1

a) **Assign IP addresses in the network, first (i) using FLSM (fixed length subnet mask) and then (ii) using VLSM (variable length subnet mask). In each of these two cases and for each subnet, show how many IPs required, what IP you assign to which host/interface in the network, the network id, broadcast IP, first and last host IPs, and possible number of hosts. Explain how you came up with those numbers. In your implementation, use a contiguous block of private IP addresses of the form 192.168.x.x. Based on the resulting IP assignments with FLSM and VLSM, can you explain which addressing method is the best one? If you were to use public IPs, what IP block size you need to buy in those two cases?**

FLSM (fixed length subnet mask)

We need to have 6 subnets:

1. Department A
2. Department B
3. Department C
4. Between routers R1 and R2
5. Between routers R2 and R3
6. Between routers R1 and R3

So biggest amount of hosts is 260 + 2 (for network ID and broadcast IP) + 1 (for router interface) = 263 = $2^{9.}$ So, we need 9 bits required to support 263 hosts and 3 more bits for network IDs 32 – 12 = 20 bits.

192.168.0.0/20

11111111.11111111.11110000.00000000 ⇒ 255.255.240.0

Network IDs will be:

1. NetA (Department A): 192.168.0.0/23

   11111111.11111111.11111110.00000000

   255.255.254.0

   From all possible addresses 256 – 254 = 2. So, size of subnet will be from 192.168.0.0 to 192.168.1.255. And every other subnet will be of same size because it is fixed length subnet mask.
2. NetB (Department B): 192.168.2.0/23
3. NetC (Department C): 192.168.4.0/23
4. NetD (Between routers R1 and R2): 192.168.6.0/23
5. NetE (Between routers R2 and R3): 192.168.8.0/23
6. NetF (Between routers R1 and R3): 192.168.10.0/23

A broadcast address is an address that we use when we want to send information to all hosts that are in same LAN. And it is the last IP address in the subnet.

Broadcast IP address:

1. NetA: 2 - 1 = 1 ⇒ 192.168.1.255
2. NetB: 4 – 1 = 3 ⇒ 192.168.3.255
3. NetC: 6 – 1 = 5 ⇒ 192.168.5.255
4. NetD: 8 – 1 = 7 ⇒ 192.168.7.255
5. NetE: 10 – 1 = 9 ⇒ 192.168.9.255
6. NetF: 12 – 1 = 11 ⇒ 192.168.11.255

Network ID	**First host IP**	**Last host IP**	**Possible number of hosts**
1. 192.168.0.0	192.168.0.1	192.168.1.254	256*2 – 2 = 510
2. 192.168.2.0	192.168.2.1	192.168.3.254	256*2 – 2 = 510
3. 192.168.4.0	192.168.4.1	192.168.5.254	256*2 – 2 = 510
4. 192.168.6.0	192.168.6.1	192.168.7.254	256*2 – 2 = 510
5. 192.168.8.0	192.168.8.1	192.168.9.254	256*2 – 2 = 510
6. 192.168.10.0	192.168.10.1	192.168.11.254	256*2 – 2 = 510

In each subnet second IP address will be assigned to router interface in that subnet, first and last IP addresses are reserved for network ID and broadcast IP addresses.

NetA 192.168.0.0	
Router interface	192.168.0.1
A1 interface	192.168.0.2
A260 interface	192.168.1.5

NetB 192.168.2.0	
Router interface	192.168.2.1
B1 interface	192.168.2.2
B58	192.168.2.59

NetC 192.168.4.0	
Router interface	192.168.4.1
C1 interface	192.168.4.2
C26	192.168.4.27

NetD 192.168.6.0	
R1	192.168.6.1
R2	192.168.6.2

NetE 192.168.8.0	
R2	192.168.8.1
R3	192.168.8.2

NetF 192.168.10.0	
R1	192.168.10.1
R3	192.168.10.2

So, in total we have 3072 possible IP addresses, some of them we can't use, because they are reserved for network IDs and broadcast IP addresses. In our situation this is amount of wasted addresses:

1. Network 192.168.0.0: 249 wasted
2. Network 192.168.32.0: 451 wasted
3. Network 192.168.64.0: 483 wasted
4. Network 192.168.96.0: 508 wasted
5. Network 192.168.128.0: 508 wasted
6. Network 192.168.160.0: 508 wasted

So, in total 2707 IP addresses are wasted.

VLSM (variable length subnet mask)

In VLSM subnetting all the subnets may have different subnet mask, depending on how many IP addresses they need. So first I have to check how many IPs I need for every subnet and which subnet mask it will have:

netA - department A [260 + 3 = 263 IPs]: 2^9 requires a/23 255.255.254.0 mask to support 260 hosts. 256- 254 = 2, so for first netA I need address range from 192.168.0.0 to 192.168.1.255

netB - department B [58 + 3 = 61 IPs]: 2^6 requires a/26 255.255.255.192 mask to support 58 hosts. 256 – 192 = 64, so for netB I need address range from 192.168.2.0 to 192.168.2.63

netC - department C [26 + 3 = 29 IPs]: 2^5 requires a/27 255.255.255.224 mask to support 26 hosts. 256 – 224 = 32, so for netC I need address range from 192.168.2.64 to 192.168.2.95. I start from 64 because network part is 27 bits, that is why in the third octet it will stay 2 and I have address range for hosts in the fourth octet only.

netD - R1 ⇔ R2 [2 + 2 = 4 IPs]: 2^2 requires a/30 255.255.255.252 to support 2 hosts. 256 – 252 = 4, so for netD I need address range from 192.168.2.96 to 192.168.2.99.

netE - R2 ⇔ R3 [2 + 2 = 4 IPs]: 2^2 requires a/30 255.255.255.252 to support 2 hosts. 256 – 252 = 4, so for netE I need address range from 192.168.2.100 to 192.168.2.103.

netF - R1 ⇔ R3 [2 + 2 = 4 IPs]: 2^2 requires a/30 255.255.255.252 to support 2 hosts. 256 – 252 = 4, so for netF I need address range from 192.168.2.104 to 192.168.2.107.

Broadcast IP:

1. netA: 192.168.1.255
2. netB:192.168.2.63

3. netC:192.168.2.95
4. netD:192.168.2.99
5. netE:192.168.2.103
6. netF: 192.168.2.107

Network ID	**First host IP**	**Last host IP**	**Possible number of hosts**
192.168.0.0	192.168.0.1	192.168.1.254	510
192.168.2.0	192.168.2.1	192.168.2.62	62
192.168.2.64	192.168.2.65	192.168.2.94	30
192.168.2.96	192.168.2.97	192.168.2.98	2
192.168.2.100	192.168.2.101	192.168.2.102	2
192.168.2.104	192.168.2.105	192.168.2.106	2

netA 192.168.0.0	
Router interface	192.168.0.1
A1	192.168.0.2
A260	192.168.1.5

netB 192.168.2.0	
Router interface	192.168.2.1
B1	192.168.2.2
B58	192.168.2.59

netC 192.168.2.64	
Router interface	192.168.2.65
C1	192.168.2.66
C26	192.168.2.91

netD 192.168.2.96	
R1	192.168.6.97
R2	192.168.6.98

netE 192.168.2.100	
R2	192.168.2.101
R3	192.168.2.102

netF 192.168.2.104	
R1	192.168.2.106
R3	192.168.2.105

In total in VLSM implementation I have 768 possible IP addresses, so 768 – 365 = 403 are wasted.

So, for our network it is better to use VLSM subnetting since we need less hosts and less amount of hosts is wasted.

If I need to buy blocks of public IPs, then I need first to calculate how many IP addresses I need for my network and then to find closest binary number.
2*256*6 = 3072, closest binary representation is 2^{12} = 4096
So, for FLSM I need to buy 12 blocks of public IP addresses.
For VLSM closest binary representation 2^{10} = 1074, so I need to buy 10 blocks of public IP addresses.

b) What is longest prefix matching in the context of routing in a network? In the case of VLSM above in a.(ii), provide the routing/forwarding tables for each of the three routers based on longest prefix matching so that each host can communicate with all other hosts and connect to the Internet. Explain how the tables are created. How the packet is forwarded when the computer A1 sends a packet to the computer C26.

"**Longest prefix match** (also called **Maximum prefix length match**) refers to an algorithm used by routers in Internet Protocol (IP) networking to select an entry from a forwarding table."[1]

Forwarding table for R1

[1] https://en.wikipedia.org/wiki/Longest_prefix_match

Router R1 has 2 links, to R2 and R3, and via R3 it has access to internet. Here I need to convert network ID to binary number and extract subnet mask and this number will be longest prefix. If longest prefix doesn't match any of longest prefixes then router will use link 1, via R3 to access internet.

Destination address range	**Link number**
[192.168.0.0/23] 11000000 10101000 0000000**0** 00000000 Through [192.168.1.255] 11000000 10101000 00000001 11111111	Local
[192.168.2.0/26] 11000000 10101000 00000010 **00**000000 Through [192.168.2.63] 11000000 10101000 00000010 00111111	0 192.168.2.98
[192.168.2.64/27] 11000000 10101000 00000010 **01**000000 Through [192.168.2.94] 11000000 10101000 00000010 10111100	1 192.168.2.105
Otherwise 0.0.0.0	1

Destination address range	**Link number**

11000000 10101000 0000000/23	Local
11000000 10101000 00000010 00/26	0 192.168.2.98
11000000 10101000 00000010 010/27	1 192.168.2.105
Otherwise 0.0.0.0	1 192.168.2.105

Forwarding table for R2

Router R2 has 2 links, to R1 and R3, and via R3 it has access to internet. To create routing table, I need first convert network's ID to binary number and then from this number I extract subnet mask and this number will be longest prefix.

Destination address range	**Link number**
[192.168.2.0/26] 11000000 10101000 00000010 00000000 Through [192.168.2.63] 11000000 10101000 00000010 00111111	Local
[192.168.0.0/23] 11000000 10101000 00000000 00000000 Through [192.168.1.255] 11000000 10101000 00000001.11111111	0 192.168.2.97
[192.168.2.64/27] 11000000 10101000 00000010 010000000 Through [192.168.2.94]	1 192.168.2.102

11000000 10101000 00000010 10111100	
Otherwise 0.0.0.0	1

Destination address range	**Link number**
11000000 10101000 00000010 00/26	Local
11000000 10101000 0000000/23	0 192.168.2.97
11000000 10101000 00000010 010/27	1 192.168.2.102
Otherwise 0.0.0.0	1 192.168.2.102

Forwarding table for R3

Router R3 has 3 links, to R2, R3 and the one that connects it to internet. Here again I need to convert network ID to binary number and extract subnet mask and this number will be longest prefix. If longest prefix doesn't match any of longest prefixes, then router will use link 2 for access to internet.

Destination address range	**Link number**
[192.168.2.64/27]	

11000000 10101000 00000010 010000000 Through [192.168.2.94] 11000000 10101000 00000010 10111100	
[192.168.0.0/23] 11000000 10101000 0000000 00000000 Through [192.168.1.255] 11000000 10101000 00000001 11111111	0 192.168.2.106
[192.168.2.0/26] 11000000 10101000 00000010 00000000 Through [192.168.2.63] 11000000 10101000 00000010 00111111	1 192.168.2.101
Otherwise	2 158.109.95.150

Destination address range	**Link number**
11000000 10101000 00000010 010/27	Local
11000000 10101000 0000000/23	0 192.168.2.106
11000000 10101000 00000010 00/26	1 192.168.2.101
Otherwise	2 158.109.95.150

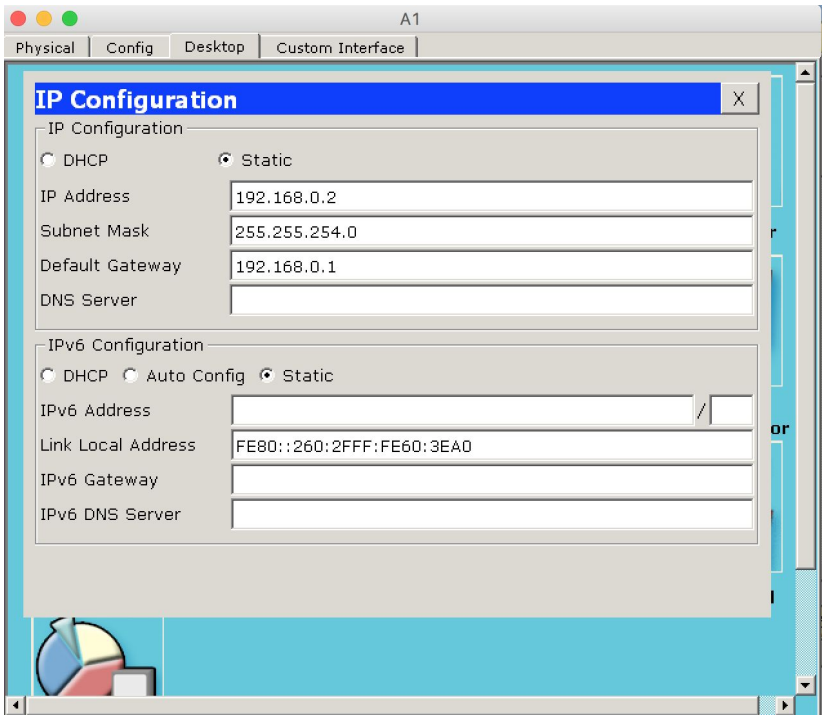
If A1 sends a packet to C26 first it will check if subnet mask of C26 is same as its, it will see that C26 has a different subnet mask and send a packet to

the default gateway, which is 192.168.0.1 – R1. Then R1 will convert IP address of C26 to binary number:
192.168.2.91 ⇒ 11000000 10101000 00000010 01011011 and check the longest prefix match from its routing table, it will find out that the longest prefix match is 11000000 10101000 00000010 01 and will send a packet via link 1 to R3. R3 will receive a packet, check subnet mask and since C26 has same subnet mask as R3 will forward directly a packet to C26.

Using the IP addresses and routing tables from your answers above in 1.) with VLSM, implement the network in Cisco packet tracer v6.2 using two different routing methods:

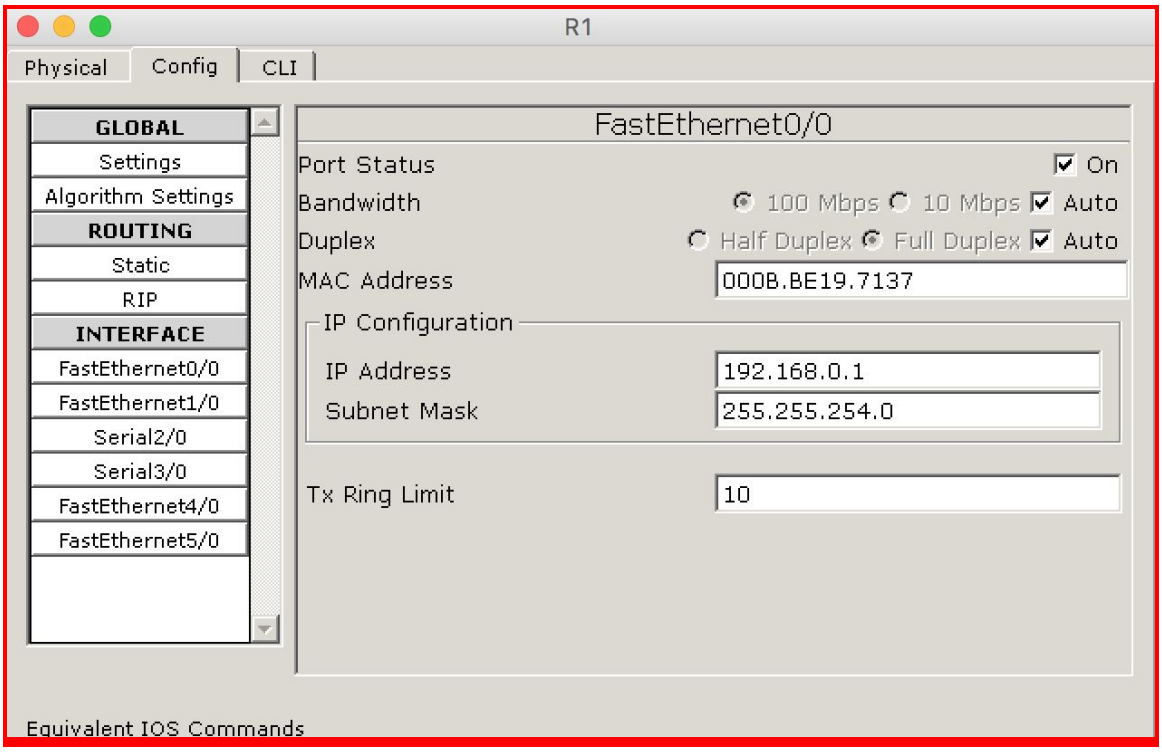
a. static routing

Static routing is a form of routing that occurs when a router uses a manually-configured routing entry, rather than information from a dynamic routing traffic.[2] So in order to configure static routing I need to configure manually hosts/PCs and routers. First in cisco packet tracer I need to put all elements of my network (like PCs, routers, switches). After I made this I started to configure hosts/PCs:



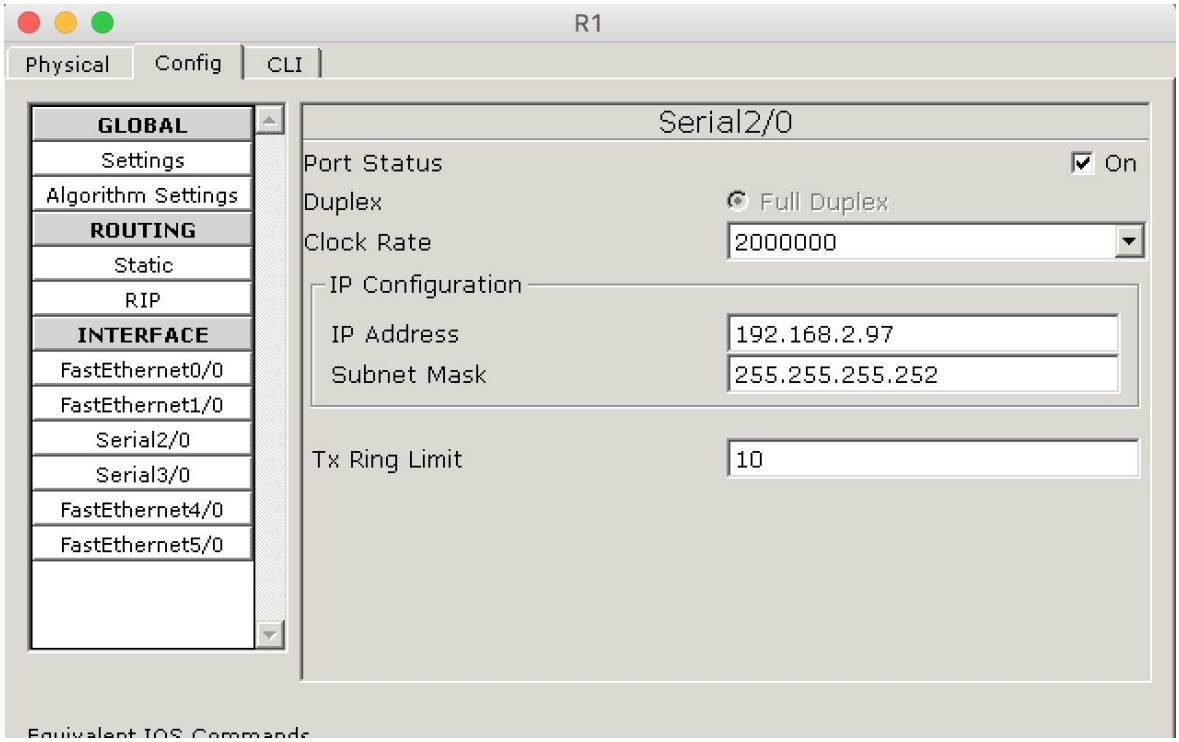
[2] TCP/IP Tutorial and technical Overview (IBM RedBooks Series)

On the picture above, you can see how I configured A1 host. I put IP address of this host, then subnet mask and default gateway which in this situation is IP address of router R1. I had to do same for all the hosts in department A, B and C, of course I have only the first and the last hosts in each department. After that when hosts in departments are configured I moved to routers.

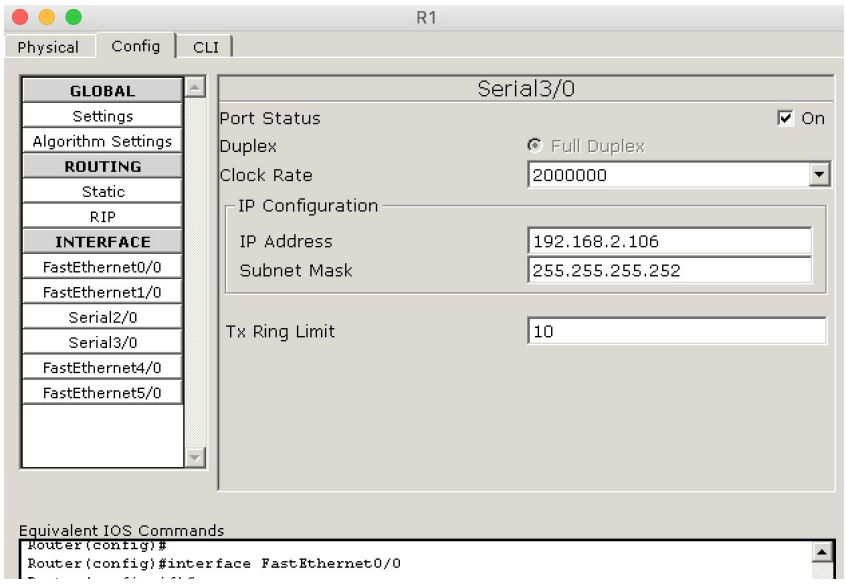


Here I configured FastEthernet0/0 (“Ethernet/Fast Ethernet/Gigabit Ethernet: Standard network interfaces used to connect different network segments.”[3]).

[3] http://www.dummies.com/programming/networking/cisco/standard-router-ports/

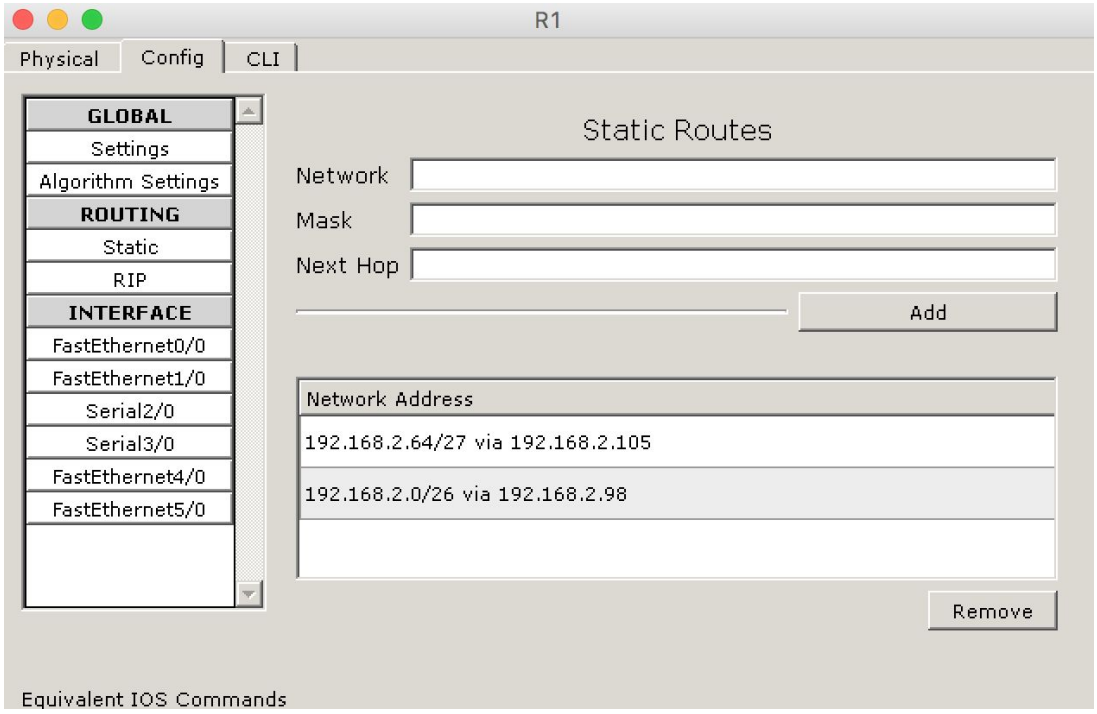


Above you can see how I configured Serial2/0 (serial interface is used to connect two different routers, in my situation it is R1 and R2). I put IP address and subnet mask and made port status “on”.



Above is configuration of Serial3/0 (serial interface between routers R1 and R3).

After that in config tab I went to ROUTING ⇒ Static and configured Static Routes



After I configured routers R2 and R3 in the same way but with different data (IP addresses, network addresses in static routing).

Routing tables:

R1

Routing Table for R1

Type	Network	Port	Next Hop IP	Metric
C	192.168.0.0/23	FastEthernet0/0	---	0/0
S	192.168.2.0/26	---	192.168.2.98	1/0
S	192.168.2.64/27	---	192.168.2.105	1/0
C	192.168.2.96/30	Serial2/0	---	0/0
C	192.168.2.104/30	Serial3/0	---	0/0

R2

Routing Table for R2

Type	Network	Port	Next Hop IP	Metric
S	192.168.0.0/23	---	192.168.2.97	1/0
C	192.168.2.0/26	FastEthernet0/0	---	0/0
S	192.168.2.64/27	---	192.168.2.102	1/0
C	192.168.2.96/30	Serial2/0	---	0/0
C	192.168.2.100/30	Serial3/0	---	0/0

R3

Routing Table for R3

Type	Network	Port	Next Hop IP	Metric
S	192.168.0.0/23	---	192.168.2.106	1/0
S	192.168.2.0/26	---	192.168.2.101	1/0
C	192.168.2.64/27	FastEthernet0/0	---	0/0
C	192.168.2.100/30	Serial3/0	---	0/0
C	192.168.2.104/30	Serial2/0	---	0/0

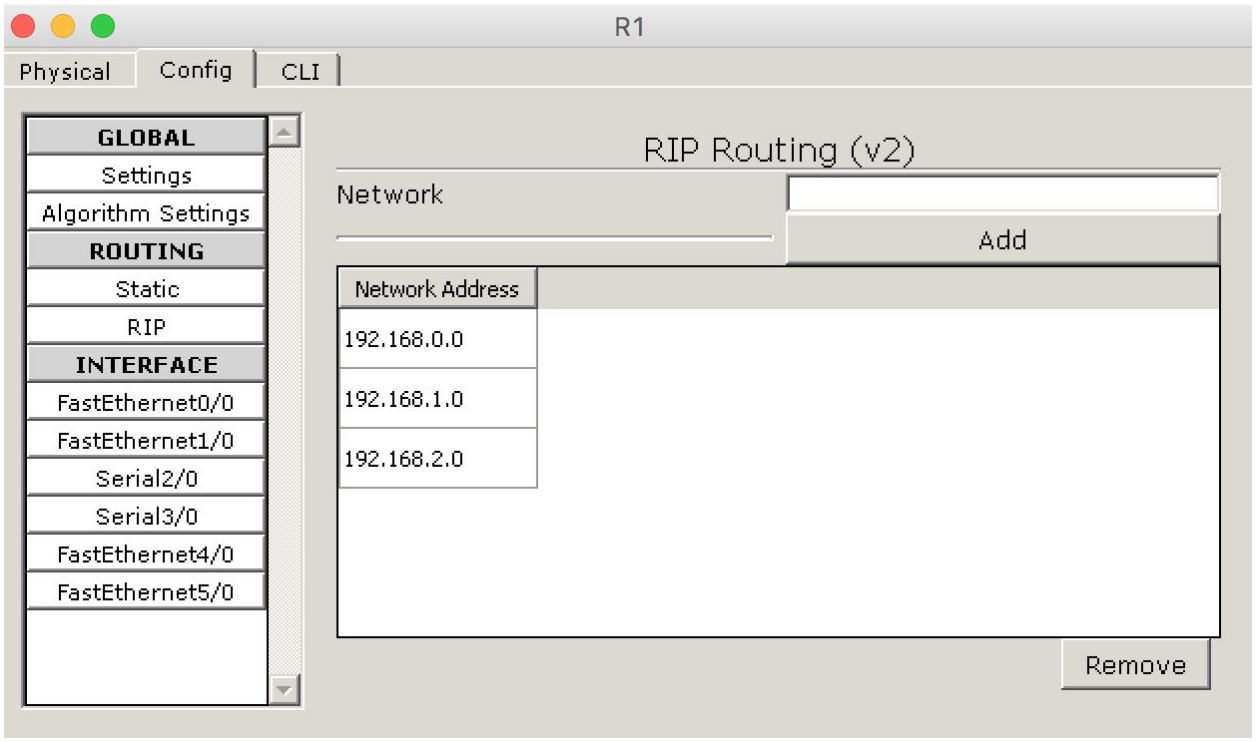
b. dynamic routing using RIP

"**Dynamic routing** is a networking technique that provides optimal data routing. Unlike static routing, dynamic routing enables routers to select paths according to real-time logical network layout changes. In dynamic routing, the routing protocol operating on the router is responsible for the creation, maintenance and updating of the dynamic routing table. In static routing, all these jobs are manually done by the system administrator. Dynamic routing uses multiple algorithms and protocols. The most popular are Routing Information Protocol (RIP) and Open Shortest Path First (OSPF)."[4]

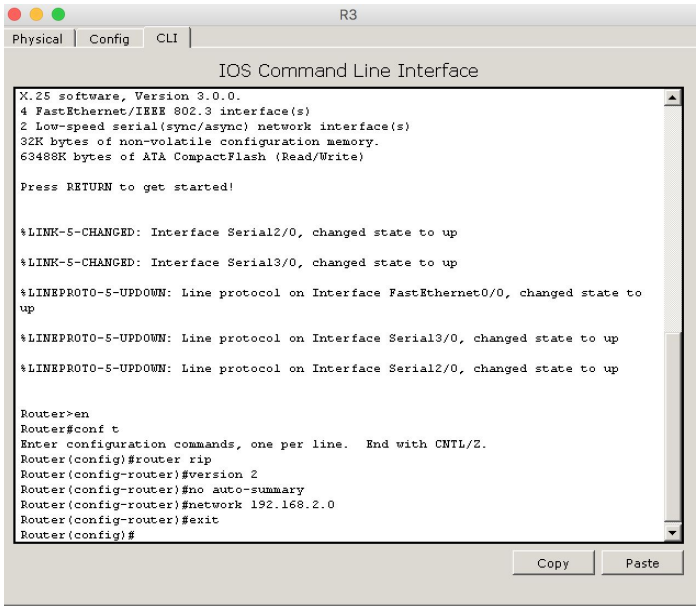
So basically, as I understood the difference is that I don't need to fill in routing tables manually, I just provide addresses of networks and RIP makes routing tables automatically for me, which is much more convenient, I spend less time on configuration and avoid human mistakes also, like mistyping something and things like that and it is easier to maintain.

In dynamic routing using RIP I again set all IP addresses and subnet masks for hosts/PCs and routers, this part is same, it is different when I come to routing tables configuration. It is possible to do it in two different ways. First: you can configure it using GUI interface, that looks like

[4] https://www.techopedia.com/definition/19047/dynamic-routing



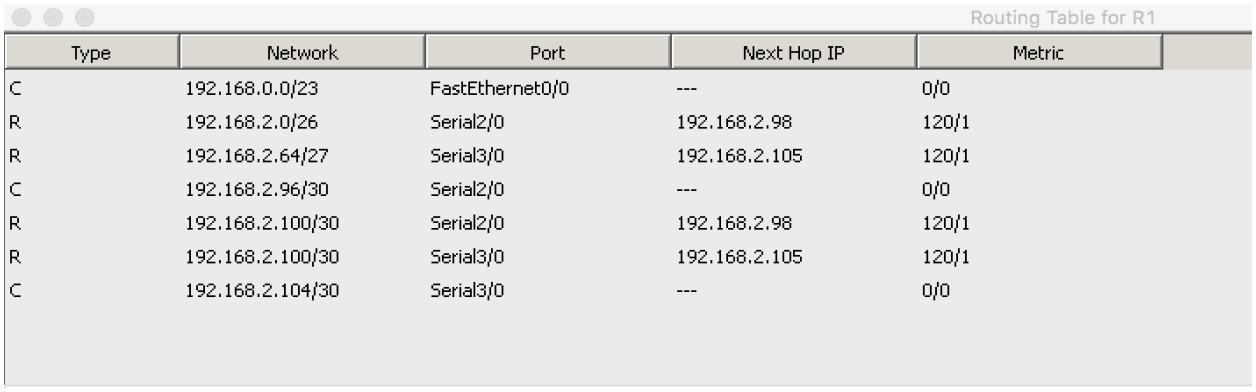
In picture above, you can see settings for router R1. Those three subnet addresses are subnets that router R1 knows. Or I can do this in CLI (command line interface), that will look like



In the picture above, I configured router R3. After I did same for R2 router.

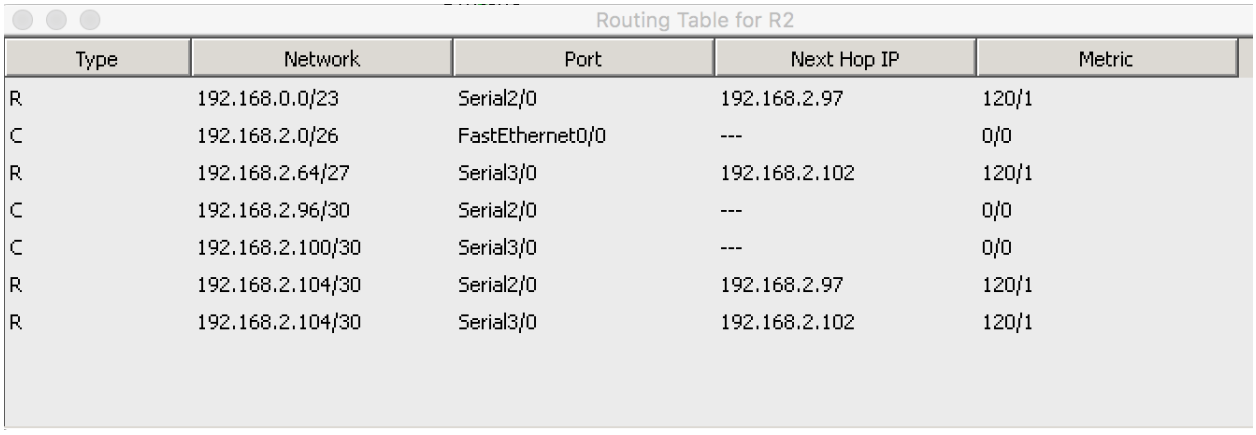
Routing tables:

R1



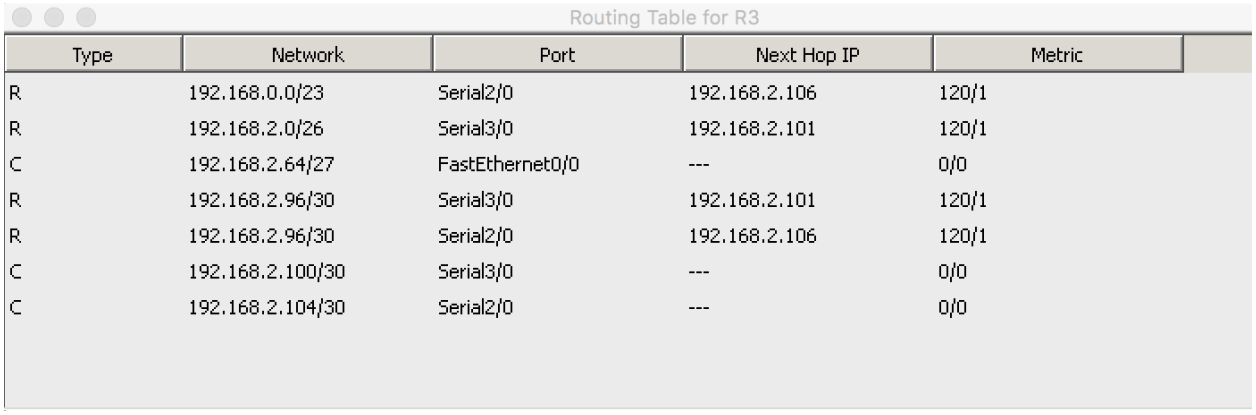
Type	Network	Port	Next Hop IP	Metric
C	192.168.0.0/23	FastEthernet0/0	---	0/0
R	192.168.2.0/26	Serial2/0	192.168.2.98	120/1
R	192.168.2.64/27	Serial3/0	192.168.2.105	120/1
C	192.168.2.96/30	Serial2/0	---	0/0
R	192.168.2.100/30	Serial2/0	192.168.2.98	120/1
R	192.168.2.100/30	Serial3/0	192.168.2.105	120/1
C	192.168.2.104/30	Serial3/0	---	0/0

R2



Type	Network	Port	Next Hop IP	Metric
R	192.168.0.0/23	Serial2/0	192.168.2.97	120/1
C	192.168.2.0/26	FastEthernet0/0	---	0/0
R	192.168.2.64/27	Serial3/0	192.168.2.102	120/1
C	192.168.2.96/30	Serial2/0	---	0/0
C	192.168.2.100/30	Serial3/0	---	0/0
R	192.168.2.104/30	Serial2/0	192.168.2.97	120/1
R	192.168.2.104/30	Serial3/0	192.168.2.102	120/1

R3



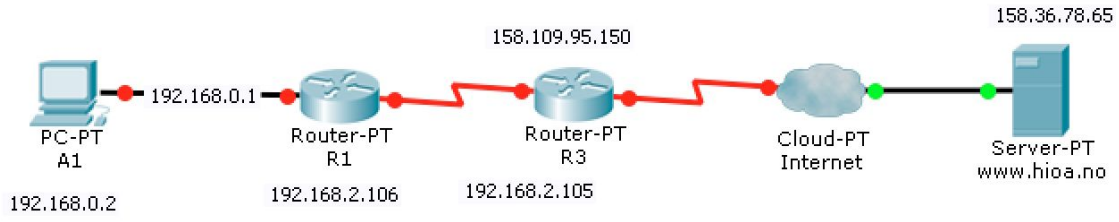
Type	Network	Port	Next Hop IP	Metric
R	192.168.0.0/23	Serial2/0	192.168.2.106	120/1
R	192.168.2.0/26	Serial3/0	192.168.2.101	120/1
C	192.168.2.64/27	FastEthernet0/0	---	0/0
R	192.168.2.96/30	Serial3/0	192.168.2.101	120/1
R	192.168.2.96/30	Serial2/0	192.168.2.106	120/1
C	192.168.2.100/30	Serial3/0	---	0/0
C	192.168.2.104/30	Serial2/0	---	0/0

3

Internet access is provided by subscribing a dedicated Internet connection from a local internet service provider (ISP) through the router R3, which supports NAT. Explain with diagram(s), NAT table, and IP addresses you assigned, how NAT works when the host computer A1 browse the HiOA website (www.hioa.no). Suppose that the R3 router interface received a dynamic IP 158.109.95.150 from the ISP. You can assume appropriate port numbers in your explanation.

Network address translation (NAT) is a method of remapping one IP address space into another by modifying network address information in IP header of packets while they are in transit across a traffic routing device.[5]

1. First A1 host sends a request to www.hioa.no via router R3 with help of R1.

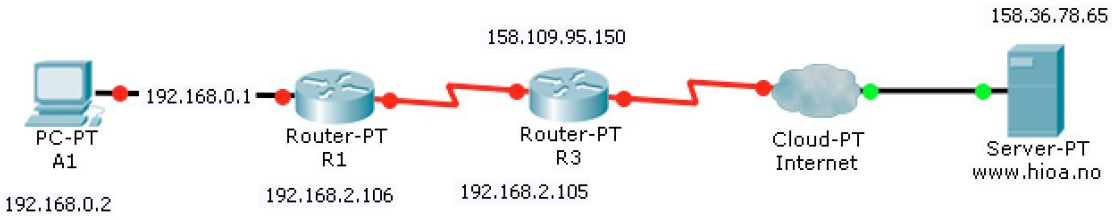



192.168.0.2: 55333 to 158.36.78.65: 80

[5] https://en.wikipedia.org/wiki/Network_address_translation

2. Then in router R3, router changes source address and port number of A1 to its dynamic IP from ISP and another port number.

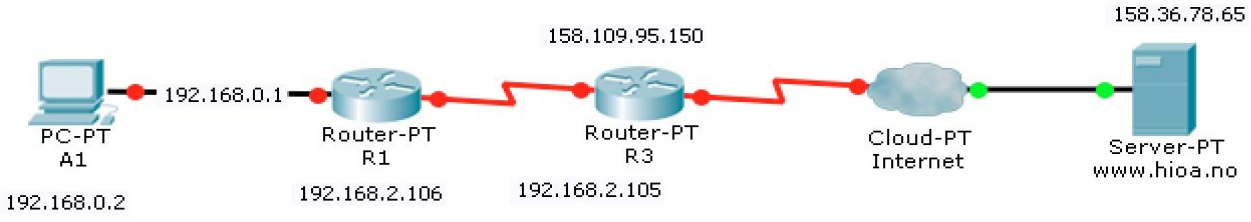
158.109.95.150: 66666 to 158.36.78.65: 80



3. And it writes down following information to NAT forwarding table:

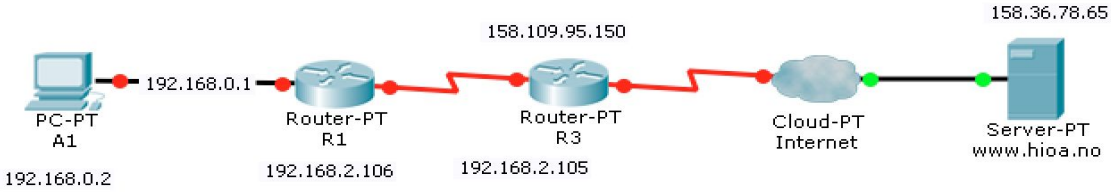
Private side	Public side
192.168.0.2: 55333	158.109.95.150: 66666

4. After that R3 sends this request with changed information to server. Server receives request and sends respond.

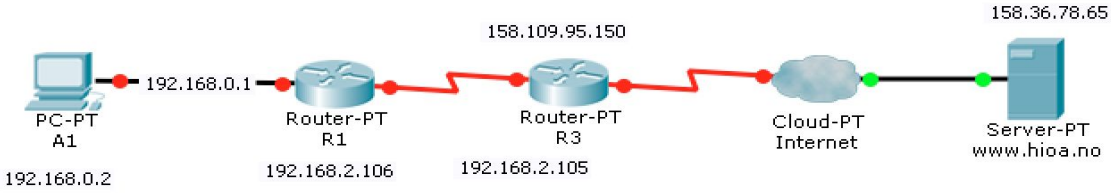


5. Server sends respond to R3.

158.36.78.65: 80 to 158.109.95.150: 66666

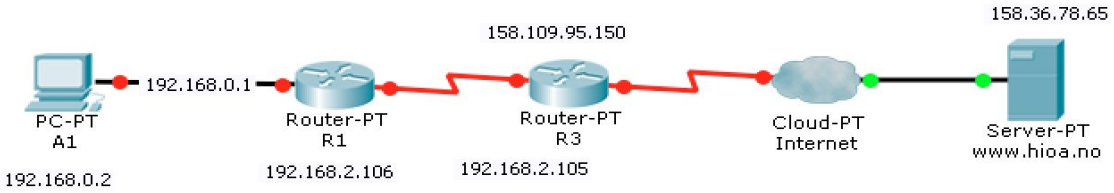


6. Then R3 retrieves information from NAT forwarding table and sends respond to A1



158.36.78.65: 80 to 192.168.0.2: 55333

7. A1 receives respond.



4

In your Cisco packet trace implementation Oblig1-PT-static.pkt above in 2.), when a data packet from host A1 is sent to host B1, enumerate the steps taken for resolving the addresses in data-link layer when

a) **all ARP tables are initially empty.**

First A1 creates a packet where as a source address it writes its IP address 192.168.0.2 and as a destination address it writes IP address of B1 192.168.2.2. Because B1 has a different subnet mask, A1 will send a packet to default gateway 192.168.0.1, in frame as a source MAC address it writes its MAC address 0060.2F60.3EA0, but it doesn't know MAC address of R1, so it broadcasts ARP request to the whole LAN via switch, where it asks host with IP address 192.168.0.1 to send its MAC address. R1 receives this request and sends response with its MAC address. A1 writes down this information to its ARP table, then puts mac address of R1 000B.BE19.7137 as a destination MAC address in frame and sends a packet. Then R1 receives a packet, deletes previous source and destination MAC addresses, checks destination IP address, then checks it's routing table to understand where to send it further. Since only Ethernet interface has a MAC address and I am using Serial interface between R1 and R2, R1 does not ask R2 about its MAC address and just uses routing table to forward a packet to R2. R2 receives the packet, deletes previous source and destination MAC addresses, checks destination IP address, it is same LAN, so it sends

ARP request to all other hosts in this LAN to find out MAC address of 192.168.2.2, B1 receives request, sends response with its MAC address. R2 saves this information to its ARP table, then writes source MAC address 0003.E457.5ACA and destination MAC address of B1 0005.5E7D.67DD and sends a packet. After that B1 receives packet and saves R2 MAC address to ARP table.

b) ARP table is up-to-date.

First A1 creates a packet where as a source IP address it writes its IP address 192.168.0.2 and as a destination IP address it writes IP address of B1 192.168.2.2. Because B1 has a different subnet mask, A1 will send a packet to default gateway 192.168.0.1, in frame as a source MAC address it writes MAC address of A1 0060.2F60.3EA0 and as a destination MAC address MAC address of R1 000B.BE19.7137, A1 does not need to send ARP request to get MAC address of R1, because ARP table is up-to-date, so it just checks MAC address of host 192.168.0.1. Then R1 receives the packet, deletes source and destination MAC addresses, checks destination IP address, checks in routing table where to send a packet, sends a packet (no ARP requests here also because of Serial interface between R1 and R2). After that R2 receives the packet, deletes source and destination MAC addresses, checks that it is in the same LAN with destination host, writes its MAC address as a source MAC address and MAC address of B1 0005.5E7D.67DD as a destination MAC address and sends it (no ARP requests here also, ARP table is up-to-date). B1 receives the packet.

c) If host A1 wants to send IP datagram to host A 260, will A1 ask router R1 to help forward the datagram? Why? In the Ethernet frame containing the IP datagram, what will be the source and destination IP and MAC addresses?

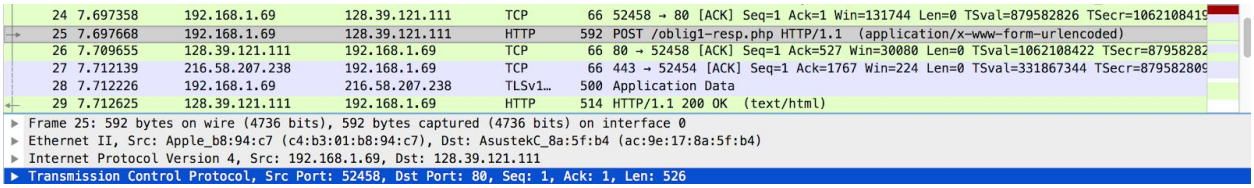
No, A1 will not ask router R1 to help to forward the datagram, because A1 will broadcast ARP request via switch to all other hosts in its LAN, and after that A260 will response with its MAC address, A1 will fill in source and destination MAC addresses and send a IP datagram.

Source MAC address	**Destination MAC address**	**Source IP address**	**Destination IP address**
0060.2F60.3EA0	0001.4335.73D1	192.168.0.2	192.168.1.5

Use Wireshark to capture packets transferred when you visit the web page http://128.39.121.111/oblig1.php from your computer, and then enter and submit 4-digit postal code of your address (a valid unique address is expected). Save the captured file as Oblig1-Wireshark.pcapng. Based on the captured packets [consider the last two HTTP packets only: POST/oblig1resp.php and its response], answer the following questions. You should provide supporting screenshots with places from where you got the answer(s) MARKED.

a. **What type of transport protocol is used? Explain with timing diagrams how TCP connection setup and termination works, using real values of SYN, ACK, Seq, Ack, FIN fields (captured by Wireshark) when the web page http://128.39.121.111/oblig1.php is browsed from your computer.**

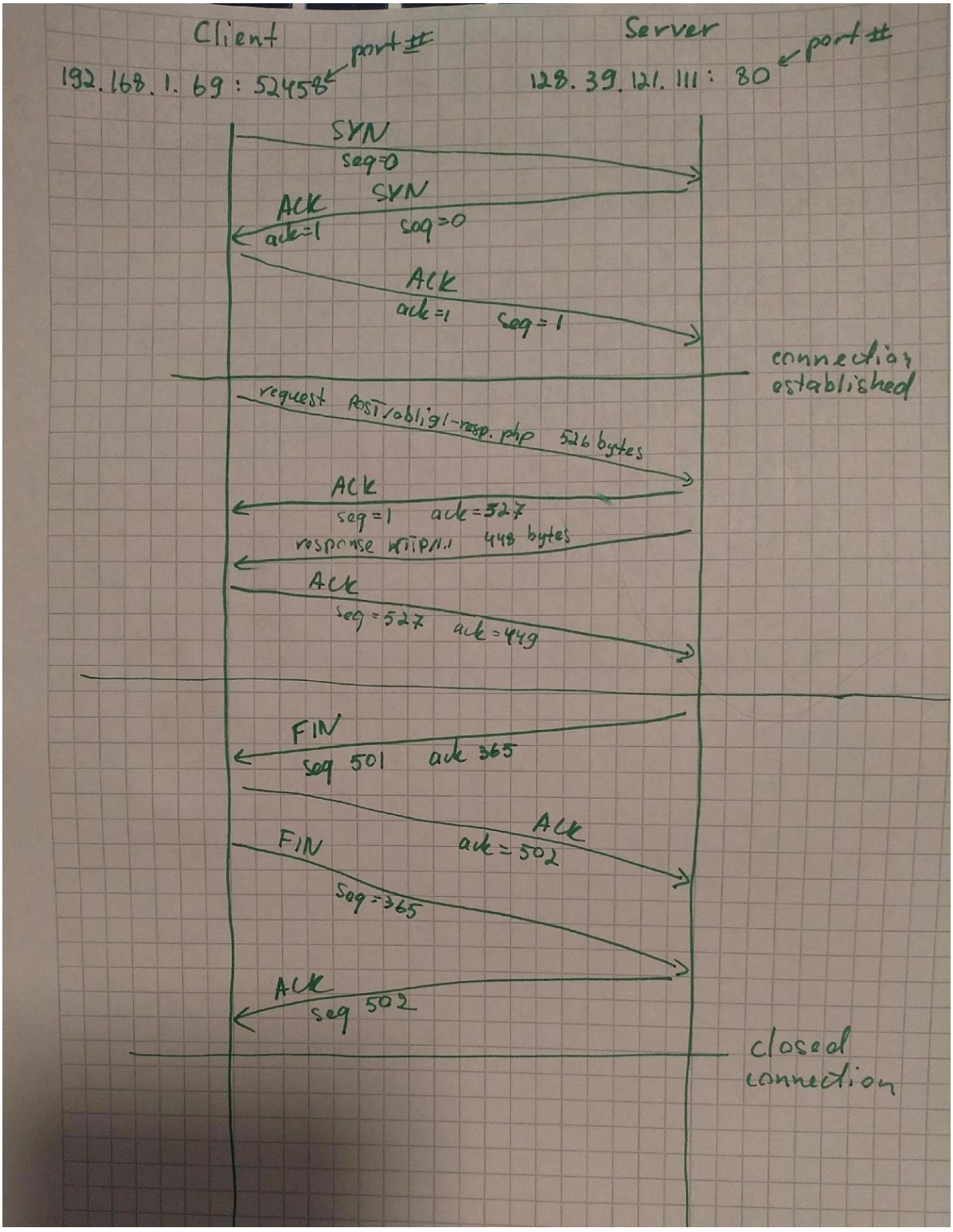
TCP protocol is used.



SYN (Synchronize) – initiates a connection, FIN (Final) – terminates a connection and ACK acknowledges received data.

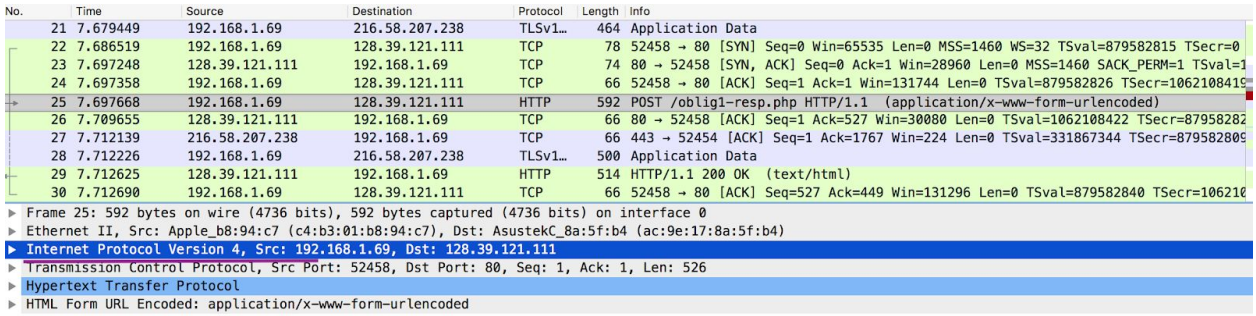
First to establish connection it will appear a three-way handshaking between server and client. Client sends a packet to a server with seq num = 0 and sends a SYNbit that equals 1 to initialize a connection. And then the server will respond by acknowledging that it received a packet, server

will set ACKnum to 1, the acknowledgement number is set to 1 to indicate the receipt of the client's SYN flag in packet #1. And also, server will set his own SYNbit to 1 and send its seq num which equals 0. After that a client will acknowledge that with an ACK, with ACKnum = 1 and it will set its seq num to 1. And at this point the connection is established. When connection is established the next packet that will be send will carry actual payload, in my situation HTTP request from client.
When a server decides that no further communication is needed it sets FIN bit in a packet to 1 and seq num is 501. After that client will acknowledge that he received a packet, increment ACKnum by 1 and then also add FINbit to his packet. Then server sends its final sequence number of 502 and acknowledges the client's FIN packet by incrementing the acknowledgement number by 1. At this moment connection is terminated.


Client
port #
192.168.1.69 : 52458
Server
port #
128.39.121.111 : 80
SYN
seq=0
ACK
ack=1
SYN
seq=0
ACK
ack=1
seq=1
connection established
request POST/obli/1-resp.php 526 bytes
ACK
seq=1 ack=527
response HTTP/1.1 448 bytes
ACK
seq=527 ack=449
FIN
seq 501 ack 365
ACK
ack=502
FIN
seq=365
ACK
seq 502
closed connection

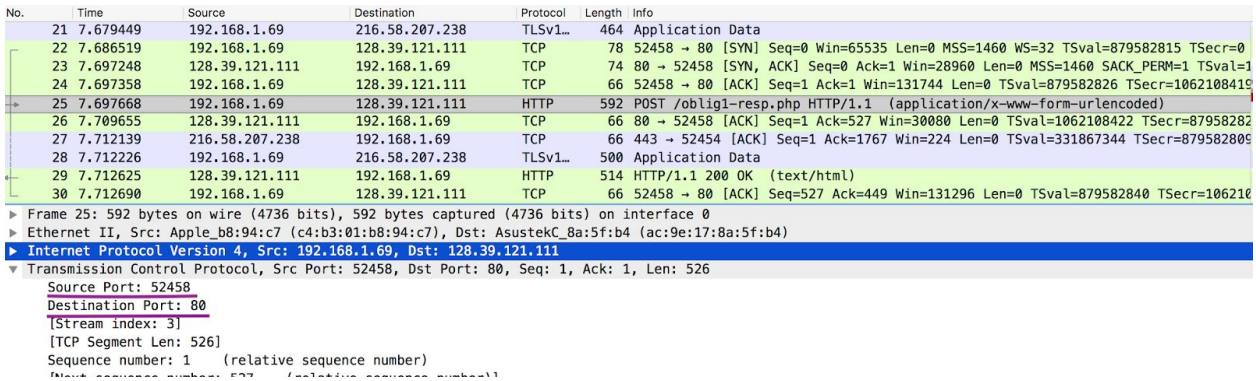
b. What IP addressing did it use, IPv4 or IPv6?

It used IPv4

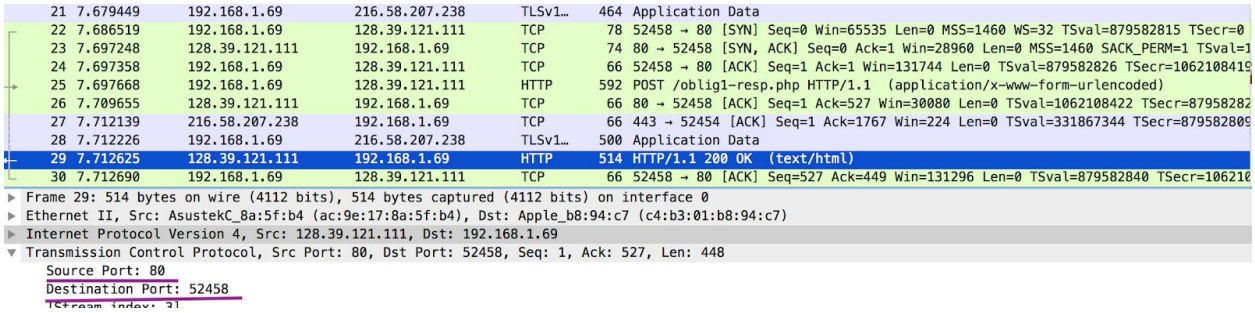


c. What is the source and destination port addresses used in the request and response packets?

Request packets source and destination ports are 52458 and 80

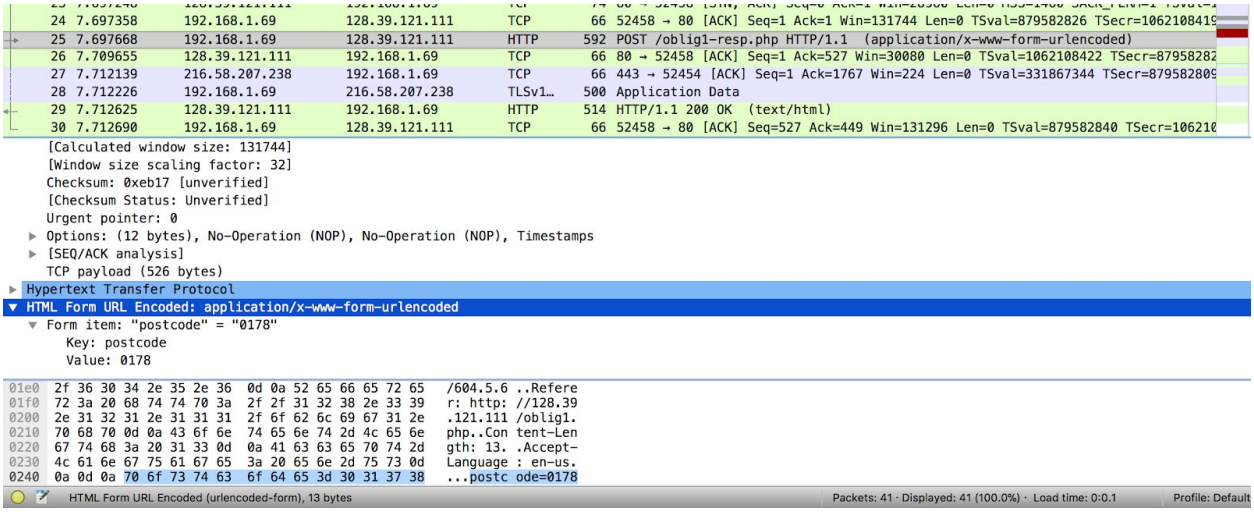


Response packets source and destination ports are 80 and 52458:

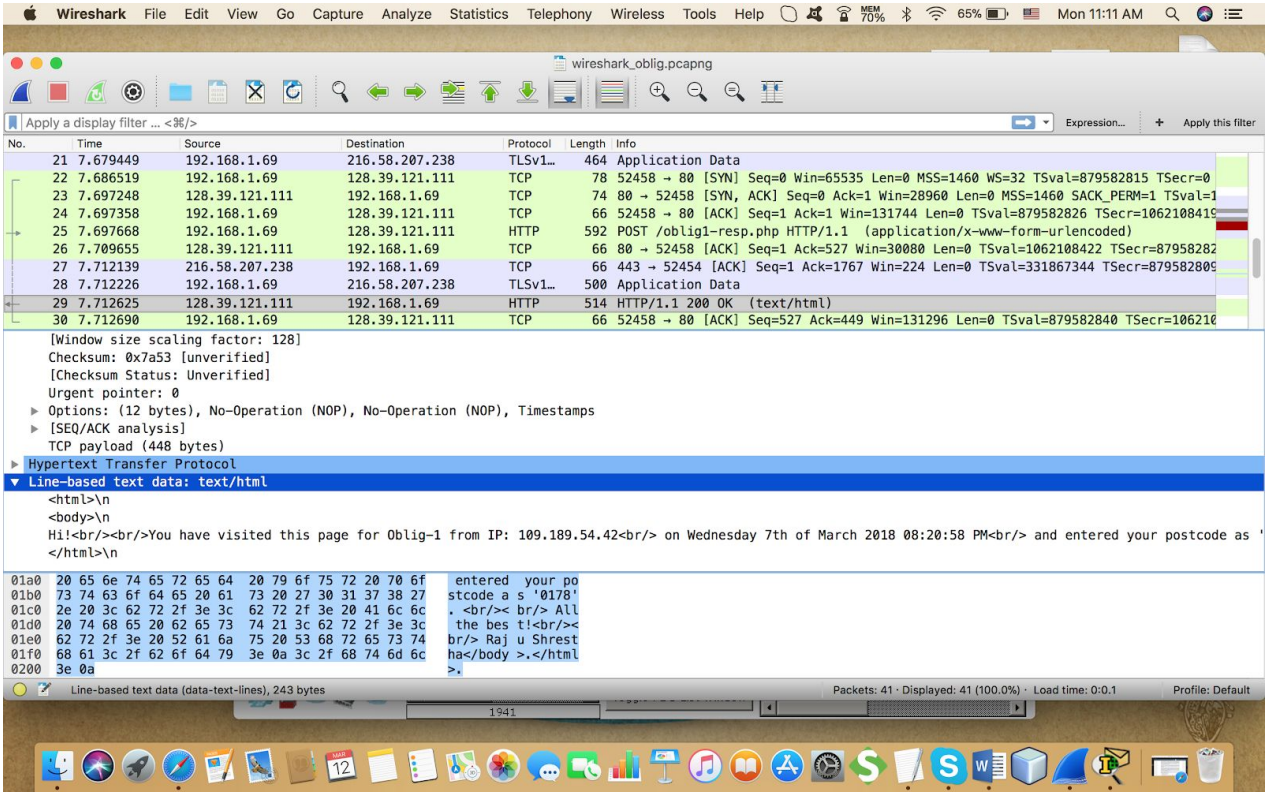


d. What is the size of the user data (payload) in the case of sending (HTTP request) and receiving (HTTP response) packets?

User data (payload) in the case of sending (HTTP request) packets is 13 bytes



User data (payload) in the case of receiving (HTTP response) packets is 243 bytes:



e. For the response packet, show the sizes in bytes when it reaches different layers (from link layer up to application layer and then to the browser) in your computer. From these number, can you explain what are the overheads (extra bytes) taken by the HTTP, TCP, IP and Ethernet headers in addition to the original user data payload?

1. When my computer receives a respond from server, it receives it first on the link layer and size of the packet is 514 bytes + 8 bytes of Preamble + 4 bytes of CRC = 526 bytes;

2. After that packet is forwarded to the network layer, where information from previous layer will be deleted, and size of the packet will be 526 bytes minus:
   a) Preamble – 8 bytes
   b) CRC – 4 bytes
   c) Destination address – 6 bytes;
   d) Source address – 6 bytes;
   e) Type – 2 bytes;

   And now size of the packet is 526 – 8 – 4 – 6 – 6 – 2 = 500 bytes.
3. After that packet is forwarded to the transport layer, where information from network layer will be deleted, and size of the packet will be 500 minus:
   a) Header info – 4 bytes (version, header length, type of service, length)
   b) 16-bit identifier – 2 bytes;
   c) Flags – 3 bits;
   d) Fragment offset – 13 bits;
   e) Time to live – 1 byte;
   f) Upper layer – 1 byte;
   g) Header checksum – 2 bytes;
   h) Source IP address – 4 bytes;
   i) Destination IP address – 4 bytes;

   And now size of the packet is 500 – 4 – 2 – 3bits – 13bits – 1 – 1 – 2 – 4 – 4 = 480 bytes

4. After that packet is forwarded to the application layer, where information from transport layer will be deleted, and size of the packet will be 480 minus:
   a) Source Port – 2 bytes;
   b) Destination port – 2 bytes;
   c) Sequence number – 4 bytes;
   d) Acknowledgement number – 4 bytes;
   e) Header length – 1 byte;
   f) Flags – 12 bits;
   g) Window size value – 2 bytes;
   h) Checksum – 2 bytes;
   i) Urgent pointer – 2 bytes;
   j) TCP Options – 12 bytes;

   And now size of the packet is 480 – 2 – 2 – 4 – 4 – 1 – 12bits – 2 – 2 – 2 – 12 = 448 bytes
5. After that the packet is forwarded to the browser on my computer and information from application layer will be deleted, and size of the packet will be 448 minus:

1) status line – 17 bytes;
2) header lines – 188 bytes:
   a) HTTP Date – 37 bytes;
   b) HTTP Server – 32 bytes;
   c) HTTP Content-Length header – 21 bytes;
   d) Response line – 32 bytes;
   e) HTTP Connection – 24 bytes;

f) HTTP Content-Type header – 40 bytes;

g) Text item – 2 bytes.

So original amount of data without all headers from different layers will be 448 – 17 – 188 = 243 bytes. Difference in size of the packet between the layers is the overhead. So overhead for:

1. Application layer 448 – 243 = 205 bytes
2. Transport layer 480 – 448 = 32 bytes
3. Network layer 500 – 480 = 20 bytes
4. Link layer 526 – 500 = 26 bytes

f. How many TCP segments used by the request and the response packets to send the whole data?

Since I don’t have any reassembling of segments in response and request packets, this means, that it is only one TCP segment in each of them.

g. Why can't you see Preamble and CRC in the Wireshark captured Ethernet frame?

As the Ethernet hardware filters the preamble and CRC, it is not given to Wireshark or any other application.

h. Recall HTTP, TCP segment, IP datagram, and Ethernet frame formats/structure from the lecture slides:

Ethernet frame format:

Preamble	**Destination address**	**Source address**	**Type/Length**	**Data payload**	**CRC**
	c4:b3:01:b8:94:c7	ac:9e:17:8a:5f:b4	IPv4 0x0800	243 bytes	

IP datagram format:

version 4	**Header length** 20 bytes	**Type of service** 0	**Length** 500
16 – bit identifier 0xd78c	**flags** 0x02	**Fragment offset** 0	
Time to live 55	**Upper layer** TCP	**Header checksum** 0xaef3	
32-bit source IP address 128.39.121.111			
32-bit destination IP address 192.168.1.69			
options (if any)			
data (variable length, typically a TCP or UDP segment)			

TCP segment structure:

<table>
<tr><td colspan="4">Source port #
80</td><td colspan="5">Destination port #
52458</td></tr>
<tr><td colspan="9">Sequence number
1</td></tr>
<tr><td colspan="9">Acknowledgement number
527</td></tr>
<tr><td>Header length
32</td><td>Not used</td><td>U
0</td><td>A
1</td><td>P
1</td><td>R
0</td><td>S
0</td><td>F
0</td><td>Receive window
235</td></tr>
<tr><td colspan="5">Checksum
0x7a53</td><td colspan="4">Urgent data pointer
0</td></tr>
<tr><td colspan="9">Options
12 bytes</td></tr>
<tr><td colspan="9">Application data
448 bytes</td></tr>
</table>

HTTP response format:

Version	sp	Status code	sp	Phrase	cr	lf
HTTP/1.1		200		OK		

Header field name		value	cr	lf
Date: Wed, 07 Mar 2018 19:20:58 GMT\r\n Server: Apache/2.4.18 (Ubuntu)\r\n Content-Length: 243\r\n Keep-Alive: timeout=5, max=100\r\n Connection: Keep-Alive\r\n Content-Type: text/html; charset=UTF-8\r\n \r\n File Data: 243 bytes				

Header field name		value	cr	lf

cr	lf

Entity body

<html>\n
<body>\nHi!

You have visited this page for Oblig-1 from IP: 109.189.54.42
 on Wednesday 7th of March 2018 08:20:58 PM
 and entered your postcode as '0178'.

 All the best!

 Raju Shrestha</body>\n
</html>\n

6

Sections (Full points)	Expected points	Comments
1.a. IP addressing (20)	18	+ I think I learnt a lot about FLSM and VLSM subnetting, did correct calculations and explained all steps -My explanation sometimes may be difficult to understand, but I tried to explain everything in simple way, how I understood it.
1.b. Longest prefix (10)	9	+ I understood how routing/forwarding tables are created and used by routers, made correct routing tables -I still have some more theoretical small details that are a little bit unclear, but it is something minor that does not influence how I see the whole picture, how this works.
2.a. PT (static) (12)	12	+ Static implementation works, I can send packets between all hosts

		-I do not think that there are some minus, everything seems to work
2.b. PT (RIP) (12)	12	+RIP implementation works, all hosts can communicate with each other, everything that was in assignment is done -I had some small problems in the beginning before I changed version 1 to version 2, but now everything works
3. NAT (12)	12	+I explained everything with diagrams, understand this concept, how everything works
4. ARP (12)	12	+this topic I understood very well, and I think that I explained everything correct -sometimes explanation can be too detailed
5. Wireshark (17)	14	+Did all of the assignments -I understood how server and client establish connection and terminate it, but have some small questions about seq and ack numbers.

Report & submission quality (3)	3	+I think I structured everything in a good and readable way, tried to use images and diagrams where it was possible -have a picture of handwritten diagram in 5th exercise, and report is big, but I hope I covered all details and showed that I have understanding of how everything works.
Self-evaluation (2)	2	+Tried to be honest and write when I had some problems -Maybe I am too optimistic about my knowledge…
Total (100)	94	I learnt a lot during this oblig