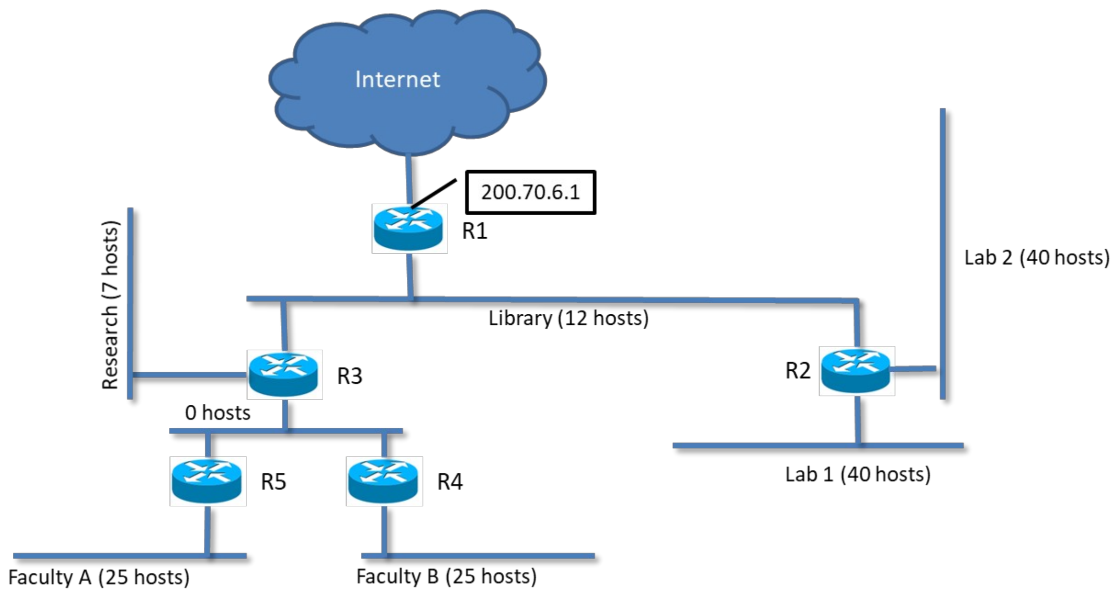
Name: Jakub Hejhal

Group: B – Erasmus

Exexcise Unit 3

Alice, our network administrator of a building at the University of Málagahas deployed the network of the figure. Alice has allocated 12 hosts for the Library, 40 PC for two labs, one connectedto an Ethernet network and the other use WiFi, 25 hosts in each of the two rooms used by lecturers (Faculty A and Faculty B) and 7 hosts in a room where research students work. Faculty B is divided in two zones connected by the bridge B1.



Address blocks:

	IP range
Lab 1 & 2	192.168.1.0 – 192.168.1.127
Library	192.168.1.128 – 192.168.1.255
Research	192.168.2.0 – 192.168.2.63
Faculty A & B	192.168.2.128 – 192.168.2.255

(a) Calculate the network id (including the prefix) and the broadcast addresses for each of the rooms, trying to adapt the prefix to each subnet size (i.e. number of IPs needed):

	Network ID	Mask	Broadcast Address	Used IP / Allowed host IPs
Lab 1	192.168.1.0	/26	192.168.1.63	41 / 62
Lab 2	192.168.1.64	/26	192.168.1.127	41 / 62
Library	192.168.1.128	/27	192.168.1.159	15 / 30
Research	192.168.2.0	/28	192.168.2.15	8 / 14
Faculty A	192.168.2.128	/27	192.168.2.159	26 / 30
Faculty B	192.168.2.160	/27	192.168.2.191	26 / 30
ZeroNet	192.168.3.0	/29	192.168.3.7	3 / 6

(b) Make the routing tables for R2, R5 and for one host of Lab 1.

Note: Assign IP addresses to those network interfaces that you need to complete the routing tables (put these addresses in the figure). You must use the subnet id 192.168.3.0/29 to complete the IP assignment of routers R3, R4 and R5 connected to the network with 0 hosts.

Let's assign the IPs:

R1:
Library IP: 192.168.1.129

R2:
Lab1 IP: 192.168.1.1
Lab2 IP: 192.168.65
Library: 192.168.1.130

R3:
Library: 192.168.1.131
Research: 192.168.2.1
ZeroNet: 192.168.3.1

R4:
Faculty B: 192.168.2.161
ZeroNet: 192.168.3.2

R5:
Faculty A: 192.168.2.129
ZeroNet: 192.168.3.3

Routing table for R2:

Destination	**Next-hop**	**Iterface**
192.168.1.0/26	Direct	Lab1
192.168.1.64/26	Direct	Lab2
192.168.1.128/27	Direct	Library
192.168.2.0/23	192.168.1.131	Library
Default	192.168.1.129	Library

Routing table for R5:

Destination	**Next-hop**	**Iterface**
192.168.2.128/27	192.168.2.129	Faculty A

192.168.2.160/27	192.168.3.2	ZeroNet
Default	192.168.1.129	Library

Routing table for Host in Lab 1:

Destination	**Next-hop**	**Iterface**
192.168.1.0/26	Direct	Lab 1
Default	192.168.1.1	Lab 1

(c) The machine A sends an IP datagram with a payload of 2100 bytes, no options,and a TTL equal 64 from its wireless interface to the Ethernet interface of machine B. Considering that the MTU of Lab 1 Ethernet segment is 1500 bytes and the MTU of Lab 2 WiFi is 1284 bytes (including the WPA encryption), how many fragments arrive to the Ethernet interface of machine B? For each of the fragments show the value of the following fields: IP source address, IP target address, total length, flags, offset and TTL.

I am going to use IP A and IP B as placeholders for the real IPs of A and B, because it's not specified in the task description, what those should be.

MTU Lab1 = 1500B
MTU Lab2 = 1284B

Whether or not B receives anything depends on whether the "Don't fragment" bit is set or no.
If it's not set, then the datagram is fragmented. It it is set, datagram will not leave the A's network card and will be dropped immediately.

First the 2100B datagram is split into two parts

Part1:
size: 1480B + 20B header
IP source: IP of A (it isn't said anywhere, what IP does A have)
IP target: IP of B (we don't know the IP of B either)
TTL: 64
Offset: 0B
Flags: Don't fragment - 0, More fragments - 1

Part2:
size: 620B + 20B header
IP source: IP of A
IP target: IP of B
TTL: 64
Offset: 1480B
Flags: Don't fragment - 0, More fragments - 0

After the packets are forwarded to B on the WiFi by R2, Part1 is further fragmented, resulting in 3 final fragments:

Part1:
size: 1264B + 20B header = 1284B (MTU)
IP source: IP of A
IP target: IP of B
TTL: 63
Offset: 0B
Flags: Don't fragment - 0, More fragments – 1

Part2:

size: 216B + 20B header = 236B
IP source: IP of A
IP target: IP of B
TTL: 63
Offset: 1264B
Flags: Don’t fragment - 0, More fragments - 1

Part3:
size: 620B + 20B header = 640B
IP source: IP of A
IP target: IP of B
TTL: 63
Offset: 1480B
Flags: Don’t fragment - 0, More fragments - 0

d)
A looks into it’s routing table and sees, that message to B must go though it’s default gateway. It knows its IP (192.168.1.1), but doesn’t know it MAC yet.
A sends ARP to Broadcast MAC: Who has 192.168.1.1? Tell IP A

R2 saves A’s MAC to its ARP cache
R2 responds to A: 192.168.1.1 is at MAC 22:22
A updates it’s ARP cache with record 192.168.1.1 – 22:22

A then sends Ping Echo-Request for B through R2
Source IP: IP A
Dest IP: IP B

R2 looks into it’s routing table and sees, that B is on its Lab2 interface. It doesn’t know its MAC yet. So it does exactly the same thing A just did.

R2 sends ARP to Broadcast MAC: Who has IP B? Tell 192.168.65
B saves R2’s MAC to its ARP cache: 192.168.65 - MAC 22:22

B responds to R2: IP B is at MAC 44:44
R2 updates it’s ARP cache with record IP B – 22:22

R2 then sends frame with Ping Echo-Request from A to B directly to B.

B replies with Ping Echo-Reply.
Source IP: IP B
Dest IP: IP A

B sends this frame containing the Ping message to it’s default gateway, which is R2.
B remembers it’s MAC from the last time R2 asked for B’s MAC.
R2 forwards the IP datagram to A, still having A’s MAC in it’s ARP cache as well.

A happily recieves ping reply, calculates RTT and show the info to user.