# Redes de Computadores

## Phase 2

# Connecting Devices (Using LAN A and B only)

| | |
|---|---|
| N. 49423 | Rafael Pegacho |
| N. 49431 | Tiago Neto |

Licenciatura em Engenharia Informática e de Computadores

Semestre de Verão 2021/2022

13/05/2022

# Contents

# 1 Introduction

In this phase, we will be mainly configuring devices in 2 LANs (Lan A and Lan B) using the EVE-NG emulator running on a web browser and a Telnet client.

The topology is already provided to us, so we only need to configure and test the devices on both LANs.

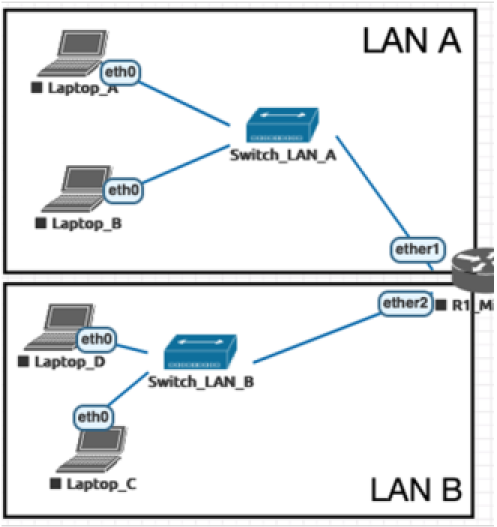


Figure 1: Lan A and Lan B Topology

# 2 Reconnaissance

Before getting started with the configurations, we need to take a look at the topology and gather as much information as we can from it.

When we're looking into the topology we aim to configure, we first have to ask ourselves these questions:

- How many IP's do I need for each LAN?
- How can I **subnet** each Network, giving It an address space that wastes as few IP addresses as possible?
- Can IP's be randomly given?
- Do IP addressing have rules?
- What If I'm trying to communicate with an IP outside of my Host network?

## 2.1 Needed IP addresses

As we can see in Figure 2 we have four machines in Lan A, two laptops, one Switch and one Router.

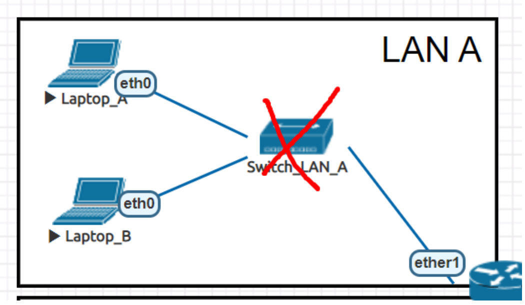


Figure 2: Lan A Topology

Do we actually need an IP for each of them? Well, the answer is ”No.” Switches aren't able to have IP addresses (we will see why soon).

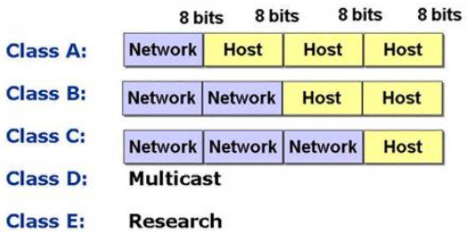
With that said, we only need one IP for each of the laptops; one for the router interface that both laptops are connected to; and as in every network, we need an IP for the broadcast and the network.

In total, we will need 10 IP addresses as soon as they have similar topologies, and we've just seen that Lan A needs 5 of them.

## 2.2 Subnetting Networks

Our group's address range is 192.168.10.0/24, which means that the entire network where the Lan's are included has a Subnet Mask of **24** 1's (11111111.11111111.11111111.00000000), which when converted to decimal represents the number 255.255.255.0.

This Subnet Mask means that our network is a **Class C** network and we can only use the last octet (byte) to address IP's So we have a total of 2 power of 8 (256) IP addresses that we can address on our network. and every IP will look like 192.168.10.x.



| | 8 bits | 8 bits | 8 bits | 8 bits |
|---|---|---|---|---|
| Class A: | Network | Host | Host | Host |
| Class B: | Network | Network | Host | Host |
| Class C: | Network | Network | Network | Host |
| Class D: | Multicast | | | |
| Class E: | Research | | | |

Figure 3: Network Classes and Octets division

The last octet is known as the **Host ID** (in the case of a Class C Network), and it is unique for each host in a private network; the first three octets are known as the testbfNetwork ID because they are not mutable.(you can subnet a network into other smaller networks.)

Now that we understand IP ranges and subnetting a bit better, we can quickly understand that for each one of the LAN's we can subnet our network into something with a 192.168.10.x/y format.

Lan A needs 5 IP addresses, and as we know, the number of IP's that can be addressed to a network are $2^z$(table 1). So the closest number to 5 that isn't lower than 5 and can be obtained by making $2^z$ is 8, because to get to Lan A we need to **address 8 IP**'s.

”z”, known now as $3 \times 2^3$, this will be the number of 0's present on the subnet mask, so it will be 11111111.11111111.11111111.111111000.

In this case, we will be only wasting 3 IP's (8-5).

| CIDR Block Size | Exponential Notation | Number of Addresses |
|---|---|---|
| /24 | $2^8$ | 256 |
| /23 | $2^9$ | 512 |
| /22 | $2^{10}$ | 1024 |
| /21 | $2^{11}$ | 2048 |
| /20 | $2^{12}$ | 4096 |
| /19 | $2^{13}$ | 8192 |
| /18 | $2^{14}$ | 16384 |
| /17 | $2^{15}$ | 32768 |
| /16 | $2^{16}$ | 65536 |

Table 1: CDIR and Its Number Addresses

The “$y$” in the 192.168.10.x/y can now be calculated with the following expression:

$$32 - z = 32 - 3 = 29$$

The “$x$” in this particular case will be 0, because those are the first addresses that are being allocated and we finally know the subnet mask and address range given to Lan A:

- Lan A Address Range: 192.168.10.0/29
- Lan A Subnet Mask: 255.255.255.248

As we saw before, Lan B needs as many addresses as Lan A, so the subnet mask is the same as ”$y$”, but if we've allocated 8 addresses to Lan A, the ”$x$” will now be 8 because the past 8 were already attributed to.

- Lan B Address Range: 192.168.10.8/29
- Lan B Subnet Mask: 255.255.255.248

## 2.3 Allocating the IP's to the Hosts

Before we begin allocating IP addresses in each LAN, we must be aware of the following rules:

1. We can never use the First IP of an address range, It's known as the Subnet Zero and it represents the Network itself.
2. We can never use the Last IP of an address range, It's known as the BroadCast IP (broadcast is a way to send a message to every ip present in the subnet, for example an **ARP Request**).
3. Usually the IP before the last one is used by the Routing Interface.

Now we can start addressing the IP's.

**Lan A:**

| | |
|---|---|
| Subnet Zero | ->192.168.10.0 |
| Laptop A | ->192.168.10.1 |
| Laptop B | ->192.168.10.2 |
| Ether 1 | ->192.168.10.6 |
| Broadcast | ->192.168.10.7 |

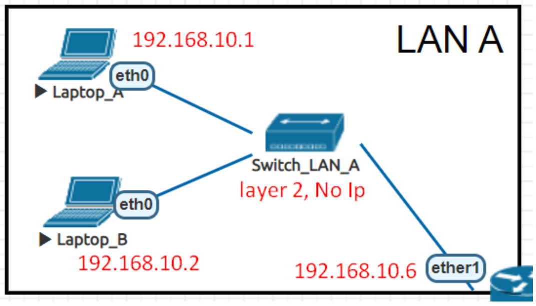


Figure 4: Lan A addressing

**Lan A:**

| | |
|---|---|
| Subnet Zero | ->192.168.10.8 |
| Laptop C | ->192.168.10.9 |
| Laptop D | ->192.168.10.10 |
| Ether 2 | ->192.168.10.14 |
| BroadCast | ->192.168.10.15 |

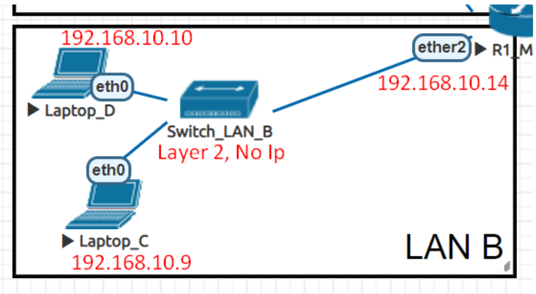


Figure 5: Lan B addressing

### 2.4 Gateways

How can hosts on LAN A communicate with hosts on LAN B and how can they communicate with the ISP? In this case, the router we're using isn't connected to any ISP, so we will only focus on communication between hosts.

When looking for a host that's on another network, we use what's called a "gateway". The host doesn't find the destination, so it sends the message to another host to deal with it. Generally, this host is the router interface, and the router itself will have gateways, either it is another router interface to try to find out if the IP is anywhere in another neighboring network or just sending it out to the internet to find the destination address outside of the private network.

Now we will define the gateways for laptops A, B, C, and D:

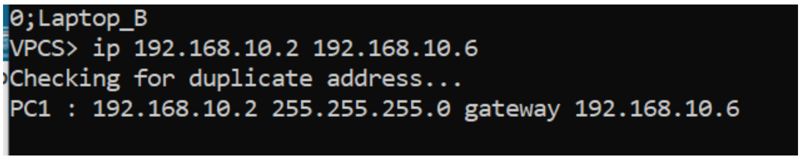
- Laptop A/B ->192.168.10.6
- Laptop C/D ->192.168.10.14

**No gateway needs to be defined for now in R1, but it will be the fa0/0 interface IP later in the project.**

## 3 Configuring the Topology

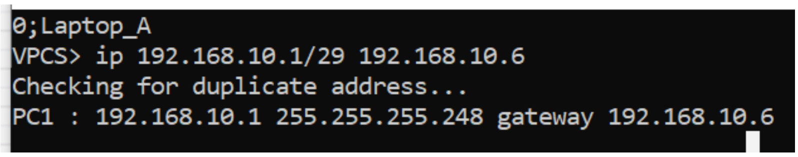
### 3.1 Configuring Lan A

Let's start by configuring the laptops A and B. We already know the IP and gateway for both, we just need to allocate them with telnet.



```
0;Laptop_B
VPCS> ip 192.168.10.2 192.168.10.6
Checking for duplicate address...
PC1 : 192.168.10.2 255.255.255.0 gateway 192.168.10.6
```

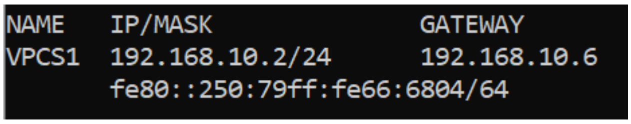
Figure 6: Addressing IP and gateway to Laptop_B



```
0;Laptop_A
VPCS> ip 192.168.10.1/29 192.168.10.6
Checking for duplicate address...
PC1 : 192.168.10.1 255.255.255.248 gateway 192.168.10.6
```

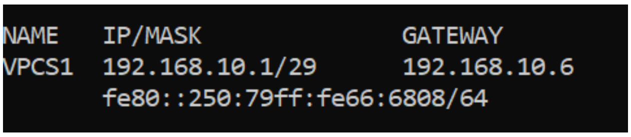
Figure 7: Addressing IP and gateway to Laptop_A

Everything seems right, so let's use the command "show" to check if it is everything set up.



```
NAME   IP/MASK              GATEWAY
VPCS1  192.168.10.2/24      192.168.10.6
       fe80::250:79ff:fe66:6804/64
```

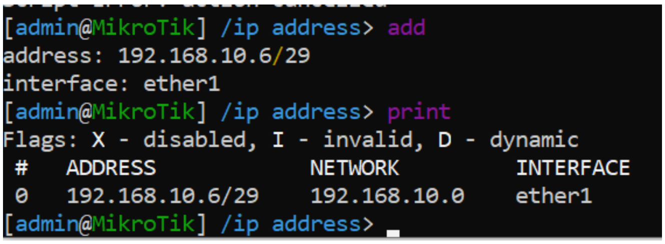
Figure 8: "Show" command print in Laptop_A



```
NAME   IP/MASK              GATEWAY
VPCS1  192.168.10.1/29      192.168.10.6
       fe80::250:79ff:fe66:6808/64
```

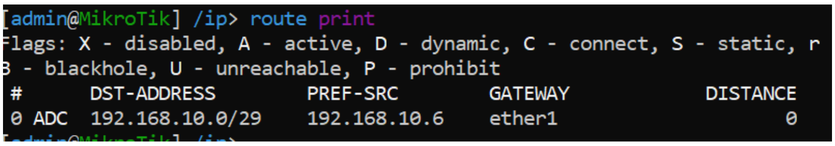
Figure 9: "Show" command print in Laptop_B

Now that we have them configured, before pinging anything or checking any arp cache, configuring ether1 will help us see it clearly later.



```
[admin@MikroTik] /ip address> add
address: 192.168.10.6/29
interface: ether1
[admin@MikroTik] /ip address> print
Flags: X - disabled, I - invalid, D - dynamic
 #   ADDRESS            NETWORK         INTERFACE
 0   192.168.10.6/29    192.168.10.0    ether1
[admin@MikroTik] /ip address>
```

Figure 10: Addressing ether1 in Mikrotik R1 configurations

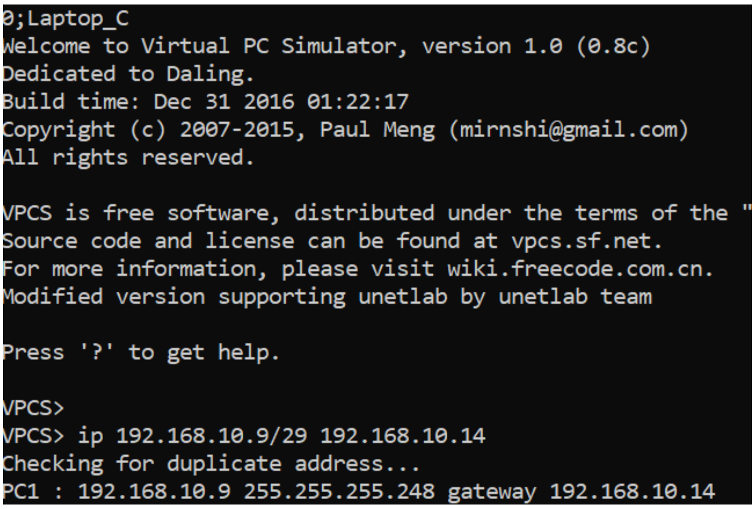


```
[admin@MikroTik] /ip> route print
Flags: X - disabled, A - active, D - dynamic, C - connect, S - static, r
B - blackhole, U - unreachable, P - prohibit
 #      DST-ADDRESS        PREF-SRC        GATEWAY            DISTANCE
 0 ADC  192.168.10.0/29    192.168.10.6    ether1                    0
```

Figure 11: Addressing ether1 in Mikrotik R1 configurations

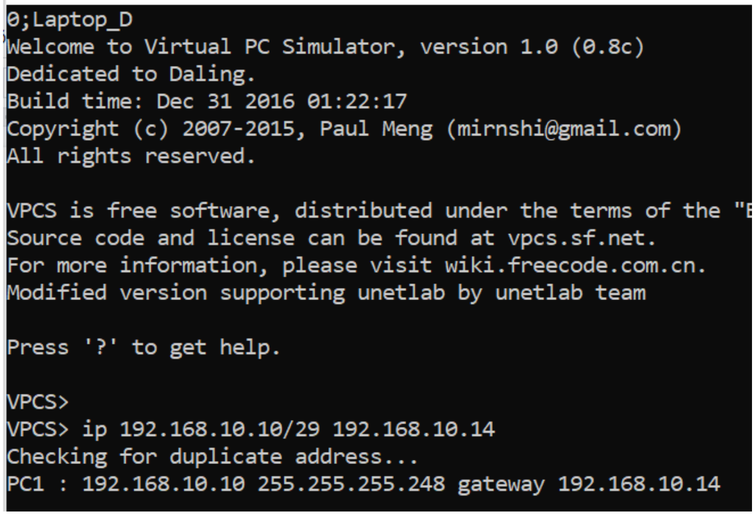
## 3.2 Configuring Lan B

Let's start by configuring the laptops C and D. We already know the IP and gateway for both, we just need to allocate them with telnet.



```
0;Laptop_C
Welcome to Virtual PC Simulator, version 1.0 (0.8c)
Dedicated to Daling.
Build time: Dec 31 2016 01:22:17
Copyright (c) 2007-2015, Paul Meng (mirnshi@gmail.com)
All rights reserved.

VPCS is free software, distributed under the terms of the "
Source code and license can be found at vpcs.sf.net.
For more information, please visit wiki.freecode.com.cn.
Modified version supporting unetlab by unetlab team

Press '?' to get help.

VPCS>
VPCS> ip 192.168.10.9/29 192.168.10.14
Checking for duplicate address...
PC1 : 192.168.10.9 255.255.255.248 gateway 192.168.10.14
```

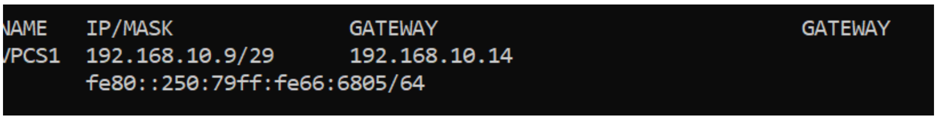
Figure 12: Addressing IP and gateway to Laptop_C



```
0;Laptop_D
Welcome to Virtual PC Simulator, version 1.0 (0.8c)
Dedicated to Daling.
Build time: Dec 31 2016 01:22:17
Copyright (c) 2007-2015, Paul Meng (mirnshi@gmail.com)
All rights reserved.

VPCS is free software, distributed under the terms of the "E
Source code and license can be found at vpcs.sf.net.
For more information, please visit wiki.freecode.com.cn.
Modified version supporting unetlab by unetlab team

Press '?' to get help.

VPCS>
VPCS> ip 192.168.10.10/29 192.168.10.14
Checking for duplicate address...
PC1 : 192.168.10.10 255.255.255.248 gateway 192.168.10.14
```

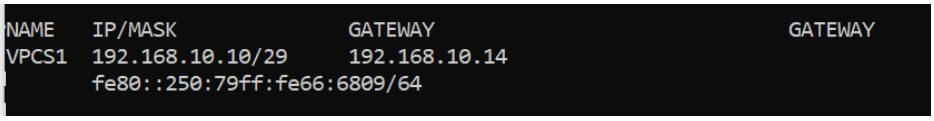
Figure 13: Addressing IP and gateway to Laptop_D

Everything seems right, so let’s use the command ”show” to check if it is everything set up.



```
NAME   IP/MASK              GATEWAY                           GATEWAY
VPCS1  192.168.10.9/29      192.168.10.14
       fe80::250:79ff:fe66:6805/64
```

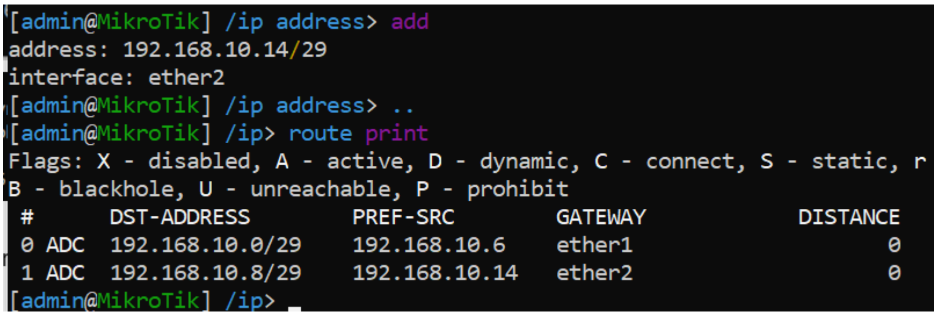
Figure 14: “Show” command print in Laptop_C



```
NAME   IP/MASK              GATEWAY                           GATEWAY
VPCS1  192.168.10.10/29     192.168.10.14
       fe80::250:79ff:fe66:6809/64
```

Figure 15: “Show” command print in Laptop_D

Everything seems right, so we will finally configure the ether2 (router interface 2) and print the IP routing table on the R1_Mikrotik.



```
[admin@MikroTik] /ip address> add
address: 192.168.10.14/29
interface: ether2
[admin@MikroTik] /ip address> ..
[admin@MikroTik] /ip> route print
Flags: X - disabled, A - active, D - dynamic, C - connect, S - static, r
B - blackhole, U - unreachable, P - prohibit
 #      DST-ADDRESS        PREF-SRC        GATEWAY            DISTANCE
 0 ADC  192.168.10.0/29    192.168.10.6    ether1                    0
 1 ADC  192.168.10.8/29    192.168.10.14   ether2                    0
[admin@MikroTik] /ip>
```

Figure 16: Addressing R1_Mikrotik ether2 and printing IP Routing Table

# 4 ARP Caching

## 4.1 ARP Table/Cache analyzes

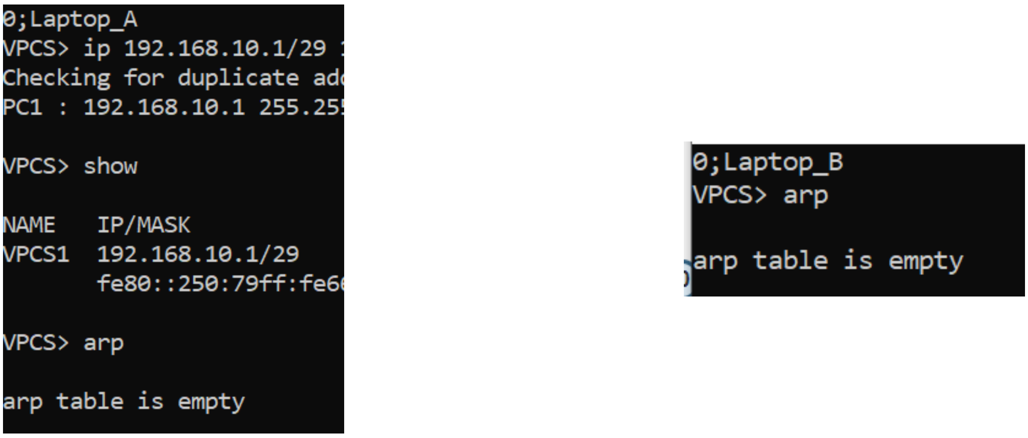
### 4.1.1 ARP Between Hosts in same Network

Lan A is completely configured, and we will now analyze the behavior of ARP caches before and after hosts ping each other.

ARP caches contain entries that map a host’s IP address to its MAC address.

In order to understand this ARP Cache/Table behavior, let’s ping Laptop_A with both Laptop_B and R1_mikrotik.

Let’s see the ARP tables before the pings:

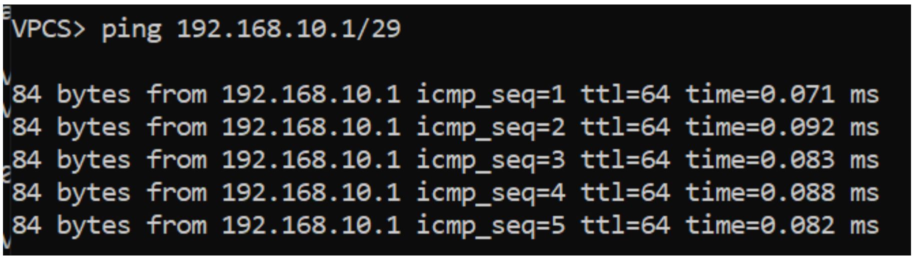


```
0;Laptop_A
VPCS> ip 192.168.10.1/29 1
Checking for duplicate add
PC1 : 192.168.10.1 255.25

VPCS> show

NAME   IP/MASK
VPCS1  192.168.10.1/29
       fe80::250:79ff:fe6

VPCS> arp

arp table is empty
```

```
0;Laptop_B
VPCS> arp

arp table is empty
```

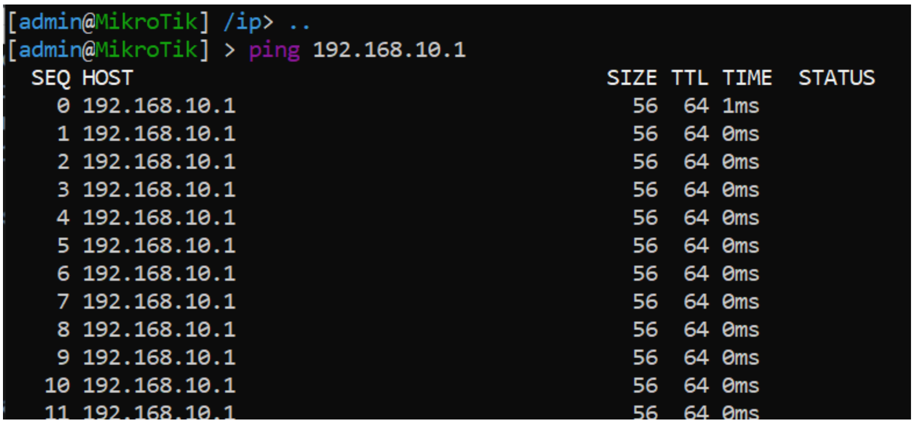
Figure 17: Addressing R1_Mikrotik ether2 and printing IP Routing Table

Now that we saw that the table is empty before any ping we will ping Laptop_A



```
VPCS> ping 192.168.10.1/29

84 bytes from 192.168.10.1 icmp_seq=1 ttl=64 time=0.071 ms
84 bytes from 192.168.10.1 icmp_seq=2 ttl=64 time=0.092 ms
84 bytes from 192.168.10.1 icmp_seq=3 ttl=64 time=0.083 ms
84 bytes from 192.168.10.1 icmp_seq=4 ttl=64 time=0.088 ms
84 bytes from 192.168.10.1 icmp_seq=5 ttl=64 time=0.082 ms
```

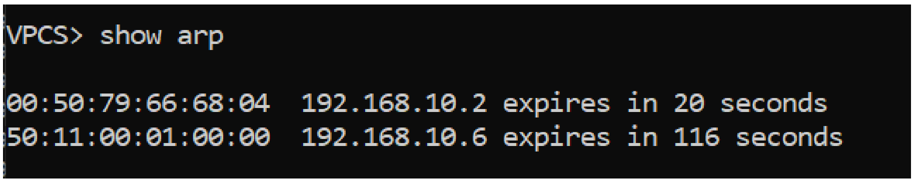
Figure 18: Pinging Laptop$_{A}withLaptop_{B}andR1_{M}ikrotik$



```
[admin@MikroTik] /ip> ..
[admin@MikroTik] > ping 192.168.10.1
  SEQ HOST                                     SIZE TTL TIME  STATUS
    0 192.168.10.1                               56  64 1ms
    1 192.168.10.1                               56  64 0ms
    2 192.168.10.1                               56  64 0ms
    3 192.168.10.1                               56  64 0ms
    4 192.168.10.1                               56  64 0ms
    5 192.168.10.1                               56  64 0ms
    6 192.168.10.1                               56  64 0ms
    7 192.168.10.1                               56  64 0ms
    8 192.168.10.1                               56  64 0ms
    9 192.168.10.1                               56  64 0ms
   10 192.168.10.1                               56  64 0ms
   11 192.168.10.1                               56  64 0ms
```

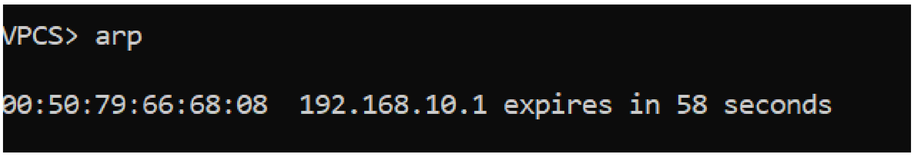
Figure 19: Pinging Laptop$_{A}withLaptop_{B}andR1_{M}ikrotik$

Coming back to Laptop_A and Laptop_B checking the Arp Table



```
VPCS> show arp

00:50:79:66:68:04  192.168.10.2 expires in 20 seconds
50:11:00:01:00:00  192.168.10.6 expires in 116 seconds
```

Figure 20: Laptop_A ARP table after Ping



```
VPCS> arp

00:50:79:66:68:08  192.168.10.1 expires in 58 seconds
```

Figure 21: Laptop_B ARP table after Ping

We can see that the ARP Table of Laptop_A is now contains MAC Addresses (or Physical Addresses) from Laptop_B and R1_Mikrotik and Laptop_B ARP Table contains only the MAC Addresses from Laptop_A.

This means that the first ping message between two hosts that don't know each other's MAC Addresses is a **ARP Request** sent as a **Broadcast** message questioning "who has" the IP that's being pinged, as well as the sender's Mac Address. After that, the ARP Response is a **Unicast** message given by the pinged host containing already containing his Mac Address to identify himself. The response is Unicast because the host answers already knowing the MAC Address of the sender.

The ARP Table avoids some repeated **Broadcast** messages in consistent communications.

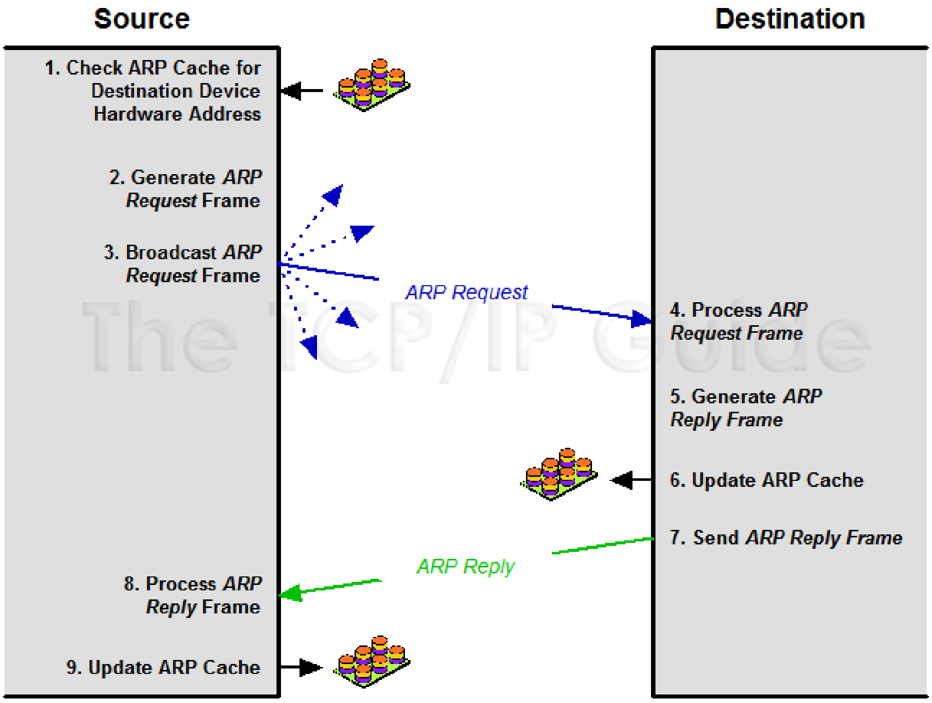
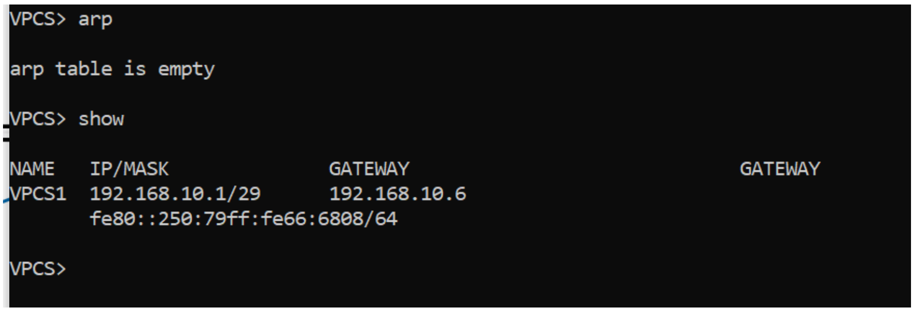

Source
Destination
1. Check ARP Cache for Destination Device Hardware Address
2. Generate ARP Request Frame
3. Broadcast ARP Request Frame
ARP Request
4. Process ARP Request Frame
5. Generate ARP Reply Frame
6. Update ARP Cache
7. Send ARP Reply Frame
ARP Reply
8. Process ARP Reply Frame
9. Update ARP Cache
The TCP/IP Guide

Figure 22: ARP Transaction Protocol

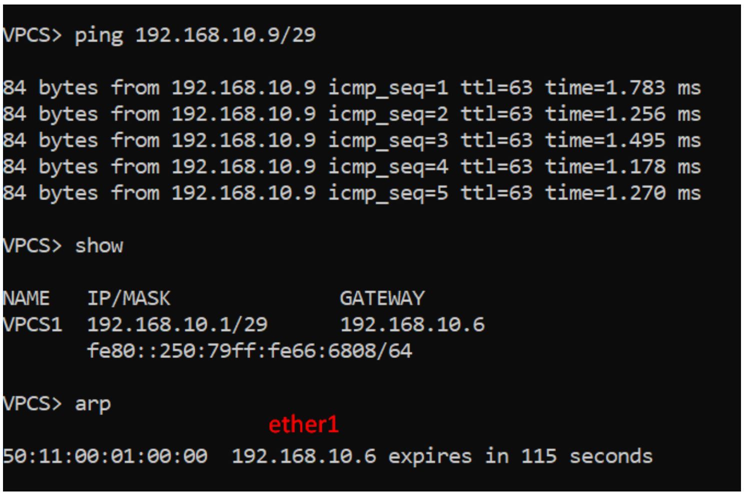
### 4.1.2 ARP Between Hosts in different Network

In the 4.1.1. section we explained how the ARP works, but will it cache anything if the ping goes between hosts in different networks? Let's just look at what happens.



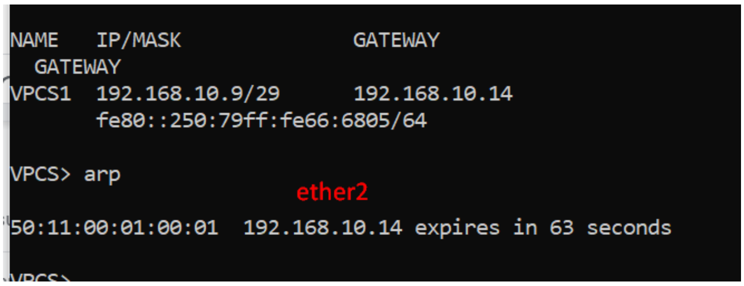
```
VPCS> arp

arp table is empty

VPCS> show

NAME   IP/MASK              GATEWAY                            GATEWAY
VPCS1  192.168.10.1/29      192.168.10.6
       fe80::250:79ff:fe66:6808/64

VPCS>
```

Figure 23: Laptop_A ARP table before pinging



```
VPCS> ping 192.168.10.9/29

84 bytes from 192.168.10.9 icmp_seq=1 ttl=63 time=1.783 ms
84 bytes from 192.168.10.9 icmp_seq=2 ttl=63 time=1.256 ms
84 bytes from 192.168.10.9 icmp_seq=3 ttl=63 time=1.495 ms
84 bytes from 192.168.10.9 icmp_seq=4 ttl=63 time=1.178 ms
84 bytes from 192.168.10.9 icmp_seq=5 ttl=63 time=1.270 ms

VPCS> show

NAME   IP/MASK              GATEWAY
VPCS1  192.168.10.1/29      192.168.10.6
       fe80::250:79ff:fe66:6808/64

VPCS> arp
                      ether1
50:11:00:01:00:00  192.168.10.6 expires in 115 seconds
```

Figure 24: Laptop_A ARP table after pingin Laptop_C



```
NAME   IP/MASK              GATEWAY
  GATEWAY
VPCS1  192.168.10.9/29      192.168.10.14
       fe80::250:79ff:fe66:6805/64

VPCS> arp
                       ether2
50:11:00:01:00:01  192.168.10.14 expires in 63 seconds

VPCS>
```

Figure 25: Laptop_C ARP table after being pinged by Laptop_A

Looking at the 3 pictures above, we can quickly see that the IP and MAC addresses that are cached are the ones that belong to the router interfaces presented in each LAN. With this said, we conclude that between different LANs we can't get host MAC addresses.

## 4.2 ARP Cache Vulnerabilities

When Address Resolution Protocol (ARP) was first created, security wasn't much of a problem, and the ARP transactions weren't made to pursue any authentication. As we said before, the reply to an ARP request is **Unicast** and it comes from the computer that has the IP address that the source of the request asked for.

But what if the reply is given from another computer? How could the source know that the reply wasn't the one he wanted to be without having his MAC address? He simply can't be sure of that.

Exploiting this vulnerability in ARP and ARP Caches, one of the most common attacks that's out there is ARP Poisoning, a type of Man In The Middle (MITM) attack.

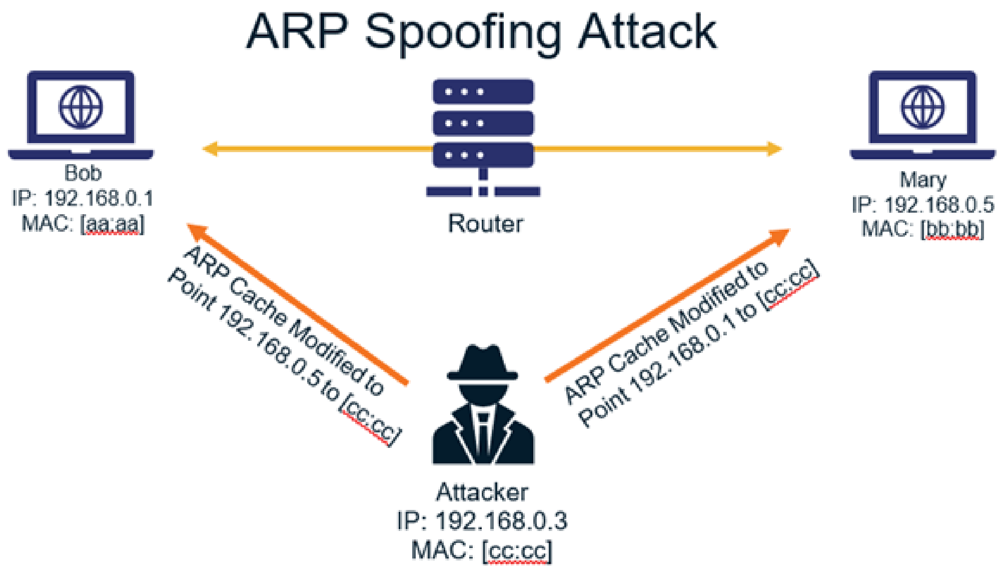


Figure 26: MITM attack

# 5 Trace Route

## 5.1 Watch is TraceRoute

TraceRoute is simple. It's a command used in troubleshooting and analyzing a network that basically just does what the name says. It "traces" the route of a packet from a source to a destination.

While tracking that route, It shows us the **HOP**'s, a HOP is a path that an at least **Layer 3** device makes to another at least **Layer 3** device.

Example:

A packet going from **PC1** to **Router10** is one HOP. Going from Router10 to Router11 is another HOP.

If PC1 has no direct connection to Router 11, from PC1 to Router 11 we will have 2 HOPs.

## 5.2 Switches on TraceRouting

"How can a PC know if it is connected to a switch? Is traceroute useful in this situation? "

Answering to the question present in the statement, a switch will have no influence on a traceroute because switches are layer 2 devices. This means they have no IP, but only a MAC Address or Physical Address.

The TraceRoute uses IP's and no MAC Addresses. Those are present in Layer 3 (Network Layer).

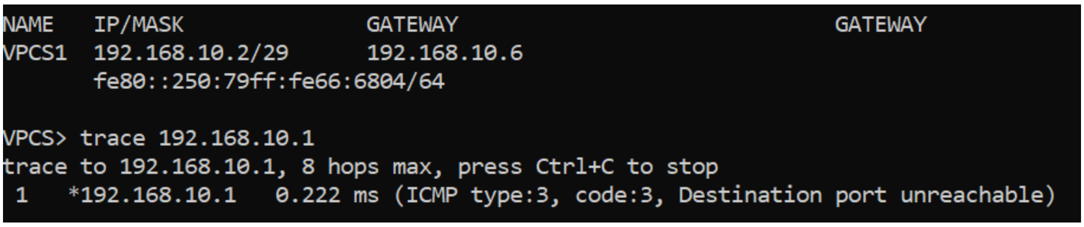
If you want to check the connection between two laptops and a switch, just turn the router off and try to ping each laptop from the other one using their static IP addresses (Section 6). (As we saw, ping works with MAC-Addresses, so it's possible to check the connection to layer 2 devices using this same command.)

## 5.3 TraceRoute Between Laptops

Having traceroute explained, we can now predict how many hops we will have between hosts. Let's see the theoretical predicted results and test them on laptops in the same lan and different lans.

### 5.3.1 Laptops In the Same Lan

First we will traceroute the Laptop_A from the Laptop_B, as we saw before, switches don't matter in traceroute (section 5.2) and both laptops are on the same LAN. This means that theoretically the traceroute between these two laptops will have a single **HOP**.



```
NAME   IP/MASK              GATEWAY                           GATEWAY
VPCS1  192.168.10.2/29      192.168.10.6
       fe80::250:79ff:fe66:6804/64

VPCS> trace 192.168.10.1
trace to 192.168.10.1, 8 hops max, press Ctrl+C to stop
 1   *192.168.10.1   0.222 ms (ICMP type:3, code:3, Destination port unreachable)
```

Figure 27: MITM attack

As expected, there is only 1 HOP in this traceroute; everything is working as we predicted.

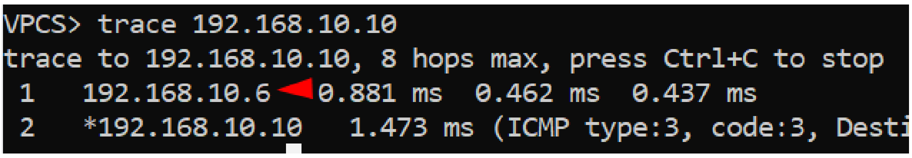
### 5.3.2 Laptops in Different Lan's

Now that we know how many HOP's there are in the traceroute between Laptop_B and Laptop_A, we are going to test the traceroute between Laptop_B that is on Lan A and Laptop_D that is on Lan B.

To go from one LAN to another, we know the packet needs to pass through the R1_Mikrotik. This Router, like every router, is a Layer 3 Host, so it will theoretically count as a HOP, so the path that the packet will take is:

1. Laptop_B ->R1_Mikrotik
2. R1_Mikrotik ->Laptop_D

Total of 2 hops.



```
VPCS> trace 192.168.10.10
trace to 192.168.10.10, 8 hops max, press Ctrl+C to stop
 1   192.168.10.6   0.881 ms  0.462 ms  0.437 ms
 2   *192.168.10.10   1.473 ms (ICMP type:3, code:3, Desti
```

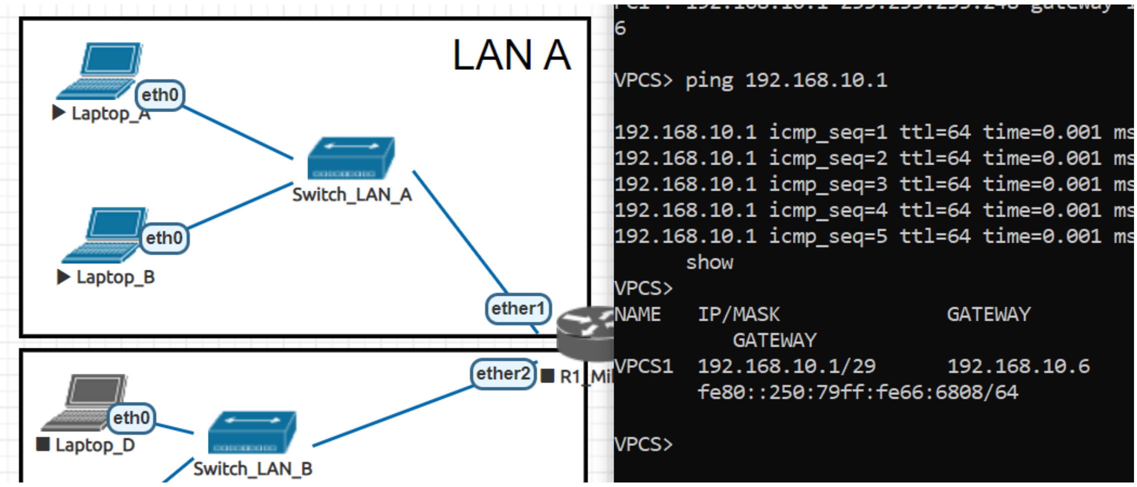
Figure 28: Traceroute between Laptop_B and Laptop_D

As we expected, there were 2 hops in this traceroute, but we can see one more thing too. The Ether1 address is the first hop that was counted. This happens because that's the gateway address of Laptop_B, otherwise the packet wouldn't get to R1_Mikrotik even if the interface was well addressed.

## 6 Laptops Connectivity to Switch

To end all the tests for the topology configuration, we will test the connectivity of the laptops to the switch. We will do it on the Lan A.

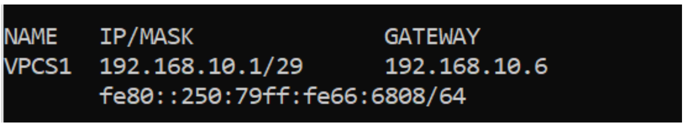
Starting by turning off the router, we will just ping the laptop_B from the laptop_A and see if it works.



```
PC1 : 192.168.10.1 255.255.255.248 gateway 1
6

VPCS> ping 192.168.10.1

192.168.10.1 icmp_seq=1 ttl=64 time=0.001 ms
192.168.10.1 icmp_seq=2 ttl=64 time=0.001 ms
192.168.10.1 icmp_seq=3 ttl=64 time=0.001 ms
192.168.10.1 icmp_seq=4 ttl=64 time=0.001 ms
192.168.10.1 icmp_seq=5 ttl=64 time=0.001 ms
          show
VPCS>
NAME   IP/MASK              GATEWAY
           GATEWAY
VPCS1  192.168.10.1/29      192.168.10.6
       fe80::250:79ff:fe66:6808/64

VPCS>
```

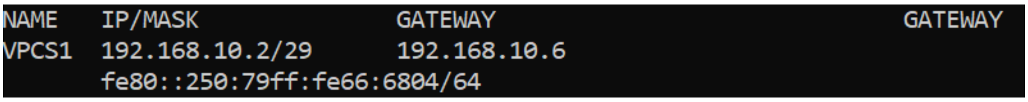
Figure 29: Pinging Laptops with R1_Mikrotik Off.

## 7 Final Configurations



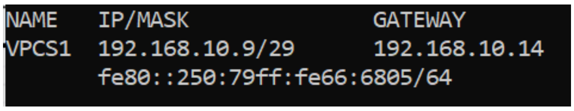
```
NAME   IP/MASK              GATEWAY
VPCS1  192.168.10.1/29      192.168.10.6
       fe80::250:79ff:fe66:6808/64
```

Figure 30: Laptop_A Configuration.



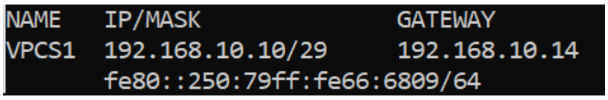
```
NAME   IP/MASK              GATEWAY                                  GATEWAY
VPCS1  192.168.10.2/29      192.168.10.6
       fe80::250:79ff:fe66:6804/64
```

Figure 31: Laptop_B Configuration.



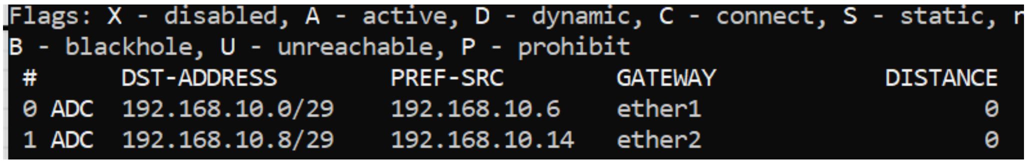
```
NAME   IP/MASK              GATEWAY
VPCS1  192.168.10.9/29      192.168.10.14
       fe80::250:79ff:fe66:6805/64
```

Figure 32: Laptop_C Configuration.



```
NAME   IP/MASK              GATEWAY
VPCS1  192.168.10.10/29     192.168.10.14
       fe80::250:79ff:fe66:6809/64
```

Figure 33: Laptop_D Configuration.



```
Flags: X - disabled, A - active, D - dynamic, C - connect, S - static, r
B - blackhole, U - unreachable, P - prohibit
 #      DST-ADDRESS        PREF-SRC        GATEWAY            DISTANCE
 0 ADC  192.168.10.0/29    192.168.10.6    ether1                    0
 1 ADC  192.168.10.8/29    192.168.10.14   ether2                    0
```

Figure 34: Route Print of R1_Mikrotik.

# 8 Conclusion

With this project, we were able to test our knowledge of configuring small networks and subnetting, as well as understanding broadcast, unicast, and the differences between Ping and Traceroute. Even though it was not necessary work, we felt it was important to include it, as it is security on one of the biggest network problems.