabilities of web browsers and therefore most server oriented operating systems usually include an FTP server application as part of the software suite. Linux is no exception.

FTP Overview

FTP relies on a pair of TCP ports to get the job done. It operates in two connection channels as

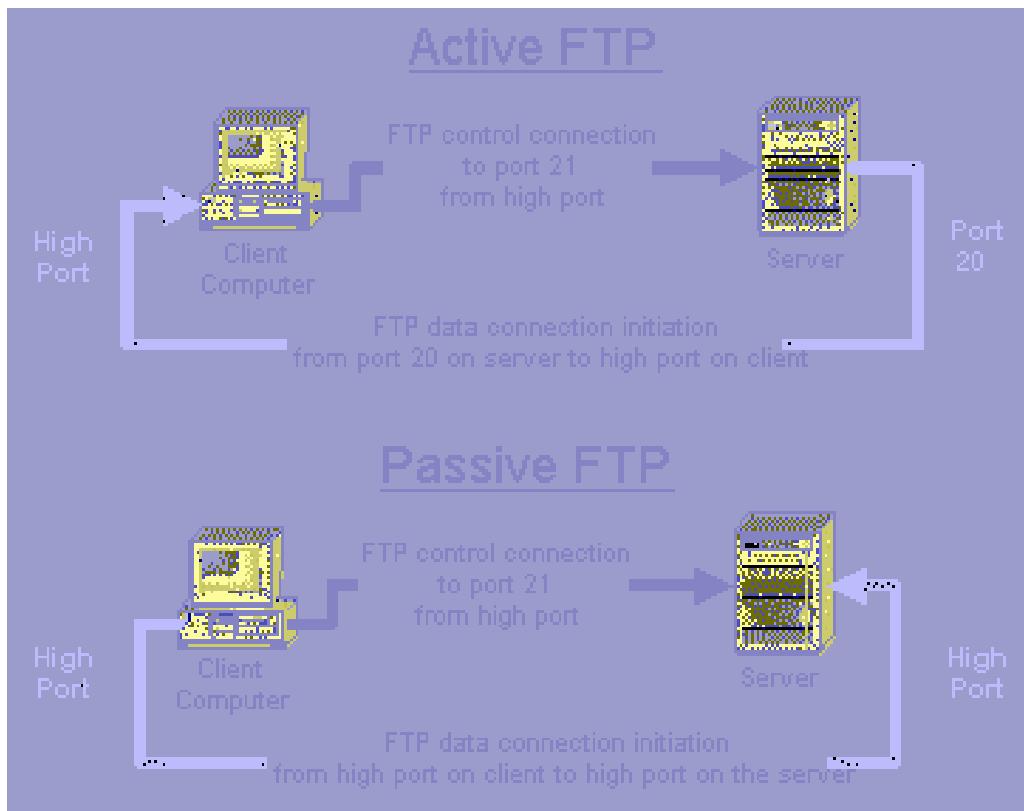
FTP Control Channel, TCP Port 21: All commands you send and the ftp server's responses to those commands will go over the control connection, but any data sent back (such as "ls" directory lists or actual file data in either direction) will go over the data connection.

FTP Data Channel, TCP Port 20: This port is used for all subsequent data transfers between the client and server.

In addition to these channels, there are several varieties of FTP.

Types of FTP

From a networking perspective, the two main types of FTP are active and passive. In active FTP, the FTP server initiates a data transfer connection back to the client. For passive FTP, the connection is initiated from the FTP client. These are illustrated in Figure



From a user management perspective there are also two types of FTP: regular FTP in which files are transferred using the username and password of a regular user FTP server, and anonymous FTP in which general access is provided to the FTP server using a well known universal login method.

Take a closer look at each type.

Active FTP

The sequence of events for active FTP is:

1. Your client connects to the FTP server by establishing an FTP control connection to port 21 of the server. Your commands such as 'ls' and 'get' are sent over this connection.
2. Whenever the client requests data over the control connection, the server initiates data transfer connections back to the client. The source port of these data transfer connections is always port 20 on the server, and the destination port is a high port (greater than 1024) on the client.
3. Thus the ls listing that you asked for comes back over the port 20 to high port connection, not the port 21 control connection.

FTP active mode therefore transfers data in a counter intuitive way to the TCP standard, as it selects port 20 as its source port (not a random high port that's greater than 1024) and connects back to the client on a random high port that has been pre-negotiated on the port 21 control connection.

Active FTP may fail in cases where the client is protected from the Internet via many to one NAT (masquerading). This is because the firewall will not know which of the many servers behind it should receive the return connection.

Passive FTP

Passive FTP works differently:

1. Your client connects to the FTP server by establishing an FTP control connection to port 21 of the server. Your commands such as ls and get are sent over that connection.
2. Whenever the client requests data over the control connection, the client initiates the data transfer connections to the server. The source port of these data transfer connections is always a high port on the client with a destination port of a high port on the server.

Passive FTP should be viewed as the server never making an active attempt to connect to the client for FTP data transfers. Because client always initiates the required connections, passive FTP works better for clients protected by a firewall.

As Windows defaults to active FTP, and Linux defaults to passive, you'll probably have to accommodate both forms when deciding upon a security policy for your FTP server.

Regular FTP

By default, the VSFTPD package allows regular Linux users to copy files to and from their home directories with an FTP client using their Linux usernames and passwords as their login credentials.

VSFTPD also has the option of allowing this type of access to only a group of Linux users, enabling you to restrict the addition of new files to your system to authorized personnel.

The disadvantage of regular FTP is that it isn't suitable for general download distribution of software as everyone either has to get a unique Linux user account or has to use a shared username and password. Anonymous FTP allows you to avoid this difficulty.

Anonymous FTP

Anonymous FTP is the choice of Web sites that need to exchange files with numerous unknown remote users. Common uses include downloading software updates and MP3s and uploading diagnostic information for a technical support engineers' attention. Unlike regular FTP where you login with a preconfigured Linux username and password, anonymous FTP requires only a username of anonymous and your email address for the password. Once logged in to a VSFTPD server, you automatically have access to only the default anonymous FTP directory (/var/ftp in the case of VSFTPD) and all its subdirectories.

C. DHCP Server

The **Dynamic Host Configuration Protocol (DHCP)** is a standardized networking protocol used on Internet Protocol (IP) networks for dynamically distributing network configuration parameters, such as IP addresses for interfaces and services. 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.

Depending on implementation, the DHCP server may have three methods of allocating IP-addresses:

- ***Dynamic allocation:*** A network administrator reserves a range of IP addresses for DHCP, and each client computer on the LAN is configured to request an IP address from the DHCP server during network initialization. The request-and-grant process uses a lease concept with a controllable time period, allowing the DHCP server to reclaim (and then reallocate) IP addresses that are not renewed.
- ***Automatic allocation:*** The DHCP server permanently assigns an IP address to a requesting client from the range defined by the administrator. This is like dynamic allocation, but the DHCP server keeps a table of past IP address assignments, so that it can preferentially assign to a client the same IP address that the client previously had.
- ***Static allocation:*** The DHCP server allocates an IP address based on a preconfigured mapping to each client's MAC address. This feature is variously called *static DHCP assignment* by DD-WRT, *fixed-address* by the *dhcpd* documentation, *address reservation* by Netgear, *DHCP reservation* or *static DHCP* by Cisco and Linksys, and *IP address reservation* or *MAC/IP address binding* by various other router manufacturers.

DHCP is used for Internet Protocol version 4 (IPv4), as well as IPv6. While both versions serve the same purpose, the details of the protocol for IPv4 and IPv6 are sufficiently different that they may be considered separate protocols. For IPv6 operation, devices may alternatively use stateless address autoconfiguration. IPv4 hosts may also use link-local addressing to achieve operation restricted to the local network link.

Configuration of Telnet Service in Linux / Ubuntu

1. Install telnet use this command in terminal(Appllications/Accessories/Terminal):

```
sudo apt-get install xinetd telnetd
```

2. Edit */etc/inetd.conf* using your favourite file editor with root permission, add this line:

```
telnet stream tcp nowait telnetd /usr/sbin/tcpd /usr/sbin/in.telnetd
```

3. Edit */etc/xinetd.conf*, make its content look like following:

```
# Simple configuration file for xinetd
#
# Some defaults, and include /etc/xinetd.d/
defaults
{
    # Please note that you need a log_type line to be able to use log_on_success
    # and log_on_failure. The default is the following :
    # log_type = SYSLOG daemon info
    instances = 60
    log_type = SYSLOG authpriv
    log_on_success = HOST PID
    log_on_failure = HOST
    cps = 25 30
}
```

4. You can change telnet port number by edit */etc/services* with this line:

```
telnet    23/tcp
```

5. If you're not satisfied with default configuration. Edit *etc/xinetd.d/telnet*, add following:

```
# default: on
# description: The telnet server serves telnet sessions; it uses
# unencrypted username/password pairs for authentication.
service telnet
{
    disable = no
    flags = REUSE
    socket_type = stream
```

```
wait = no  
user = root  
server = /usr/sbin/in.telnetd  
log_on_failure += USERID  
}
```

add these lines as you like:

```
only_from = 192.168.120.0/24 #Only users in 192.168.120.0 can access to  
only_from = .bob.com #allow access from bob.com  
no_access = 192.168.120.{101,105} #not allow access from the two IP.  
access_times = 8:00-9:00 20:00-21:00 #allow access in the two times  
.....
```

6. Use this command to start telnet server:

```
sudo /etc/init.d/xinetd restart
```

Configuration of FTP Server in Linux using VSFTPD

INSTALLING VSFTPD

While there are a variety of FTP server tools available for Linux, one of the most popular and mature options is vsftpd.

Begin by SSHing into your server as root and use the apt-get command to install *vsftpd*:

```
$ apt-get update
$ apt-get install vsftpd
Reading package lists... Done
Building dependency tree
Reading state information... Done
[...]
The following NEW packages will be installed:
vsftpd
0 upgraded, 1 newly installed, 0 to remove and 18 not upgraded.
Need to get 111 kB of archives.
After this operation, 361 kB of additional disk space will be used.
Get:1 http://mirrors.digitalocean.com/ubuntu/ trusty-updates/main vsftpd amd64 3.0.2-1ubuntu2.14.04.1 [111 kB]
Fetched 111 kB in 0s (231 kB/s)
Preconfiguring packages ...
Selecting previously unselected package vsftpd.
(Reading database ... 175600 files and directories currently installed.)
Preparing to unpack .../vsftpd_3.0.2-1ubuntu2.14.04.1_amd64.deb ...
Unpacking vsftpd (3.0.2-1ubuntu2.14.04.1) ...
Processing triggers for man-db (2.6.7.1-1) ...
Processing triggers for ureadahead (0.100.0-16) ...
Setting up vsftpd (3.0.2-1ubuntu2.14.04.1) ...
vsftpd start/running, process 18690
Processing triggers for ureadahead (0.100.0-16) ...
```

CONFIGURATION

The next step is to change any configuration settings for vsftpd. Open the /etc/vsftpd.conf file in your preferred text editor:

```
nano /etc/vsftpd.conf
```

```
# Example config file /etc/vsftpd.conf
#
# Run standalone? vsftpd can run either from an inetc or as a standalone
# daemon started from an initscript.
listen=YES
#
# Allow anonymous FTP? (Disabled by default)
anonymous_enable=NO
#
# Uncomment this to allow local users to log in.
local_enable=YES
#
# Uncomment this to enable any form of FTP write command.
#write_enable=YES
#
# You may restrict local users to their home directories. See the FAQ for
# the possible risks in this before using chroot_local_user or
# chroot_list_enable below.
#chroot_local_user=YES
```

The critical settings seen above are outlined below:

listen=YES tells vsftpd to run as a standalone daemon (the simplest method for getting up and running).anonymous_enable=NO disallows anonymous FTP users, which is generally preferred for security reasons but can be enabled for testing purposes.

local_enable=YES allows any user account defined in the /etc/passwd file access to the FTP server and is generally how most FTP users will connect.

write_enable=YES is commented out by default, but removing the hash (#) allows files to be uploaded to the FTP server.chroot_local_user=YES restricts users to their home directory and is also commented out by default.

To begin your testing and make sure everything is working, start with the following settings for the above parameters:

```
listen=YES
anonymous_enable=YES
local_enable=YES
write_enable=YES
chroot_local_user=YES
```

Save the vsftpd.conf file then restart the vsftpd service for the changes to take effect:

```
sudo service vsftpd restart
vsftpd stop/waiting
vsftpd start/running, process 18954
```

TESTING YOUR FTP SERVER

To quickly determine if your server was installed properly and is up and running, try to connect to the FTP server from your active shell, using the name anonymous and a blank password:

```
ftp localhost
Connected to localhost.
220 (vsFTPD 3.0.2)
Name (localhost:root): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

With both anonymous_enable and local_enable set to "YES" in the configuration, you should be able to successfully login to your local FTP server as seen above!

With that out of the way, simply enter quit at the ftp> prompt to cancel out:

```
ftp> quit
221 Goodbye.
```

With the test complete, you may wish to disable anonymous access once again by setting anonymous_enable=NO in the /etc/vsftpd.conf file and restarting the service:

```
nano /etc/vsftpd.conf
# Set to NO to disable anonymous access
```

```
anonymous_enable=NO  
sudo service vsftpd restart  
vsftpd stop/waiting  
vsftpd start/running, process 18996
```

ADDING AN FTP USER

If this is a new server it may be advisable to add a specific user for FTP access. Doing so is a fairly simple process but begin by creating a new user:

```
sudo adduser foobarAdding user `foobar' ...Adding new group `foobar' (1000) ...Adding new user `foobar' (1000) with group `foobar' ...Creating home directory `/home/foobar' ...Copying files from `/etc/skel' ...Enter new UNIX password:Retype new UNIX password:passwd: password updated successfullyChanging the user information for foobarEnter the new value, or press ENTER for the default      Full Name []:      Room Number []:      Work Phone []:      Home Phone []:      Other []:Is the information correct? [Y/n] Y
```

With a new user added you can now connect to your server remotely with an FTP client such as FileZilla, but you will immediately run into an error:

```
Status: Connecting to 104.131.170.253:21...Status: Connection established, waiting for welcome message...Response: 220 (vsFTPd 3.0.2)Command: USER foobarResponse: 331 Please specify the password.Command: PASS *****Response: 500 OOPS: vsftpd: refusing to run with writable root inside chroot()
```

The "500 OOPS" error vsftpd returns is a security measure designed to prevent *writable* root access for FTP users by default. To resolve this issue there are two main options available.

Allowing Writable User-root Access

The simplest method is to alter the /etc/vsftpd.conf file once again and enable one particular setting:

nano /etc/vsftpd.conf

```
# Allow users to write to their root directory  
allow_writeable_chroot=YES
```

With allow_writeable_chroot enabled following a service vsftpd restart, you can now successfully FTP into your server remotely as your newly created user:

```
Status: Connecting to 104.131.170.253:21...Status: Connection established, waiting for  
welcome message...Response: 220 (vsFTPd 3.0.2)Command: USER foobarResponse: 331  
Please specify the password.Command: PASS *****Response: 230 Login  
successful.
```

Using Writeable Subdirectories

The other option to maintain slightly stronger security is not to enable `allow_writeable_chroot` as outlined above, but instead to create a new subdirectory in the user's root directory with write access:

```
sudo chown root:root /home/foobar  
sudo mkdir /home/foobar/uploads  
sudo chown foobar:foobar /home/foobar/uploads  
sudo service vsftpd restart
```

Now when you connect remotely to your FTP server as the new user, that user will not have write access to the root directory, but will instead have full write access to upload files into the newly created `uploads` directory instead.

SECURING YOUR FTP WITH SSL

While standard unencrypted FTP access as outlined so far is sufficient in many cases, when transferring sensitive information over FTP it is useful to utilize a more secure connection using SSL.

To begin you'll likely need to generate a new SSL certificate with the following command, following the prompts as appropriate to complete the process:

```
sudo openssl req -x509 -nodes -days 365 -newkey rsa:1024 -keyout /etc/ssl/private/vsftpd.pem  
-out /etc/ssl/private/vsftpd.pem
```

Now you must ensure that `vsftpd` is aware of the SSL certificate. Open the `/etc/vsftpd.conf` file once again:

```
sudo nano /etc/vsftpd.conf
```

Look near the bottom of the file for two `rsa_` settings like this, indicating the location of the SSL certificate that was just created:

```
rsa_cert_file=/etc/ssl/private/vsftpd.pem  
rsa_private_key_file=/etc/ssl/private/vsftpd.pem
```

If those lines don't exist or match the appropriate path to the SSL certificate created, update them accordingly.

Additionally, there are a number of configuration settings to handle SSL connections, particularly forcing use of the TLS protocol which is ideal:

```
ssl_enable=YES  
allow_anon_ssl=NO  
force_local_data_ssl=YES  
force_local_logins_ssl=YES  
ssl_tlsv1=YES  
ssl_sslv2=NO  
ssl_sslv3=NO  
require_ssl_reuse=NO  
ssl_ciphers=HIGH
```

Some of the settings are self-explanatory, but the key components are the overall enabling of SSL, the restriction to use only TLS, and disallowing anonymous access.

With the settings added and the file saved, once again restart the vsftpd service:

sudo service vsftpd restart

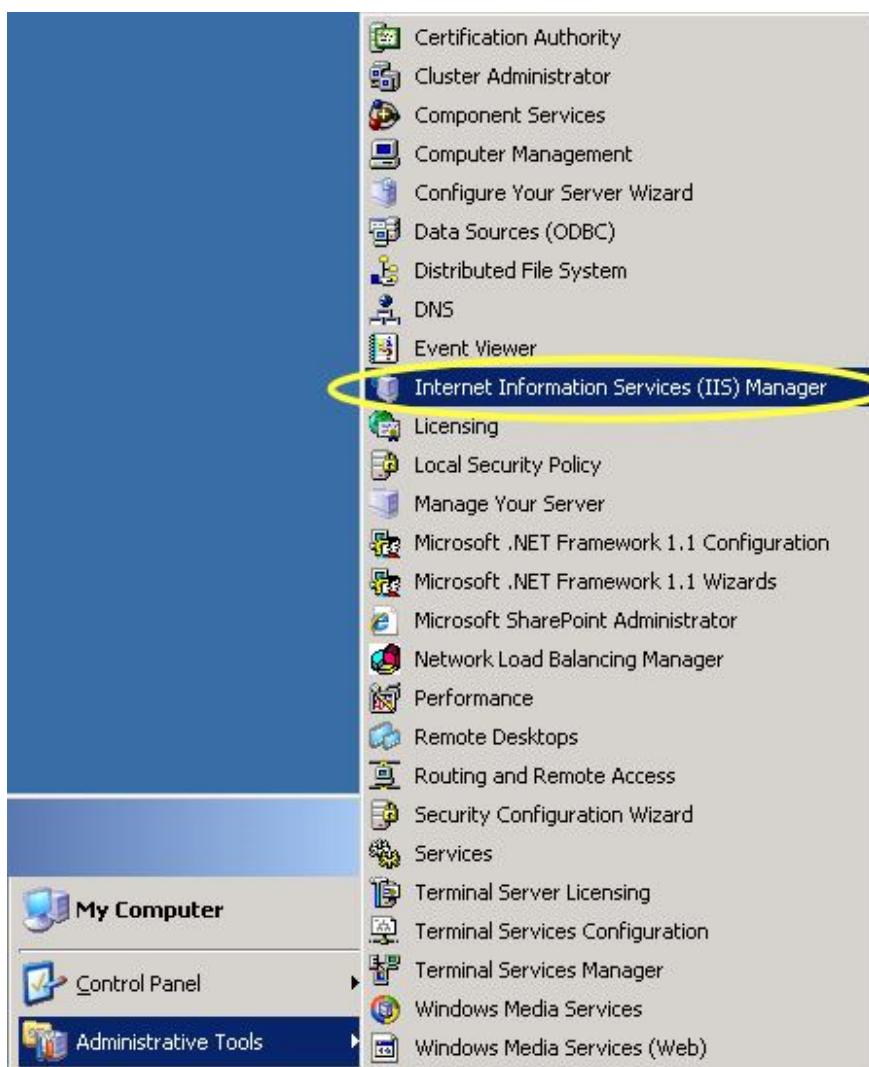
Now your FTP server is ready to accept secure connections using "FTP over TLS" encryption. Using a client such as FileZilla, you will be presented with a certificate popup asking to verify the newly created SSL certification.

Upon accepting you will now be securely connected and transfers will be encrypted via SSL:

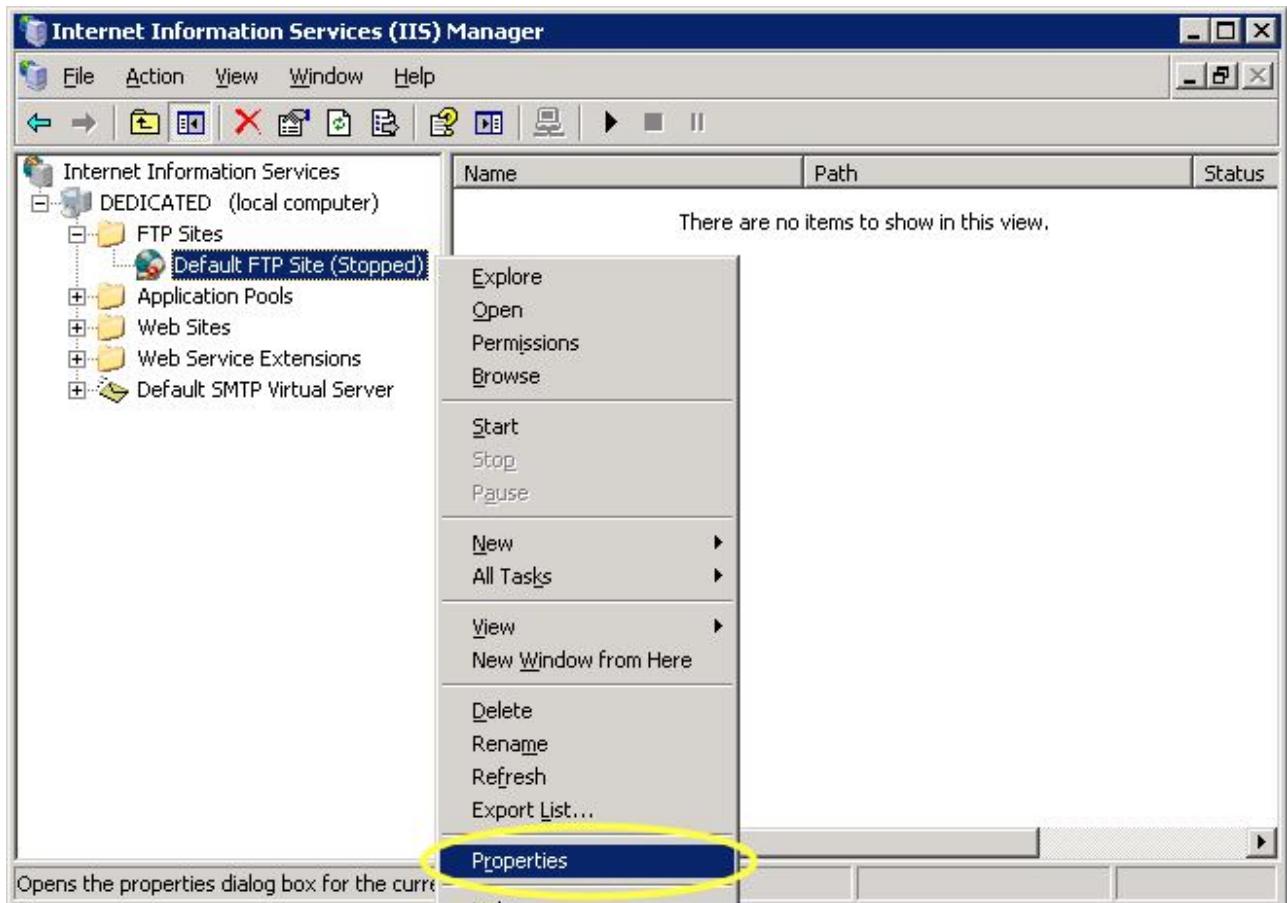
```
Status: Connecting to 104.131.170.253:21...Status: Connection established, waiting for welcome message...Response: 220 (vsFTPD 3.0.2)Command: AUTH TLSResponse: 234 Proceed with negotiation.Status: Initializing TLS...Status: Verifying certificate...Command: USER foobarStatus: TLS/SSL connection established.Response: 331 Please specify the password.Command: PASS *****Response: 230 Login successful.
```

Creating and Configuring FTP Server in Windows Server 2003

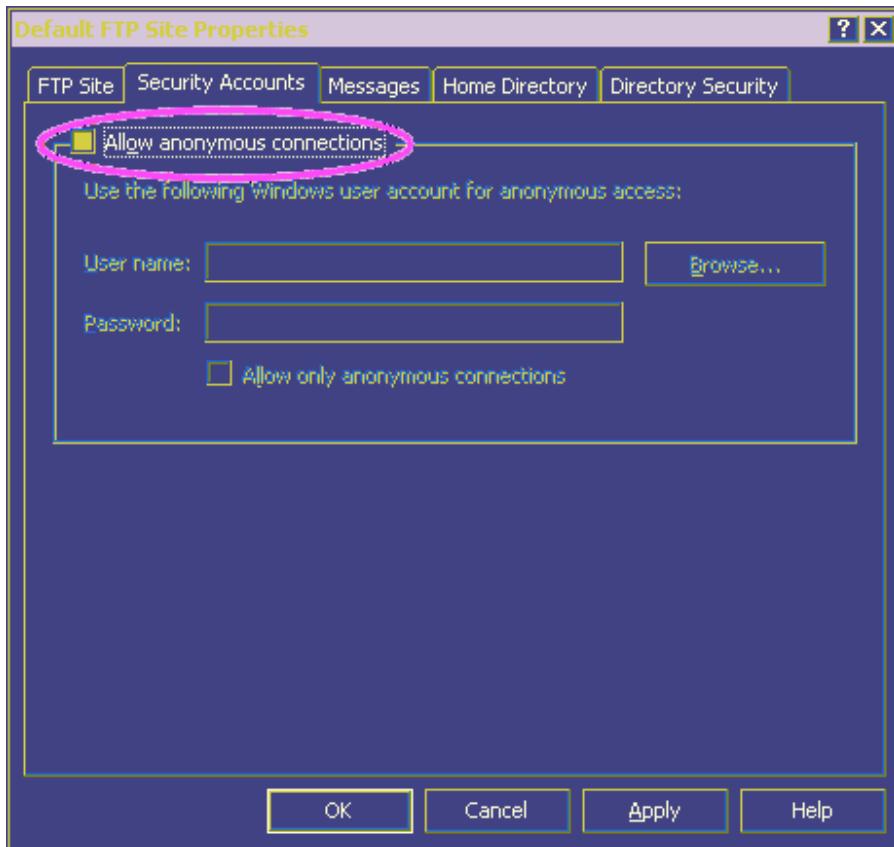
To enable the FTP service, go to the start menu -> Administrative Tools -> Internet Information Services (IIS) Manager.



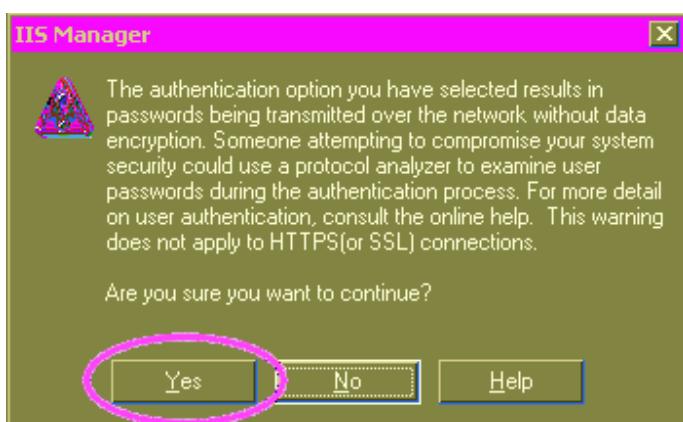
Tree open your server in the Internet Information Services list. Here my server is called "DEDICATED". Then tree open FTP Sites and right-click Default FTP Site (Stopped). Click Properties from the context menu.



On the dialog, choose the Security Accounts tab. Make sure to uncheck Allow anonymous connections. We don't want to allow anonymous access to the FTP server or we will have spammers, porn-servers, and who knows what else on here in a matter of days. We only want to allow authenticated user accounts to connect.

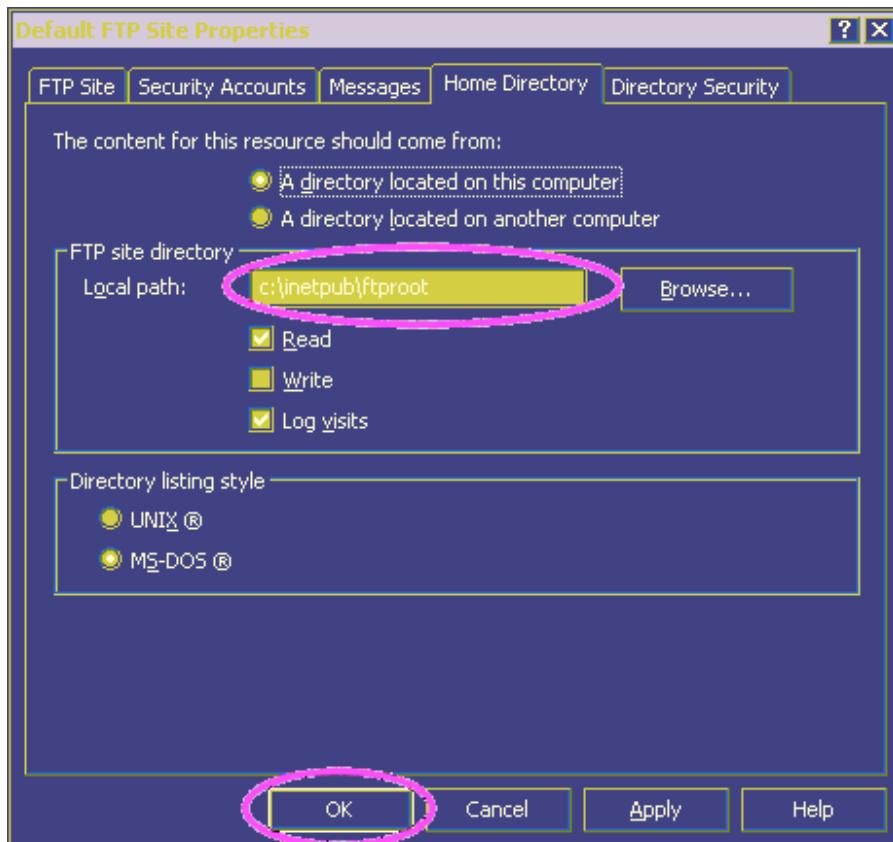


When you uncheck this box, you will see a dialog like this. Basically, this is just telling you that since you don't want to allow anonymous access, you will have to login with a UserName and Password. Since FTP is not a secure protocol, these credentials will be passed in clear text and there is a remote possibility that someone could see the credentials. In other words, this is saying, make sure that you don't use base windows accounts that you want to be secure. I recommend using a dummy ftp account that you change on a regular basis instead. Just click Yes on this dialog.

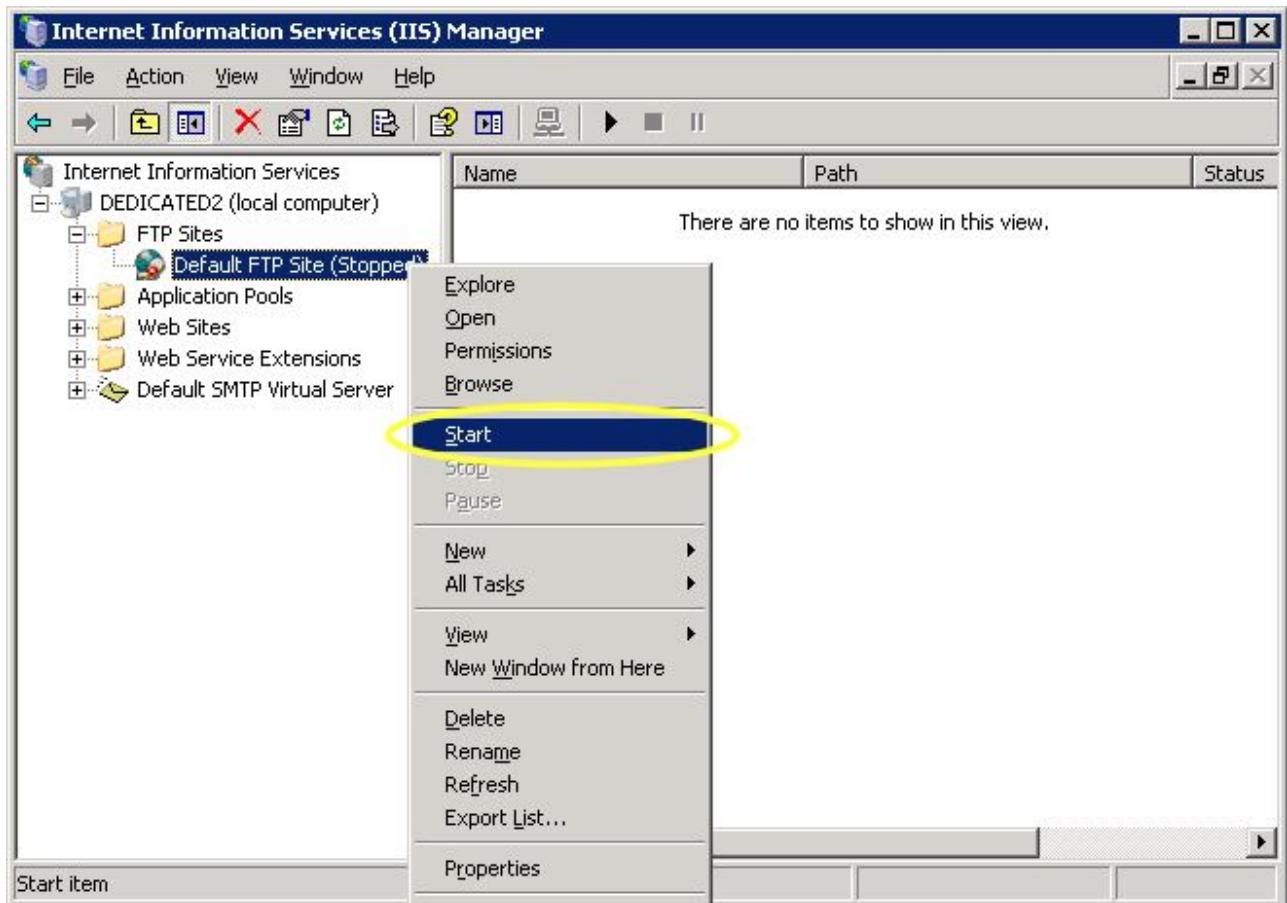


On the Home Directory tab, set the path to where you want your FTP files to be placed. NOTE: By default the path is set to `inetpub\ftproot`. If you want to allow users to create directories and add files instead of just downloading, make sure the Write box is checked. Then click OK to

apply all these changes.

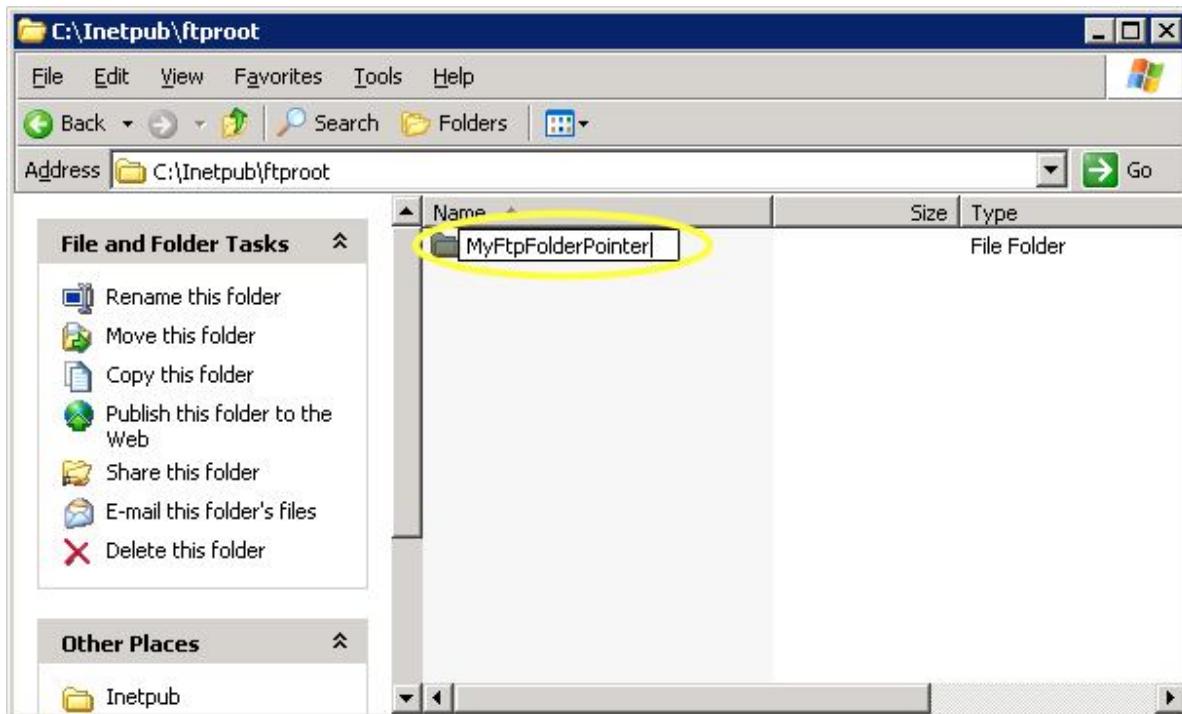


Now we want to start our FTP service. Right-click the Default FTP Site (Stopped) in the tree view and select Start to run the FTP server.

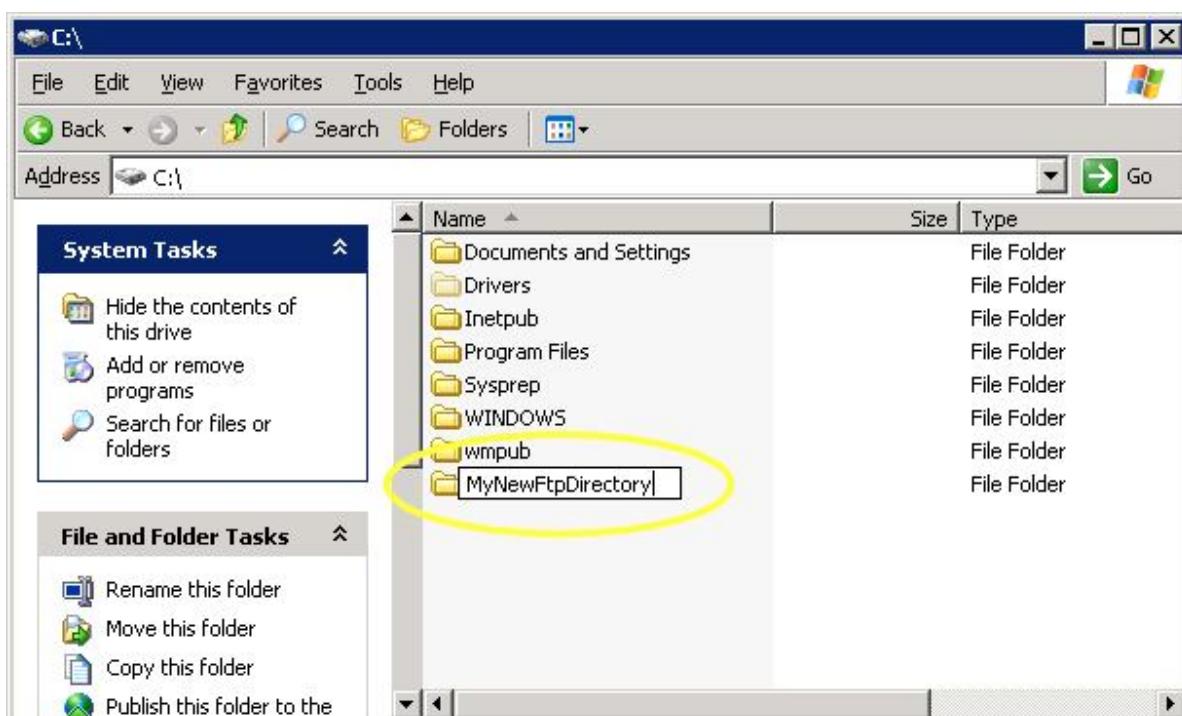


Setting Up FTP Directories & Permissions

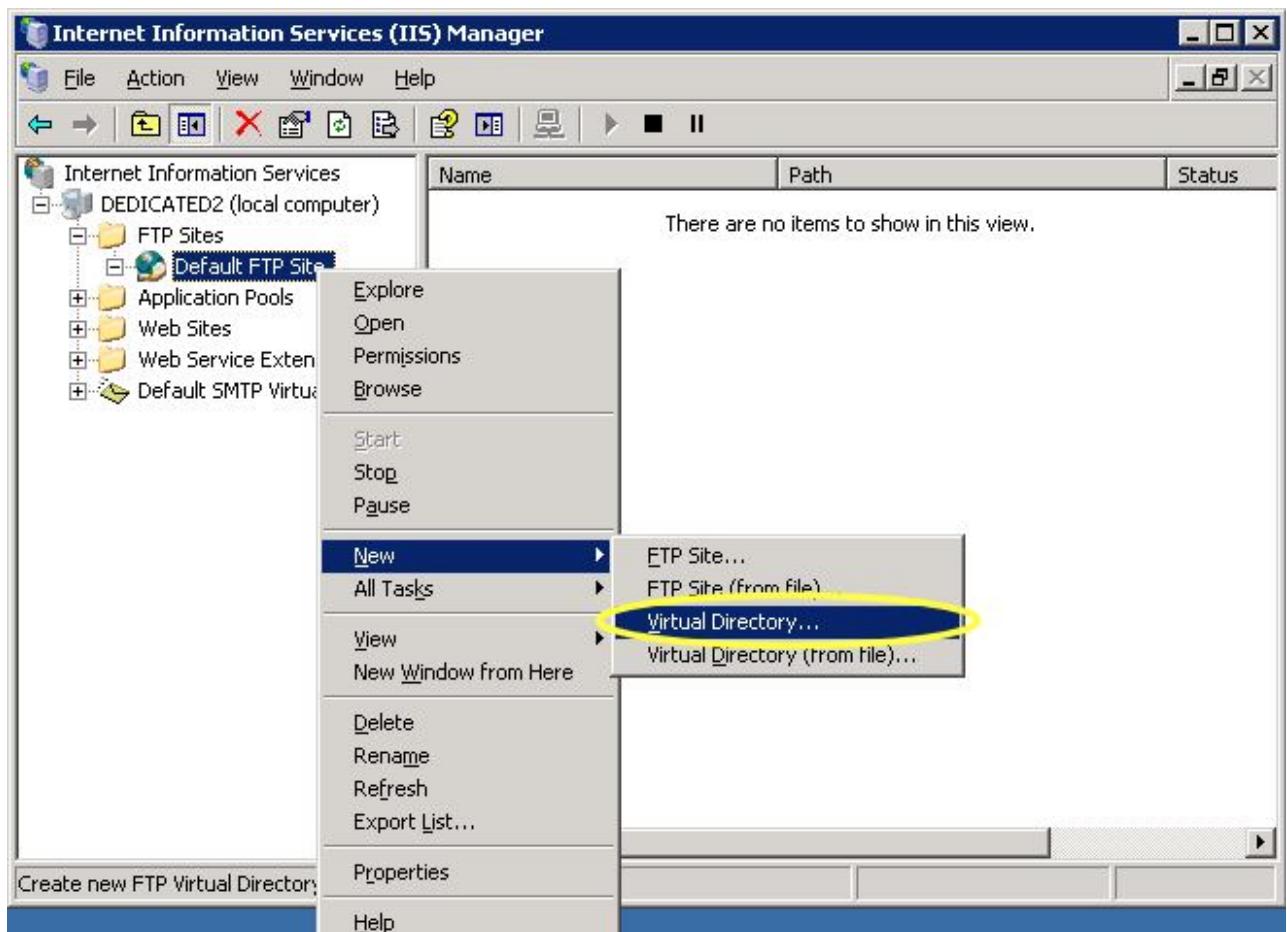
The previous tasks are all you need to do if you want to just put files in the mail FTP directory. But sometimes you want to set up specific directories for users that actually put the files in different directories than the default directory. The way you do this is to set up a "pointer" directory in your default `inetpub\ftproot` that will just be an empty folder (FTP Service requires this for a virtual directory). Here, I created a new folder in my default FTP root folder called "MyFtpFolderPointer".



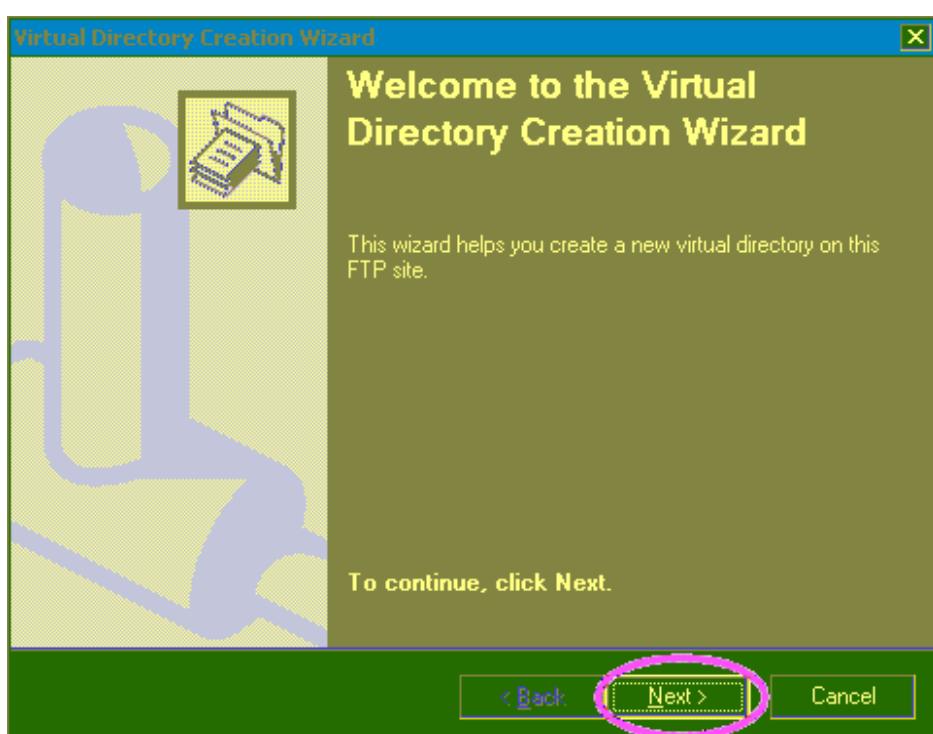
Now, we create a folder where we actually want our files to be placed when they are uploaded/downloaded. So I put a folder in the C:\ drive and called it "MyNewFtpDirectory". This is the place where the FTP files will actually go and the folder we created in the previous step will point to this folder.



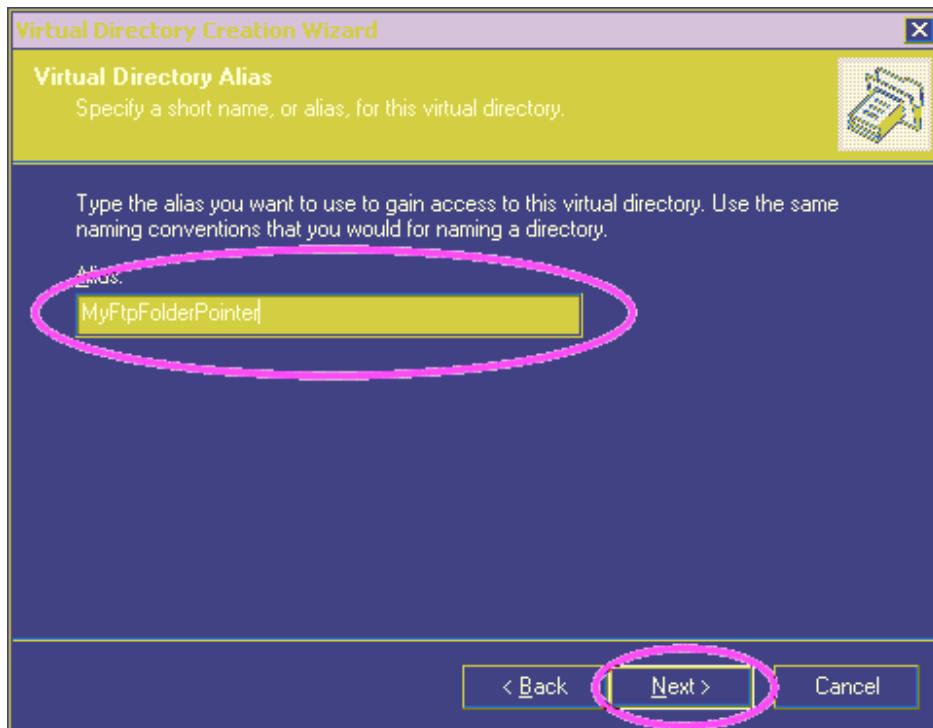
Now go back to the Internet Information Services (IIS) Manager and right-click the Default FTP Site. Choose New -> Virtual Directory... to start the virtual directory wizard.



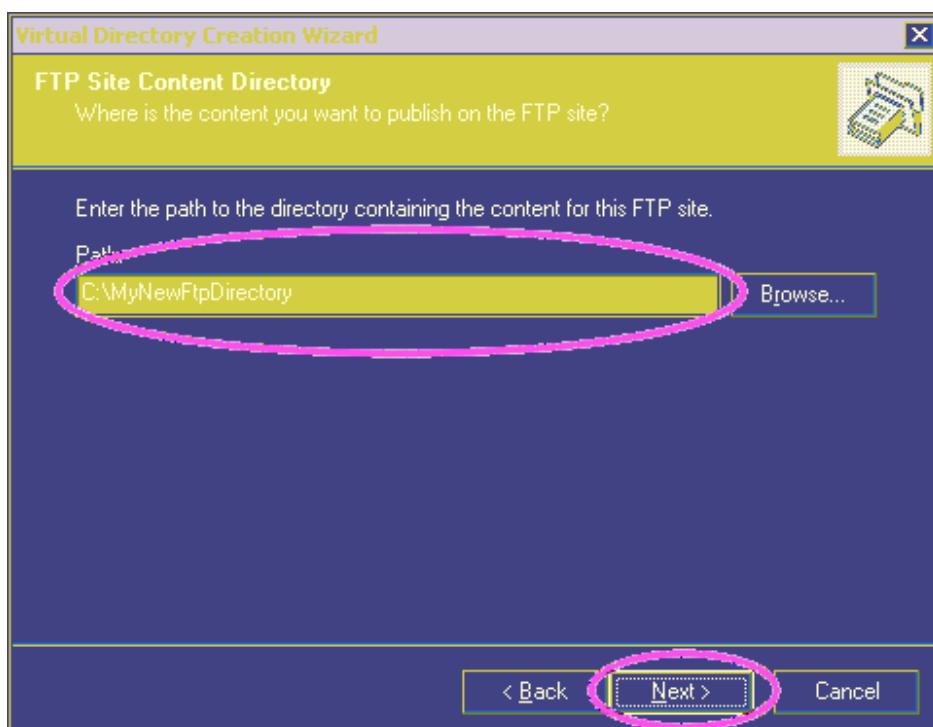
Click Next to start the wizard.



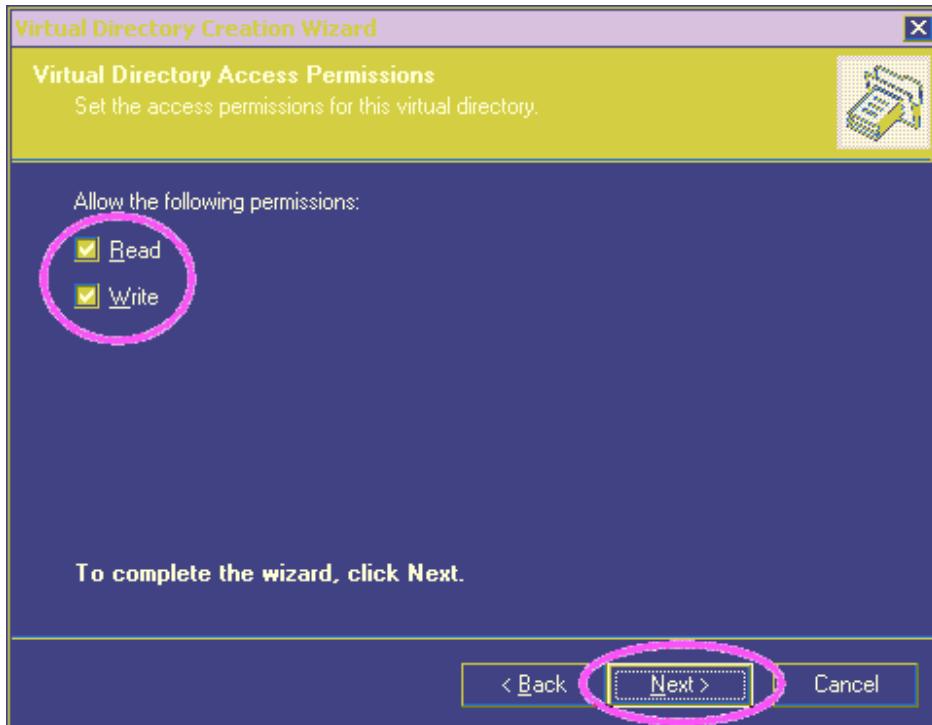
Name your alias for this virtual directory the EXACT same name you named the Virtual Directory folder we created in the FTP root since this is the one we want to point to the C:\ drive folder. So here, we name our Virtual Directory "MyFtpFolderPointer". Click Next.



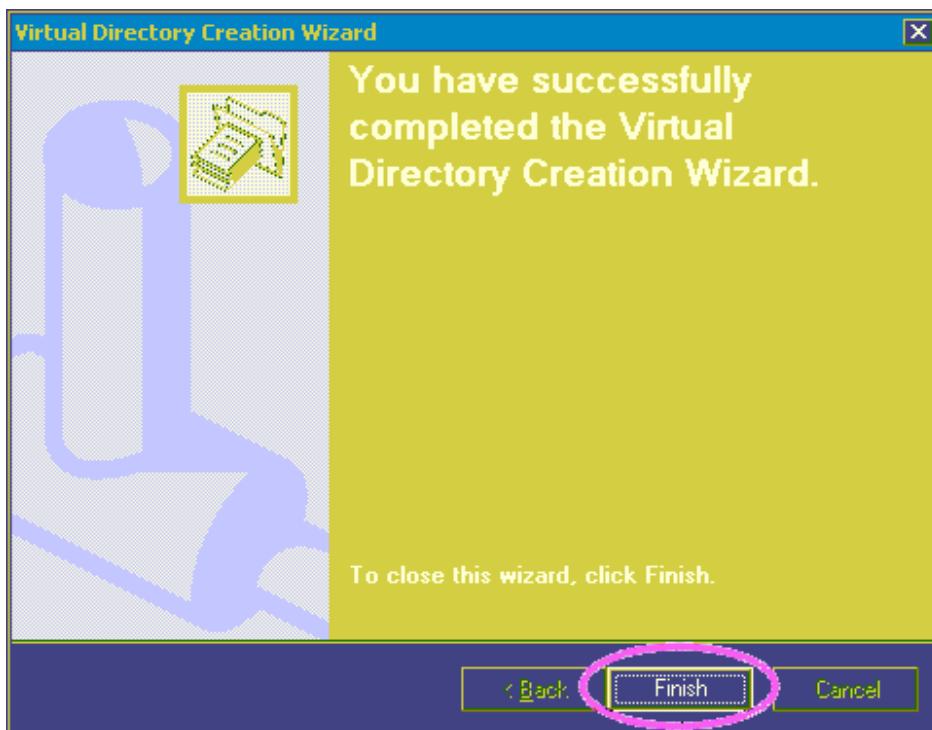
Now we tell this virtual directory what it's actual path should be. So point this path to the folder where you actually want the FTP files to go. It can be any path you want. Here we put the path to our folder on C:\MyNewFtpDirectory. Click Next.



If you want users to be able to both download and upload files to this FTP directory, check both the Read and Write boxes. Click Next.



Click Finish to complete the wizard and apply the virtual directory settings.



By default, FTP used Port 21 so you will need make sure that your Windows Firewall (if that is what you are using) is configured to allow Port 21 for FTP.

Setting up a DHCP server in Ubuntu

Installation

At a terminal prompt, enter the following command to install dhcpcd:

```
sudo apt-get install isc-dhcp-server
```

You will probably need to change the default configuration by editing */etc/dhcp3/dhcpd.conf* to suit your needs and particular configuration.

You also need to edit */etc/default/isc-dhcp-server* to specify the interfaces **dhcpcd** should listen to.

By default it listens to eth0.

Also, you have to assign a static ip to the interface that you will use for dhcp. If you will use eth0 for providing addresses in the 192.168.1.x subnet then you should assign for instance ip 192.168.1.1 to the eth0 interface using NetworkManager. Without this step you will get an error from dhcpcd when starting the service.

Configuration

The error message the installation ends with might be a little confusing, but the following steps will help you configure the service:

Most commonly, what you want to do is assign an IP address randomly. This can be done with settings as follows:

```
nano -w /etc/dhcp/dhcpd.conf
# Sample /etc/dhcpd.conf
# (add your comments here)
default-lease-time 600;
max-lease-time 7200;
option subnet-mask 255.255.255.0;
option broadcast-address 192.168.1.255;
option routers 192.168.1.254;
option domain-name-servers 192.168.1.1, 192.168.1.2;
option domain-name "mydomain.example";
```

```
subnet 192.168.1.0 netmask 255.255.255.0 {  
    range 192.168.1.10 192.168.1.100;  
    range 192.168.1.150 192.168.1.200;  
}
```

This will result in the DHCP server giving a client an IP address from the range 192.168.1.10-192.168.1.100 or 192.168.1.150-192.168.1.200. It will lease an IP address for 600 seconds if the client doesn't ask for a specific time frame. Otherwise the maximum (allowed) lease will be 7200 seconds. The server will also "advise" the client that it should use 255.255.255.0 as its subnet mask, 192.168.1.255 as its broadcast address, 192.168.1.254 as the router/gateway and 192.168.1.1 and 192.168.1.2 as its DNS servers.

If you need to specify a WINS server for your Windows clients, you will need to include the netbios-name-servers option, e.g.

```
nano -w /etc/dhcp/dhcpd.conf
```

```
option netbios-name-servers 192.168.1.1;
```

Start and stop service

```
sudo service isc-dhcp-server restart  
sudo service isc-dhcp-server start  
sudo service isc-dhcp-server stop
```

dhcp3-server and multiple interfaces

multiple interfaces example

Interface

```
nano -w /etc/network/interfaces
```

```
auto lo  
iface lo inet loopback
```

mapping hotplug

```
script grep
```

```
map eth1

iface eth1 inet dhcp

auto eth0
iface eth0 inet static
    address 10.152.187.1
    netmask 255.255.255.0

auto wlan0
iface wlan0 inet static
    address 192.168.1.1
    netmask 255.255.255.0
    up /sbin/iwconfig wlan0 mode TTTTT && /sbin/iwconfig wlan0 enc
restricted && /sbin/iwconfig wlan0 key [Y] XXXXXXXX && /sbin/iwconfig
wlan0 essid SSSSSSS

auto eth1
```

Select Interface card

```
nano -w /etc/default/isc-dhcp-server
INTERFACES="wlan0 eth0"
```

Configure Subnet

```
nano -w /etc/dhcp3/dhcpd.conf
ddns-update-style none;
log-facility local7;

subnet 192.168.1.0 netmask 255.255.255.0 {
```

```

option routers          192.168.1.1;
option subnet-mask      255.255.255.0;
option broadcast-address 192.168.1.255;
option domain-name-servers 194.168.4.100;
option ntp-servers      192.168.1.1;
option netbios-name-servers 192.168.1.1;
option netbios-node-type 2;
default-lease-time 86400;
max-lease-time 86400;

host bla1 {
    hardware ethernet DD:GH:DF:E5:F7:D7;
    fixed-address 192.168.1.2;
}

host bla2 {
    hardware ethernet 00:JJ:YU:38:AC:45;
    fixed-address 192.168.1.20;
}

}

subnet 10.152.187.0 netmask 255.255.255.0 {

    option routers          10.152.187.1;
    option subnet-mask      255.255.255.0;
    option broadcast-address 10.152.187.255;
    option domain-name-servers 194.168.4.100;
    option ntp-servers      10.152.187.1;
    option netbios-name-servers 10.152.187.1;
    option netbios-node-type 2;

    default-lease-time 86400;
}

```

```
max-lease-time 86400;

host bla3 {
    hardware ethernet 00:KK:HD:66:55:9B;
    fixed-address 10.152.187.2;
}
}
```

Check Route

ip route

```
192.168.1.0/24 dev wlan0 scope link
82.16.TT.0/24 dev eth1 scope link
10.152.187.0/24 dev eth0 scope link
default via 82.16.TT.UU dev eth1
```

Permission issues with ISC-DHCP server

Sometimes upon rising DHCP server informs about permission errors like

Can't open /etc/dhcp/dhcp.conf: permission denied

or

Can't open /var/lib/dhcp/dhcpd.leases: permission denied.

If after checking the permissions are found to be correct, check **apparmor** profile for dhcpcd:

shell# sudo apparmor_status

apparmor module is loaded.

15 profiles are loaded.

15 profiles are in enforce mode.

/sbin/dhclient

/usr/bin/evince

/usr/bin/evince-previewer

/usr/bin/evince-thumbnailer

```
/usr/lib/NetworkManager/nm-dhcp-client.action
```

```
/usr/lib/connman/scripts/dhclient-script
```

```
/usr/lib/cups/backend/cups-pdf
```

```
/usr/lib/telepathy/mission-control-5
```

```
/usr/lib/telepathy/telepathy-*
```

```
/usr/sbin/cupsd
```

```
/usr/sbin/dhcpcd
```

```
/usr/sbin/mysqld-akonadi
```

```
/usr/sbin/mysqld-akonadi///usr/sbin/mysqld
```

```
/usr/sbin/tcpdump
```

```
/usr/share/gdm/guest-session/Xsession
```

0 profiles are in complain mode.

4 processes have profiles defined.

4 processes are in enforce mode.

```
/sbin/dhclient (1092)
```

```
/sbin/dhclient (1093)
```

```
/usr/sbin/cupsd (978)
```

```
/usr/sbin/mysqld-akonadi///usr/sbin/mysqld (2136)
```

0 processes are in complain mode.

0 processes are unconfined but have a profile defined.

If **/usr/sbin/dhcpcd** is in the list of profiles do the following:

1. Stop apparmor deamon

```
sudo /etc/init.d/apparmor stop
```

2. Edit /etc/apparmor.d/usr.sbin.dhcpcd with root permissions and ensure that file has following lines:

```
/var/lib/dhcp/dhcpcd.leases* rwl,
```

```
/var/lib/dhcp/dhcpcd6.leases* rwl,
```

```
/etc/dhcp/dhcpcd.conf r,
```

```
/etc/dhcp/dhcpcd6.conf r,
```

1. /var/lib/dhcp/dhcpd6.leases and /etc/dhcp/dhcpd6.conf are needed to run DHCP server in IPV6 mode, for example:

```
dhcpd -6 -cf /etc/dhcp/dhcpd6.conf -lf /var/lib/dhcp/dhcpd6.leases eth0
```

1. 3.Start apparmor deamon

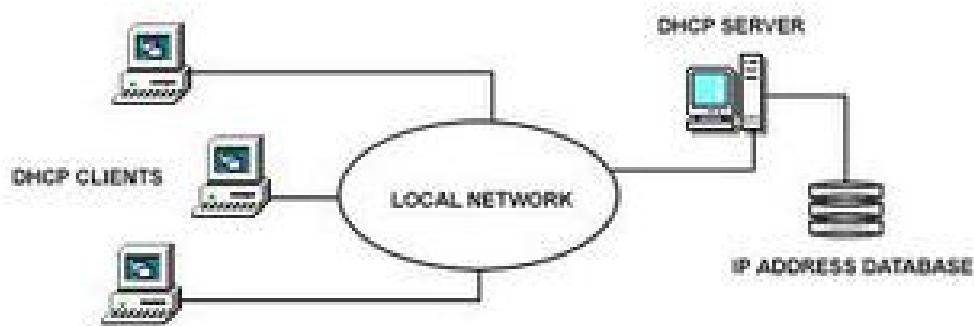
```
sudo /etc/init.d/apparmor start
```

After this operation apparmor deamon will allow dhcp server to open /etc/dhcp/dhcpd.conf or /var/lib/dhcp/dhcpd.leases files.

Setting up a DHCP server in Windows 2003

Dynamic Host Configuration Protocol (DHCP) is an IP standard designed to reduce the complexity of administering IP address configurations. - Microsoft's definition. A DHCP server would be set up with the appropriate settings for a given network. Such settings would include a set of fundamental parameters such as the gateway, DNS, subnet masks, and a range of IP addresses. Using DHCP on a network means administrators don't need to configure these settings individually for each client on the network. The DHCP would automatically distribute them to the clients itself.

The DHCP server assigns a client an IP address taken from a predefined scope for a given amount of time. If an IP address is required for longer than the lease has been set for, the client must request an extension before the lease expires. If the client has not requested an extension on the lease time, the IP address will be considered free and can be assigned to another client. If the user wishes to change IP address then they can do so by typing "ipconfig /release", followed by "ipconfig /renew" in the command prompt. This will remove the current IP address and request a new one. Reservations can be defined on the DHCP server to allow certain clients to have their own IP address (this will be discussed a little later on). Addresses can be reserved for a MAC address or a host name so these clients will have a fixed IP address that is configured automatically. Most Internet Service Providers use DHCP to assign new IP addresses to client computers when a customer connects to the internet - this simplifies things at user level.



The above diagram displays a simple structure consisting of a DHCP server and a number of client computers on a network.

DHCP (Dynamic Host Configuration Protocol) is a protocol that allows clients on a network to request network configuration settings from a server running the DHCP server service which, in our case, will be Windows Server 2003. Additionally the protocol allows the clients to self-configure those network configuration settings without the intervention of an administrator. Some of the settings that a DHCP server can provide to its clients include the IP addresses for the DNS servers, the IP addresses for the WINS servers, the IP address for the default gateway (usually a router) and, of course, an IP address for the client itself.

This assignment will discuss the steps of installing and configuring DHCP on a Windows Server 2003 member server, specifically focusing on setting up a scope and its accompanying settings. The same configuration can be applied to a standalone server even though the step-by-step details differ slightly. The upcoming '**Advanced DHCP Server Configuration on Windows 2003**' article will discuss other DHCP options and features such as superscopes, multicast scopes, dynamic DNS, DHCP Backup and more.

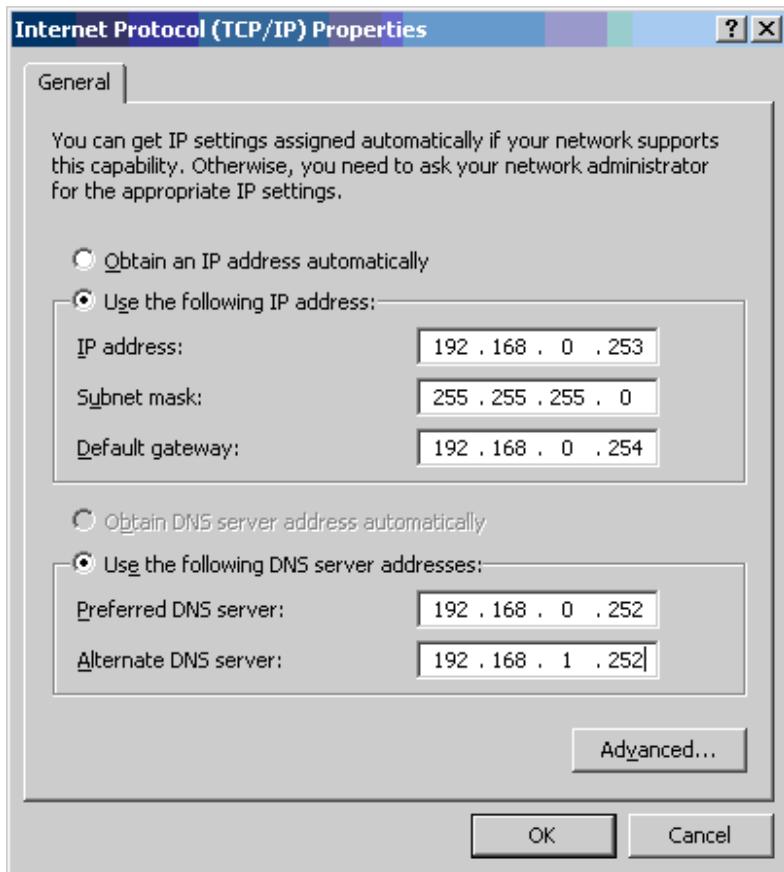
While we make use of specific IP addresses and network settings, you can change these settings as needed to make them compatible with your LAN – This won't require you to make changes to your LAN, but you'll need to have a slightly stronger understanding of DHCP and TCP/IP.

ASSIGNING THE SERVER A STATIC IP ADDRESS

Before we install the DHCP server service on Windows Server 2003, we need to assign the Windows server a static IP address. To do this:

1. Go to **Start > Control Panel > Network Connections**, right-click **Local Area Connection** and choose **Properties**.
2. When the **Local Area Connection Properties** window comes up, select **Internet Protocol (TCP/IP)** and click the **Properties** button.
3. When the **Internet Protocol (TCP/IP)** window comes up, enter an **IP address**, **subnet mask** and **default gateway** IP address that is compatible with your LAN.

We've configured our settings according to our network, as shown below:



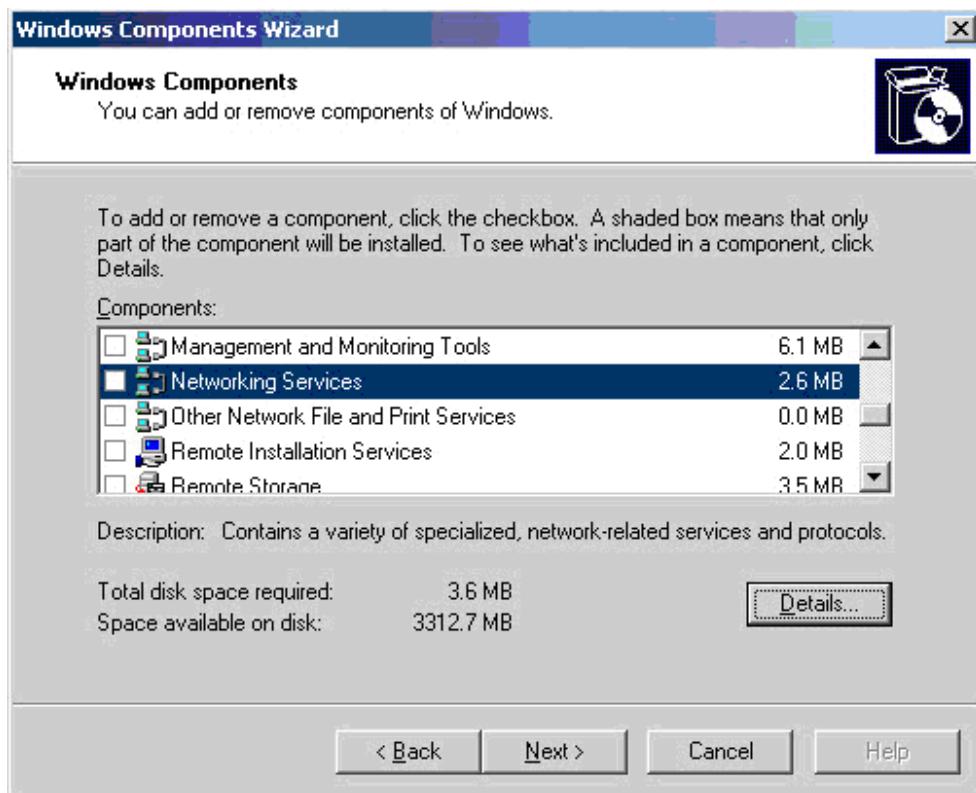
4. Enter **192.168.0.252** for the Preferred DNS server and **192.168.1.252** for the Alternate DNS server. The Preferred and Alternate DNS server IP addresses are optional for the functionality of the DHCP server, but we will populate them since you typically would in a real-world network. Usually these fields are populated with the IP addresses of your Active Directory domain controllers.

5. After filling out those fields, click **OK** and **OK** to save and close all windows.

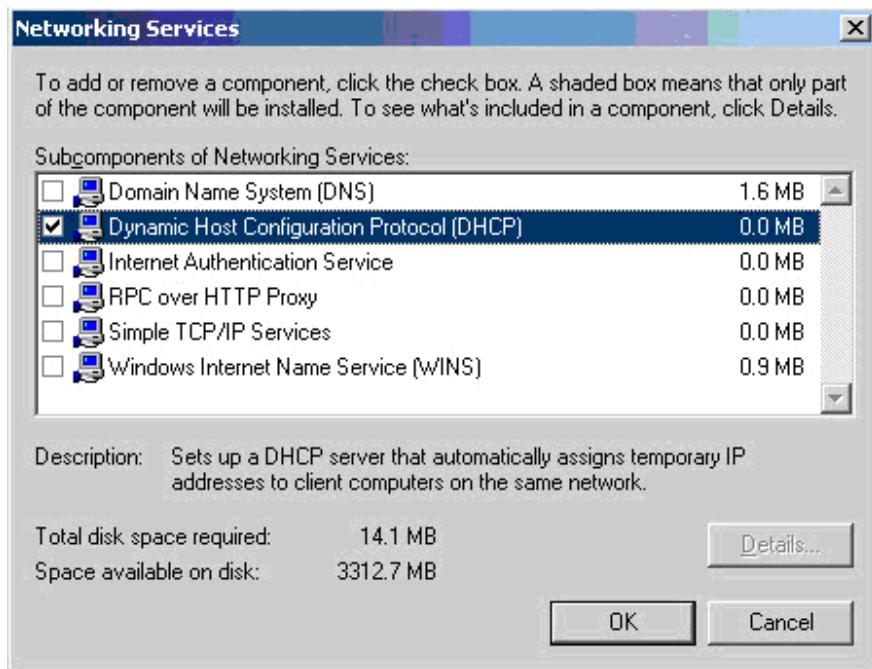
INSTALL DHCP SERVER SERVICE ON WINDOWS SERVER 2003

Our server now has a static IP address and we are now ready to install the DHCP server service. To do this:

1. Go to **Start > Control Panel > Add or Remove Programs**.
2. When the **Add or Remove Programs** window launches, click **Add/Remove Windows Components** in the left pane.
3. When the **Windows Components Wizard** comes up, scroll down and highlight **Networking Services** and then click the **Details** button.



4. When the **Networking Services** window comes up, place a check mark next to **Dynamic Host Configuration Protocol (DHCP)** and click **OK** and **OK** again.



Note that, during the install, Windows may generate an error claiming that it could not find a file needed for DHCP installation. If this happens, insert your Windows Server 2003 CD into the server's

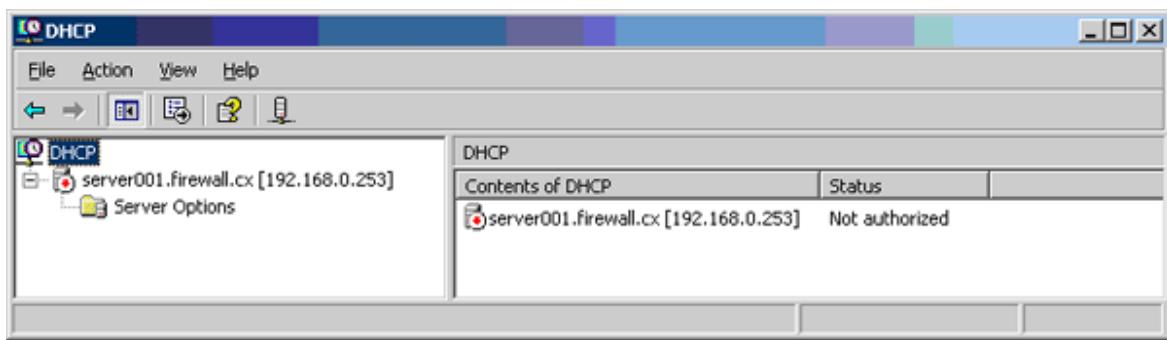
CD-ROM drive and browse to the i386 directory. The wizard should automatically find the file and allow you to select it. After that, the wizard should resume the installation process.

CONFIGURE DHCP ON WINDOWS SERVER 2003

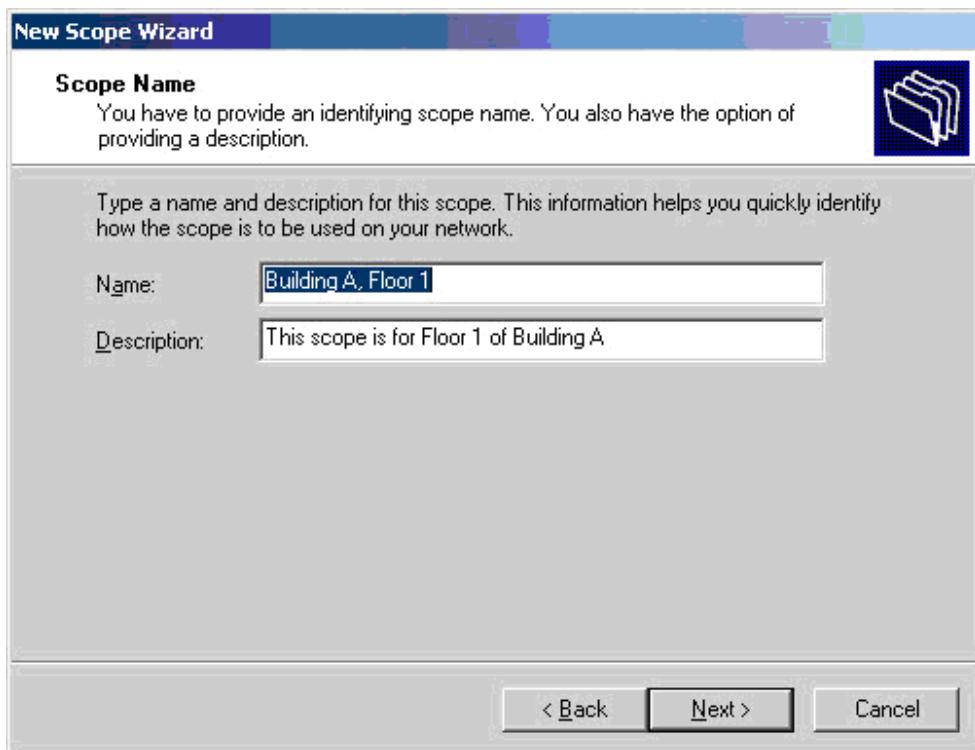
DHCP has now been successfully installed and we are ready to configure it. We will create a new scope and configure some of the scope's options. To begin:

1. Launch the DHCP MMC by going to **Start > Administrative Tools > DHCP**.

Currently, the DHCP MMC looks empty and the server node in the left pane has a red arrow pointing down. Keep that in mind because it will be significant later on.



2. Right-click the server node in the left pane and choose **New Scope**. This will launch the New Scope Wizard.
3. On the New Scope Wizard, click **Next**.
4. Specify a scope name and scope description. For the scope **Name**, enter "**Building A, Floor 1**". For the scope **Description**, enter "**This scope is for Floor 1 of Building A**." Afterwards, click **Next**.

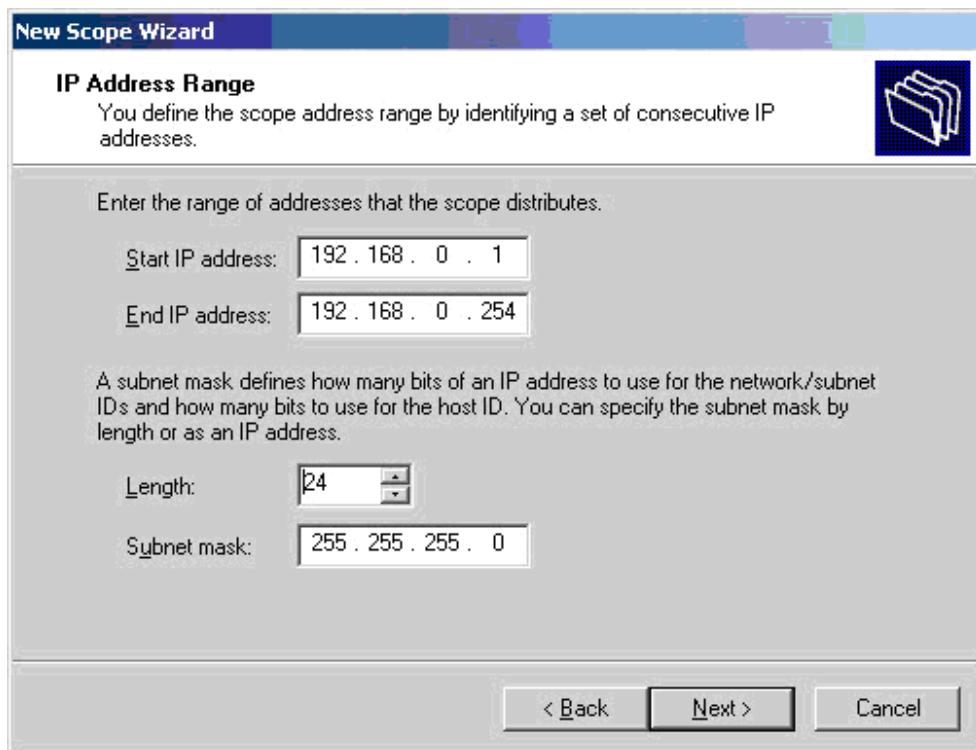


The scope name can be anything, but we certainly want to name it something that describes the scope's purpose. The scope Description is not required. It is there in case we needed to provide a broader description of the scope.

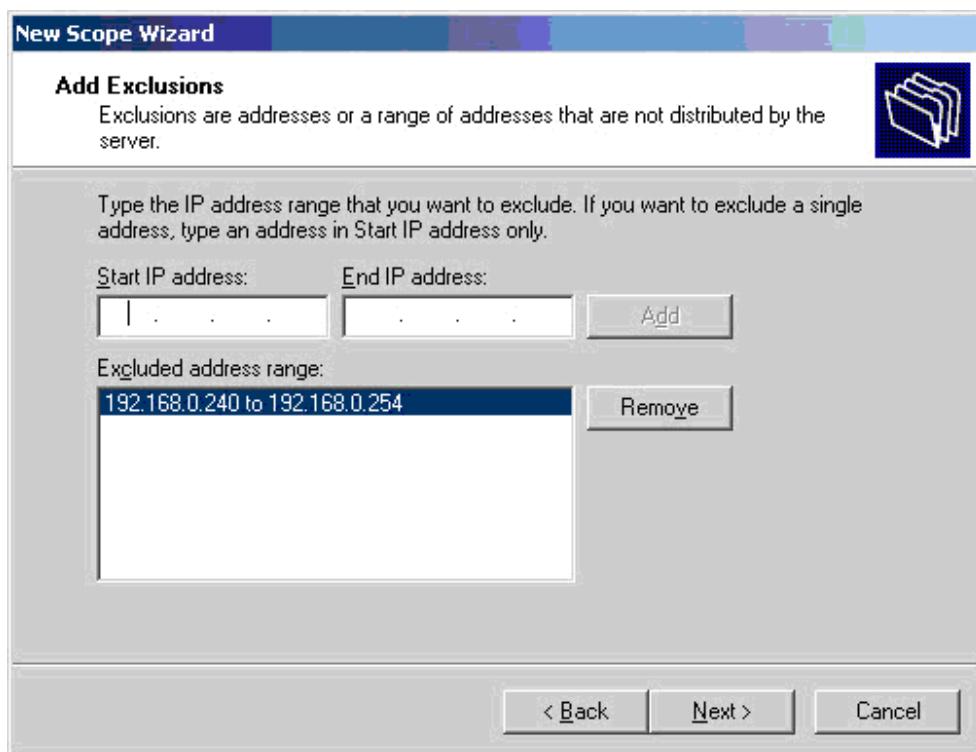
5. Specify an IP address range and subnet mask. For the **Start IP address** enter **192.168.0.1**, for the **End IP address** enter **192.168.0.254**. Finally, specify a **subnet mask** of **255.255.255.0** and click **Next**.

Specifying the IP address range of a scope requires some knowledge of subnetting. Each scope in a DHCP server holds a pool of IP addresses to give out to clients, and the range of IP addresses must be within the allowed range of the subnet (that you specify on the subnet mask field).

For simplicity we entered a classful, class C IP address range from 192.168.0.1 to 192.168.0.254. Notice that the range encompasses the IP address of our server, the DNS servers and the default gateway, meaning that the DHCP server could potentially assign a client an IP address that is already in use! Do not worry -- we will take care of that later.



6. Specify IP addresses to exclude from assignment. For the **Start IP address**, enter **192.168.0.240** and for the **End IP address** enter **192.168.0.254**, click **Add**, and then click **Next**.

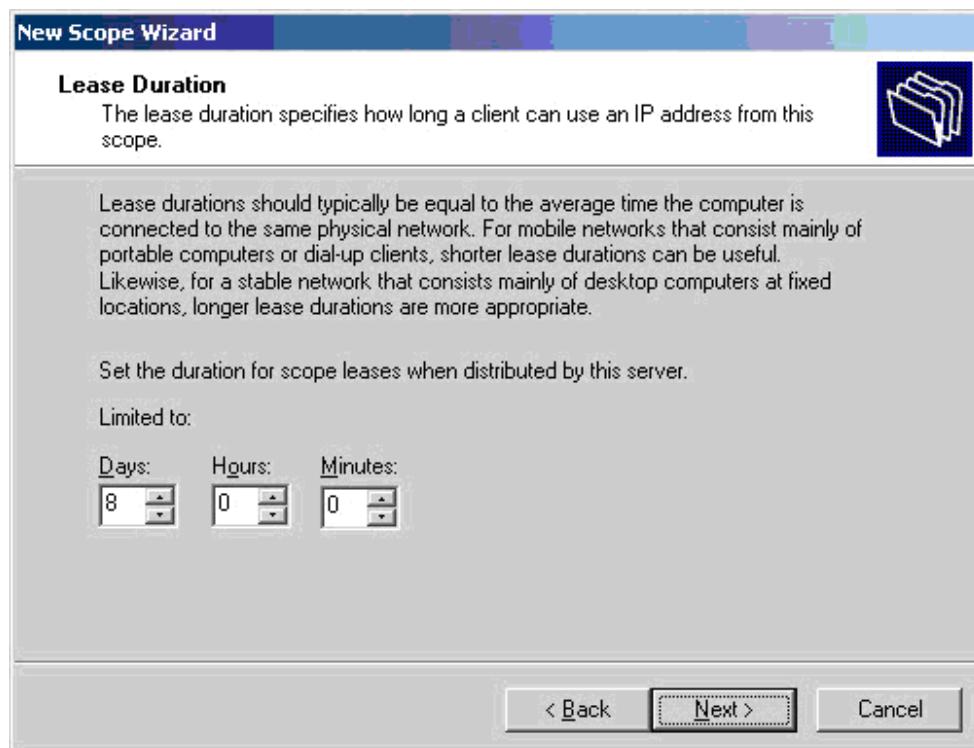


Certain network devices, such as servers, will need statically configured IP addresses. The IP addresses may sometimes be within the range of IP addresses defined for a scope. In those cases, you have to exclude the IP addresses from being assigned out by DHCP.

We have the opportunity here to define those IP addresses that are to be excluded. We specified IP addresses 192.168.0.240 to 192.168.0.254 to ensure we've included our servers plus a few spare IP addresses for future use.

7. Specify the lease duration for the scope. Verify that **Days** is **8** and click **Next**.

The lease duration is how long clients should keep their IP addresses before having to renew them.



There are a few considerations at this point. If a short lease duration is configured, clients will be renewing their IP addresses more frequently. The result will be additional network traffic and additional strain on the DHCP server. On the other hand if a long lease duration is configured, IP addresses previously obtained by decommissioned clients would remain leased and unavailable to future clients until the leases either expire or are manually deleted.

Additionally if network changes occur, such as the implementation of a new DNS server, those clients would not receive those updates until their leases expire or the computers are restarted.

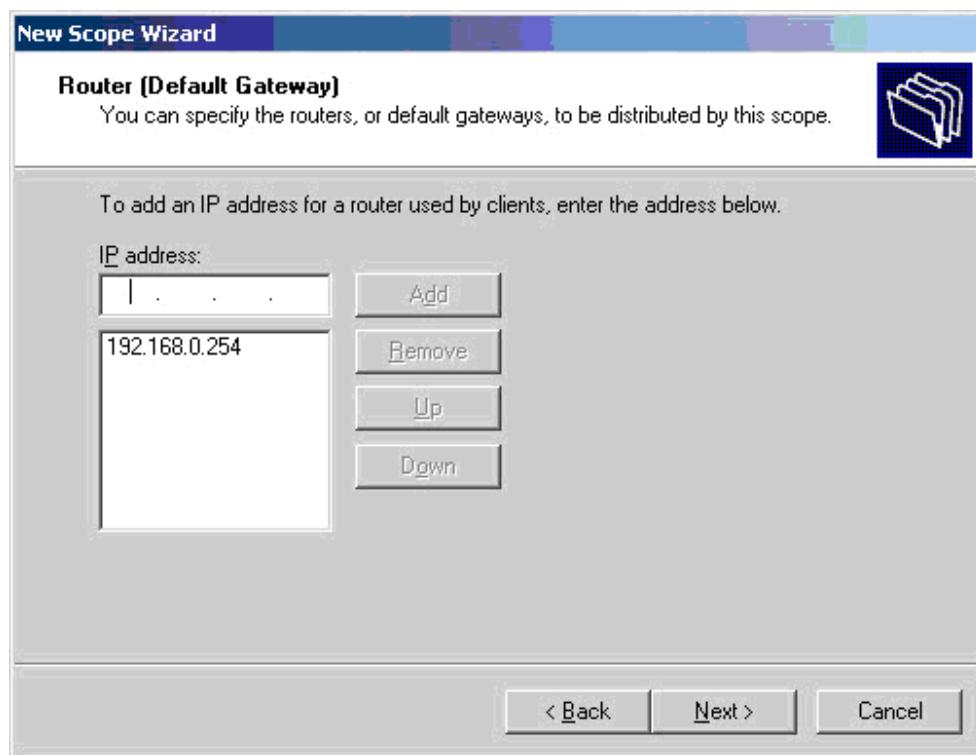
As Microsoft states, "*lease durations should typically be equal to the average time the computer is connected to the same physical network.*" You would typically leave the default lease duration in an environment where computers are rarely moved or replaced, such as a wired network. In an environment where computers are often moved and replaced, such as a wireless network, you would want to specify a short duration since a new wireless client could roam within range at any time.

8. Configure DHCP Options. Make sure "**Yes, I want to configure these settings now**" is selected and click **Next** to begin configuring DHCP options.

DHCP options are additional settings that the DHCP server can provide to clients when it issues them with IP addresses. These are the other settings that help clients communicate on the network. In the New Scope Wizard we can only configure a few options but from the DHCP MMC we have several more options.

9. Specify the router IP address. Enter **192.168.0.254** as the IP address of the subnet's router, click **Add**, and then click **Next**.

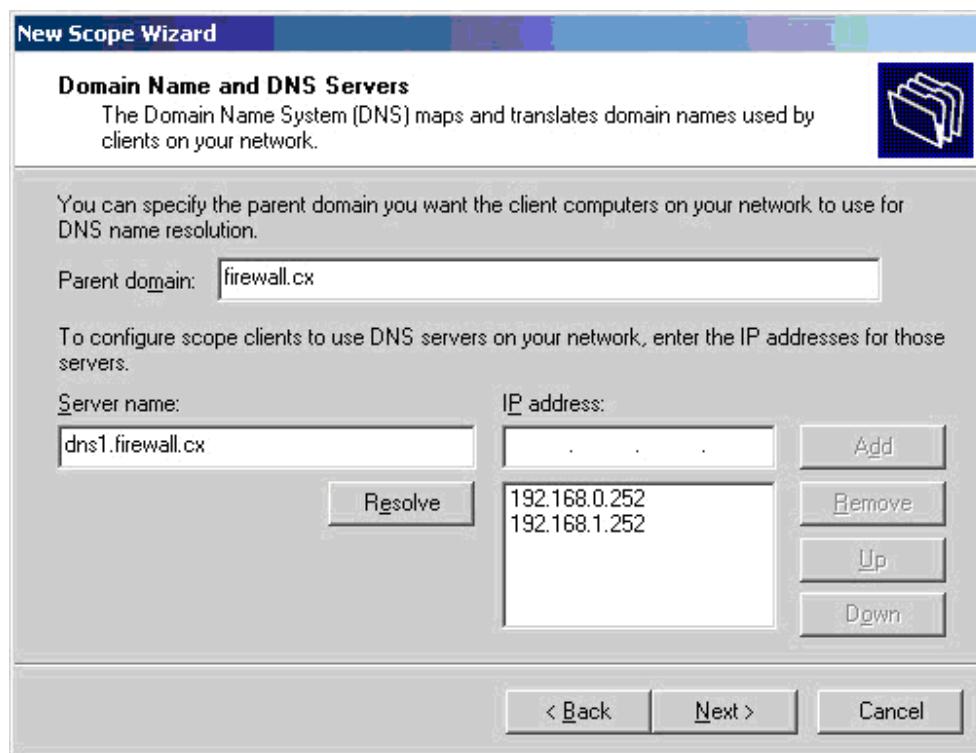
The first option we can configure is the IP address for the subnet's router for which this scope is providing IP addresses. Keep in mind that this IP address must be in the same network as the IP addresses in the range that we created earlier.



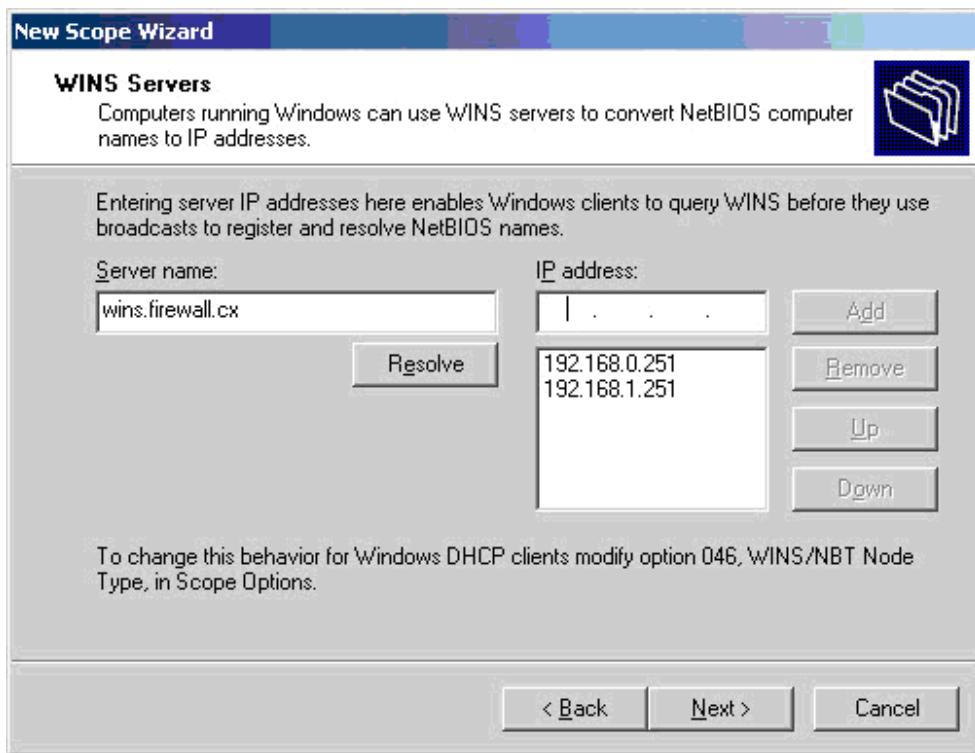
10. Configure domain name and DNS servers. On the next page, enter “**firewall.cx**” for the domain name. Then enter **192.168.0.252** for the IP address of a DNS server, click **Add**, enter **192.168.1.252** as the IP address for another DNS server, and click **Add** again. When finished, click **Next**.

If you had a DNS infrastructure in place, you could have simply typed in the fully qualified domain name of the DNS server and clicked**Resolve** .

The DNS servers will be used by clients primarily for name resolution, but also for other purposes that are beyond the scope of this article. The DNS domain name will be used by clients when registering their hostnames to the DNS zones on the DNS servers (covered in the '**Advanced DHCP Server Configuration on Windows 2003**' article).



11. Configure WINS servers. On the next screen, enter **192.168.0.251** as the IP address for the first WINS server, click **Add** , enter**192.168.1.251** as the IP address for the second WINS server, click **Add** again, and then click **Finish** .



12. Finally, the wizard asks whether you want to activate the scope. For now, choose “**No, I will activate this scope later**” and click**Next** and then **Finish** to conclude the New Scope Wizard and return to the DHCP MMC.

At this point we almost have a functional DHCP server. Let us go ahead and expand the scope node in the left pane of the DHCP MMC to see the new available nodes:

Address Pool – Shows the IP address range the scope offers along with any IP address exclusions.

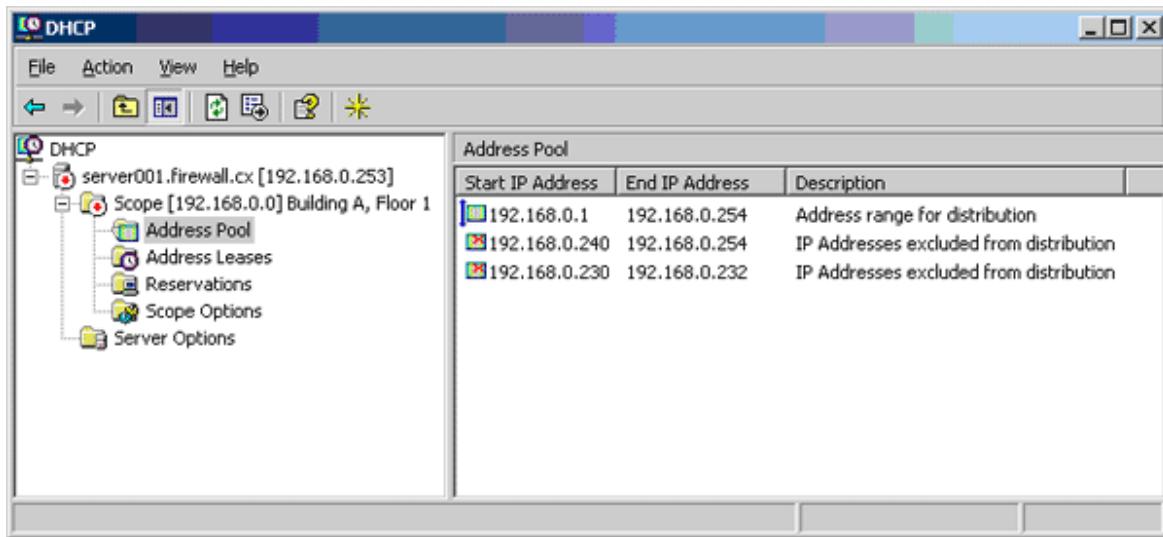
Address Leases – Shows all the leased IP addresses.

Reservations – Shows the IP addresses that are reserved. Reservations are made by specifying the MAC address that the server would “listen to” when IP address requests are received by the server. Certain network devices, such as networked printers, are best configured with reserved IP addresses rather than static IP addresses.

Scope Options – Shows configured scope options. Some of the visible options now are router, DNS, domain name and WINS options.

13. Select and right-click **Address Pool** and choose **New Exclusion Range**.

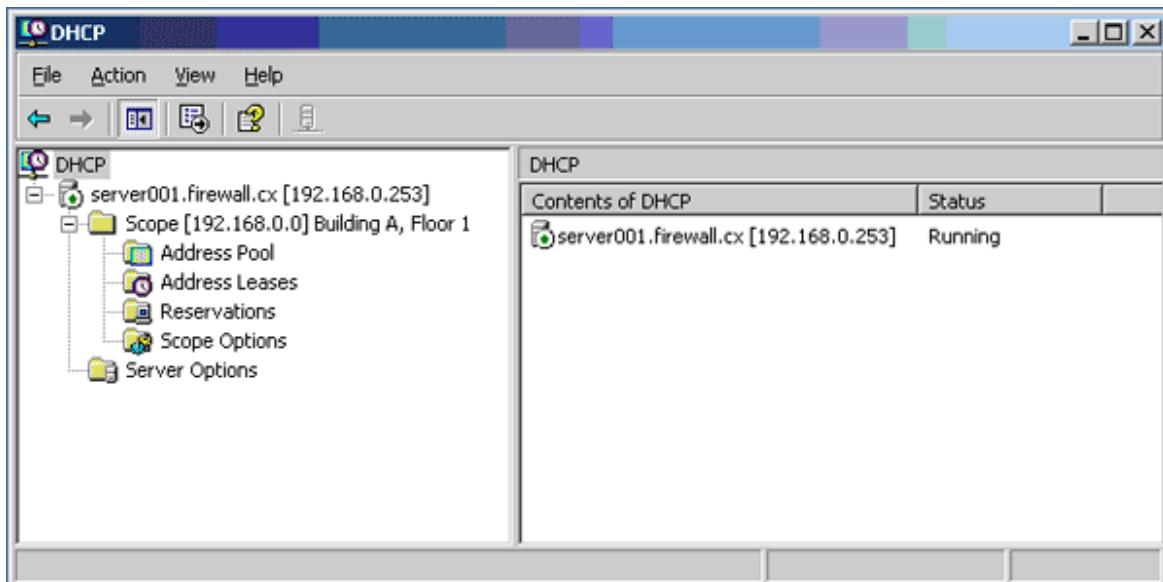
14. When the Add Exclusion window comes up, enter the required range and then click **Add**. In our example, we've excluded the addition range 192.168.0.230 - 192.168.0.232.



Notice that the server node and scope node still has a red arrow pointing down. These red arrows pointing down mean that the server and scope are not “turned on”.

The concept of “turning on” the scope is called “activating” and the concept of “turning on” the server for DHCP service is called “authorizing”. Security has some influence in the concept of authorizing a DHCP server and, to authorize a DHCP server, you must be a member of the Enterprise Admins Active Directory group.

15. Right-click the server (server001.firewall.cx) and choose **Authorize**, then right-click the scope (Building A, Floor 1) and choose**Activate**. If the red arrows remain, refresh the MMC by going to **Action > Refresh**.



How to create Wireless Sensor Network (WSN) in ns2

A wireless sensor network (WSN) consists of a large number of small sensor nodes that are deployed in the area in which a factor is to be monitored. In wireless sensor network, energy model is one of the optional attributes of a node. The energy model denotes the level of energy in a mobile node. The components required for designing energy model includes initialEnergy, txPower, rxPower, and idlePower. The “initialEnergy” represents the level of energy the node has at the initial stage of simulation. “txPower” and “rxPower” denotes the energy consumed for transmitting and receiving the packets. If the node is a sensor, the energy model should include a special component called “sensePower”. It denotes the energy consumed during the sensing operation. Apart from these components, it is important to specify the communication range (RXThresh_) and sensing range of a node (CSThresh_). The sample 18.tcl designs a WSN in which sensor nodes are configured with different communication and sensing range. Base Station is configured with highest communication range. Data Transmission is established between nodes using UDP agent and CBR traffic.

Sample Program

```
#Filename: sample18.tcl

*****SENSOR NETWORK *****

*****ENERGY MODEL *****88
*****Multiple node Creation and communication model using

UDP (User Datagram Protocol)and CBR (Constant Bit Rate)
*****88

# Simulator Instance Creation
set ns [new Simulator]

#Fixing the co-ordinate of simulation area
set val(x) 600
set val(y) 600
# Define options
set val(chan) Channel/WirelessChannel ;# channel type
set val(prop) Propagation/TwoRayGround ;# radio-propagation model
set val(netif1) Phy/WirelessPhy ;# network interface type
set val(netif2) Phy/WirelessPhy ;# network interface type
set val(mac) Mac/802_11 ;# MAC type
set val(ifq) Queue/DropTail/PriQueue ;# interface queue type
```

```

set val(ll) LL ;# link layer type
set val(ant) Antenna/OmniAntenna ;# antenna model
set val(ifqlen) 50 ;# max packet in ifq
set val(nn) 10 ;# number of mobilenodes
set val(rp) AODV ;# routing protocol
set val(x) 600 ;# X dimension of topography
set val(y) 600 ;# Y dimension of topography
set val(stop) 10.0 ;# time of simulation end
set val(energymodel) EnergyModel ;#Energy set up
# set up topography object
set topo [new Topography]
$topo load_flatgrid $val(x) $val(y)

# general operational descriptor- storing the hop details in the network
create-god $val(nn)

#Transmission range setup

***** UNITY GAIN, 1.5m HEIGHT OMNI
DIRECTIONAL ANTENNA SET UP *****

Antenna/OmniAntenna set X_ 0
Antenna/OmniAntenna set Y_ 0
Antenna/OmniAntenna set Z_ 1.5
Antenna/OmniAntenna set Gt_ 1.0
Antenna/OmniAntenna set Gr_ 1.0

***** SET UP COMMUNICATION AND
SENSING RANGE *****

#default communication range 250m

# Initialize the SharedMedia interface with parameters to make
# it work like the 914MHz Lucent WaveLAN DSSS radio interface
$val(netif1) set CPTthresh_ 10.0
$val(netif1) set CSTthresh_ 2.28289e-11 ;#sensing range of 500m
$val(netif1) set RXThresh_ 2.28289e-11 ;#communication range of 500m
$val(netif1) set Rb_ 2*1e6
$val(netif1) set Pt_ 0.2818
$val(netif1) set freq_ 914e+6
$val(netif1) set L_ 1.0

```

```

# Initialize the SharedMedia interface with parameters to make
# it work like the 914MHz Lucent WaveLAN DSSS radio interface
$val(netif2) set CPTthresh_ 10.0
$val(netif2) set CSThresh_ 8.91754e-10 ;#sensing range of 200m
$val(netif2) set RXThresh_ 8.91754e-10 ;#communication range of 200m
$val(netif2) set Rb_ 2*1e6
$val(netif2) set Pt_ 0.2818
$val(netif2) set freq_ 914e+6
$val(netif2) set L_ 1.0

# configure the first 5 nodes with transmission range of 500m

$ns node-config -adhocRouting $val(rp) \
-llType $val(ll) \
-macType $val(mac) \
-ifqType $val(ifq) \
-ifqLen $val(ifqlen) \
-antType $val(ant) \
-propType $val(prop) \
-phyType $val(netif1) \
-channelType $val(chan) \
-topoInstance $topo \
-energyModel $val(energymodel) \
-initialEnergy 10 \
-rxPower 0.5 \
-txPower 1.0 \
-idlePower 0.0 \
-sensePower 0.3 \
-agentTrace ON \
-routerTrace ON \
-macTrace OFF \
-movementTrace ON

# Node Creation

set energy(0) 1000

$ns node-config -initialEnergy 1000 \
-rxPower 0.5 \
-txPower 1.0 \

```

```

-idlePower 0.0 \
-sensePower 0.3

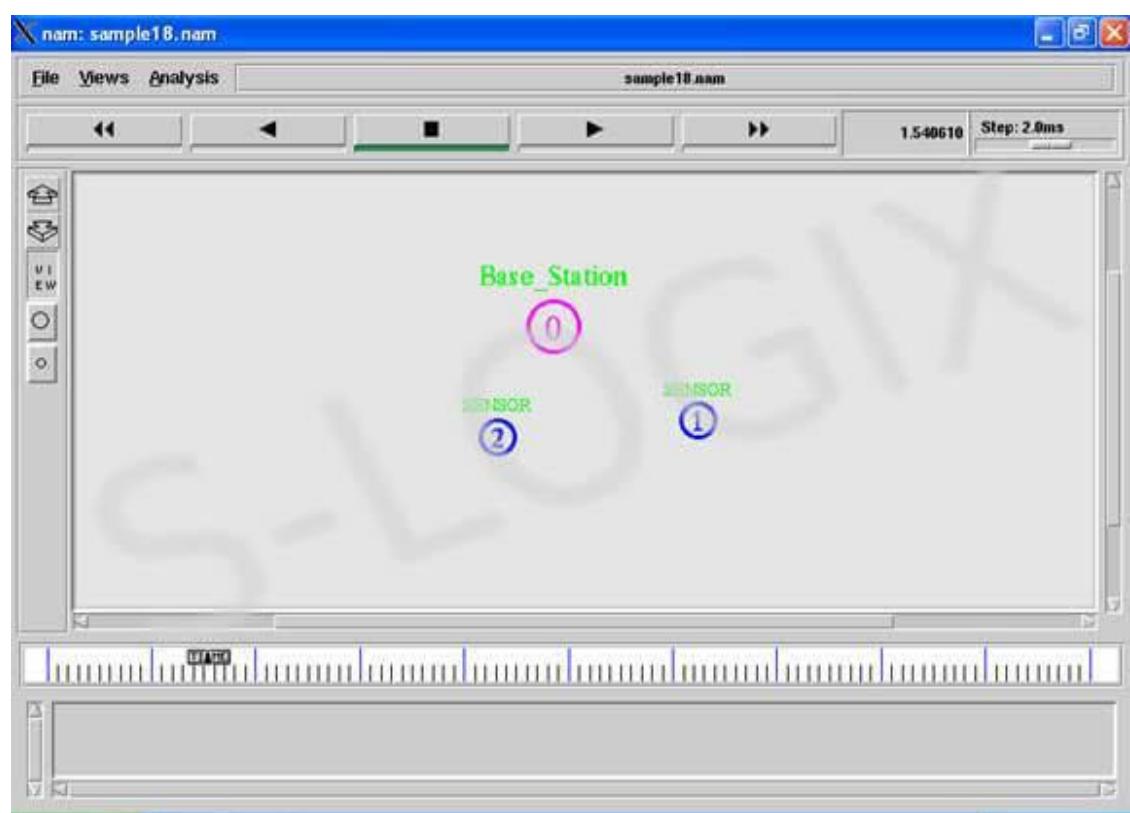
set node_(0) [$ns node]
$node_(0) color black
# configure the remaining 5 nodes with transmission range of 200m

$ns node-config -adhocRouting $val(rp) \
-llType $val(ll) \
-macType $val(mac) \
-ifqType $val(ifq) \
-ifqLen $val(ifqlen) \
-antType $val(ant) \
-propType $val(prop) \
-phyType $val(netif2) \
-channelType $val(chan) \
-topoInstance $topo \
-energyModel $val(energymodel) \
-initialEnergy 10 \
-rxPower 0.5 \
-txPower 1.0 \
-idlePower 0.0 \
-sensePower 0.3 \
-agentTrace ON \
-routerTrace ON \
-macTrace OFF \
-movementTrace ON
for {set i 1} {$i < 3} {incr i} {

    set energy($i) [expr rand()*500]

$ns node-config -initialEnergy $energy($i) \
-rxPower 0.5 \
-txPower 1.0 \
-idlePower 0.0 \
-sensePower 0.3
set node_($i) [$ns node]
$node_($i) color black
}

```



Assignment No: 9

Problem statement : Write a program using Arduino / Raspberry 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

9.1 Hardware Requirements:

- Raspberry Pi B+/2/3
- HDMI Monitor or HDMI to VGA converter
- Digital Relay switch
- 12V Battery supply
- Jumper cables(Male to Male (10), Male to Female(10), Female to Female(10)
- Bread Board
- Ethernet Cable or Wi-Fi adaptor
- USB web-cam

9.2. Starting Raspberry Pi:

Connect power supply, USB Keyboard and Mouse and HDMI display to connectors as shown in figure 9.1. Insert NOOBS (New Out Of the Box Software) preinstalled Micro SD card in Micro SD card slot (Please Note: Micro SD slot is at backside of board). If you don't have NOOBS preinstalled Micro SD card or it is corrupted then follow steps in section 9.2.1 otherwise go to section 9.2.2 directly.

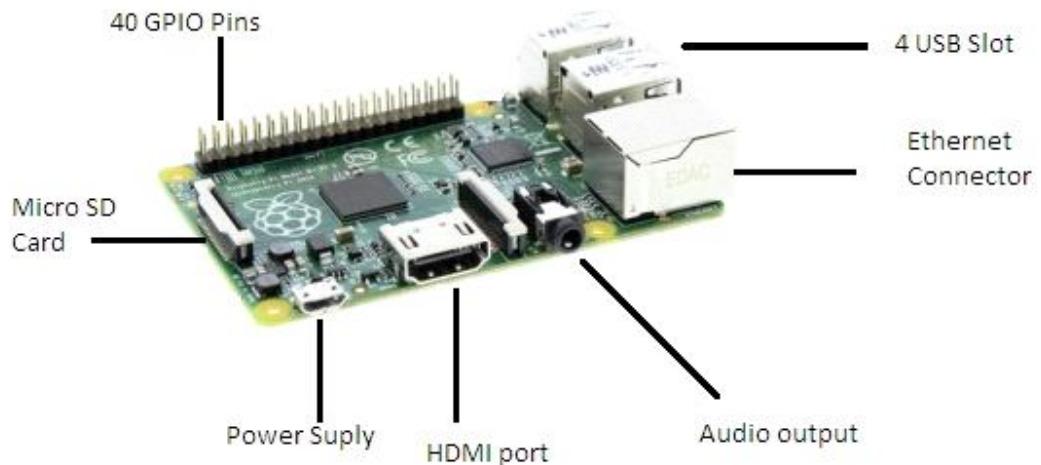


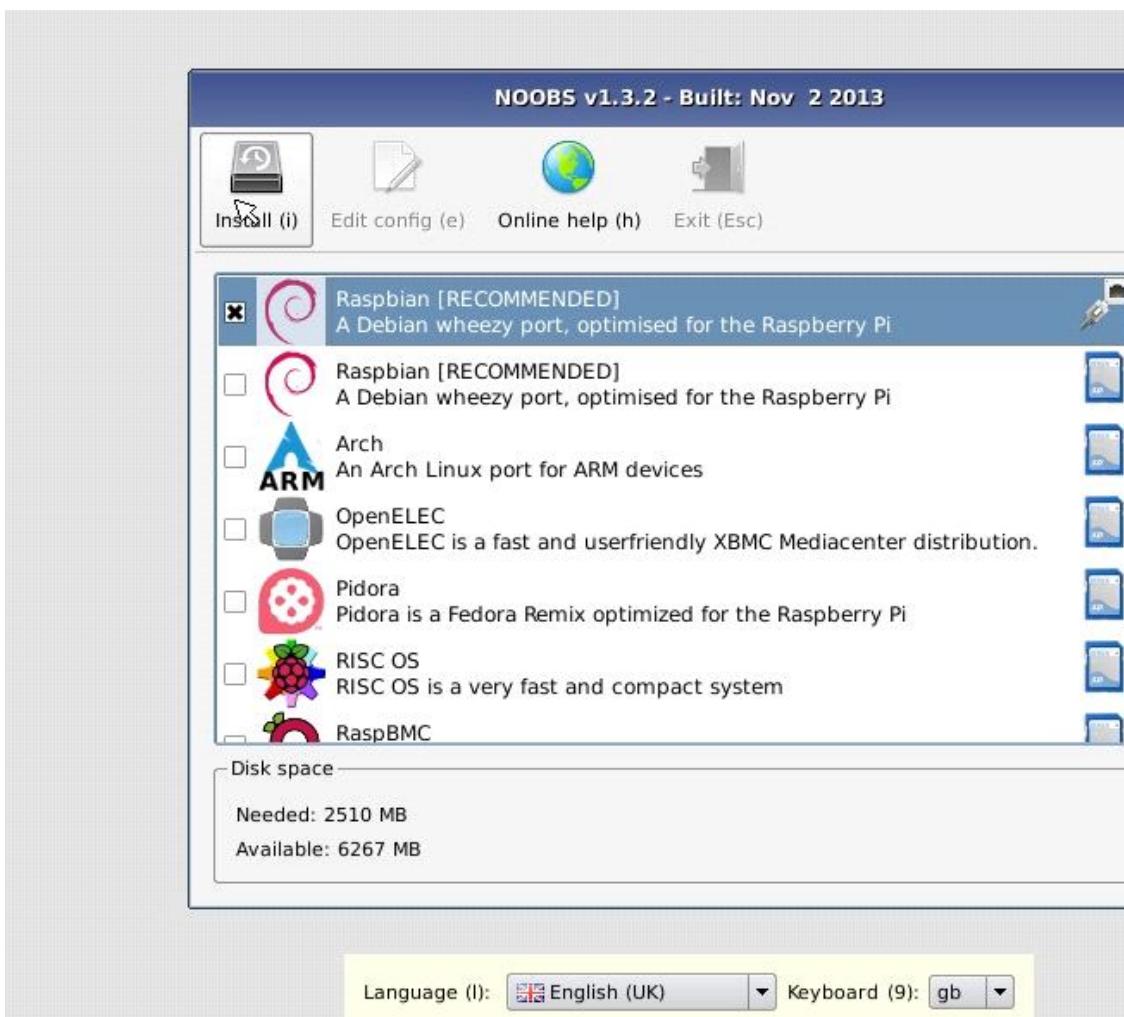
Figure 9.1

9.2.1 Loading NOOBS in SD Card

1. Using a computer with an SD card reader, visit the www.raspberrypi.org/downloads/ page.
2. Click on the Download ZIP button under 'NOOBS (offline and network install)', and select a folder to save it to.
3. Extract the files from the zip.
4. Format your SD card.
5. Drag and drop NOOBS files.

9.2.2 Installing Operating Systems using NOOBS

Now turn on power supply, Raspberry PI will start with NOOBS to install operating system. NOOBS screen is shown in figure 9.2.



Select Raspbian and click on Install.

Please note that default login for Raspbian is username **pi** with the password **raspberry**. Command to start graphical user interface is **startx**

9.3. Installing Prerequisite Packages on Raspberry Pi

9.3.1 Update system repositories using following command

```
sudo apt-get update
```

9.3.2 Installing python development kit using following command

```
sudo apt-get install build-essential python-dev python-openssl
```

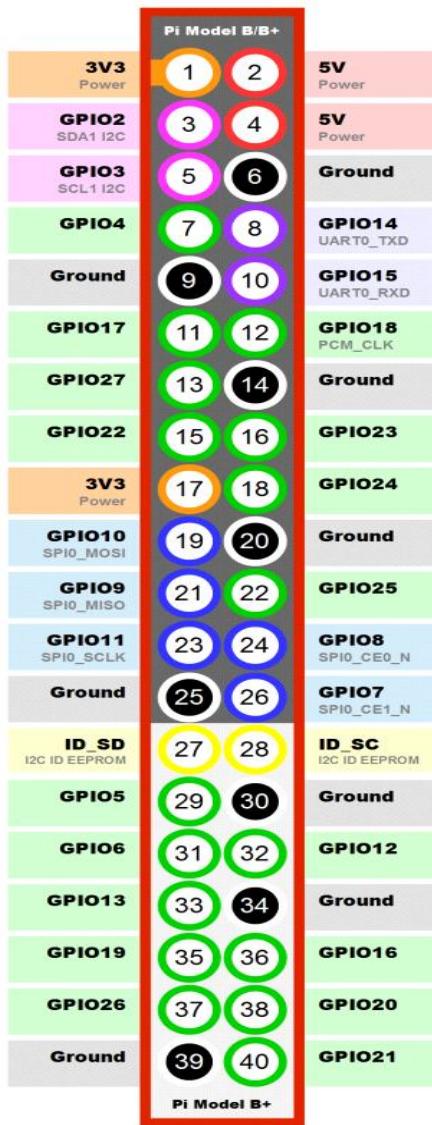
9.3.3 Install Pygame for Camera using following command

```
sudo apt-get install python-pygame
```

9.3.4 Install GPIO Libraries using following commands

```
wget http://pypi.python.org/packages/source/R/RPi.GPIO/RPi.GPIO-0.1.0.tar.gz
tar -xvf Rpi.GPIO-0.1.0.tar.gz
cd Rpi.GPIO-0.1.0
sudo ./setup.py install
```

9.4 Pin-out Diagram of Raspberry Pi B+/2/3



www.raspberrypi-spy.co.uk

