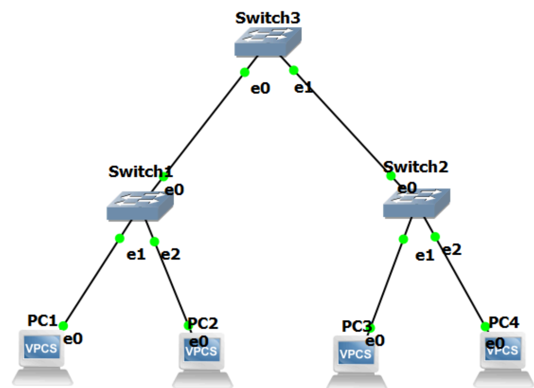
Computer Networks CA6 Report

Arian Firoozi 810100196

In this project, some basic network functionalities are implemented and tested using GNS3 emulator and WireShark. The router used for this assignment is Cisco c7200 and all switches and vpcs' are default GNS3 devices.

Phase 1:

A simple topology is implemented in which all devices share a common IP address, the network does not have to deal with any routing, and pinging vpcs' is simply done by 'ping' command in the terminal. The overall topology is shown in image 1.1.



1.1 topology

As instructed, device state has been observed in the GNS3 terminal using 'show device' command. Part of the result is shown below. Full result was to long to put in the report but it is accessible in the file './Q1/show_device_output.txt'.

```
show device
Ethernet switch Switch1 is always-on
  Running on server DESKTOP-JG8BREE with port 3080
  Local ID is 2 and server ID is efe36c2d-9eec-4150-906c-5b26ac77fc43
  Console is on port 5000 and type is none
   Port Ethernet0 is in access  mode, with VLAN ID 1,
    connected to Switch3 on port Ethernet0
   Port Ethernet1 is in access  mode, with VLAN ID 1,
    connected to PC1 on port Ethernet0
   Port Ethernet2 is in access  mode, with VLAN ID 1,
    connected to PC2 on port Ethernet0
   Port Ethernet3 is empty
   Port Ethernet4 is empty
   Port Ethernet5 is empty
   Port Ethernet6 is empty
   Port Ethernet7 is empty

Node PC1 is started
  Running on server DESKTOP-JG8BREE with port 3080
  Local ID is 5 and server ID is 77376902-8635-4b7f-8503-b137721fef56
  Console is on port 5003 and type is telnet
     Ethernet0 connected to Switch1 on port Ethernet1
```

Both vpcs pinged each other successfully, terminal output for PC1 pinging PC2 is shown in image 1.2, other instances of ping operation can be found in the './Q1' directory.

```
PC1> show

NAME   IP/MASK              GATEWAY           MAC                LPORT  RHOST:PORT
PC1    192.168.1.0/8        255.0.0.0         00:50:79:66:68:00  10012  127.0.0.1:10013
       fe80::250:79ff:fe66:6800/64

PC1> show

NAME   IP/MASK              GATEWAY           MAC                LPORT  RHOST:PORT
PC1    192.168.1.0/8        255.0.0.0         00:50:79:66:68:00  10012  127.0.0.1:10013
       fe80::250:79ff:fe66:6800/64

PC1> ping 192.168.1.0
192.168.1.0 icmp_seq=1 ttl=64 time=0.001 ms
192.168.1.0 icmp_seq=2 ttl=64 time=0.001 ms
192.168.1.0 icmp_seq=3 ttl=64 time=0.001 ms
192.168.1.0 icmp_seq=4 ttl=64 time=0.001 ms
192.168.1.0 icmp_seq=5 ttl=64 time=0.001 ms

PC1> ping 192.168.1.1
84 bytes from 192.168.1.1 icmp_seq=1 ttl=64 time=1.697 ms
84 bytes from 192.168.1.1 icmp_seq=2 ttl=64 time=1.656 ms
84 bytes from 192.168.1.1 icmp_seq=3 ttl=64 time=1.706 ms
84 bytes from 192.168.1.1 icmp_seq=4 ttl=64 time=1.607 ms
84 bytes from 192.168.1.1 icmp_seq=5 ttl=64 time=1.437 ms

PC1>
```

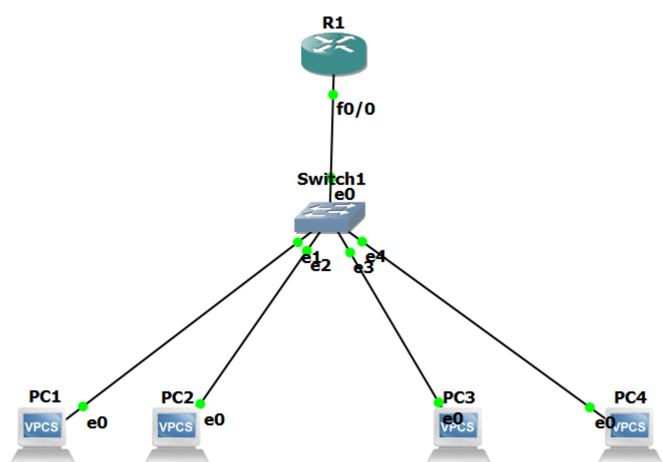
1.2 PC1 pinging PC2

Phase 2:

The toplogy of this phase includes a router and 2 VLANs: VLAN 10 and VLAN 20. In order to create VLANs, switch ports were modified to have 3 different types:

- one port for connecting the router, which is a dot1q port to allow different VLANs to communicate through it
- Two ports for VLAN 10
- Two ports for VLAN 20

These ports then used to connect VPCS in such way that PC1 and PC2 are in VLAN 10 and PC3 and PC4 are in VLAN 20.
After building the basic topology, VLANs have been introduced to router with following commands:



2.1 topology

```
interface FastEthernet0/0.10
encapsulation dot1Q 10
ip address 192.168.10.254 255.255.255.0
interface FastEthernet0/0.20
encapsulation dot1Q 20
ip address 192.168.20.254 255.255.255.0
```

Finally routing between two VLANs configured with ‘router rip’ and networks were added to router. Full router configuration commands are available in ‘./Q2/router_config.txt’ file.
As shown below, PC2 pings both networks successfully.

```
PC2> show

NAME   IP/MASK              GATEWAY           MAC                LPORT  RHOST:PORT
PC2    192.168.10.2/24      192.168.10.254    00:50:79:66:68:01  10014  127.0.0.1:10015
       fe80::250:79ff:fe66:6801/64

PC2> ping 192.168.20.1
84 bytes from 192.168.20.1 icmp_seq=1 ttl=63 time=62.138 ms
84 bytes from 192.168.20.1 icmp_seq=2 ttl=63 time=30.970 ms
84 bytes from 192.168.20.1 icmp_seq=3 ttl=63 time=30.757 ms
84 bytes from 192.168.20.1 icmp_seq=4 ttl=63 time=31.408 ms
84 bytes from 192.168.20.1 icmp_seq=5 ttl=63 time=30.547 ms

PC2> ping 192.168.10.1
84 bytes from 192.168.10.1 icmp_seq=1 ttl=64 time=1.301 ms
84 bytes from 192.168.10.1 icmp_seq=2 ttl=64 time=1.030 ms
84 bytes from 192.168.10.1 icmp_seq=3 ttl=64 time=1.090 ms
84 bytes from 192.168.10.1 icmp_seq=4 ttl=64 time=1.030 ms
84 bytes from 192.168.10.1 icmp_seq=5 ttl=64 time=0.541 ms
```

2.2 PC2 pinging PC1 and PC3

Wireshark

Data transfer has been captured using Wireshark on the FastEthernet port of the router, results are shown in figure 2.3. Operation captured is PC3 (with IP address 192.168.20.1) pinging PC1 (with IP address 192.168.10.1). We can see that in the first ping request (No.4) routing tables are not complete, therefore router broadcasts a message to find the IP address associated with destination. The same happens for PC1’s reply message and after first ping is complete, routing table is formed for this operation and no broadcast messages are present for the next ping messages.

	Time	Source	Destination	Protocol	Length	Info
1	0.000000	ca:01:23:f0:00:00	ca:01:23:f0:00:00	LOOP	60	Reply
2	15.782029	192.168.10.254	224.0.0.9	RIPv2	70	Response
3	16.408490	192.168.20.254	224.0.0.9	RIPv2	70	Response
4	20.992552	192.168.20.1	192.168.10.1	ICMP	102	Echo (ping) request id=0xa822, seq=1/256, ttl=64 (no response found!)
5	21.004909	ca:01:23:f0:00:00	Broadcast	ARP	64	Who has 192.168.10.1? Tell 192.168.10.254
6	21.004909	00:50:79:66:68:00	ca:01:23:f0:00:00	ARP	64	192.168.10.1 is at 00:50:79:66:68:00
7	21.035159	192.168.20.1	192.168.10.1	ICMP	102	Echo (ping) request id=0xa822, seq=1/256, ttl=63 (reply in 8)
8	21.035159	192.168.10.1	192.168.20.1	ICMP	102	Echo (ping) reply id=0xa822, seq=1/256, ttl=64 (request in 7)
9	21.049930	ca:01:23:f0:00:00	Broadcast	ARP	64	Who has 192.168.20.1? Tell 192.168.20.254
10	21.049930	00:50:79:66:68:02	ca:01:23:f0:00:00	ARP	64	192.168.20.1 is at 00:50:79:66:68:02
11	21.065105	192.168.10.1	192.168.20.1	ICMP	102	Echo (ping) reply id=0xa822, seq=1/256, ttl=63
12	22.097356	192.168.20.1	192.168.10.1	ICMP	102	Echo (ping) request id=0xa922, seq=2/512, ttl=64 (no response found!)
13	22.113118	192.168.20.1	192.168.10.1	ICMP	102	Echo (ping) request id=0xa922, seq=2/512, ttl=63 (reply in 15)
14	22.113118	ca:01:23:f0:00:00	ca:01:23:f0:00:00	LOOP	60	Reply
15	22.113118	192.168.10.1	192.168.20.1	ICMP	102	Echo (ping) reply id=0xa922, seq=2/512, ttl=64 (request in 13)
16	22.127970	192.168.10.1	192.168.20.1	ICMP	102	Echo (ping) reply id=0xa922, seq=2/512, ttl=63
17	23.148536	192.168.20.1	192.168.10.1	ICMP	102	Echo (ping) request id=0xaa22, seq=3/768, ttl=64 (no response found!)
18	23.163423	192.168.20.1	192.168.10.1	ICMP	102	Echo (ping) request id=0xaa22, seq=3/768, ttl=63 (reply in 19)
19	23.164363	192.168.10.1	192.168.20.1	ICMP	102	Echo (ping) reply id=0xaa22, seq=3/768, ttl=64 (request in 18)
20	23.179150	192.168.10.1	192.168.20.1	ICMP	102	Echo (ping) reply id=0xaa22, seq=3/768, ttl=63
21	24.203648	192.168.20.1	192.168.10.1	ICMP	102	Echo (ping) request id=0xab22, seq=4/1024, ttl=64 (no response found!)
22	24.219326	192.168.20.1	192.168.10.1	ICMP	102	Echo (ping) request id=0xab22, seq=4/1024, ttl=63 (reply in 23)
23	24.219326	192.168.10.1	192.168.20.1	ICMP	102	Echo (ping) reply id=0xab22, seq=4/1024, ttl=64 (request in 22)
24	24.234523	192.168.10.1	192.168.20.1	ICMP	102	Echo (ping) reply id=0xab22, seq=4/1024, ttl=63
25	25.256170	192.168.20.1	192.168.10.1	ICMP	102	Echo (ping) request id=0xac22, seq=5/1280, ttl=64 (no response found!)
26	25.271093	192.168.20.1	192.168.10.1	ICMP	102	Echo (ping) request id=0xac22, seq=5/1280, ttl=63 (reply in 27)
27	25.271603	192.168.10.1	192.168.20.1	ICMP	102	Echo (ping) reply id=0xac22, seq=5/1280, ttl=64 (request in 26)
28	25.286661	192.168.10.1	192.168.20.1	ICMP	102	Echo (ping) reply id=0xac22, seq=5/1280, ttl=63

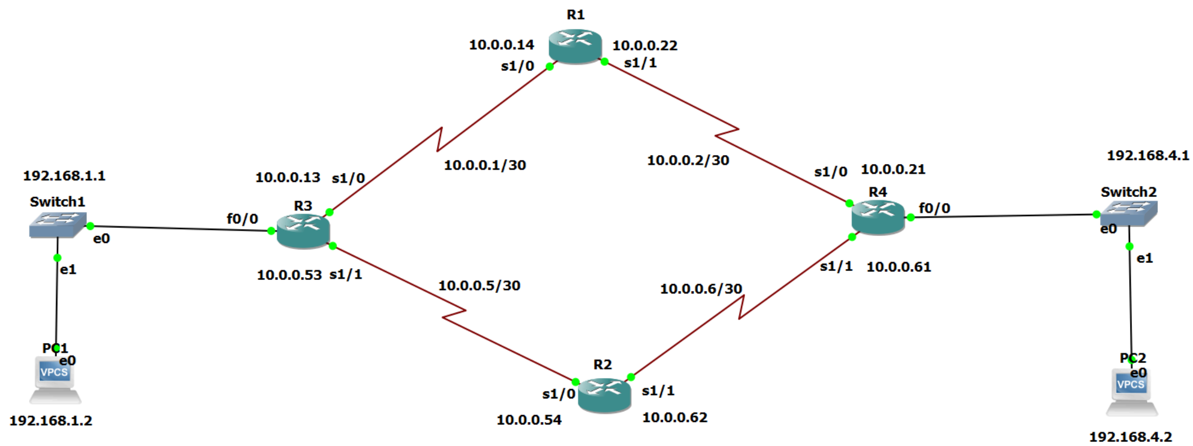
```
rame 13: 102 bytes on wire (816 bits), 102 bytes captured (816 bits) on interface -, id 0
thernet II, Src: ca:01:23:f0:00:00 (ca:01:23:f0:00:00), Dst: 00:50:79:66:68:00 (00:50:79:66:68:00)
02.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 10
nternet Protocol Version 4, Src: 192.168.20.1, Dst: 192.168.10.1
nternet Control Message Protocol
```

```
0000  00 50 79 66 68 00 ca 01  23 f0 00 00 81 00 00 0a
0010  08 00 45 00 00 54 22 a9  00 00 3f 01 b9 ad c0 a8
0020  14 01 c0 a8 0a 01 08 00  76 e7 a9 22 00 02 08 09
0030  0a 0b 0c 0d 0e 0f 10 11  12 13 14 15 16 17 18 19
0040  1a 1b 1c 1d 1e 1f 20 21  22 23 24 25 26 27 28 29
0050  2a 2b 2c 2d 2e 2f 30 31  32 33 34 35 36 37 38 39
```

2.3 Wireshark capturing data transfer on FastEthernet

Phase 3:

In this phase, routers are configured staticly and manually. 4 routers have been configured to have needed ports for connections, both FastEthernet and serial ports have been added to slots in order to connect routers to other routers and switches. The topology is shown in figure 3.1. Notice that IP addresses are not exactly as the document suggested, this is due to certain ip addresses raising errors that couldn't be fixed during the assignment (it may be due those ports being on use by some other applications).



3.1 topology

Routers then configured staticly to recognize each switch address and its route as well as subnet mask used for communication, using 'ip route' command in GNS3 terminal. Here is and example of used commands:

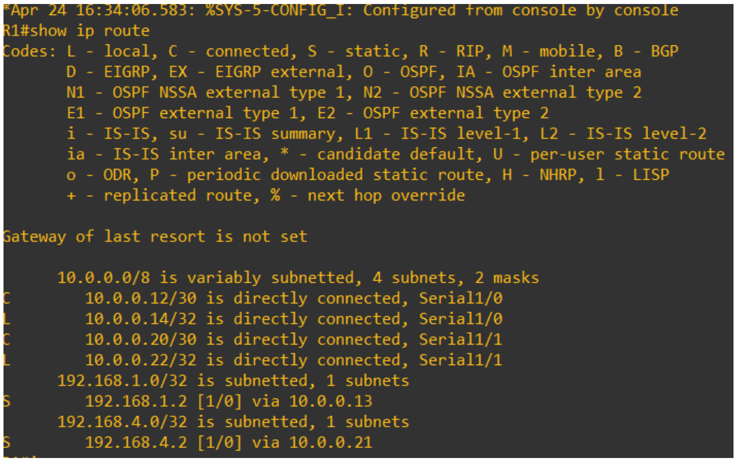
```
ip route 192.168.4.1 255.255.255.255 10.0.0.51
ip route 192.168.4.1 255.255.255.255 10.0.0.54
ip route 192.168.4.2 255.255.255.255 10.0.0.54
ip route 192.168.4.2 255.255.255.255 10.0.0.14
```

After configuration finished, PC2 was able to successfully ping PC1 via routers (figure 3.2).

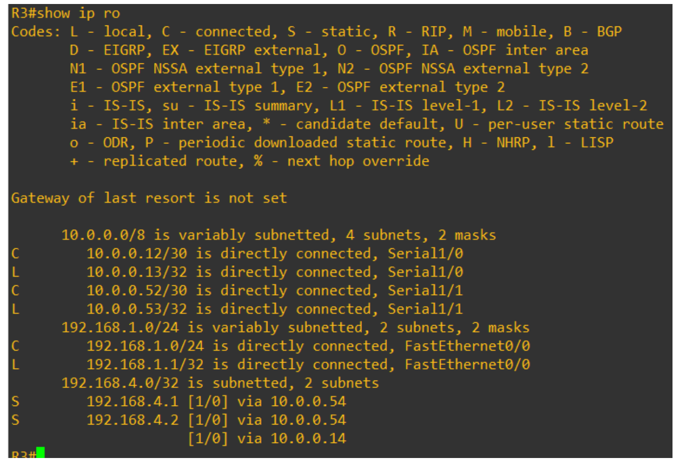
```
PC2> show

NAME   IP/MASK              GATEWAY           MAC                LPORT  RHOST:PORT
PC2    192.168.4.2/24       192.168.4.1       00:50:79:66:68:01  10038  127.0.0.1:10039
       fe80::250:79ff:fe66:6801/64

PC2> ping 192.168.1.2
192.168.1.2 icmp_seq=1 timeout
192.168.1.2 icmp_seq=2 timeout
84 bytes from 192.168.1.2 icmp_seq=3 ttl=61 time=92.102 ms
84 bytes from 192.168.1.2 icmp_seq=4 ttl=61 time=92.717 ms
84 bytes from 192.168.1.2 icmp_seq=5 ttl=61 time=75.434 ms
```

3.2 PC2 pings PC1

Routing table of routers are shown in figure 3.3. Other routing tables are accessible in './Q3/Router*.png'



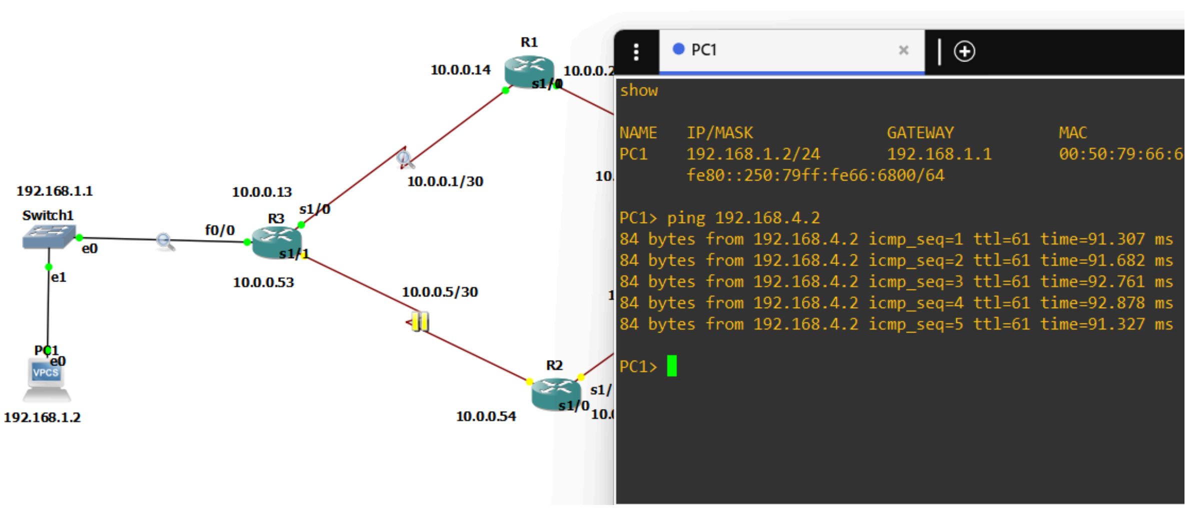
3.3.1 R1



3.3.2 R3

3.3 routing tables

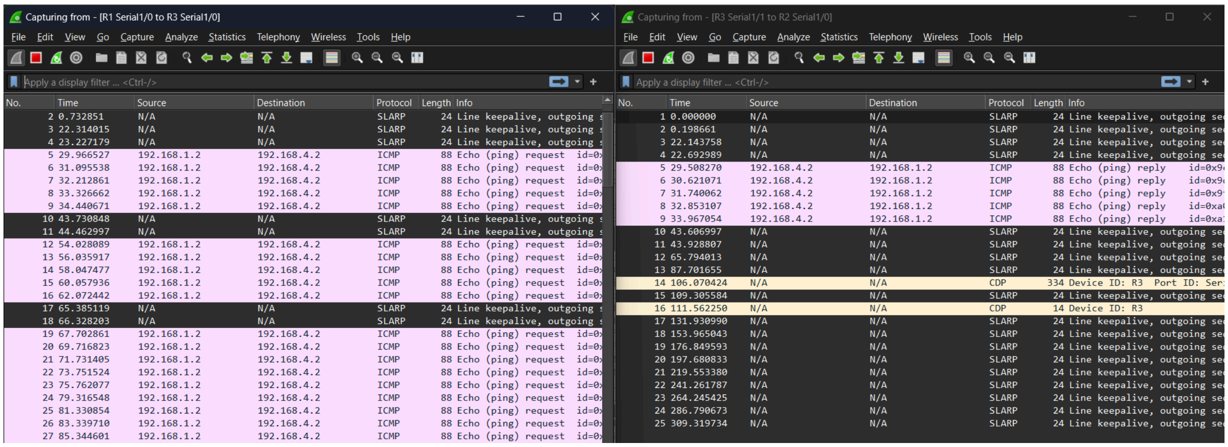
The link between R2 and R3 disconnected for testing purposes and as shown in figure 3.4, the VPCS' were still able to communicate.



3.4 R2 and R3 disconnected

Wireshark

By inspecting R3-R1 and R3-R2 connections before and after disconnecting R2-R3, it is observable that replies to ping messages were being sent via R2 router, but after disconnection, R4 re-routed replies using R1.



No.	Time	Source	Destination	Protocol	Length	Info
2	0.732851	N/A	N/A	SLARP	24	Line keepalive, outgoing s
3	22.314015	N/A	N/A	SLARP	24	Line keepalive, outgoing s
4	23.227179	N/A	N/A	SLARP	24	Line keepalive, outgoing s
5	29.966527	192.168.1.2	192.168.4.2	ICMP	88	Echo (ping) request id=0x
6	31.095538	192.168.1.2	192.168.4.2	ICMP	88	Echo (ping) request id=0x
7	32.212861	192.168.1.2	192.168.4.2	ICMP	88	Echo (ping) request id=0x
8	33.326662	192.168.1.2	192.168.4.2	ICMP	88	Echo (ping) request id=0x
9	34.440671	192.168.1.2	192.168.4.2	ICMP	88	Echo (ping) request id=0x
10	43.730848	N/A	N/A	SLARP	24	Line keepalive, outgoing s
11	44.462997	N/A	N/A	SLARP	24	Line keepalive, outgoing s
12	54.028089	192.168.1.2	192.168.4.2	ICMP	88	Echo (ping) request id=0x
13	56.035917	192.168.1.2	192.168.4.2	ICMP	88	Echo (ping) request id=0x
14	58.047477	192.168.1.2	192.168.4.2	ICMP	88	Echo (ping) request id=0x
15	60.057936	192.168.1.2	192.168.4.2	ICMP	88	Echo (ping) request id=0x
16	62.072442	192.168.1.2	192.168.4.2	ICMP	88	Echo (ping) request id=0x
17	65.385119	N/A	N/A	SLARP	24	Line keepalive, outgoing s
18	66.328203	N/A	N/A	SLARP	24	Line keepalive, outgoing s
19	67.702861	192.168.1.2	192.168.4.2	ICMP	88	Echo (ping) request id=0x
20	69.716823	192.168.1.2	192.168.4.2	ICMP	88	Echo (ping) request id=0x
21	71.731405	192.168.1.2	192.168.4.2	ICMP	88	Echo (ping) request id=0x
22	73.751524	192.168.1.2	192.168.4.2	ICMP	88	Echo (ping) request id=0x
23	75.762077	192.168.1.2	192.168.4.2	ICMP	88	Echo (ping) request id=0x
24	79.316548	192.168.1.2	192.168.4.2	ICMP	88	Echo (ping) request id=0x
25	81.330854	192.168.1.2	192.168.4.2	ICMP	88	Echo (ping) request id=0x
26	83.339710	192.168.1.2	192.168.4.2	ICMP	88	Echo (ping) request id=0x
27	85.344601	192.168.1.2	192.168.4.2	ICMP	88	Echo (ping) request id=0x

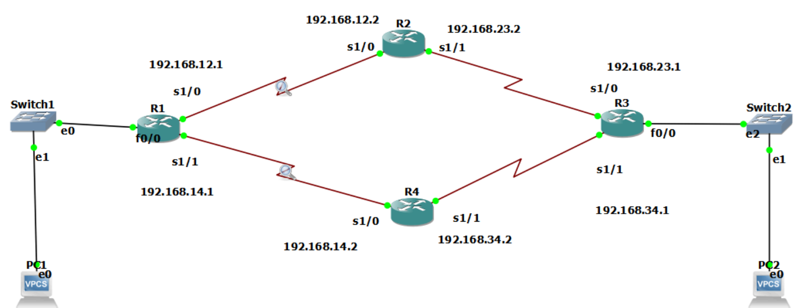
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	N/A	N/A	SLARP	24	Line keepalive, outgoing se
2	0.198661	N/A	N/A	SLARP	24	Line keepalive, outgoing se
3	22.143758	N/A	N/A	SLARP	24	Line keepalive, outgoing se
4	22.692989	N/A	N/A	SLARP	24	Line keepalive, outgoing se
5	29.508270	192.168.4.2	192.168.1.2	ICMP	88	Echo (ping) reply id=0x9
6	30.621071	192.168.4.2	192.168.1.2	ICMP	88	Echo (ping) reply id=0x9
7	31.740062	192.168.4.2	192.168.1.2	ICMP	88	Echo (ping) reply id=0x9
8	32.853107	192.168.4.2	192.168.1.2	ICMP	88	Echo (ping) reply id=0xa
9	33.967054	192.168.4.2	192.168.1.2	ICMP	88	Echo (ping) reply id=0xa
10	43.606997	N/A	N/A	SLARP	24	Line keepalive, outgoing se
11	43.928807	N/A	N/A	SLARP	24	Line keepalive, outgoing se
12	65.794013	N/A	N/A	SLARP	24	Line keepalive, outgoing se
13	87.701655	N/A	N/A	SLARP	24	Line keepalive, outgoing se
14	106.070424	N/A	N/A	CDP	334	Device ID: R3 Port ID: Ser
15	109.305584	N/A	N/A	SLARP	24	Line keepalive, outgoing se
16	111.562250	N/A	N/A	CDP	14	Device ID: R3
17	131.930990	N/A	N/A	SLARP	24	Line keepalive, outgoing se
18	153.965043	N/A	N/A	SLARP	24	Line keepalive, outgoing se
19	176.849593	N/A	N/A	SLARP	24	Line keepalive, outgoing se
20	197.680833	N/A	N/A	SLARP	24	Line keepalive, outgoing se
21	219.553380	N/A	N/A	SLARP	24	Line keepalive, outgoing se
22	241.261787	N/A	N/A	SLARP	24	Line keepalive, outgoing se
23	264.245425	N/A	N/A	SLARP	24	Line keepalive, outgoing se
24	286.790673	N/A	N/A	SLARP	24	Line keepalive, outgoing se
25	309.319734	N/A	N/A	SLARP	24	Line keepalive, outgoing se

3.5 Wireshark catch on R3-R2 and R3-R1 links

Phase 4:

The topology of this phase is almost identical to phase 3, but routing is done with OSPF protocol instead of manual routing.

Implementation of OSPF routing is relatively simple: all routers should be configured to recognize different domains in the network. These domains then get linked to each other forming areas and nodes in each area can communicate with each other with routers. Instructions to initialize this protocol are as follows:



4.1 topology

```
router ospf 1
 network 192.168.1.0 0.0.0.255 area 1
 network 192.168.2.0 0.0.0.255 area 1
 network 192.168.3.0 0.0.0.255 area 1
 network 192.168.12.0 0.0.0.255 area 1
```

Full configuration is available in importable project.

IP routing table, table of aliases and OSPF routing table for R3 router are shown in figures 4.2, 4.3 and 4.4. Tables for other routers are available in './Q4/'.

```
*Apr 28 15:49:03.975: %SYS-5-CONFIG_I: Configured from console by console
R3#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is not set

O     192.168.1.0/24 [110/129] via 192.168.34.2, 00:04:59, Serial1/1
                     [110/129] via 192.168.23.2, 00:04:59, Serial1/0
      192.168.3.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.3.0/24 is directly connected, FastEthernet0/0
L        192.168.3.1/32 is directly connected, FastEthernet0/0
O     192.168.12.0/24 [110/128] via 192.168.23.2, 00:04:59, Serial1/0
O     192.168.14.0/24 [110/128] via 192.168.34.2, 00:04:59, Serial1/1
      192.168.23.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.23.0/24 is directly connected, Serial1/0
L        192.168.23.1/32 is directly connected, Serial1/0
      192.168.34.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.34.0/24 is directly connected, Serial1/1
L        192.168.34.1/32 is directly connected, Serial1/1
R3#
```

4.2 R3 routing table

```
R3#sho ip aliases
Address Type             IP Address       Port
Interface                192.168.3.1
Interface                192.168.23.1
Interface                192.168.34.1
R3#
```

4.3 R3 aliases table

```
R3#show ip ospf route

            OSPF Router with ID (192.168.34.1) (Process ID 1)


                Base Topology (MTID 0)


    Area 1

    Intra-area Route List
*>  192.168.1.0/24, Intra, cost 129, area 1
      via 192.168.23.2, Serial1/0
      via 192.168.34.2, Serial1/1
*   192.168.3.0/24, Intra, cost 1, area 1, Connected
      via 192.168.3.1, FastEthernet0/0
*>  192.168.12.0/24, Intra, cost 128, area 1
      via 192.168.23.2, Serial1/0
*>  192.168.14.0/24, Intra, cost 128, area 1
      via 192.168.34.2, Serial1/1
*   192.168.23.0/24, Intra, cost 64, area 1, Connected
      via 192.168.23.1, Serial1/0
*   192.168.34.0/24, Intra, cost 64, area 1, Connected
      via 192.168.34.1, Serial1/1
R3#
```

4.4 R3 OSPF routing table

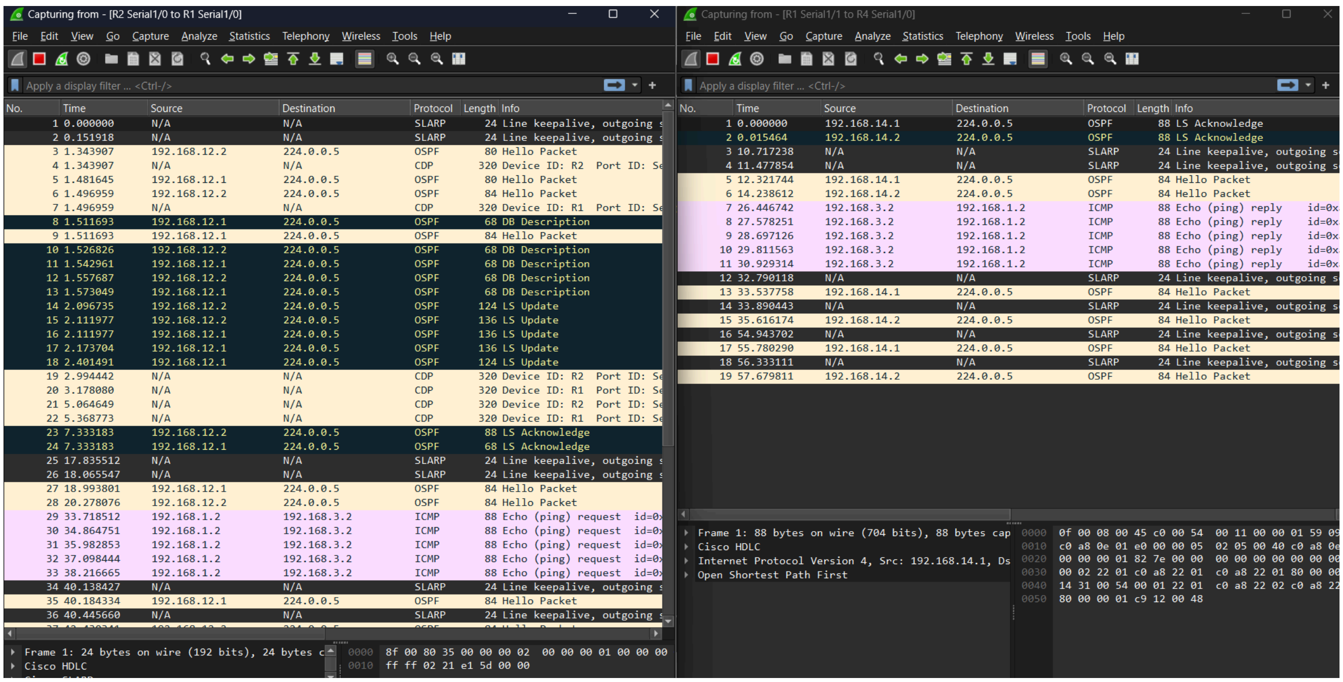
```
R1    R3    R2    R4    PC2    PC1

elcome to Virtual PC Simulator, version 0.6.2
edicated to Daling.
uild time: Apr 10 2019 02:42:20
opyright (c) 2007-2014, Paul Meng (mirnshi@gmail.com)
ll rights reserved.

PCS is free software, distributed under the terms of the "BSD" licence.
ource code and license can be found at vpcs.sf.net.
or more information, please visit wiki.freecode.com.cn.

ress '?' to get help.

xecuting the startup file

hecking for duplicate address...
C1 : 192.168.1.2 255.255.255.0 gateway 192.168.1.1

C1> ping 192.168.3.2
4 bytes from 192.168.3.2 icmp_seq=1 ttl=61 time=120.112 ms
4 bytes from 192.168.3.2 icmp_seq=2 ttl=61 time=92.996 ms
4 bytes from 192.168.3.2 icmp_seq=3 ttl=61 time=92.931 ms
4 bytes from 192.168.3.2 icmp_seq=4 ttl=61 time=92.230 ms
4 bytes from 192.168.3.2 icmp_seq=5 ttl=61 time=91.653 ms

C1>
```

4.5 PC2 pinged by PC1

After configuration, PC1 pinged PC2 successfully, which is shown in figure 4.5.

Wireshark

OSPF algorithm computes the shortest path between nodes in the network using Dijkstra's shortest path algorithm, which constructs a loop-free tree of nodes. This algorithm requires messaging between nodes to find all nodes and fill the router tables. These messages use OSPF protocol which can be seen in figure 4.6. Before sending ping message, the algorithm sends 'Hello Packets' to other routers, then constructs shortest path tree (in Wireshark capture some update and description packets are present, which indicates that the Dijkstra algorithm is being used), and after that regular ping operation is done.



4.6 Wireshark capture on R2-R1 and R4-R1 links