iu-ne-lab-06-Iskander_Nafikov

- **Name**: Iskander Nafikov
- **E-mail**: i.nafikov@innopolis.university
- **Username**: [iskanred](https://github.com/iskanred)
- **Hostname**: `lenovo`

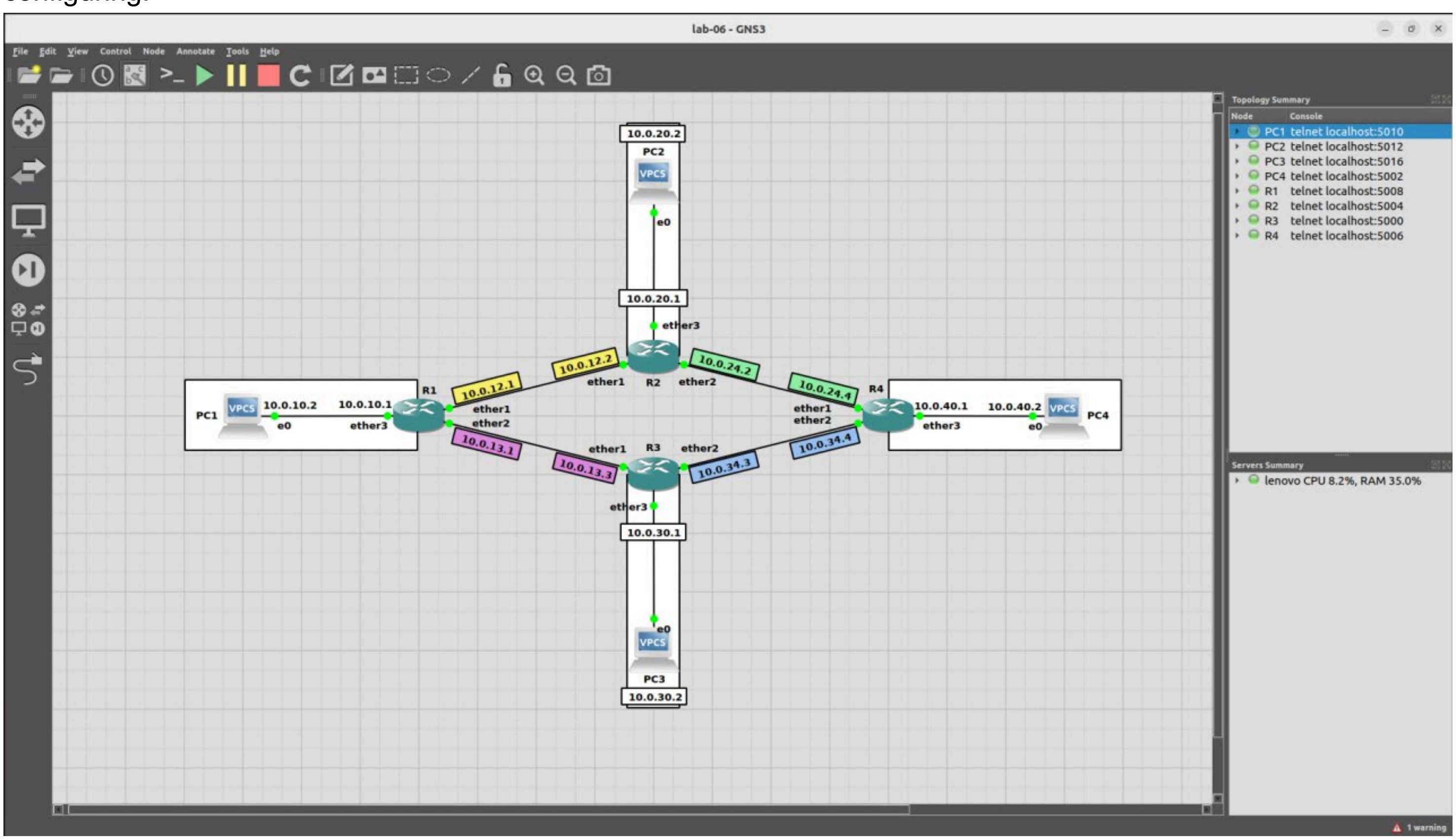
Task 1 - Prepare your network topology

> **1.** In the GNS3 project, select and install a virtual routing solution that you would like to use: for example, Mikrotik (recommended), Pfsense, vyos.

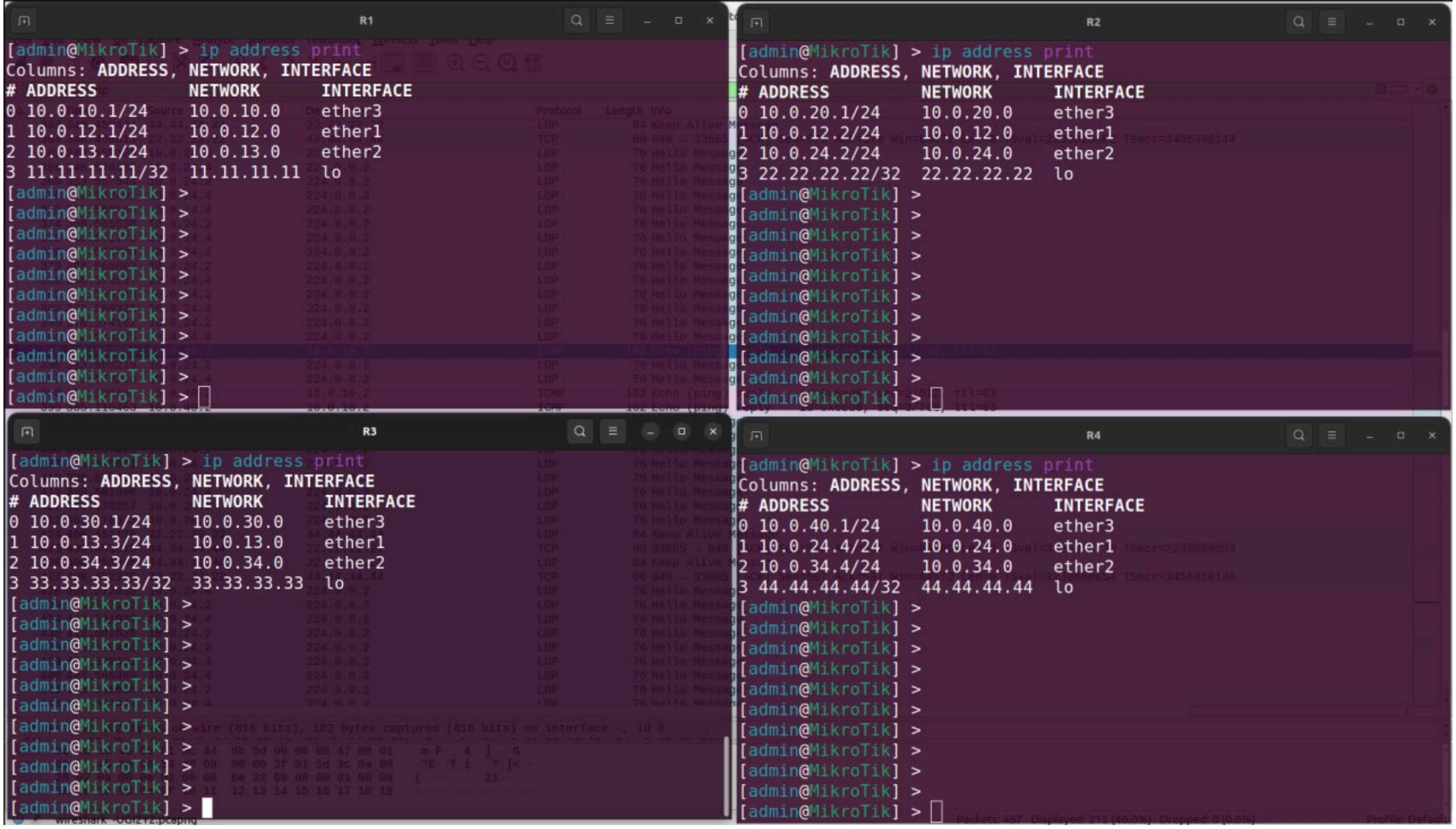
I prefer Mikrotik

> **2.** Prepare a simple network consisting of at least three router and two hosts. Each one of them has a different subnet, and the routers should be able to reach each other (for example, a bus topology with dynamic routing). Your network must have routing protocols configured.

- I decided to create a new network that will be more interesting for configuring MPLS and to practise OSPF configuring.



- This is routers' IP addresses that lay in different subnets. Pay attention to the address assigned to loopback `lo` interface. It was given to be a router-id in OSPF as a LSR-ID in LDP



Task 2 - MPLS learning & configuring

> **1.** Briefly answer the questions or give one-line description what is it: LSP, VPLS, PHP, LDP, MPLS L2VPN, CE-router, PE-router?

- **MPLS (Multiprotocol Label Switching)**: High-performance networking technology that directs data from one node to the next based on short, fixed-length labels rather than long network addresses, which speeds up the flow of traffic across the network. It enables the creation of end-to-end circuits (LPS) which can support various types of traffic, including IP packets, by efficiently managing Traffic Engineering such as managing bandwidth and ensuring quality of service (QoS). MPLS is commonly used in enterprise and service provider networks to create Virtual

Private Networks (VPNs), improve traffic management, and optimise network performance.

MPLS works by prefixing packets with an MPLS header, containing one or more labels. This is called a label stack.

MPLS packet structure

<table>
<tr><th>Offset</th><th>Octet</th><th colspan="8">0</th><th colspan="8">1</th><th colspan="8">2</th><th colspan="8">3</th></tr>
<tr><th>Octet</th><th>Bit</th><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th><th>15</th><th>16</th><th>17</th><th>18</th><th>19</th><th>20</th><th>21</th><th>22</th><th>23</th><th>24</th><th>25</th><th>26</th><th>27</th><th>28</th><th>29</th><th>30</th><th>31</th></tr>
<tr><th>0</th><th>0</th><td colspan="32">MPLS Label [1]</td></tr>
<tr><th>4</th><th>32</th><td colspan="32">MPLS Label [2]</td></tr>
<tr><th>⋮</th><th>⋮</th><td colspan="32">⋮</td></tr>
<tr><th>⋮</th><th>⋮</th><td colspan="32">MPLS Label [n]</td></tr>
<tr><th>4n</th><th>32n</th><td colspan="32" rowspan="3">Packet</td></tr>
<tr><th>⋮</th><th>⋮</th></tr>
<tr><th>⋮</th><th>⋮</th></tr>
</table>

Each entry in the label stack contains four fields:

MPLS Label

<table>
<tr><th>Offset</th><th>Octet</th><th colspan="8">0</th><th colspan="8">1</th><th colspan="8">2</th><th colspan="8">3</th></tr>
<tr><th>Octet</th><th>Bit</th><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th><th>15</th><th>16</th><th>17</th><th>18</th><th>19</th><th>20</th><th>21</th><th>22</th><th>23</th><th>24</th><th>25</th><th>26</th><th>27</th><th>28</th><th>29</th><th>30</th><th>31</th></tr>
<tr><th>0</th><th>0</th><td colspan="20">Label</td><td colspan="3">TC</td><td>S</td><td colspan="8">Time to Live</td></tr>
</table>

Label: 20 bits

A label with the value of 1 represents the router alert label.

Traffic Class (TC): 3 bits

Field for QoS (quality of service) priority and ECN (Explicit Congestion Notification). Prior to 2009 this field was called EXP.[12]

Bottom of Stack (S): 1 bit

If this flag is set, it signifies that the current label is the last in the stack.

Time to Live (TTL): 8 bits

Time to live.

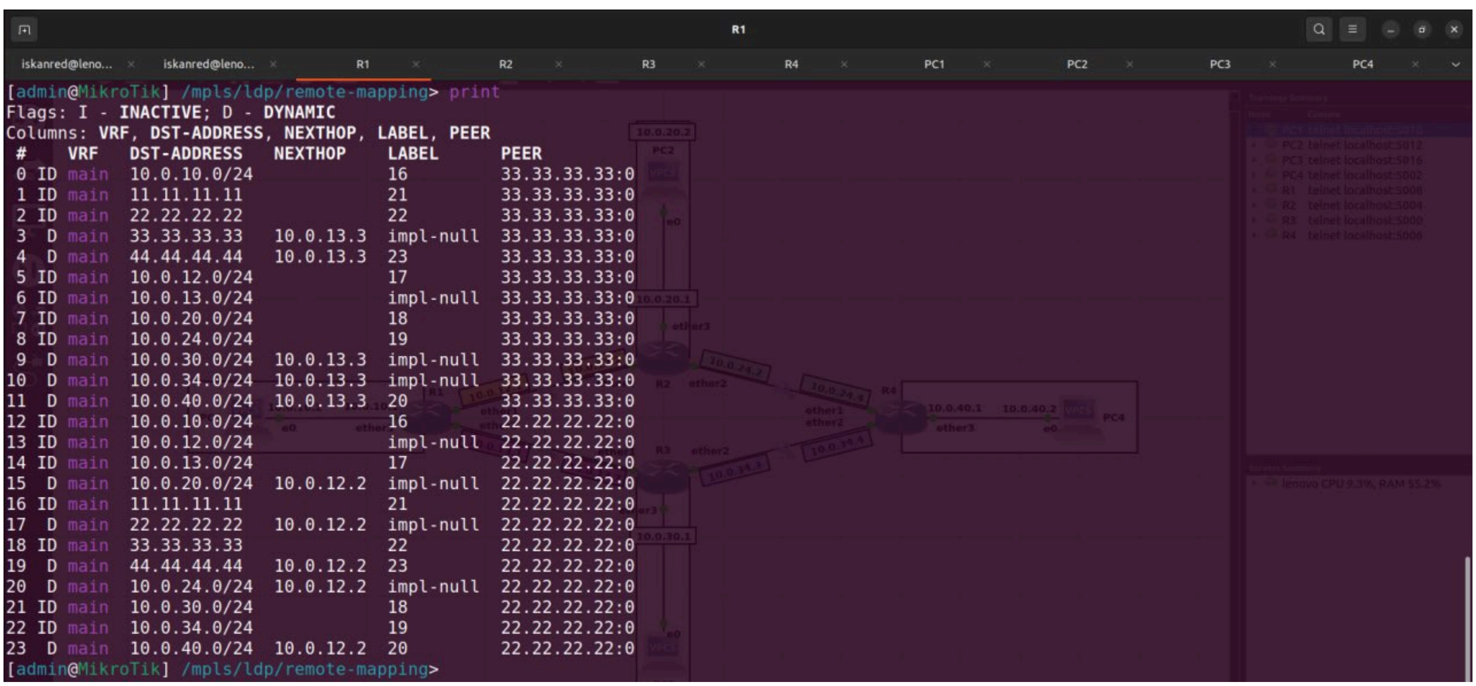
- **LSP (Label Switched Path)**: A predetermined path in an MPLS network through which labeled packets travel from a source to a destination. This path is established using special protocols such as LDP or RSVP.
- **LDP (Label Distribution Protocol)**: A protocol used in MPLS networks to establish the mapping between network layer (IP) addresses and MPLS labels.
- **VPLS (Virtual Private LAN Service)**: A Layer 2 VPN service that allows multiple customer sites to communicate as if they are on the same local area network (LAN), regardless of geographic locations. An alternative is VPWS (Virtual Private WAN Service) which implements the same idea but for a point-to-point interaction.
- **PHP (Penultimate Hop Popping)**: A technique used in MPLS networks where the penultimate router removes the MPLS label, passing the packet to the egress router in MPLS network with only the IP header instead of replacing this label to `Explicit Null` (label $= 0$) which is used by default by to signal the egress router to remove the label. For this purpose an egress routers sends an `Implicit Null` (label $= 3$) message to its penultimate router. This might be used where QoS (on the MPLS level) is not important because after stripping the MPLS label, everything `Traffic Class` field disappears.
- **CE-router (Customer Edge Router)**: A router located at the customer's premises that connects to the service provider's network and typically performs routing between the customer's internal network and the service provider's network. This router does not belong to MPLS network and does not know anything about it. It is just connected to a PE-router.
- **PE-router (Provider Edge Router)**: A router located at the edge of the service provider's network that connects to customer edge devices. It acts as a gateway for the customer to the MPLS network. Another name is **LER (Label Edge Router)**.
- **L2VPN**: A service that provides Layer 2 connectivity over an MPLS network, enabling Ethernet, Frame Relay, or ATM services across geographically dispersed sites.

2. Configure MPLS domain on your OSPF network, first without authentication.

- I configured MPLS "dynamically" using LDP

```
# R1
/mpls ldp add afi=ip lsr-id=11.11.11.11 transport-addresses=11.11.11.11
/mpls ldp interface add interface=ether1
/mpls ldp interface add interface=ether2
# ether3 is a customer network

# R2
/mpls ldp add afi=ip lsr-id=22.22.22.22 transport-addresses=22.22.22.22
/mpls ldp interface add interface=ether1
/mpls ldp interface add interface=ether2
# ether3 is a customer network

# R3
/mpls ldp add afi=ip lsr-id=33.33.33.33 transport-addresses=33.33.33.33
/mpls ldp interface add interface=ether1
/mpls ldp interface add interface=ether2
# ether3 is a customer network

# R4
/mpls ldp add afi=ip lsr-id=44.44.44.44 transport-addresses=44.44.44.44
/mpls ldp interface add interface=ether1
/mpls ldp interface add interface=ether2
# ether3 is a customer network
```

- And we see that it worked



```
[admin@MikroTik] /mpls/ldp/remote-mapping> print
Flags: I - INACTIVE; D - DYNAMIC
Columns: VRF, DST-ADDRESS, NEXTHOP, LABEL, PEER
 #    VRF   DST-ADDRESS    NEXTHOP     LABEL      PEER
 0 ID main  10.0.10.0/24               16         33.33.33.33:0
 1 ID main  11.11.11.11                21         33.33.33.33:0
 2 ID main  22.22.22.22                22         33.33.33.33:0
 3  D main  33.33.33.33    10.0.13.3   impl-null  33.33.33.33:0
 4  D main  44.44.44.44    10.0.13.3   23         33.33.33.33:0
 5 ID main  10.0.12.0/24               17         33.33.33.33:0
 6 ID main  10.0.13.0/24               impl-null  33.33.33.33:0
 7 ID main  10.0.20.0/24               18         33.33.33.33:0
 8 ID main  10.0.24.0/24               19         33.33.33.33:0
 9  D main  10.0.30.0/24   10.0.13.3   impl-null  33.33.33.33:0
10  D main  10.0.34.0/24   10.0.13.3   impl-null  33.33.33.33:0
11  D main  10.0.40.0/24   10.0.13.3   20         33.33.33.33:0
12 ID main  10.0.10.0/24               16         22.22.22.22:0
13 ID main  10.0.12.0/24               impl-null  22.22.22.22:0
14 ID main  10.0.13.0/24               17         22.22.22.22:0
15  D main  10.0.20.0/24   10.0.12.2   impl-null  22.22.22.22:0
16 ID main  11.11.11.11                21         22.22.22.22:0
17  D main  22.22.22.22    10.0.12.2   impl-null  22.22.22.22:0
18 ID main  33.33.33.33                22         22.22.22.22:0
19  D main  44.44.44.44    10.0.12.2   23         22.22.22.22:0
20  D main  10.0.24.0/24   10.0.12.2   impl-null  22.22.22.22:0
21 ID main  10.0.30.0/24               18         22.22.22.22:0
22 ID main  10.0.34.0/24               19         22.22.22.22:0
23  D main  10.0.40.0/24   10.0.12.2   20         22.22.22.22:0
[admin@MikroTik] /mpls/ldp/remote-mapping>
```

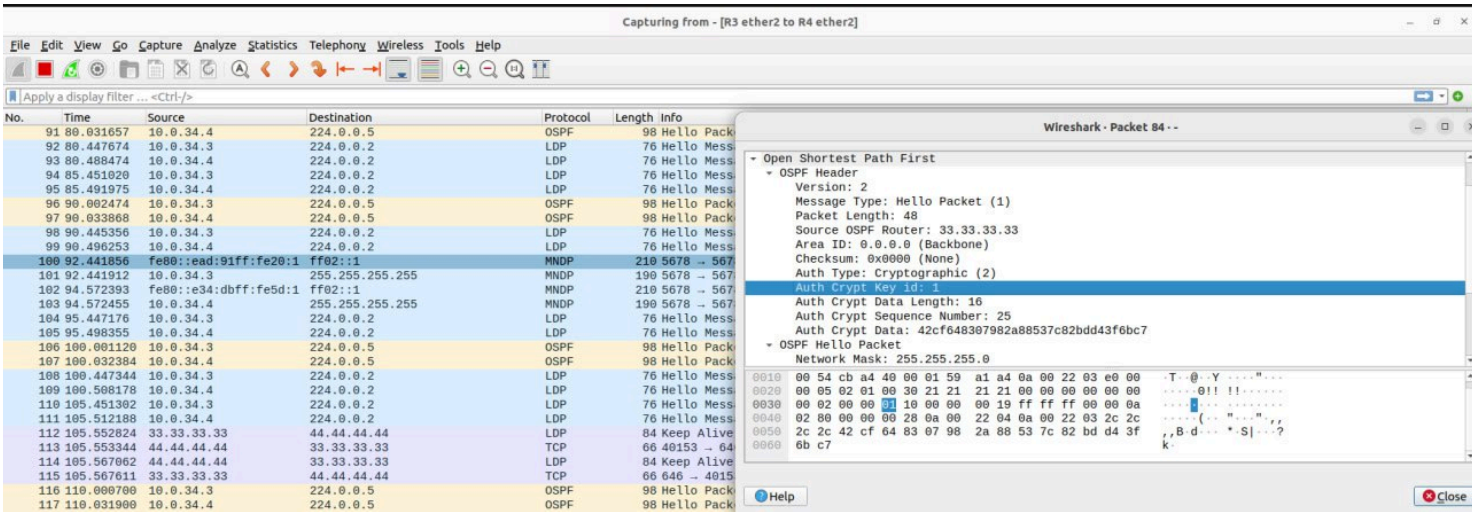
3. Enable authentication (what kind of authentication did you use)? Make sure that you can ping and trace all your network.

- I found no information on how to configure authentication for LDP's TCP packets (since they can be sniffed and corrupted) as in Cisco for example. It seems there is no such a possibility and nobody talks about it. However, we know that LDP uses OSPF for routing based on labels. Thus, if our OSPF hosts have other `auth-key`, these paths won't be distributed over all LDP peers.

- Therefore I configured `MD5` authentication on all the routers' interfaces

```
/routing ospf interface-template set numbers=0,1 auth=md5 auth-id=1 auth-key=password
```

- And we can see that it worked, the password was encrypted or better to say hashed



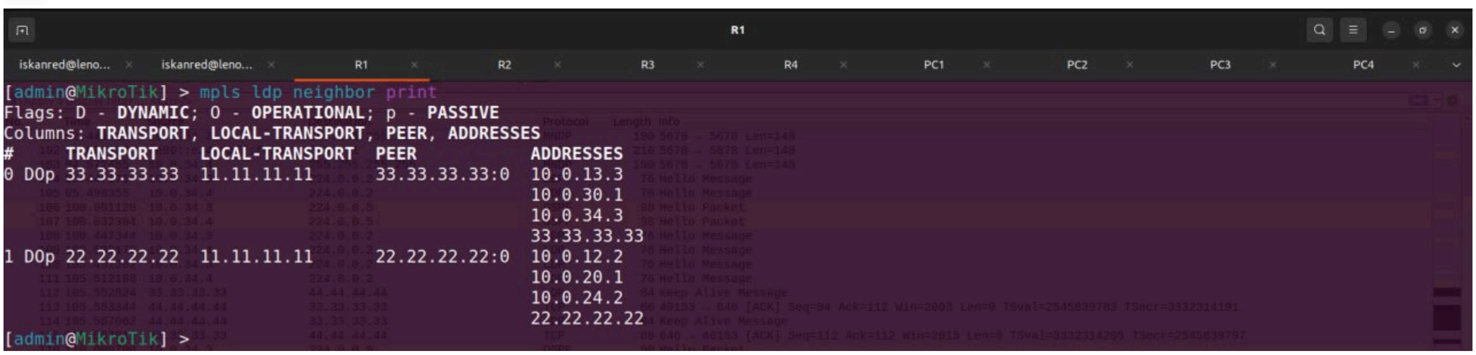
Task 2 - Verification

> **1.** Show your LDP neighbors.

- The command

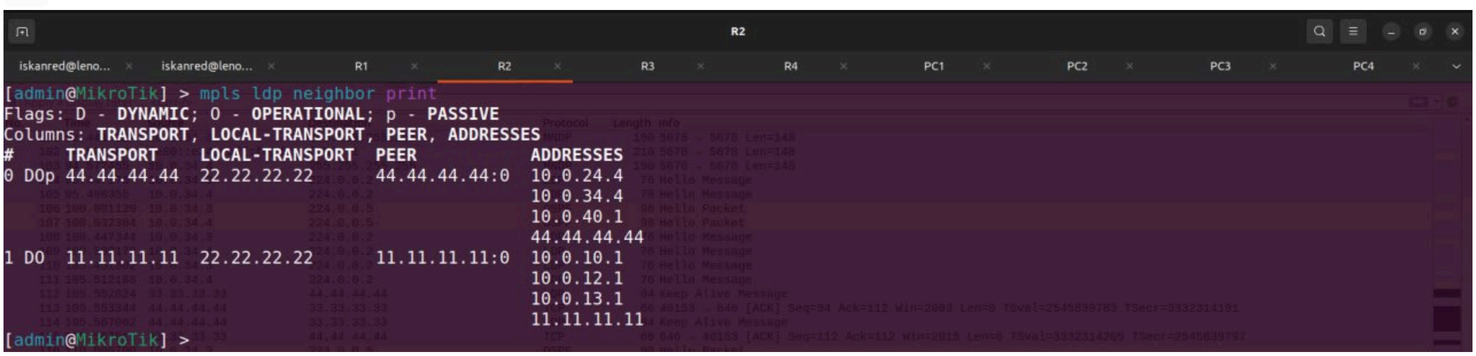
```
mpls ldp neighbor print
```

- R1



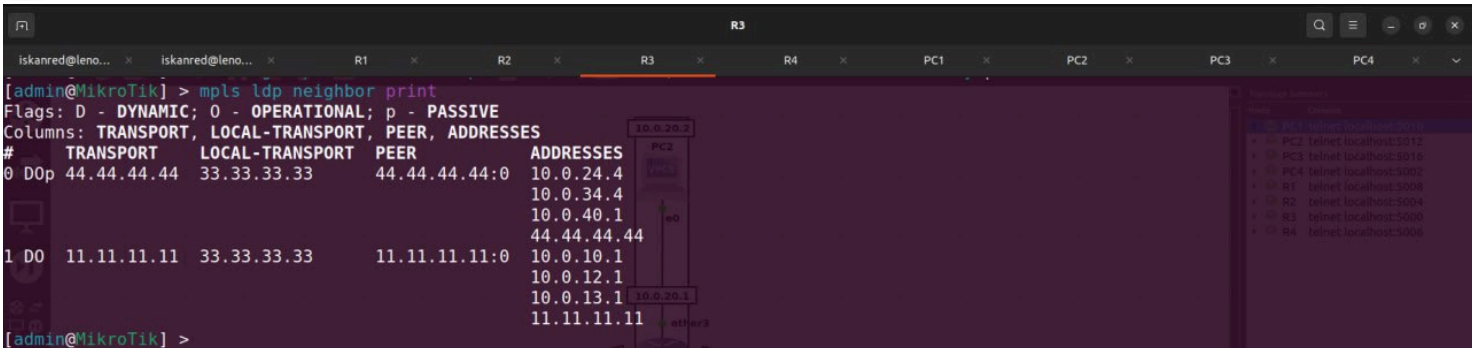
```
[admin@MikroTik] > mpls ldp neighbor print
Flags: D - DYNAMIC; O - OPERATIONAL; p - PASSIVE
Columns: TRANSPORT, LOCAL-TRANSPORT, PEER, ADDRESSES
#    TRANSPORT    LOCAL-TRANSPORT  PEER           ADDRESSES
0 DOp 33.33.33.33 11.11.11.11      33.33.33.33:0  10.0.13.3
                                                  10.0.30.1
                                                  10.0.34.3
                                                  33.33.33.33
1 DOp 22.22.22.22 11.11.11.11      22.22.22.22:0  10.0.12.2
                                                  10.0.20.1
                                                  10.0.24.2
                                                  22.22.22.22
[admin@MikroTik] >
```

- R2

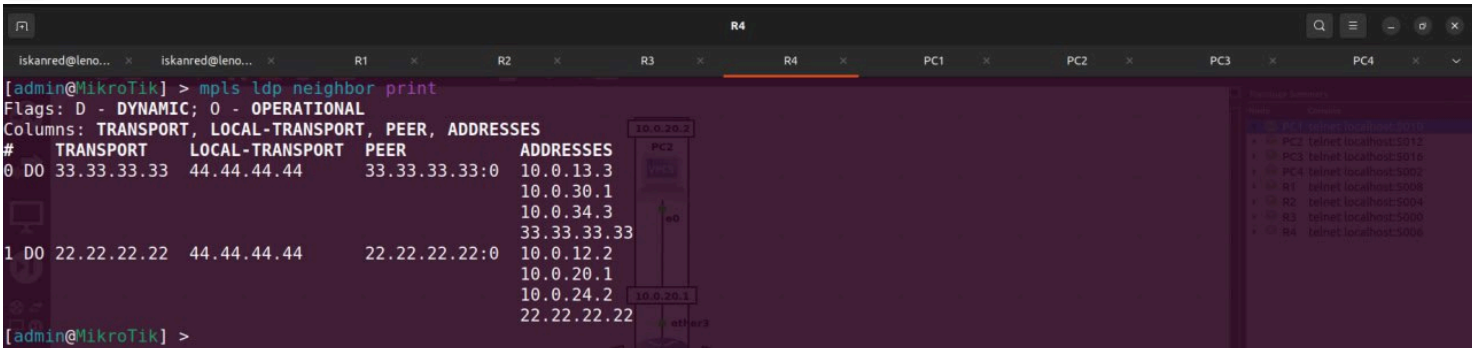


```
[admin@MikroTik] > mpls ldp neighbor print
Flags: D - DYNAMIC; O - OPERATIONAL; p - PASSIVE
Columns: TRANSPORT, LOCAL-TRANSPORT, PEER, ADDRESSES
#    TRANSPORT    LOCAL-TRANSPORT  PEER           ADDRESSES
0 DOp 44.44.44.44 22.22.22.22      44.44.44.44:0  10.0.24.4
                                                  10.0.34.4
                                                  10.0.40.1
                                                  44.44.44.44
1 DO  11.11.11.11 22.22.22.22      11.11.11.11:0  10.0.10.1
                                                  10.0.12.1
                                                  10.0.13.1
                                                  11.11.11.11
[admin@MikroTik] >
```

- R3



- R4



- We can conclude that the configuration was done correctly

> **2.** Show your local LDP bindings and remote LDP peer labels.

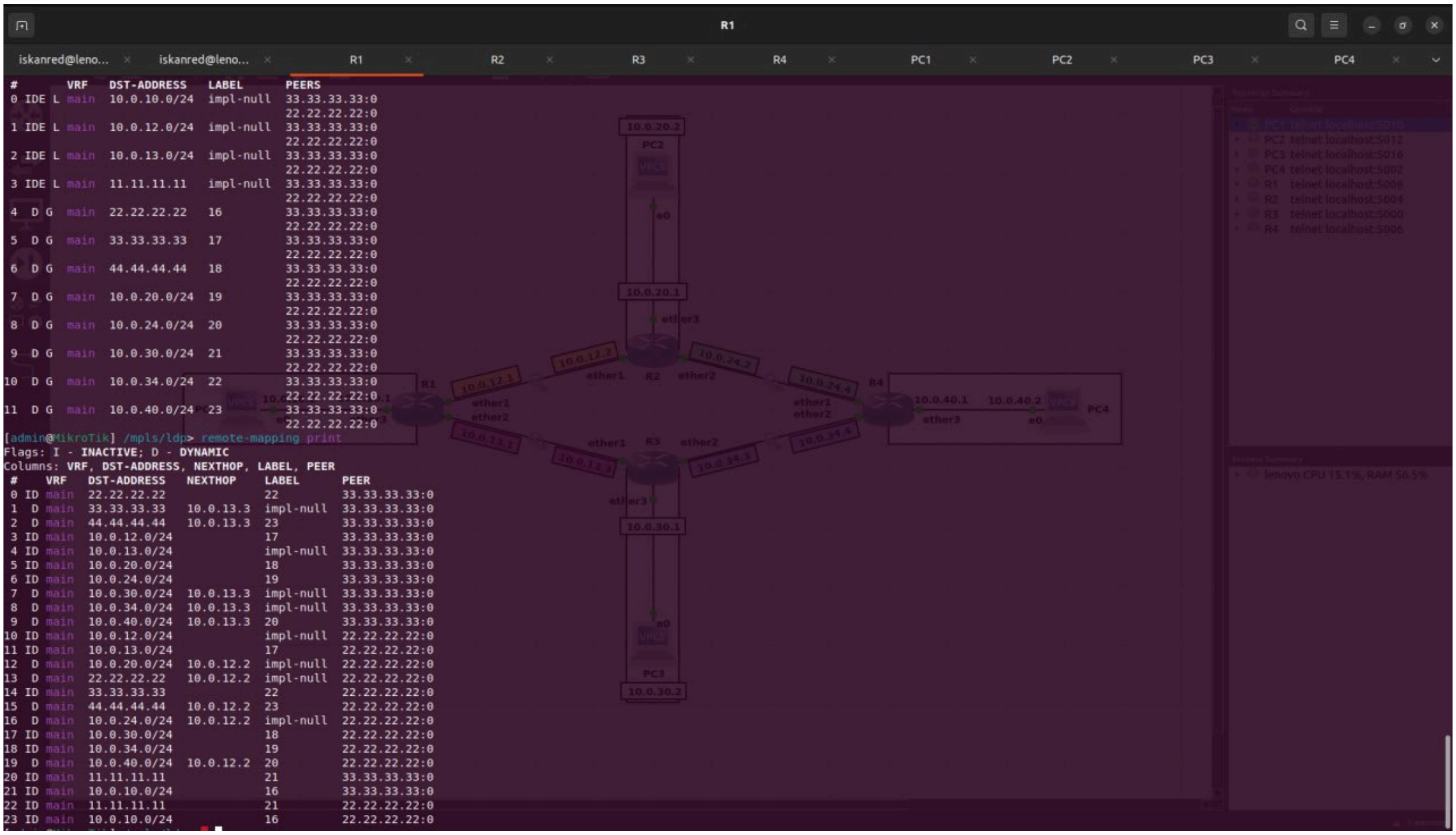
- The command for LDP local bindings

```
mpls ldp local-mapping print
```

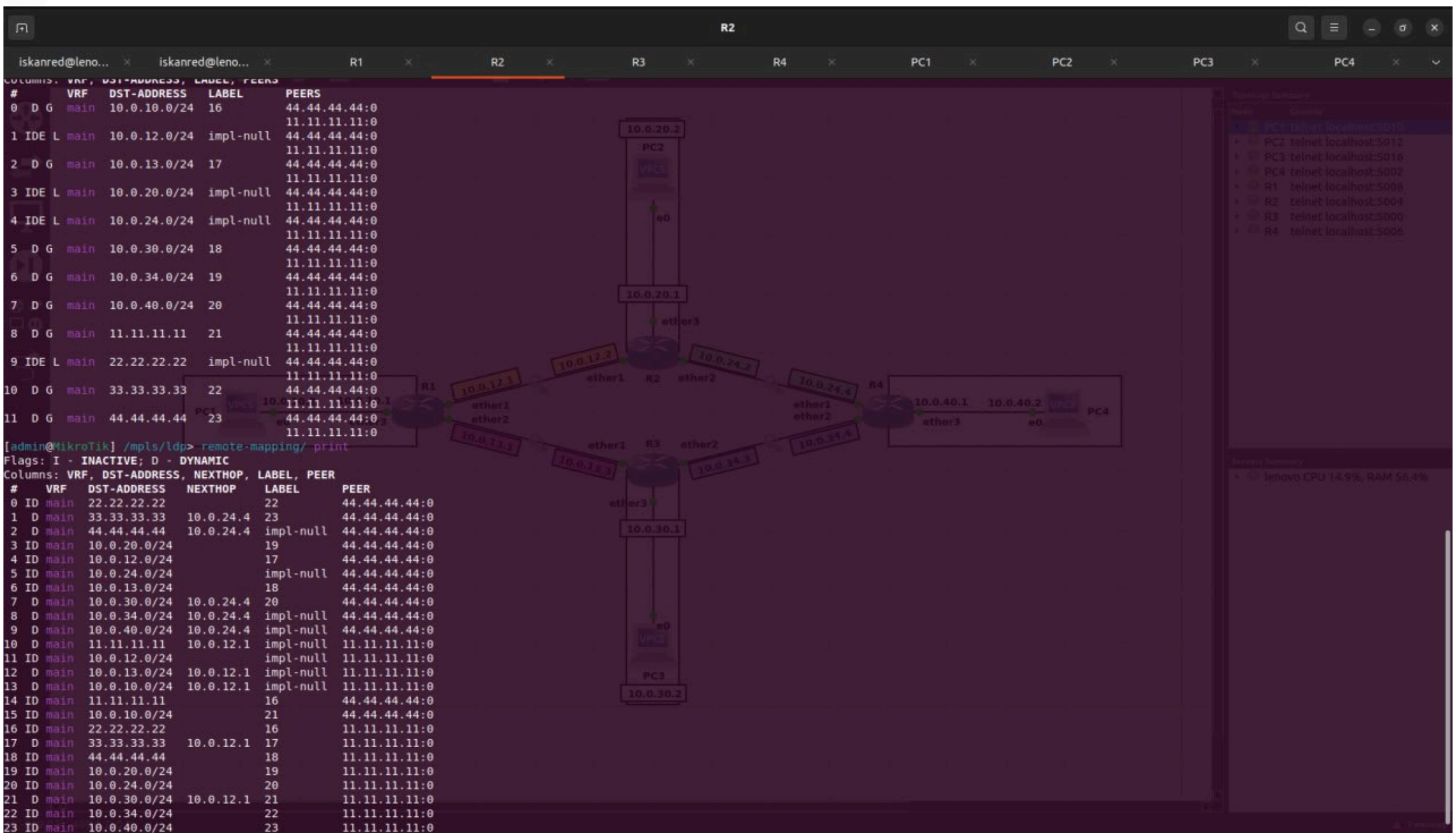
- The command for LDP remote peer labels

```
mpls ldp remote-mapping print
```

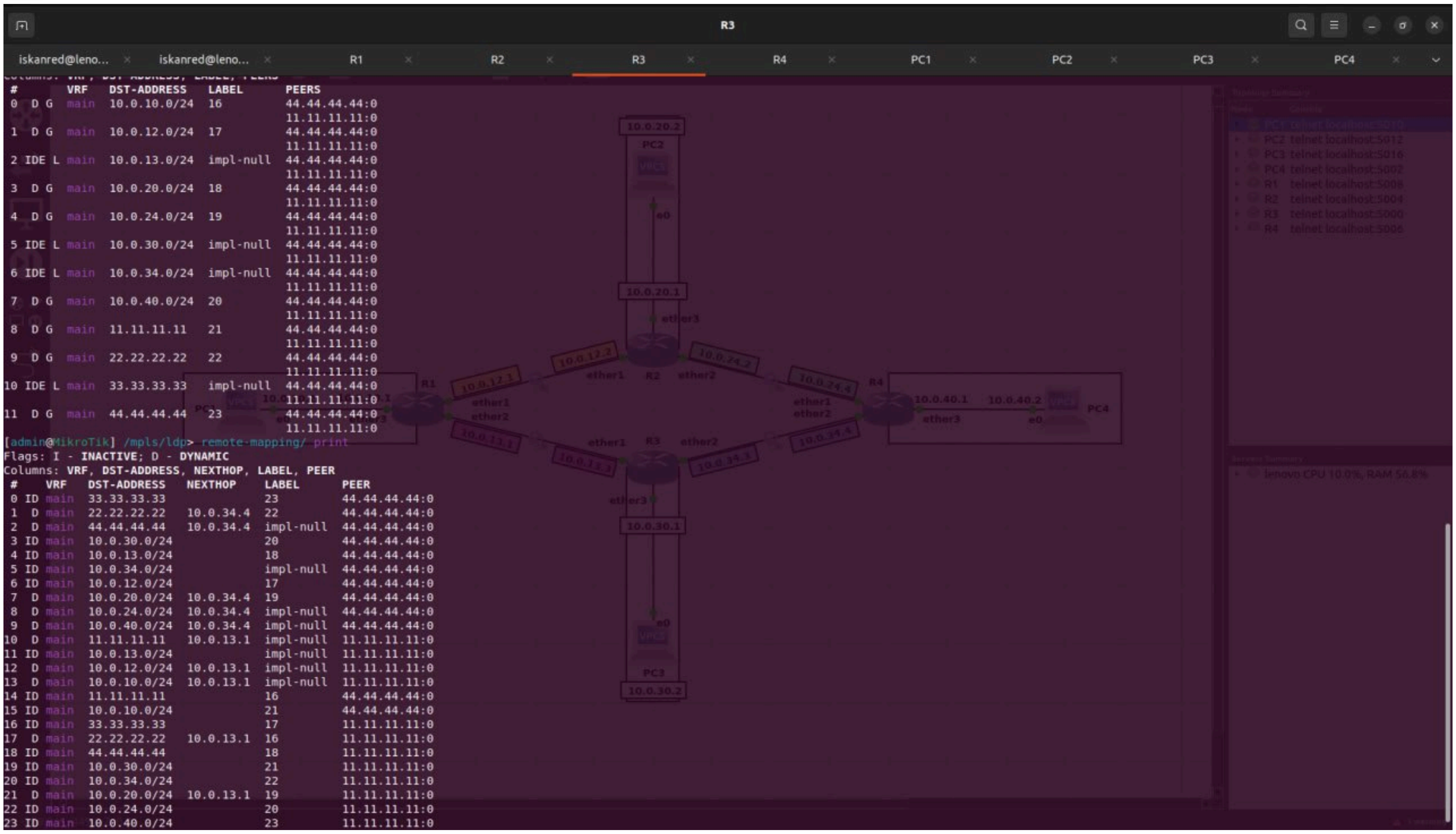
- R1


R1
iskanred@leno...
iskanred@leno...
R1
R2
R3
R4
PC1
PC2
PC3
PC4
VRF DST-ADDRESS LABEL PEERS
0 IDE L main 10.0.10.0/24 impl-null 33.33.33.33:0 22.22.22.22:0
1 IDE L main 10.0.12.0/24 impl-null 33.33.33.33:0 22.22.22.22:0
2 IDE L main 10.0.13.0/24 impl-null 33.33.33.33:0 22.22.22.22:0
3 IDE L main 11.11.11.11 impl-null 33.33.33.33:0 22.22.22.22:0
4 D G main 22.22.22.22 16 33.33.33.33:0 22.22.22.22:0
5 D G main 33.33.33.33 17 33.33.33.33:0 22.22.22.22:0
6 D G main 44.44.44.44 18 33.33.33.33:0 22.22.22.22:0
7 D G main 10.0.20.0/24 19 33.33.33.33:0 22.22.22.22:0
8 D G main 10.0.24.0/24 20 33.33.33.33:0 22.22.22.22:0
9 D G main 10.0.30.0/24 21 33.33.33.33:0 22.22.22.22:0
10 D G main 10.0.34.0/24 22 33.33.33.33:0 22.22.22.22:0
11 D G main 10.0.40.0/24 23 33.33.33.33:0 22.22.22.22:0
[admin@MikroTik] /mpls/ldp> remote-mapping print
Flags: I - INACTIVE; D - DYNAMIC
Columns: VRF, DST-ADDRESS, NEXTHOP, LABEL, PEER
VRF DST-ADDRESS NEXTHOP LABEL PEER
0 ID main 22.22.22.22 22 33.33.33.33:0
1 D main 33.33.33.33 10.0.13.3 impl-null 33.33.33.33:0
2 D main 44.44.44.44 10.0.13.3 23 33.33.33.33:0
3 ID main 10.0.12.0/24 17 33.33.33.33:0
4 ID main 10.0.13.0/24 impl-null 33.33.33.33:0
5 ID main 10.0.20.0/24 18 33.33.33.33:0
6 ID main 10.0.24.0/24 19 33.33.33.33:0
7 D main 10.0.30.0/24 10.0.13.3 impl-null 33.33.33.33:0
8 D main 10.0.34.0/24 10.0.13.3 impl-null 33.33.33.33:0
9 D main 10.0.40.0/24 10.0.13.3 20 33.33.33.33:0
10 ID main 10.0.12.0/24 impl-null 22.22.22.22:0
11 ID main 10.0.13.0/24 17 22.22.22.22:0
12 D main 10.0.20.0/24 10.0.12.2 impl-null 22.22.22.22:0
13 D main 22.22.22.22 10.0.12.2 impl-null 22.22.22.22:0
14 ID main 33.33.33.33 22 22.22.22.22:0
15 D main 44.44.44.44 10.0.12.2 23 22.22.22.22:0
16 D main 10.0.24.0/24 10.0.12.2 impl-null 22.22.22.22:0
17 ID main 10.0.30.0/24 18 22.22.22.22:0
18 ID main 10.0.34.0/24 19 22.22.22.22:0
19 D main 10.0.40.0/24 10.0.12.2 20 22.22.22.22:0
20 ID main 11.11.11.11 21 33.33.33.33:0
21 ID main 10.0.10.0/24 16 33.33.33.33:0
22 ID main 11.11.11.11 21 22.22.22.22:0
23 ID main 10.0.10.0/24 16 22.22.22.22:0

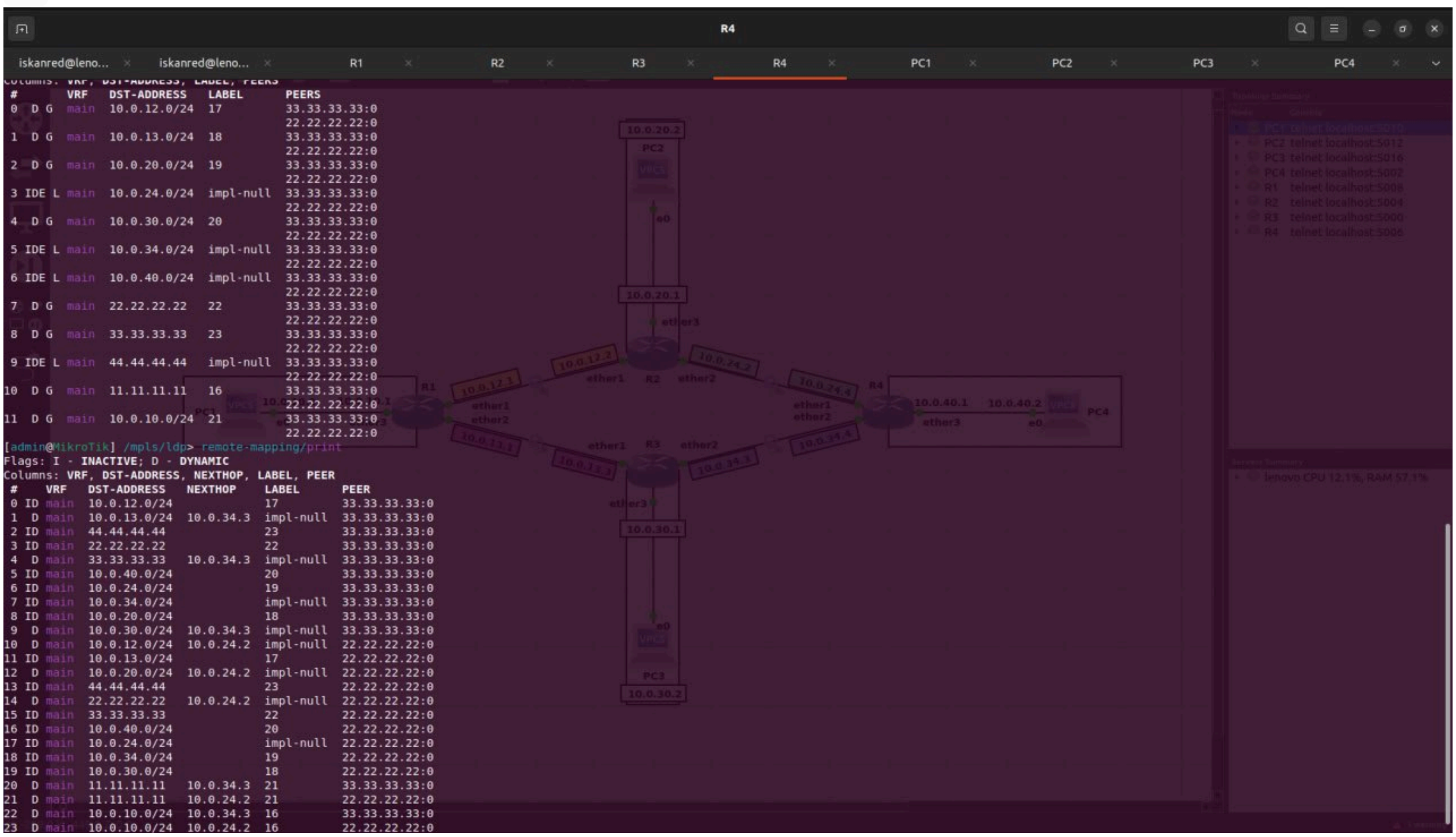
- R2


R2
iskanred@leno...
iskanred@leno...
R1
R2
R3
R4
PC1
PC2
PC3
PC4
VRF DST-ADDRESS LABEL PEERS
0 D G main 10.0.10.0/24 16 44.44.44.44:0 11.11.11.11:0
1 IDE L main 10.0.12.0/24 impl-null 44.44.44.44:0 11.11.11.11:0
2 D G main 10.0.13.0/24 17 44.44.44.44:0 11.11.11.11:0
3 IDE L main 10.0.20.0/24 impl-null 44.44.44.44:0 11.11.11.11:0
4 IDE L main 10.0.24.0/24 impl-null 44.44.44.44:0 11.11.11.11:0
5 D G main 10.0.30.0/24 18 44.44.44.44:0 11.11.11.11:0
6 D G main 10.0.34.0/24 19 44.44.44.44:0 11.11.11.11:0
7 D G main 10.0.40.0/24 20 44.44.44.44:0 11.11.11.11:0
8 D G main 11.11.11.11 21 44.44.44.44:0 11.11.11.11:0
9 IDE L main 22.22.22.22 impl-null 44.44.44.44:0 11.11.11.11:0
10 D G main 33.33.33.33 22 44.44.44.44:0 11.11.11.11:0
11 D G main 44.44.44.44 23 44.44.44.44:0 11.11.11.11:0
[admin@MikroTik] /mpls/ldp> remote-mapping/ print
Flags: I - INACTIVE; D - DYNAMIC
Columns: VRF, DST-ADDRESS, NEXTHOP, LABEL, PEER
VRF DST-ADDRESS NEXTHOP LABEL PEER
0 ID main 22.22.22.22 22 44.44.44.44:0
1 D main 33.33.33.33 10.0.24.4 23 44.44.44.44:0
2 D main 44.44.44.44 10.0.24.4 impl-null 44.44.44.44:0
3 ID main 10.0.20.0/24 19 44.44.44.44:0
4 ID main 10.0.12.0/24 17 44.44.44.44:0
5 ID main 10.0.24.0/24 impl-null 44.44.44.44:0
6 ID main 10.0.13.0/24 18 44.44.44.44:0
7 D main 10.0.30.0/24 10.0.24.4 20 44.44.44.44:0
8 D main 10.0.34.0/24 10.0.24.4 impl-null 44.44.44.44:0
9 D main 10.0.40.0/24 10.0.24.4 impl-null 44.44.44.44:0
10 D main 11.11.11.11 10.0.12.1 impl-null 11.11.11.11:0
11 ID main 10.0.12.0/24 impl-null 11.11.11.11:0
12 D main 10.0.13.0/24 10.0.12.1 impl-null 11.11.11.11:0
13 D main 10.0.10.0/24 10.0.12.1 impl-null 11.11.11.11:0
14 ID main 11.11.11.11 16 44.44.44.44:0
15 ID main 10.0.10.0/24 21 44.44.44.44:0
16 ID main 22.22.22.22 16 11.11.11.11:0
17 D main 33.33.33.33 10.0.12.1 17 11.11.11.11:0
18 ID main 44.44.44.44 18 11.11.11.11:0
19 ID main 10.0.20.0/24 19 11.11.11.11:0
20 ID main 10.0.24.0/24 20 11.11.11.11:0
21 D main 10.0.30.0/24 10.0.12.1 21 11.11.11.11:0
22 ID main 10.0.34.0/24 22 11.11.11.11:0
23 ID main 10.0.40.0/24 23 11.11.11.11:0

- R3



- R4



- We can see that we have `impl-null` which implies PHP from `R2` & `R3` to `R4` going from `R1` ingress and vice a versa: from `R2` & `R3` to `R1` going over from `R4` ingress.

> **3.** Show your MPLS labels.

W

The labels are:

- `impl-null`
- 16
- 17
- 18
- 19
- 20
- 21
- 22
- 23

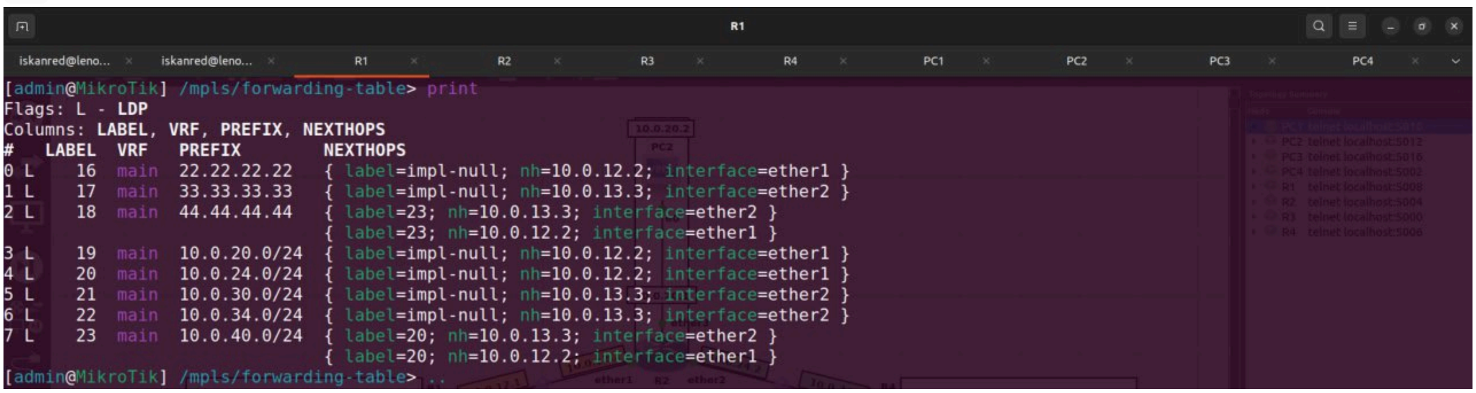
  We see that for each FEC (Forwarding Equivalence Class) or simply subnet there is a unique label

> **4.** Show your forwarding table.

- The command

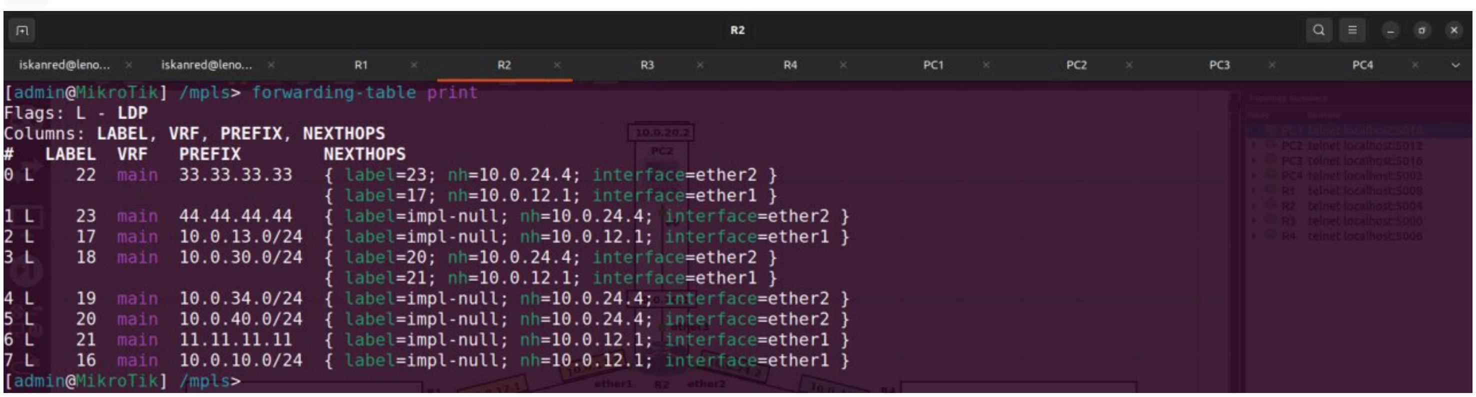
```
/mpls forwarding-table print
```

- `R1`



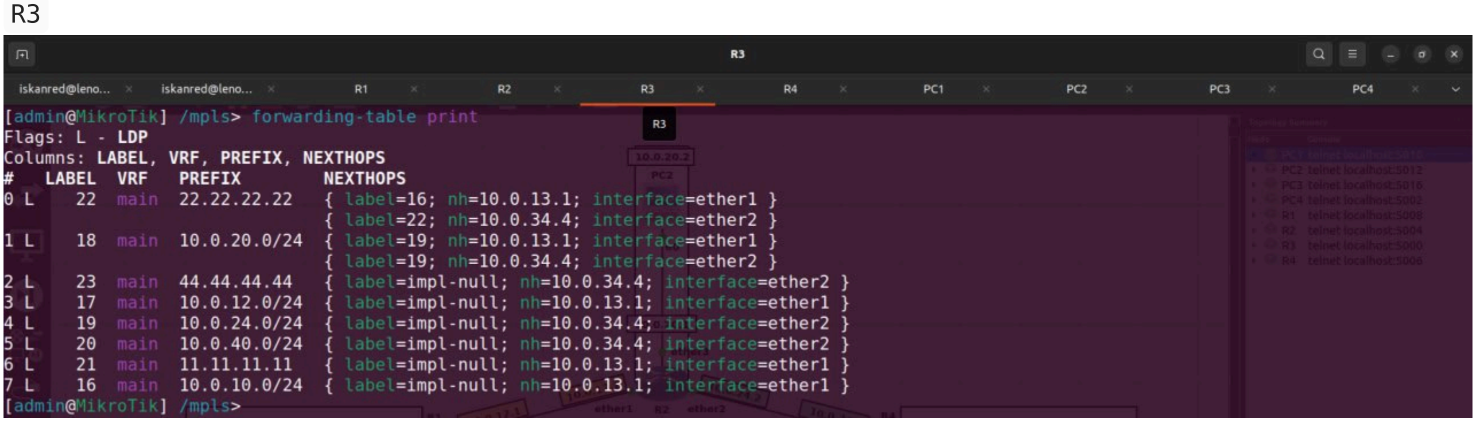
```
[admin@MikroTik] /mpls/forwarding-table> print
Flags: L - LDP
Columns: LABEL, VRF, PREFIX, NEXTHOPS
#   LABEL  VRF   PREFIX        NEXTHOPS
0 L    16  main  22.22.22.22   { label=impl-null; nh=10.0.12.2; interface=ether1 }
1 L    17  main  33.33.33.33   { label=impl-null; nh=10.0.13.3; interface=ether2 }
2 L    18  main  44.44.44.44   { label=23; nh=10.0.13.3; interface=ether2 }
                               { label=23; nh=10.0.12.2; interface=ether1 }
3 L    19  main  10.0.20.0/24  { label=impl-null; nh=10.0.12.2; interface=ether1 }
4 L    20  main  10.0.24.0/24  { label=impl-null; nh=10.0.12.2; interface=ether1 }
5 L    21  main  10.0.30.0/24  { label=impl-null; nh=10.0.13.3; interface=ether2 }
6 L    22  main  10.0.34.0/24  { label=impl-null; nh=10.0.13.3; interface=ether2 }
7 L    23  main  10.0.40.0/24  { label=20; nh=10.0.13.3; interface=ether2 }
                               { label=20; nh=10.0.12.2; interface=ether1 }
[admin@MikroTik] /mpls/forwarding-table>
```

- `R2`



```
[admin@MikroTik] /mpls> forwarding-table print
Flags: L - LDP
Columns: LABEL, VRF, PREFIX, NEXTHOPS
#   LABEL  VRF   PREFIX        NEXTHOPS
0 L    22  main  33.33.33.33   { label=23; nh=10.0.24.4; interface=ether2 }
                               { label=17; nh=10.0.12.1; interface=ether1 }
1 L    23  main  44.44.44.44   { label=impl-null; nh=10.0.24.4; interface=ether2 }
2 L    17  main  10.0.13.0/24  { label=impl-null; nh=10.0.12.1; interface=ether1 }
3 L    18  main  10.0.30.0/24  { label=20; nh=10.0.24.4; interface=ether2 }
                               { label=21; nh=10.0.12.1; interface=ether1 }
4 L    19  main  10.0.34.0/24  { label=impl-null; nh=10.0.24.4; interface=ether2 }
5 L    20  main  10.0.40.0/24  { label=impl-null; nh=10.0.24.4; interface=ether2 }
6 L    21  main  11.11.11.11   { label=impl-null; nh=10.0.12.1; interface=ether1 }
7 L    16  main  10.0.10.0/24  { label=impl-null; nh=10.0.12.1; interface=ether1 }
[admin@MikroTik] /mpls>
```

- `R3`



```
[admin@MikroTik] /mpls> forwarding-table print
Flags: L - LDP
Columns: LABEL, VRF, PREFIX, NEXTHOPS
#   LABEL  VRF   PREFIX        NEXTHOPS
0 L    22  main  22.22.22.22   { label=16; nh=10.0.13.1; interface=ether1 }
                               { label=22; nh=10.0.34.4; interface=ether2 }
1 L    18  main  10.0.20.0/24  { label=19; nh=10.0.13.1; interface=ether1 }
                               { label=19; nh=10.0.34.4; interface=ether2 }
2 L    23  main  44.44.44.44   { label=impl-null; nh=10.0.34.4; interface=ether2 }
3 L    17  main  10.0.12.0/24  { label=impl-null; nh=10.0.13.1; interface=ether1 }
4 L    19  main  10.0.24.0/24  { label=impl-null; nh=10.0.34.4; interface=ether2 }
5 L    20  main  10.0.40.0/24  { label=impl-null; nh=10.0.34.4; interface=ether2 }
6 L    21  main  11.11.11.11   { label=impl-null; nh=10.0.13.1; interface=ether1 }
7 L    16  main  10.0.10.0/24  { label=impl-null; nh=10.0.13.1; interface=ether1 }
[admin@MikroTik] /mpls>
```

- R4

```
[admin@MikroTik] /mpls> forwarding-table print
Flags: L - LDP
Columns: LABEL, VRF, PREFIX, NEXTHOPS
#   LABEL  VRF   PREFIX        NEXTHOPS
0 L    17  main  10.0.12.0/24  { label=impl-null; nh=10.0.24.2; interface=ether1 }
1 L    18  main  10.0.13.0/24  { label=impl-null; nh=10.0.34.3; interface=ether2 }
2 L    23  main  33.33.33.33   { label=impl-null; nh=10.0.34.3; interface=ether2 }
3 L    20  main  10.0.30.0/24  { label=impl-null; nh=10.0.34.3; interface=ether2 }
4 L    19  main  10.0.20.0/24  { label=impl-null; nh=10.0.24.2; interface=ether1 }
5 L    22  main  22.22.22.22   { label=impl-null; nh=10.0.24.2; interface=ether1 }
6 L    16  main  11.11.11.11   { label=21; nh=10.0.34.3; interface=ether2 }
                               { label=21; nh=10.0.24.2; interface=ether1 }
7 L    21  main  10.0.10.0/24  { label=16; nh=10.0.34.3; interface=ether2 }
                               { label=16; nh=10.0.24.2; interface=ether1 }
[admin@MikroTik] /mpls>
```

> **5.** Show your network path from one customer edge to the other customer edge.

- The command

```
/tool traceroute {dst-address} src={src-address}
```

- `10.0.10.1` (`R1 - ether3`) to `10.0.40.2` (`PC4`)

```
[admin@MikroTik] > tool traceroute 10.0.40.2 src=10.0.10.1
Columns: ADDRESS, LOSS, SENT, LAST, AVG, BEST, WORST, STD-DEV, STATUS
#  ADDRESS    LOSS  SENT  LAST   AVG  BEST  WORST  STD-DEV  STATUS
1  10.0.13.3  0%       4  1.7ms  1.8  1.5   2      0.2      <MPLS:L=20,E=0>
2  10.0.34.4  0%       4  1.3ms  1.4  1.2   1.5    0.1
3  10.0.40.2  0%       4  1.9ms  1.9  1.5   2.2    0.3

[admin@MikroTik] >
```

- We see MPLS works and the label used to transmit packets from `R1` to `R3` to reach `10.0.40.2` $= 20$. Then `R3` strips MPLS header and sends it right to `R4`. Then `R4` transmits traffic to `PC4` using IP.
- `10.0.10.1` (`R1 - ether3`) to `44.44.44.44` (`R4 - lo`)

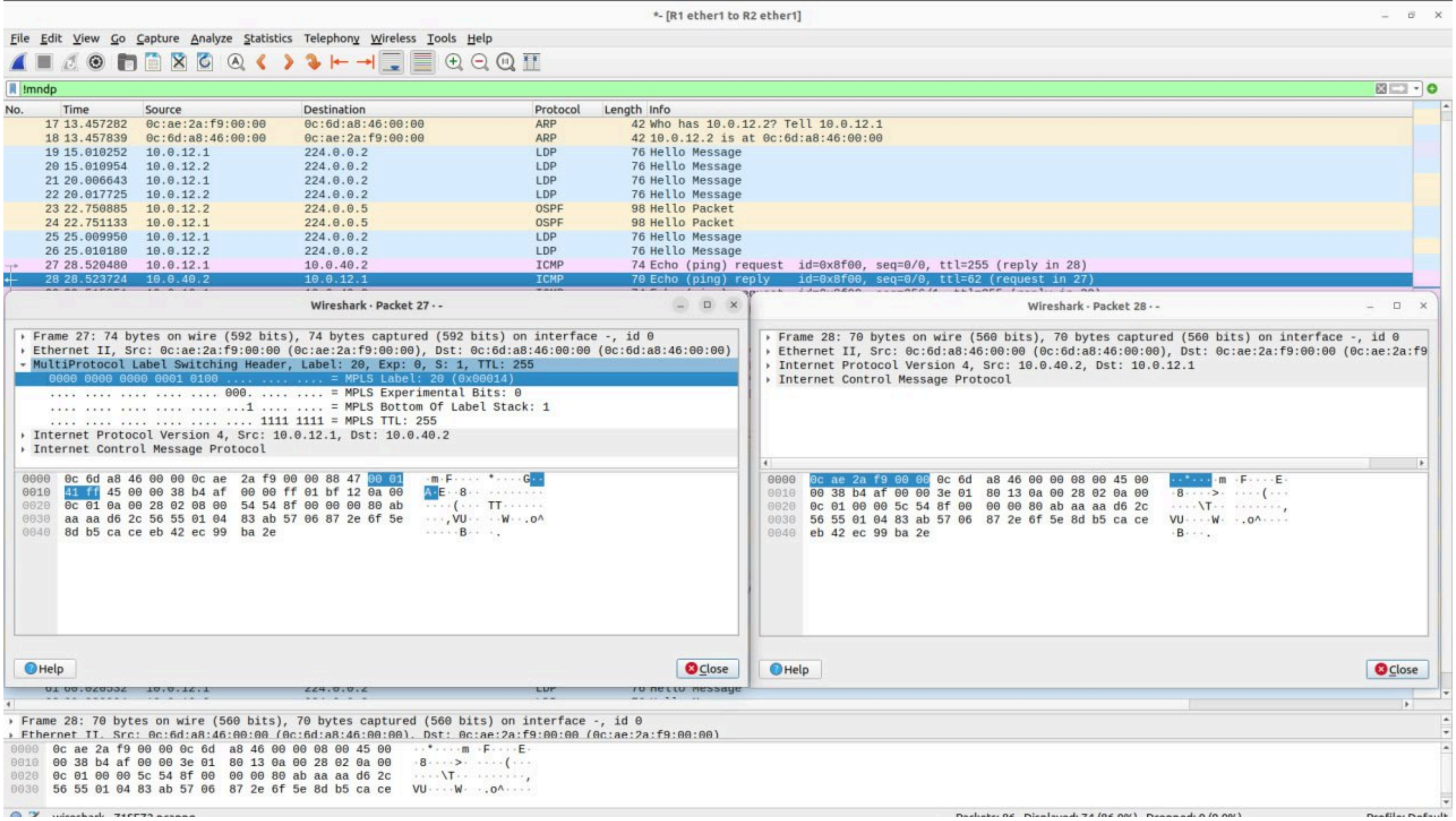
```
[admin@MikroTik] > tool traceroute 44.44.44.44 src=10.0.10.1
Columns: ADDRESS, LOSS, SENT, LAST, AVG, BEST, WORST, STD-DEV, STATUS
#  ADDRESS      LOSS  SENT  LAST   AVG  BEST  WORST  STD-DEV  STATUS
1  10.0.12.2    0%       1  2ms    2    2     2            0  <MPLS:L=23,E=0>
2  44.44.44.44  0%       1  1.5ms  1.5  1.5   1.5          0

[admin@MikroTik] >
[admin@MikroTik] >
```

- We see MPLS works and the label used to transmit packets from `R1` to `R2` to reach `44.44.44.44` $= 23$. Then `R2` strips MPLS header and send it right to `R4`.

Task 3 - MPLS packets analysis

> **1.** Can you use Wireshark to see the MPLS packets?

- Yes, e.g. link `R1` - `R2` (`10.0.13.0/24`). I did ping and we can see ICMP request on the left and ICMP reply on the right



- The key moment here is that for ICMP request `R1` gives label $= 20$ and we can see it in MPLS header. However, in the reply PHP was occurred and `R2` stripped MPLS header before transmitting it from `R4` to `R1`.
- We see that
  - Label is **20**
  - This label is the **bottom** of MPLS labels stack
  - Time To Live is **255** hops
  - And strange experimental bits which is actually a QoS field

    > The use of "experimental" bits is not specified by MPLS standards, but the most common use is to carry QoS information, similar to 802.1q priority in the VLAN tag. Note that the EXP field is 3 bits only therefore it can carry values from 0 to 7 only, which allows having 8 traffic classes.

> **2.** Look deeper into the MPLS packets: can you identify MAC address, ICMP, Ethernet header or something else useful?

In Wireshark I can identify anything actually and we can see it on the pictures above

Task 4 - VPLS

> **1.** Configure VPLS between the 2 hosts edges.

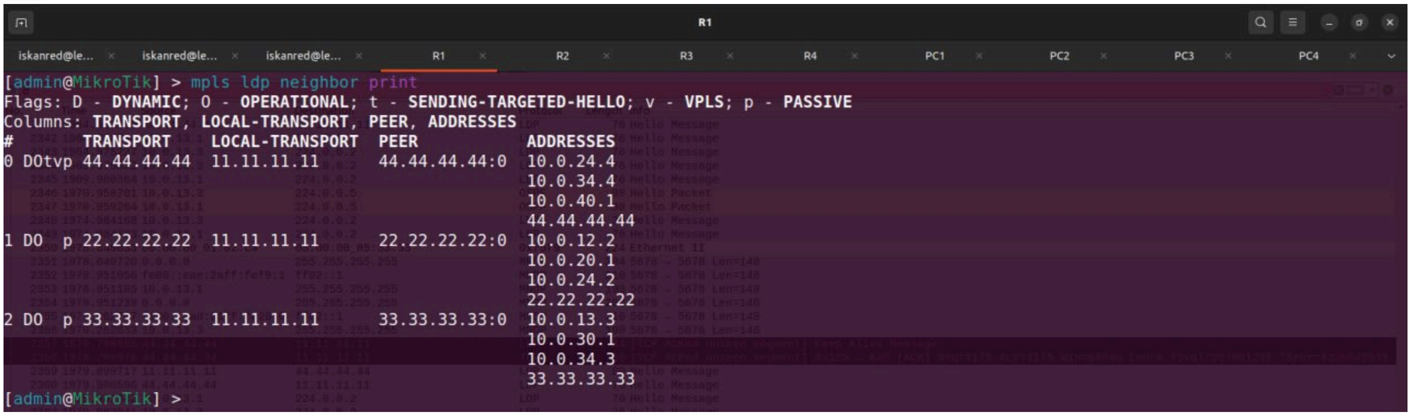
- I decided to configure VPLS between `R1` and `R4`
- The command

```
# R1
/interface vpls add name="1-to-4" peer=44.44.44.44 vpls-id=1:1
```

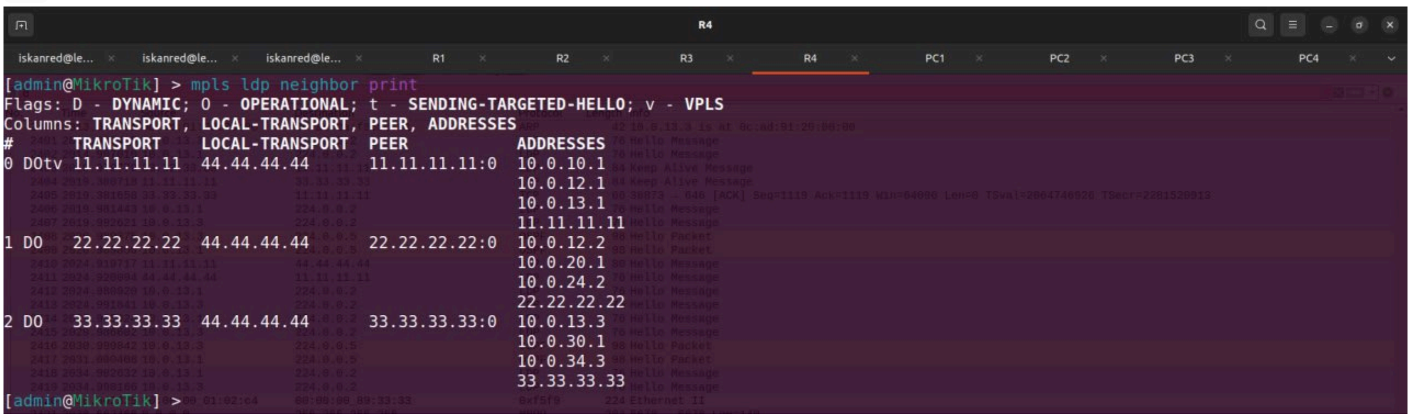
```
#R2
/interface vpls add name="4-to-1" peer=11.11.11.11 vpls-id=1:1
```

> **2.** Show your LDP neighbors again, what has been changed?

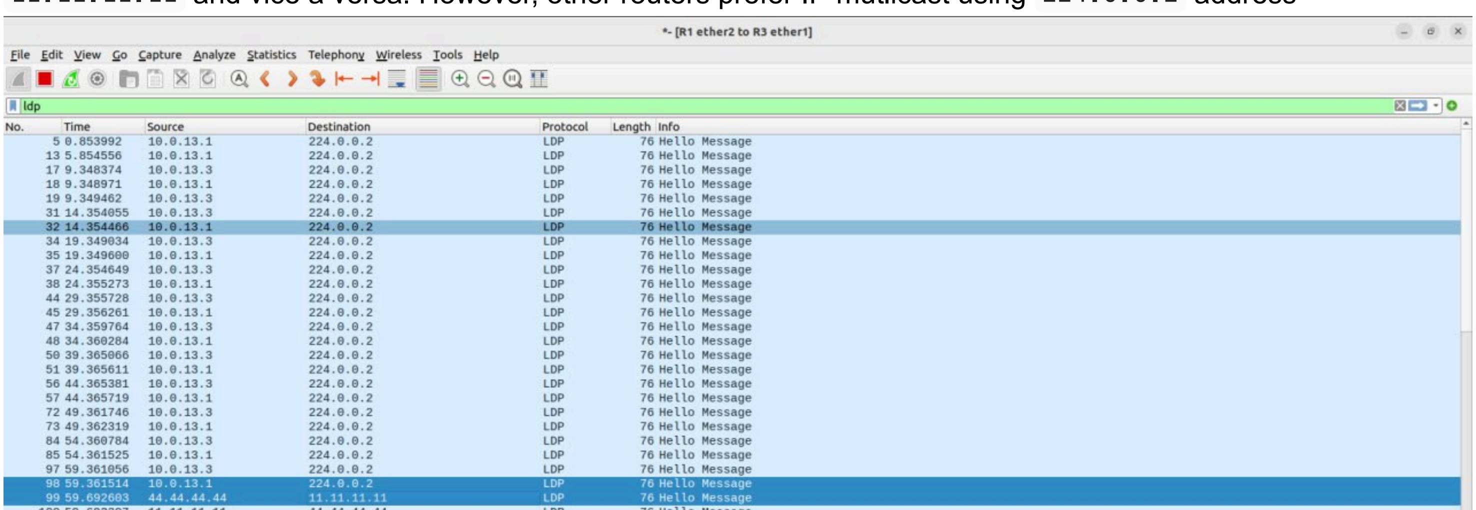
- Let's check neighbours table on our VPLS-connected routers
  - R1



  - R4



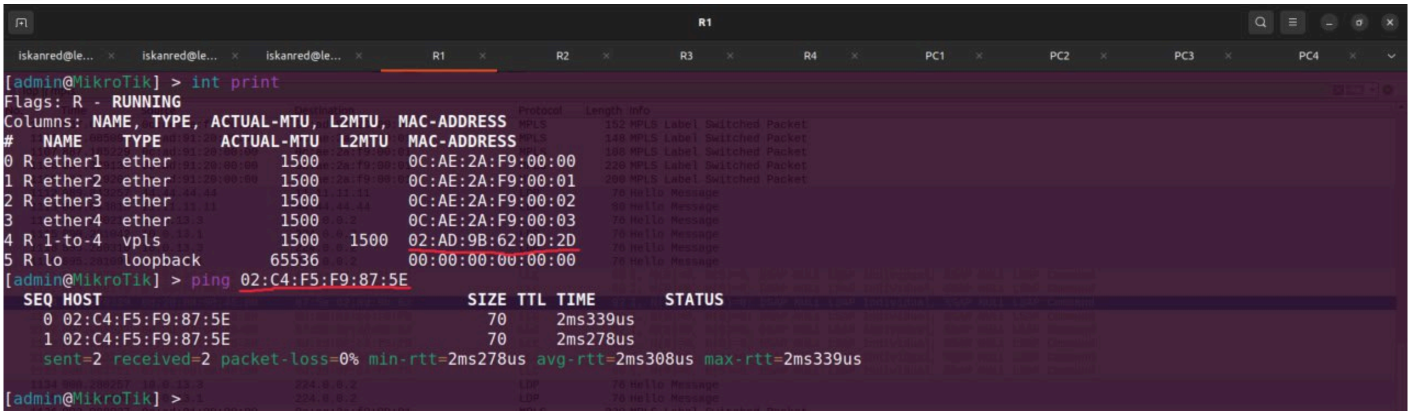
- Now we can notice that `R1` is a neighbour of `R4` and vice a versa, but earlier this was not true.
- What is also noticeable is that we see here `t – SENDING–TARGETED–HELLO` and `v – VPLS` flags.
- `t – SENDING–TARGETED–HELLO` means that LDP `Hello` message will be sent exactly from `44.44.44.44` to `11.11.11.11` and vice a versa. However, other routers prefer IP mutlicast using `224.0.0.2` address



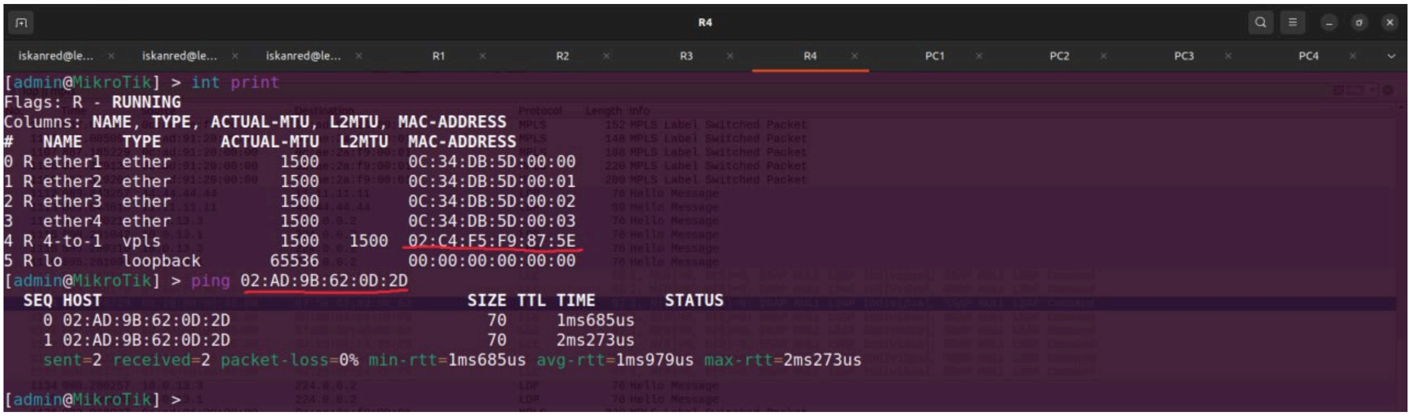
> **3.** Find a way to prove that the two customers can communicate at OSI layer 2.

- We can use "Layer 2 ping" or LLC protocol

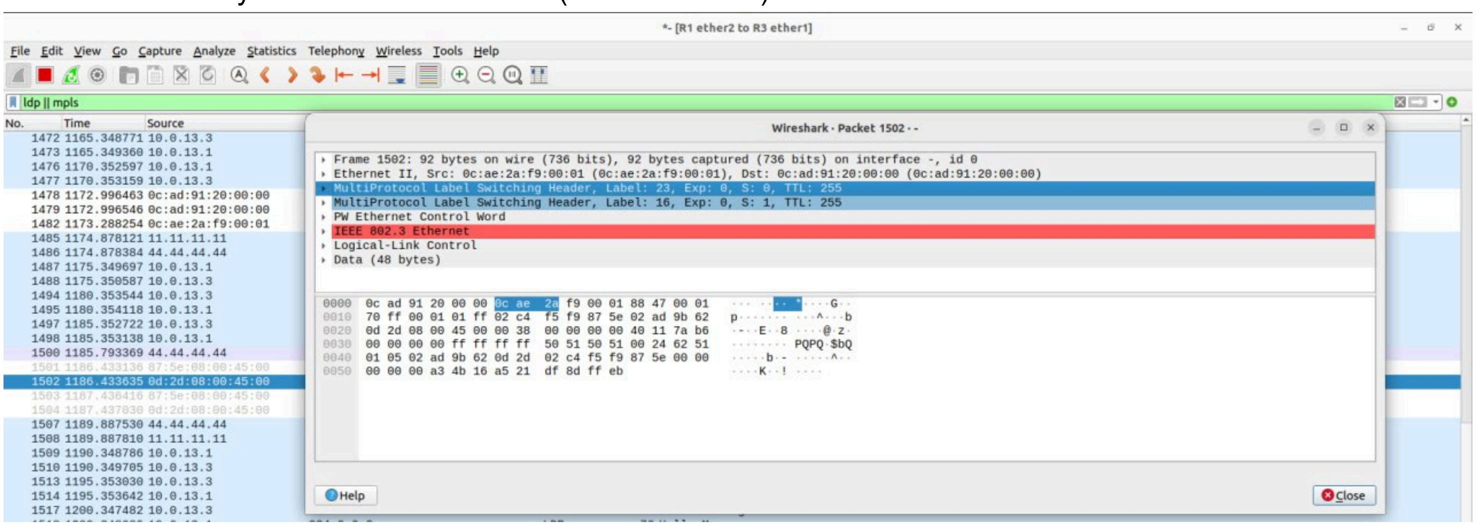
- `R1` to `R4` through the VPLS interface



- `R4` to `R1` through the VPLS interface



- And we see exactly the stack of labels now (16 and then 23)



> **4.** Is it required to disable PHP? Explain your answer.

I see 2 cases where it can be needed:

1. In MPLS networks where QoS is used. In these cases it may be needed that QoS won't be removed from penultimate hop to the next because QoS type is stored in MPLS header (also, in IP, but penultimate routers will not check IP header transmitting it to the destination).
2. In MPLS networks where label stacking is used (e.g., when using MPLS in Layer 2 VPNs or L3 VPNs), it might be necessary to disable PHP to ensure that the entire label stack is maintained until the packet reaches the appropriate egress point. This is important for proper label processing.

References

- https://en.wikipedia.org/wiki/Multiprotocol_Label_Switching
- https://help.mikrotik.com/docs/spaces/ROS/pages/40992794/MPLS+Overview
- https://www.cisco.com/c/dam/en/us/td/docs/ios/12_0/12_0sy/feature/guide/md5orig.pdf