How to reach remote devices behind NAT.
Build your VPN concentrator with MikroTik!

Marco Boschini
MikroTik Professionals Conference – Praha 2025

About me

- Marco Boschini
- from Desenzano, Italy
- MikroTik Certified Trainer
  MTCNA, MTCRE, MTCINE, MTCWE, MTCSE, MTCTCE, MTCSWE, MTCEWE, MTCIPv6, MTCUME
- Founder Corsi MikroTik
  https://corsimikrotik.it

Agenda

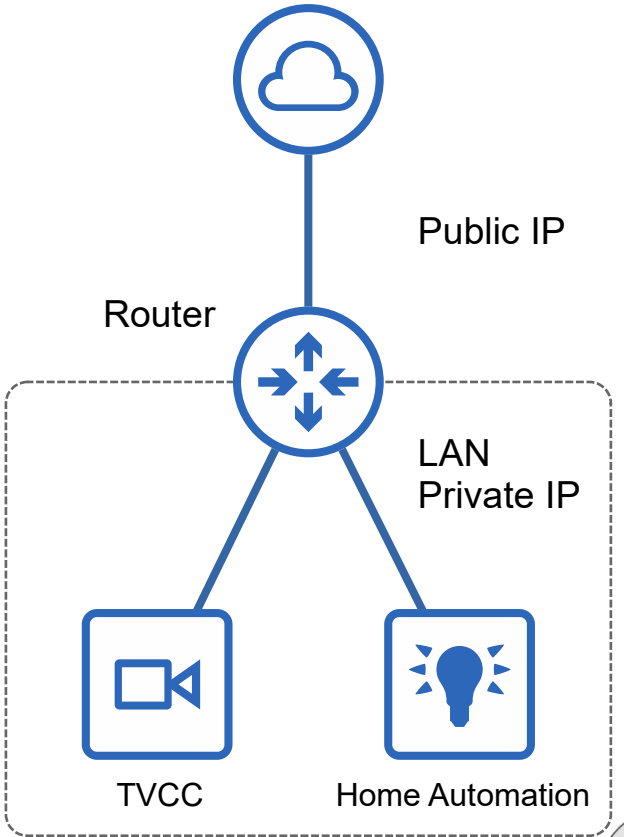
- Scenario of a remote device
- How to reach remote devices?
- NAT problem
- RouterOS as a VPN Server
- Basic routing concepts
- RouterOS as a VPN Client (main router)
- RouterOS as a VPN Client (not main router)

Agenda

- How about if two or more customers have the same network subnet?
- How about security?
- Best VPN client to connect customers

Scenario of a remote device

- Remote device can be a printer, PLC, TVCC system, Home Automation or similar device
- This device normally are behind a router, in a private network (LAN) without public IP.



Scenario of a remote device

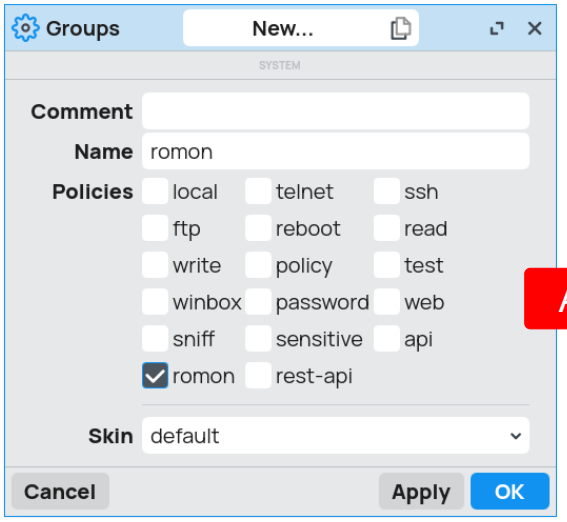
- Some devices can not implement security mechanism that are required today:
  - Authentication
  - Authorization
  - Encryption

Authentication

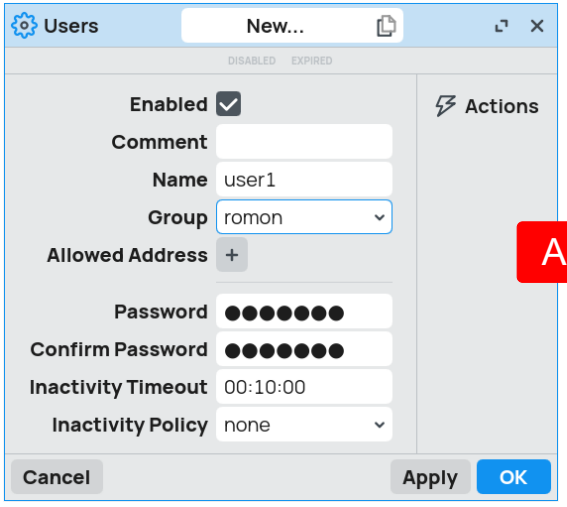
- Authentication is used by server when the server needs **to know exactly who is accessing** their information or site.
- In authentication the user or computer has to **prove its identity** to the server or client.
- Usually, authentication entails the use of a **username, password, certificate, 2FA**.

Authorization

- **Authorization** is a process by which a server determines if the client has **permission to use a resource**.
- Authorization is usually coupled with authentication so that the server has some concept of the client who is requesting access.



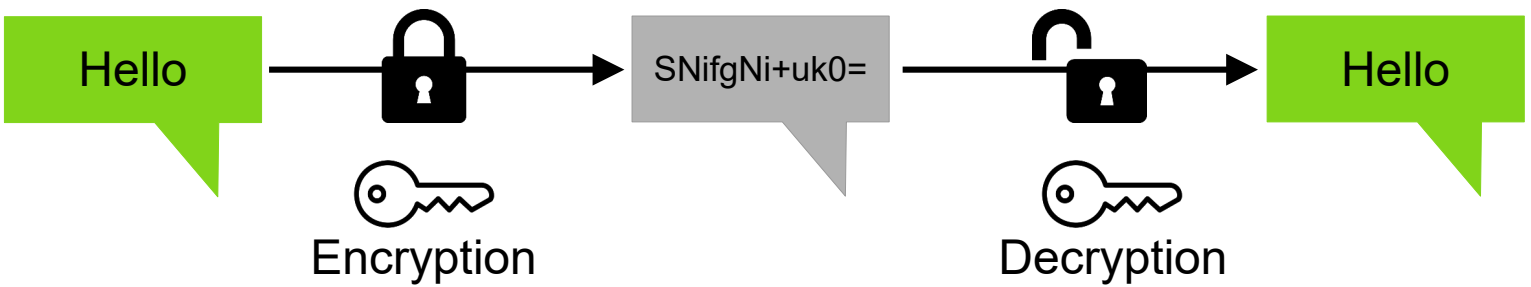
Authorization



Authentication

Encryption

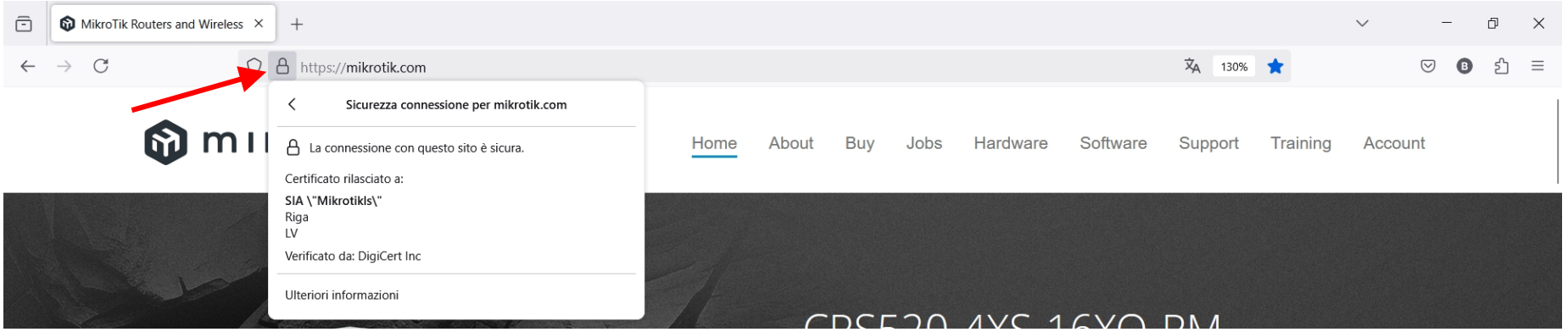
- **Encryption** involves the process of transforming data so that it is unreadable by anyone who does not have a **decryption** key.



Encryption

Corsi MikroTik

For example, HTTPS use encryption that use SSL (Secure Socket Layer) to protect data transitted over the web like credential (user/password), credit card number or information delivered as content.



De facto protocols

Corsi MikroTik

- Not all the devices/protocols support authentication, authorization and encryption.
- I worked for many years in the field of Building Automation.
- There are some protocols, developed many years ago, that is very simple, efficient and open (no royalties).
- These protocols have become a de facto standard, being implemented and used worldwide by many manufacturers.

Modbus

- One of these is Modbus
- Developed by Modicon in the 1979.
- Today is propriety of Schneider Electric
- MikroTik implement that protocol in the IoT Packages for IoT devices (es: KNOT)

https://mikrotik.com/product/knot

Modbus

- Modbus permit to exchange “raw” data within devices in a common binary language.
- You can read values from device
  - Eg. read a temperature or power
- You can write values to device
  - Eg. actuate a light or ventilation

Image from https://se.com

Image from https://www.spluss.de/

Image from https://www.johnsoncontrols.com/

Modbus

- ModbusRTU communicate over serial bus (RS485)
- ModbusTCP communicate over TCP/IP protocols, TCP/502.

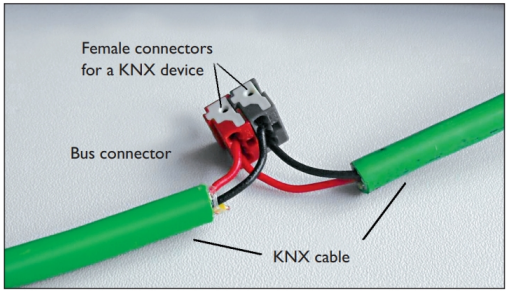
KNX

Corsi MikroTik

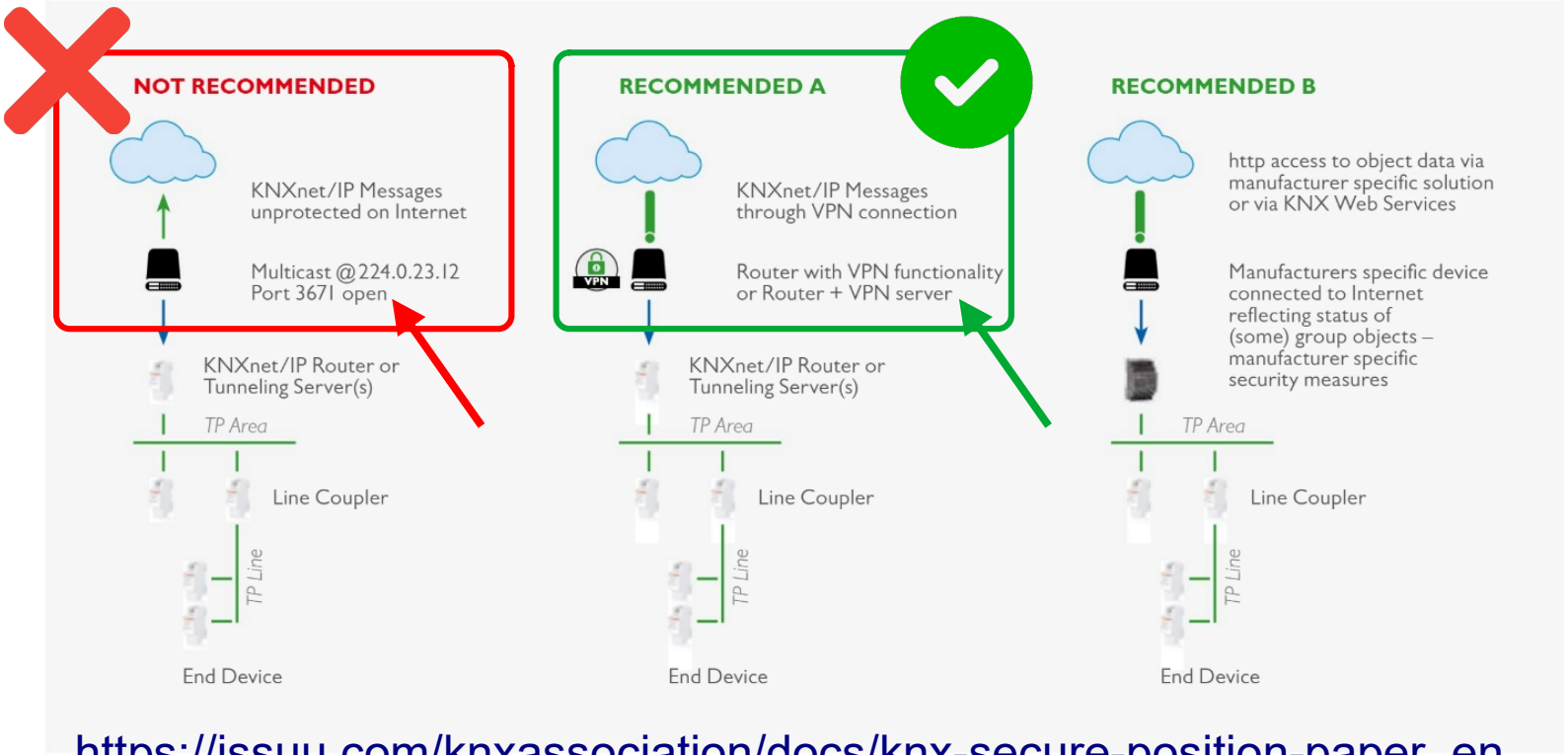
- KNX is the first worldwide standard for Building Automation.
- Developed in the ‘90
- This is like the ethernet protocols for networking: simple, robust and future proof.
- Implemented by over 500 manufacturer worldwide
- More than 500 training centres.

KNX

- The communication device-to-device is over bus cable Twisted Pair
- IP Connection is also possibile via KNXnet/IP from 2007
  - Unicast via UDP/3671
  - Multicast via IP address 224.0.23.12 UDP/3671
- **IP Secure connection** is available from 2017



PREVENTING ACCESS TO THE NETWORK TO THE VARIOUS KNX PHYSICAL MEDIA



https://issuu.com/knxassociation/docs/knx-secure-position-paper_en

Secure Access using VPN – Example #2

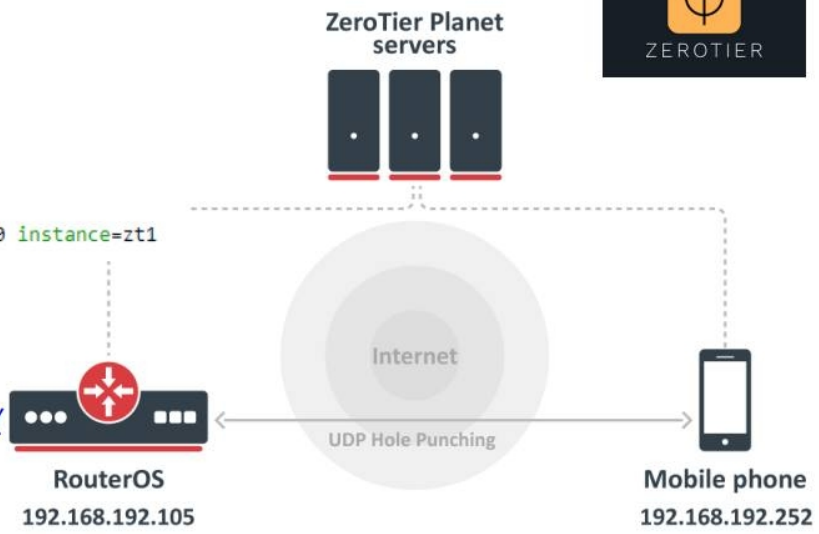
OPTION B: Use ZeroTier direct with LAN Router (Example: Mikrotik)

(Steps 1, 2, 4 and 5 are the same as for Example #1)

Step 3: Config Zerotier in MKT console:

```
[admin@mikrotik] > zerotier/enable zt1
[admin@mikrotik] zerotier/interface/add network=1d71939404912b40 instance=zt1
```

See:
https://help.mikrotik.com/docs/display/ROS/ZeroTier
https://www.zerotier.com/2021/08/31/zerotier-on-mikrotik/



Smart home and building solutions.
Global. Secure. Connected.

Join us
www.knx.org

KNX

- I’m also an Official KNX Tutor Instructor
- At some point in my life I felt it was necessary to **connect the dots** regarding these technologies and helping people to build solid competency.

The reason why

- This is the reason why I use (read “love” ❤️) MikroTik:
  - It permits everyone to study networking and grow
  - It is affordable to utilize in every situation.
    (We need practice to familiarize and better understand.)
  - It is very flexibile
  - It helped me to solve problems a lot of times
  - Almost all devices have all the functionality
    (no need for money for extra license or subscriptions)

How to reach remote devices?

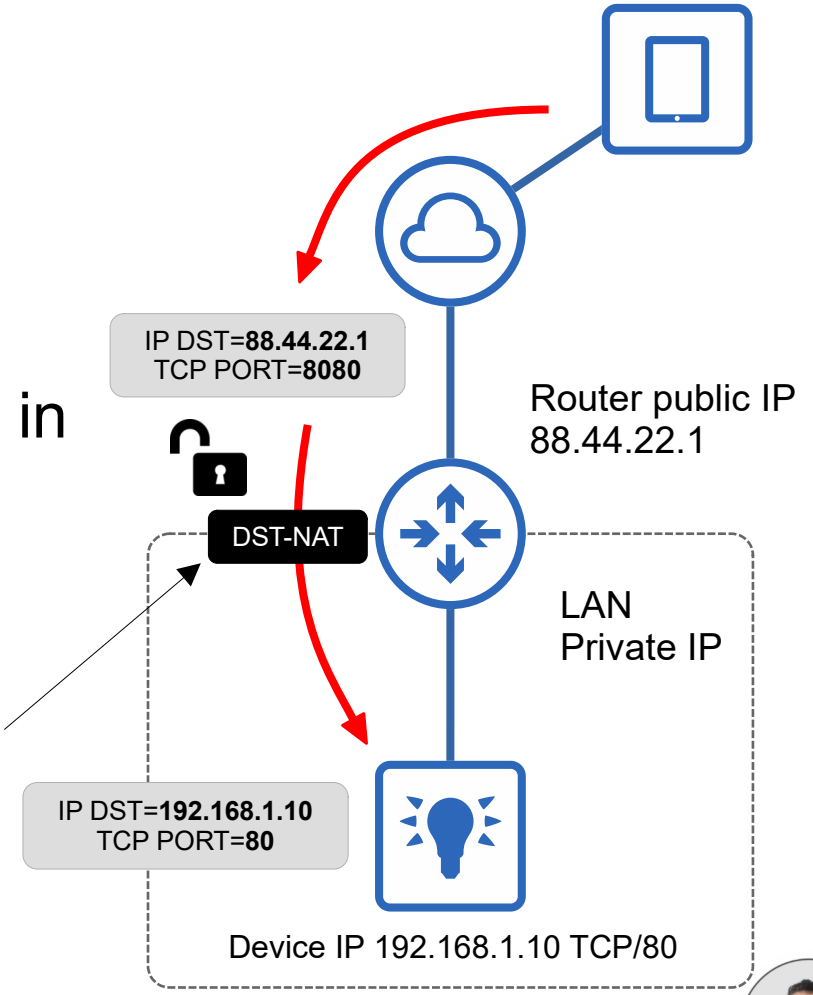
Corsi MikroTik

- To reach these devices there are multiple solutions:
  - **DST-NAT** (“port forwarding”):
    widely used in the past, not secure today.
  - **Cloud**
    only if the manufacturer have a cloud implementation, not always available.
  - **VPN**
    Virtual private network, client can reach remote devices in a private local network over a Secure Tunnel connection with Authentication, Authorization and Encryption.

DST-NAT

- DST-NAT is a function of the firewall NAT
- It permits to change two parameters in the IP packet
  - Destination IP address
  - Destination protocol port

```
/ip firewall nat
add chain=dst-nat \
in-interface=ether1 protocol=tcp dst-port=8080 \
action=dstnst to-address=192.168.1.10 to-port=80
```

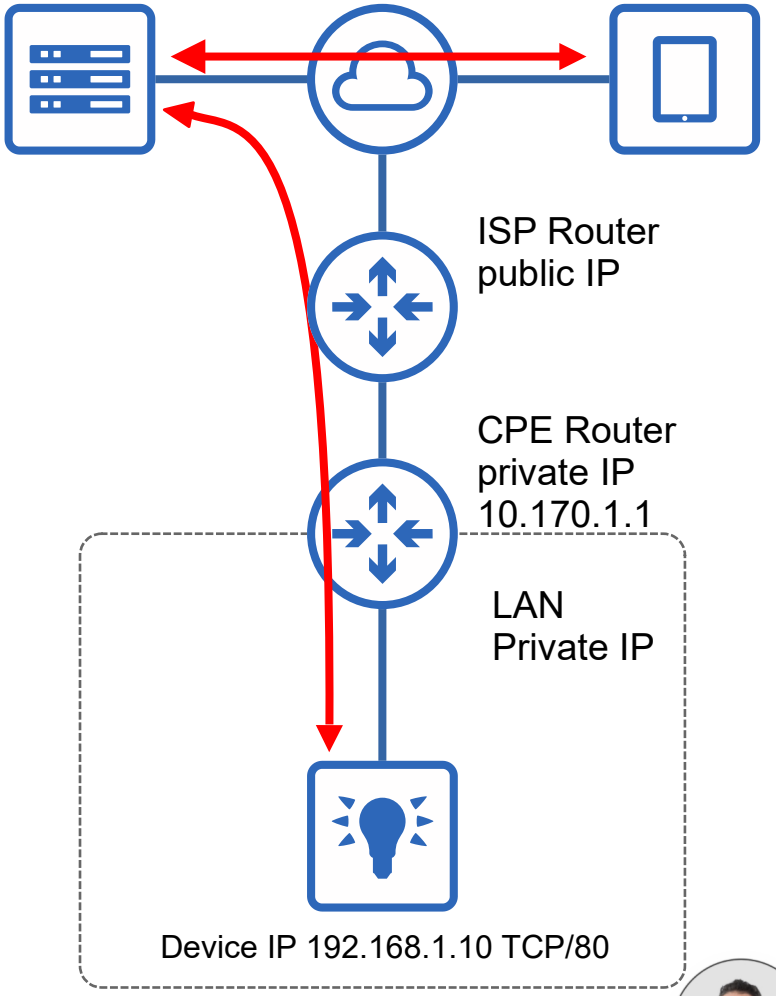


DST-NAT problem

- **It is not secure because it expose devices on the WAN**.
  - Everyone can try to login with bruteforce attack
  - Everyone can send a tons of IP packets to the device (DoS/DDoS).
  - You can limit access with firewall filter combined with Address List.
- **It requires a public IP** (also dynamic IP is OK).
  - With the lack of IPv4 address, today it can be difficult have a public IP address.
  - With Mobile data (4G/LTE) it's very hard to have a public IP address.

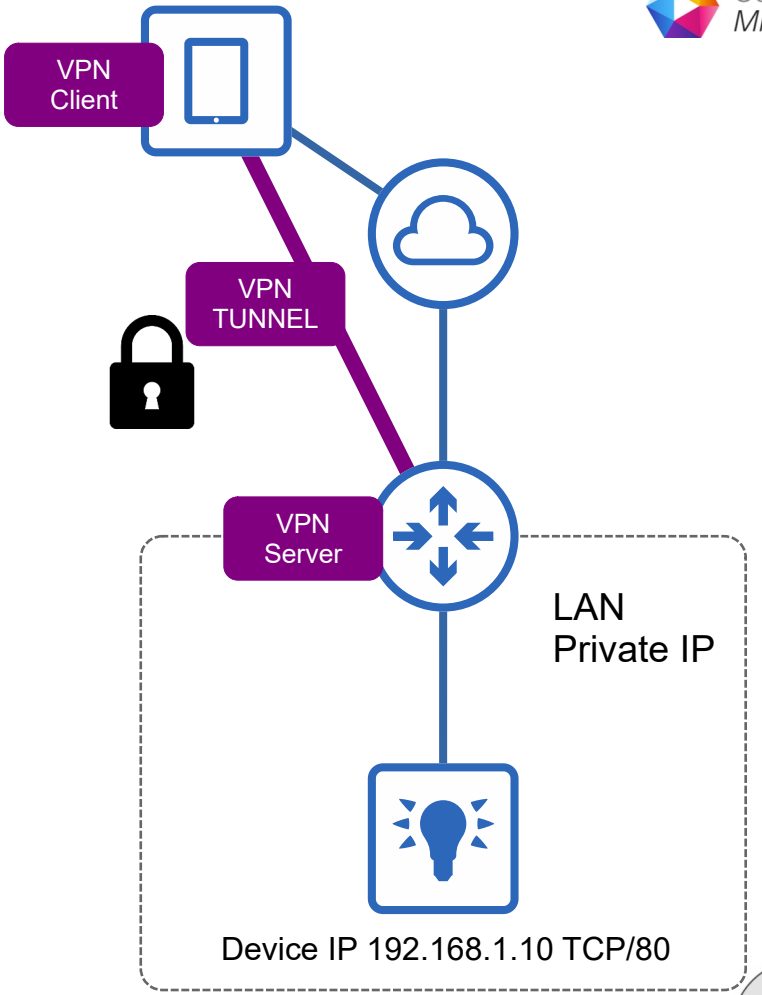
Cloud

- Require a manufacturer implementation
- Data storage is on cloud.
- If it’s available, it’s ready-to-use.
- Reads “on someone else’s computer”
- We have no control over.



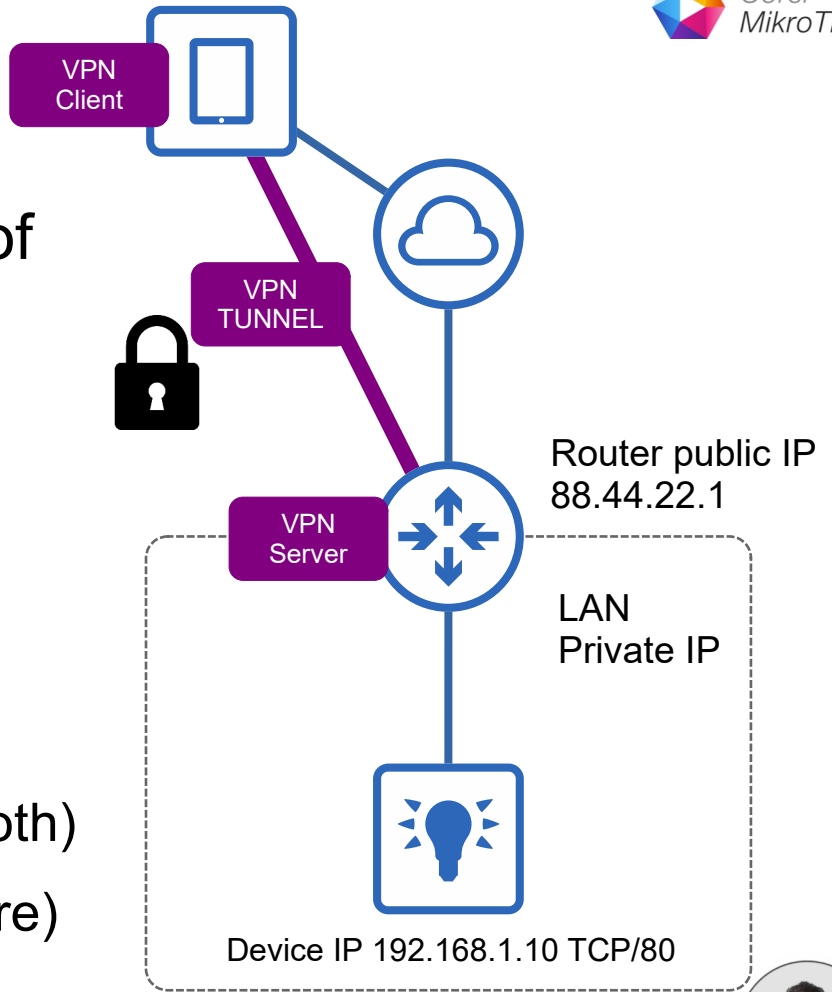
VPN

- Via VPN a **secure direct connection** can be created from a remote client (VPN Client) to the customer router (VPN Server)
- VPN Server MUST have a public IP to be reachable from remote VPN client.



VPN

- RouterOS implement different type of VPN Server for this scenario:
  - OpenVPN
  - WireGuard
  - L2TP/IPSec
  - SSTP
  - IPSec
    (require public and static address for both)
  - ~~PPTP~~ (please don't use it! It's not secure)



NAT problem

- If the customer does not have a Public IP, the VPN Server can not be reachable from the customers
- A “relay server” is required

5 minutes OpenVPN Server

- Now we are going to configure a RouterOS device to become a OpenVPN Server, with a mobile device as VPN Client.
- OpenVPN Require Certificate and credentials (user/password)
- OpenVPN permit to push routes to the client.
- It is possible to create multiple OpenVPN Server on the same device (eg. with different protocol: TCP and UDP)

OpenVPN - Certificate Authority

Corsi MikroTik

Terminal R1

```
/certificate

add name=OpenVPN-CA common-name=OpenVPN-CA \
days-valid=3650 key-usage=crl-sign,key-cert-sign

sign OpenVPN-CA ca-crl-host=127.0.0.1
```


OpenVPN - Server Certificate

Corsi MikroTik

Terminal R1

```
/certificate

add name=OpenVPN-Server common-name=OpenVPN-Server \
days-valid=3650 \
key-usage=digital-signature,key-encipherment,tls-server

sign OpenVPN-Server ca=OpenVPN-CA
```


OpenVPN - Client Certificate

Terminal R1

```
/certificate

add name=OpenVPN-Client common-name=OpenVPN-Client \
days-valid=3650 key-usage=tls-client

sign OpenVPN-Client ca=OpenVPN-CA
```


OpenVPN - Export Certificate

Corsi MikroTik

Terminal R1

```
/certificate

export-certificate OpenVPN-CA export-passphrase="" \
export-certificate OpenVPN-Client export-passphrase="key-secret"
```


OpenVPN Certificate

Certificates | Certificates | SCEP Servers | SCEP RA | Requests | OTP | CRL

New | Remove

	Name	Issuer	Common Name	Subject Alt. N...	Key Size	Days Valid	Trusted
KLAT	OpenVPN-CA		OpenVPN-CA		2048	3650	yes
KI	OpenVPN-Client		OpenVPN-Client		2048	3650	no
KI	OpenVPN-Server		OpenVPN-Server		2048	3650	no

File List | File | Cloud Backup

New | Remove

File Name	Type	Size	Creation Time
autosupout.rif	.rif file	332,9 KiB	2025-02-15 20:22:30
cert_export_OpenVPN-CA.crt	.crt file	1139 B	2025-02-15 20:45:34
cert_export_OpenVPN-Client.crt	.crt file	1111 B	2025-02-15 20:46:02
cert_export_OpenVPN-Client.key	.key file	1886 B	2025-02-15 20:46:02
skins	directory		2025-02-15 17:43:50

OpenVPN Server Configuration

Terminal R1

```
/interface/ovpn-server/server

add name=ovpn-server1 certificate=OpenVPN-Server \
protocol=udp default-profile=default-encryption \
push-routes="192.168.1.0 255.255.255.0"
```


OpenVPN Export .ovpn conf

Terminal R1

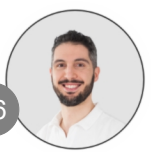
```
/interface/ovpn-server/server

export-client-configuration \
server-address=public-ip-openvpn-server \
ca-certificate=cert_export_OpenVPN-CA \
client-certificate=cert_export_OpenVPN-Client.crt \
client-cert-key=cert_export_OpenVPN-Client.key
```

Result
progress: ovpn client configuration **'client1739653470.ovpn'** file exported

OpenVPN User

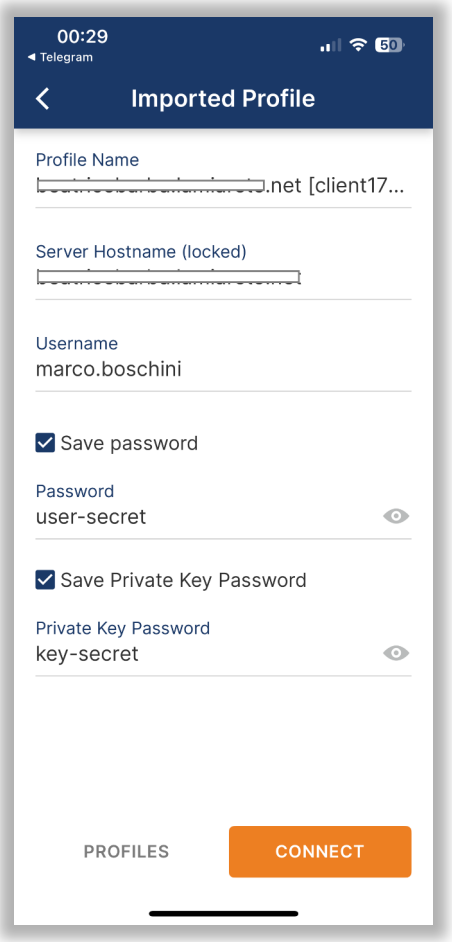
Terminal R1

```
/ppp/secret
add name=marco.boschini password="user-secret" \
service=ovpn \
local-address=10.99.99.1 remote-address=10.99.99.2
```

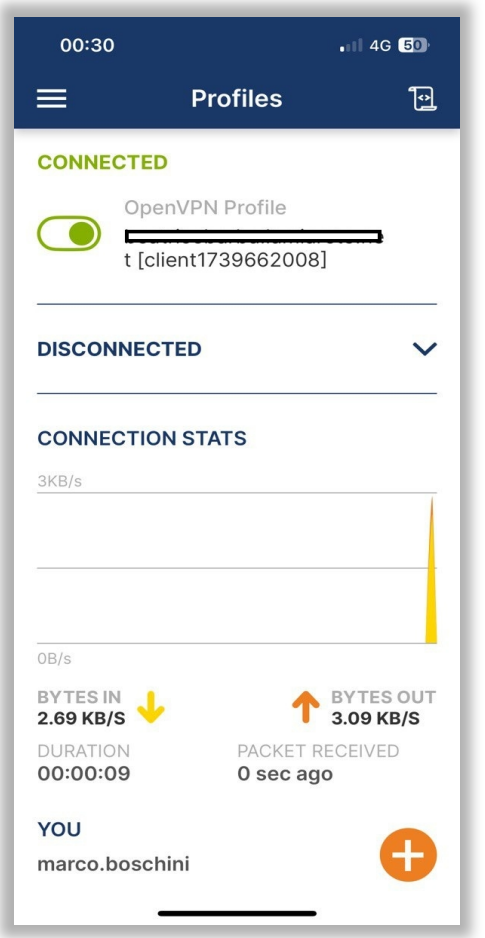
OpenVPN Connect (user application)

- Import the **.ovpn file** in OpenVPN Connect
- Insert the user-name “marco.boschini”
- Insert the user-password “user-secret”
- Insert the key-password “key-secret”
- Select “CONNECT”



OpenVPN Connect (user application)

- The connection is working!
- Connection from Windows PC is almost the same
- Connection from MacOS is almost the same



OpenVPN Client from linux

Save the remote openvpn config file in a directory eg:

```
/home/marcoboschini/
```

Via SCP (SSH is required enabled on RouterOS)

```
$ scp admin@mikrotik-ip:/client1739653470.ovpn
/home/marcoboschini/client.ovpn
```

```
marcoboschini@corsimikrotik-vm:~
$ scp admin@192.168.122.158:/client1739653470.ovpn ./client.ovpn
admin@192.168.122.158's password:
client1739653470.ovpn                         100% 4483     1.0MB/s   00:00
```


OpenVPN Client from linux

To save credentials in the same directory:

- Create a new file **“client-auth-user-pass.txt”** and write
```
marco.boschini
user-secret
```
- Create a new file **“client-askpass.txt”** and write
```
key-secret
```

OpenVPN Client from linux

Open the config file client.ovpn with a text-editor (Eg. Nano)

- find the row:

```
auth-user-pass
```

- edit with this

```
auth-user-pass client-auth-user-pass.txt
askpass client-askpass.txt
```


OpenVPN Client from linux

Corsi MikroTik

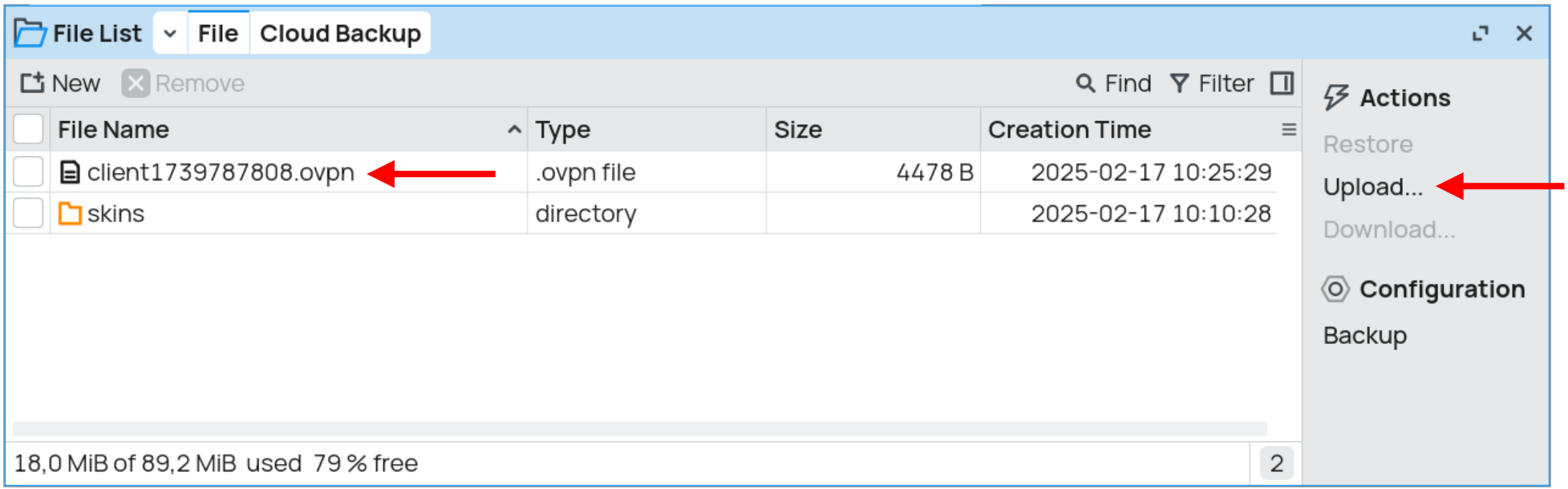
Start openvpn with this command:

```
sudo openvpn --config client.ovpn
```

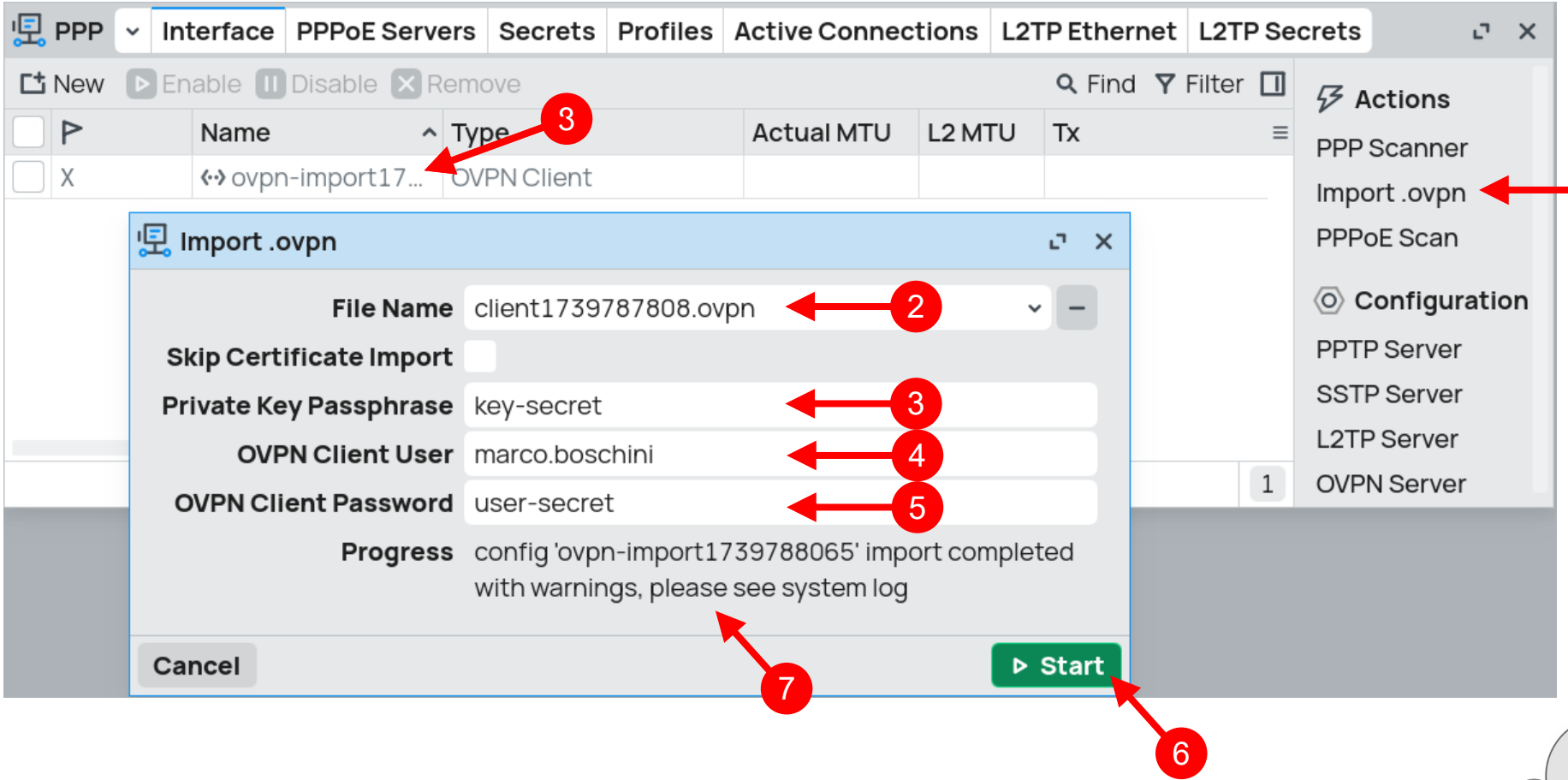
```
2025-02-16 01:33:09 ++ Certificate has EKU (str) TLS Web Server Authentication, expects TLS Web Server Authentication
2025-02-16 01:33:09 VERIFY EKU OK
2025-02-16 01:33:09 VERIFY OK: depth=0, CN=OpenVPN-Server
2025-02-16 01:33:09 Control Channel: TLSv1.2, cipher TLSv1.2 ECDHE-RSA-AES256-GCM-SHA384, 2048 bit RSA
2025-02-16 01:33:09 [OpenVPN-Server] Peer Connection Initiated with [AF_INET]192.168.122.158:1194
2025-02-16 01:33:10 SENT CONTROL [OpenVPN-Server]: 'PUSH_REQUEST' (status=1)
2025-02-16 01:33:10 PUSH: Received control message: 'PUSH_REPLY,ping 20,ping-restart 60,topology subnet,route-gateway 10.99.99.
1,route 192.168.1.0 255.255.255.0,ifconfig 10.99.99.2 255.255.255.0,peer-id 8'
2025-02-16 01:33:10 OPTIONS IMPORT: timers and/or timeouts modified
2025-02-16 01:33:10 OPTIONS IMPORT: --ifconfig/up options modified
2025-02-16 01:33:10 OPTIONS IMPORT: route options modified
2025-02-16 01:33:10 OPTIONS IMPORT: route-related options modified
2025-02-16 01:33:10 OPTIONS IMPORT: peer-id set
2025-02-16 01:33:10 OPTIONS IMPORT: adjusting link_mtu to 1624
2025-02-16 01:33:10 Using peer cipher 'AES-256-CBC'
2025-02-16 01:33:10 Outgoing Data Channel: Cipher 'AES-256-CBC' initialized with 256 bit key
2025-02-16 01:33:10 Outgoing Data Channel: Using 512 bit message hash 'SHA512' for HMAC authentication
2025-02-16 01:33:10 Incoming Data Channel: Cipher 'AES-256-CBC' initialized with 256 bit key
2025-02-16 01:33:10 Incoming Data Channel: Using 512 bit message hash 'SHA512' for HMAC authentication
2025-02-16 01:33:10 ROUTE_GATEWAY 10.0.2.2/255.255.255.0 IFACE=eth0 HWADDR=08:00:27:36:87:b9
2025-02-16 01:33:10 TUN/TAP device tun0 opened
2025-02-16 01:33:10 /sbin/ip link set dev tun0 up mtu 1500
2025-02-16 01:33:10 /sbin/ip link set dev tun0 up
2025-02-16 01:33:10 /sbin/ip addr add dev tun0 10.99.99.2/24
2025-02-16 01:33:10 /sbin/ip route add 192.168.1.0/24 via 10.99.99.1
2025-02-16 01:33:10 GID set to nogroup
2025-02-16 01:33:10 UID set to nobody
2025-02-16 01:33:10 Initialization Sequence Completed
```

OpenVPN Client RouterOS

RouterOS can be OpenVPN Client. From Files, import .ovpn config file with certificate



OpenVPN Client RouterOS



OpenVPN Client RouterOS
(without certificate)

Terminal R1

```
/interface ovpn-client

add name=ovpn-R1 \
connect-to=88.44.22.2
protocol=udp\
user=marco.boschini password=user-secret \
profile=default-encryption
```

NAT problem

- If your router (or VPN Server) is not reachable by public IP
- Some solutions:
  - ZeroTier
  - BackToHome (based on WireGuard protocol)
  - RouterOS as a VPN Server concentrator

ZeroTier

- Available only in ARM/ARM64 architecture
- Free up to 25 devices
- Can be hosted for free in your local environment without the limitation of 25 devices
- Easy to implement
- Not under your complete control
- iOS/Android/PC/MAC/Linux Apps available

BackToHome

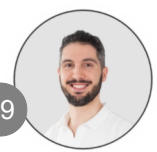
- Available only in ARM/ARM64/TILE architecture
- Very Easy to implement (2 clicks!)
- Based on WireGuard
- MikroTik mantain the relay server for free
- Not under your complete control
- iOS/Android Apps available
- PC/MAC/Linux can use WireGuard client

BackToHome

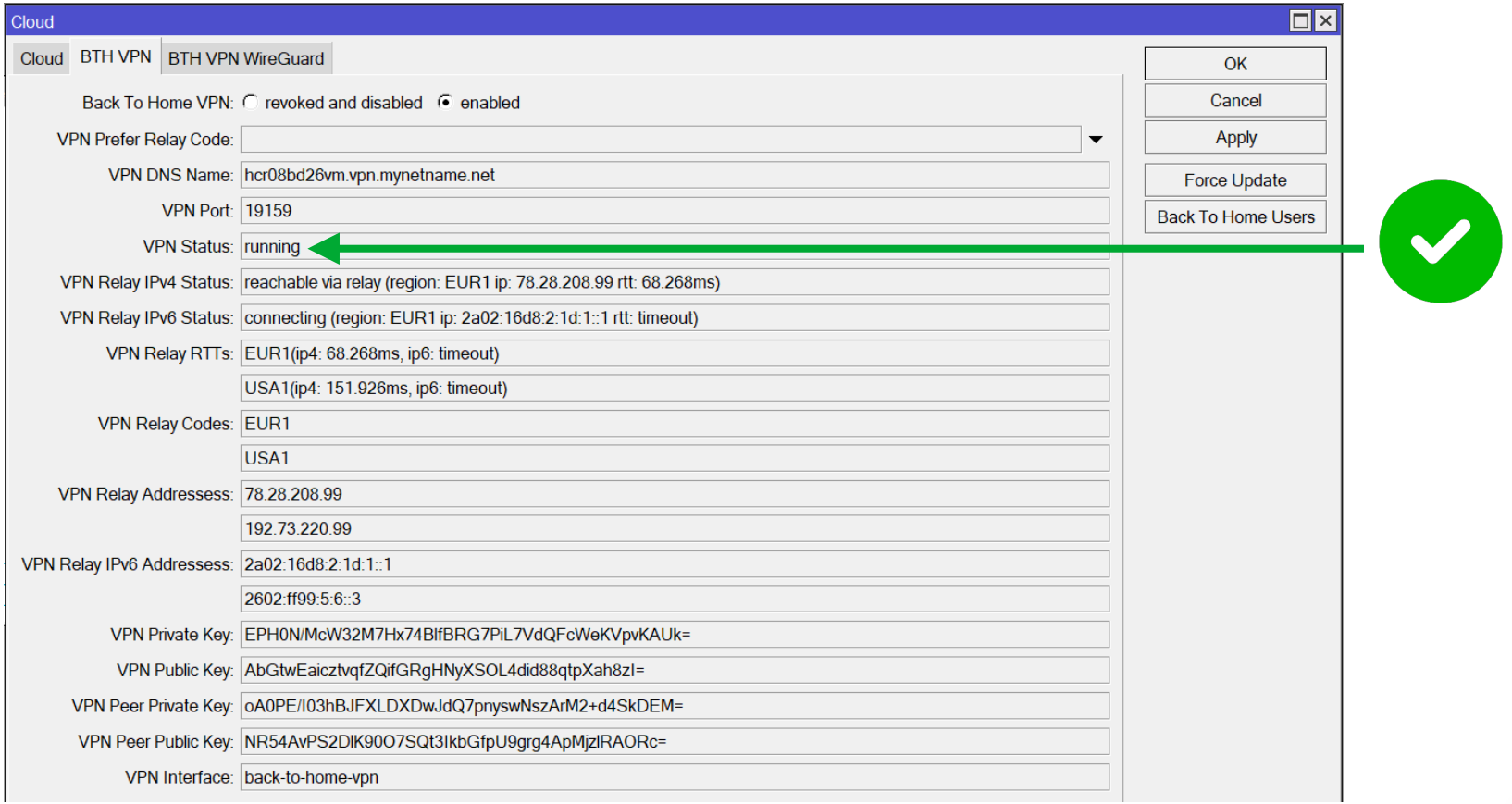
Corsi MikroTik

Terminal R1

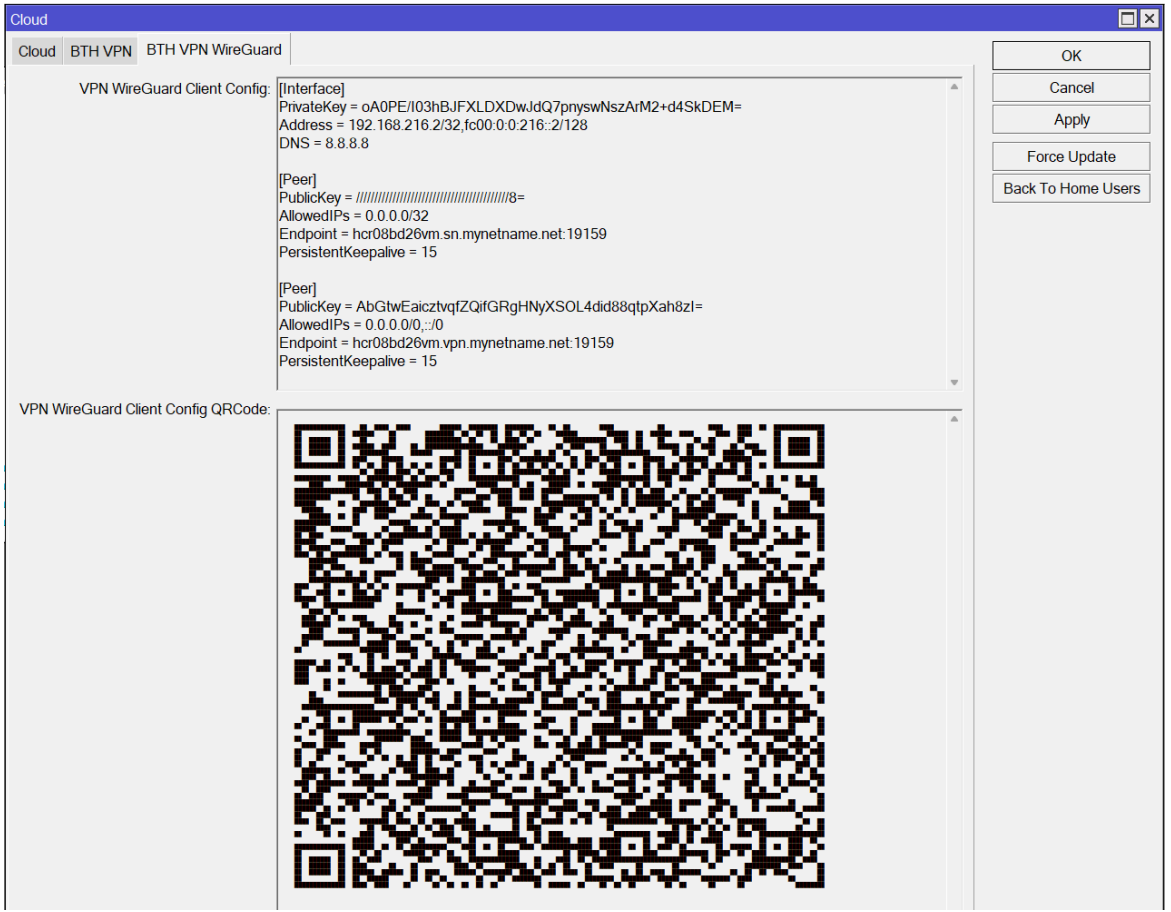
```
/ip cloud

set ddns-enabled=yes

set back-to-home-vpn=enabled
```



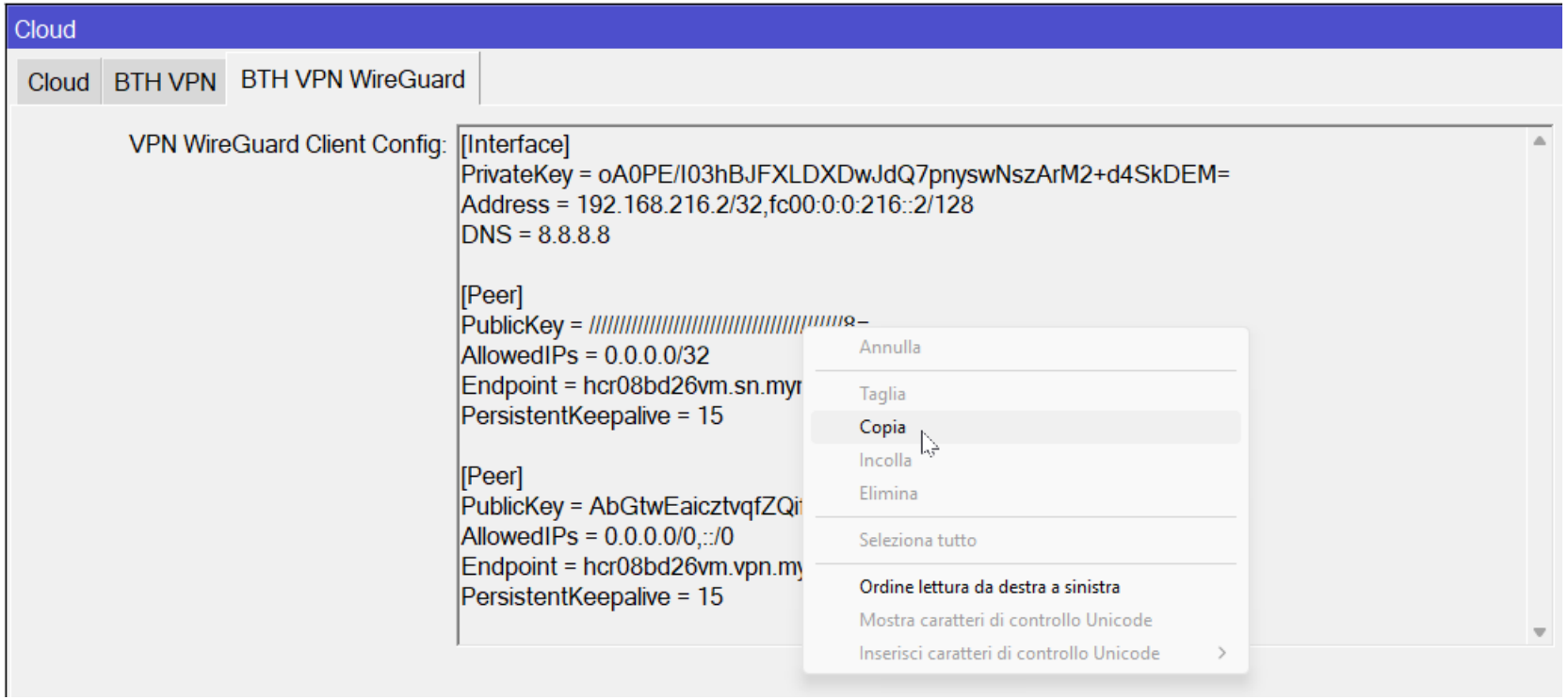
BackToHome


Cloud
Cloud
BTH VPN
BTH VPN WireGuard
OK
Cancel
Apply
Force Update
Back To Home Users
Back To Home VPN: revoked and disabled enabled
VPN Prefer Relay Code:
VPN DNS Name: hcr08bd26vm.vpn.mynetname.net
VPN Port: 19159
VPN Status: running
VPN Relay IPv4 Status: reachable via relay (region: EUR1 ip: 78.28.208.99 rtt: 68.268ms)
VPN Relay IPv6 Status: connecting (region: EUR1 ip: 2a02:16d8:2:1d:1::1 rtt: timeout)
VPN Relay RTTs: EUR1(ip4: 68.268ms, ip6: timeout)
USA1(ip4: 151.926ms, ip6: timeout)
VPN Relay Codes: EUR1
USA1
VPN Relay Addressess: 78.28.208.99
192.73.220.99
VPN Relay IPv6 Addressess: 2a02:16d8:2:1d:1::1
2602:ff99:5:6::3
VPN Private Key: EPH0N/McW32M7Hx74BlfBRG7PiL7VdQFcWeKVpvKAUk=
VPN Public Key: AbGtwEaicztvqfZQifGRgHNyXSOL4did88qtpXah8zI=
VPN Peer Private Key: oA0PE/I03hBJFXLDXDwJdQ7pnyswNszArM2+d4SkDEM=
VPN Peer Public Key: NR54AvPS2DIK90O7SQt3IkbGfpU9grg4ApMjzIRAORc=
VPN Interface: back-to-home-vpn

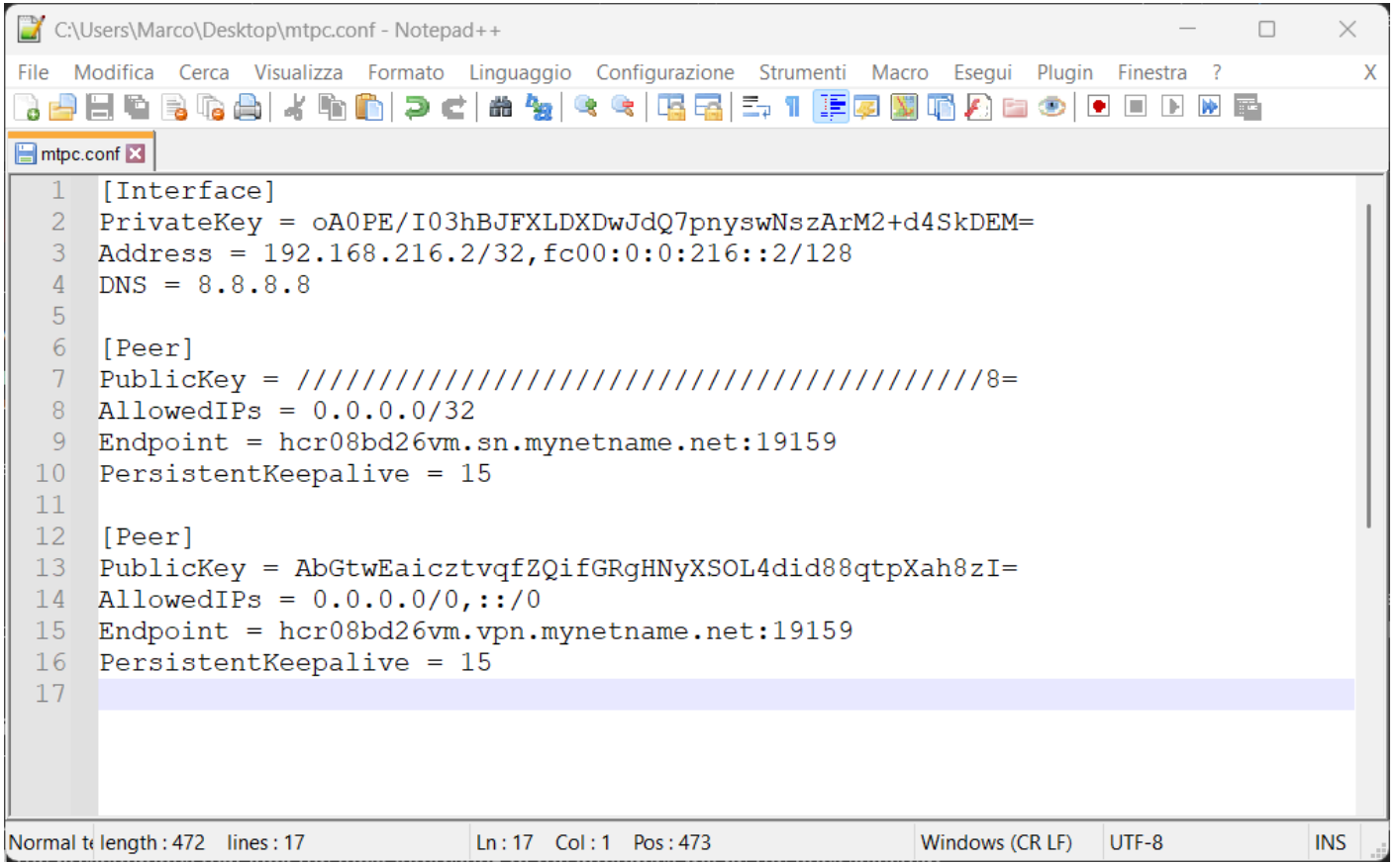
BackToHome


Cloud
Cloud
BTH VPN
BTH VPN WireGuard
OK
Cancel
Apply
Force Update
Back To Home Users
VPN WireGuard Client Config:
[Interface]
PrivateKey = oA0PE/I03hBJFXLDXDwJdQ7pnyswNszArM2+d4SkDEM=
Address = 192.168.216.2/32,fc00:0:0:216::2/128
DNS = 8.8.8.8
[Peer]
PublicKey = //8=
AllowedIPs = 0.0.0.0/32
Endpoint = hcr08bd26vm.sn.mynetname.net:19159
PersistentKeepalive = 15
[Peer]
PublicKey = AbGtwEaicztvqfZQifGRgHNyXSOL4did88qtpXah8zI=
AllowedIPs = 0.0.0.0/0,::/0
Endpoint = hcr08bd26vm.vpn.mynetname.net:19159
PersistentKeepalive = 15
VPN WireGuard Client Config QRCode:

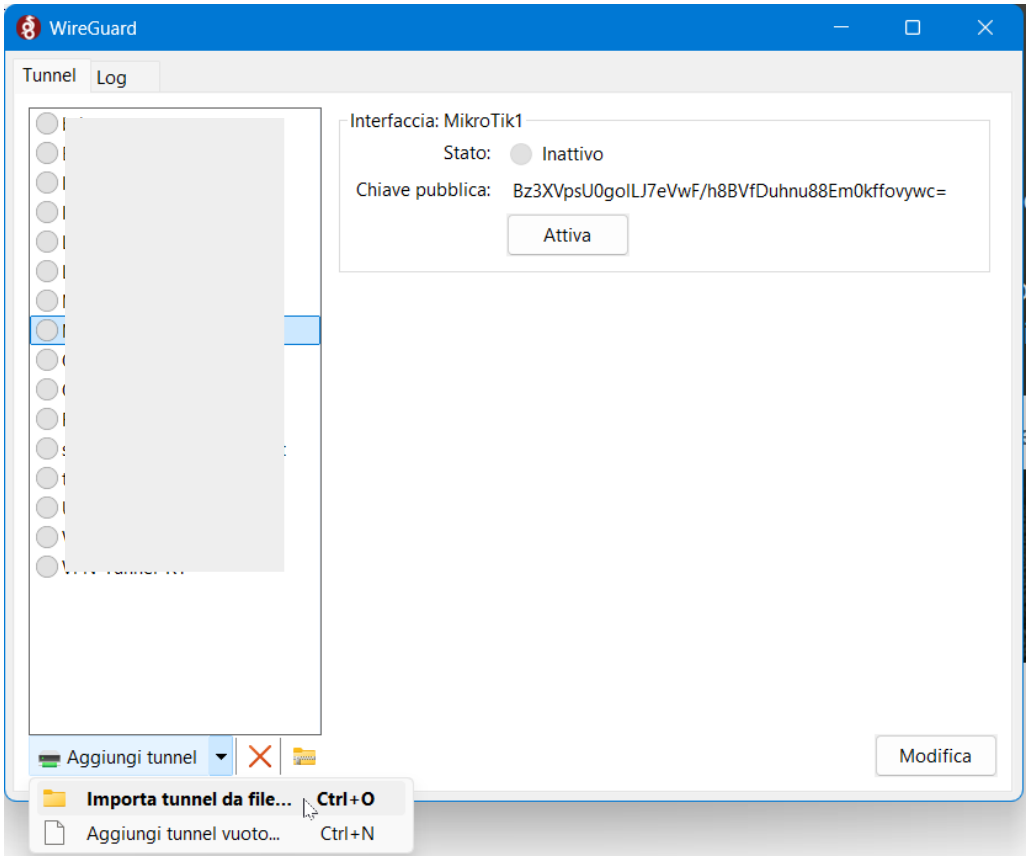
BackToHome


Cloud
Cloud
BTH VPN
BTH VPN WireGuard
VPN WireGuard Client Config:
[Interface]
PrivateKey = oA0PE/I03hBJFXLDXDwJdQ7pnyswNszArM2+d4SkDEM=
Address = 192.168.216.2/32,fc00:0:0:216::2/128
DNS = 8.8.8.8
[Peer]
PublicKey =
AllowedIPs = 0.0.0.0/32
Endpoint = hcr08bd26vm.sn.myr
PersistentKeepalive = 15
[Peer]
PublicKey = AbGtwEaicztvqfZQi
AllowedIPs = 0.0.0.0/0,::/0
Endpoint = hcr08bd26vm.vpn.my
PersistentKeepalive = 15
Annulla
Taglia
Copia
Incolla
Elimina
Seleziona tutto
Ordine lettura da destra a sinistra
Mostra caratteri di controllo Unicode
Inserisci caratteri di controllo Unicode

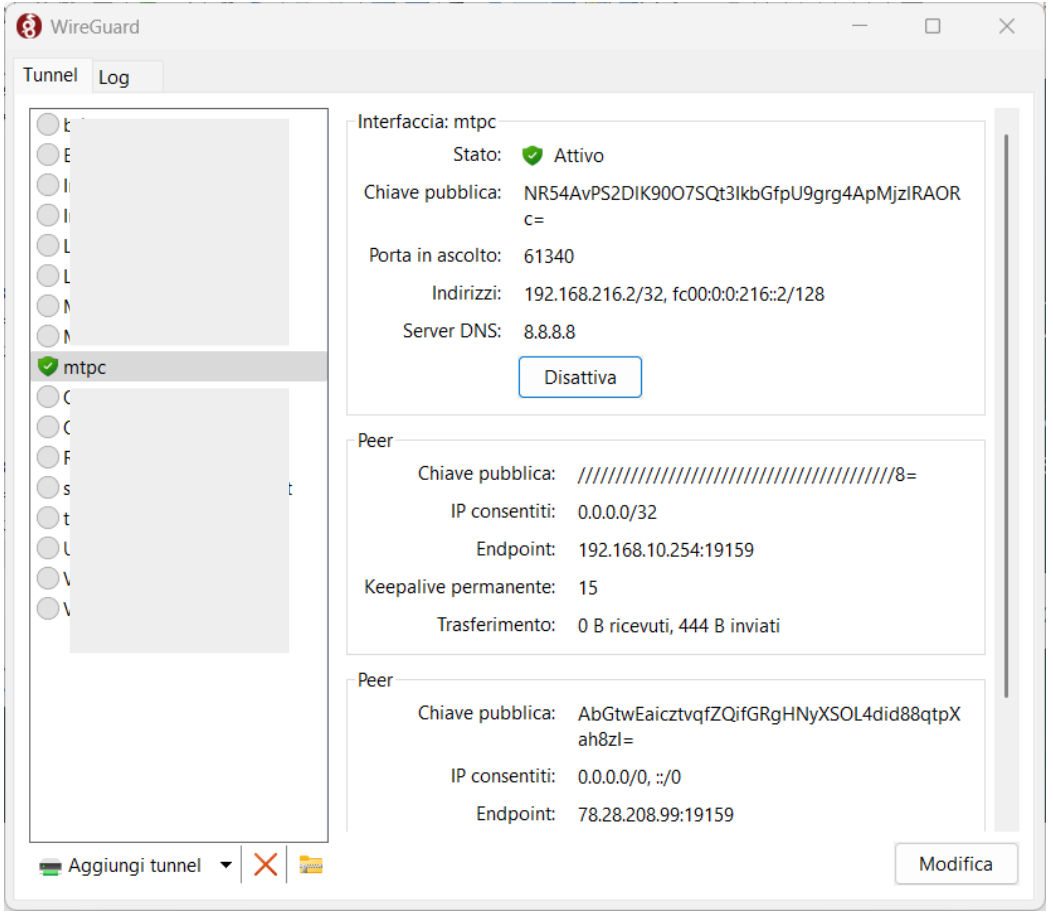
BackToHome


C:\Users\Marco\Desktop\mtpc.conf - Notepad++
File Modifica Cerca Visualizza Formato Linguaggio Configurazione Strumenti Macro Esegui Plugin Finestra ?
mtpc.conf
[Interface]
PrivateKey = oA0PE/I03hBJFXLDXDwJdQ7pnyswNszArM2+d4SkDEM=
Address = 192.168.216.2/32,fc00:0:0:216::2/128
DNS = 8.8.8.8
[Peer]
PublicKey = //8=
AllowedIPs = 0.0.0.0/32
Endpoint = hcr08bd26vm.sn.mynetname.net:19159
PersistentKeepalive = 15
[Peer]
PublicKey = AbGtwEaicztvqfZQifGRgHNyXSOL4did88qtpXah8zI=
AllowedIPs = 0.0.0.0/0,::/0
Endpoint = hcr08bd26vm.vpn.mynetname.net:19159
PersistentKeepalive = 15
length : 472 lines : 17
Ln : 17 Col : 1 Pos : 473
Windows (CR LF)
UTF-8
INS

BackToHome


WireGuard
Tunnel
Log
Interfaccia: MikroTik1
Stato:
Inattivo
Chiave pubblica:
Bz3XVpsU0goILJ7eVwF/h8BVfDuhnu88Em0kffovywc=
Attiva
Aggiungi tunnel
Modifica
Importa tunnel da file... Ctrl+O
Aggiungi tunnel vuoto... Ctrl+N

BackToHome


WireGuard
Tunnel
Log
mtpc
Interfaccia: mtpc
Stato: Attivo
Chiave pubblica: NR54AvPS2DIK90O7SQt3IkbGfpU9grg4ApMjzIRAORc=
Porta in ascolto: 61340
Indirizzi: 192.168.216.2/32, fc00:0:0:216::2/128
Server DNS: 8.8.8.8
Disattiva
Peer
Chiave pubblica: //8=
IP consentiti: 0.0.0.0/32
Endpoint: 192.168.10.254:19159
Keepalive permanente: 15
Trasferimento: 0 B ricevuti, 444 B inviati
Peer
Chiave pubblica: AbGtwEaicztvqfZQifGRgHNyXSOL4did88qtpXah8zI=
IP consentiti: 0.0.0.0/0, ::/0
Endpoint: 78.28.208.99:19159
Aggiungi tunnel
Modifica

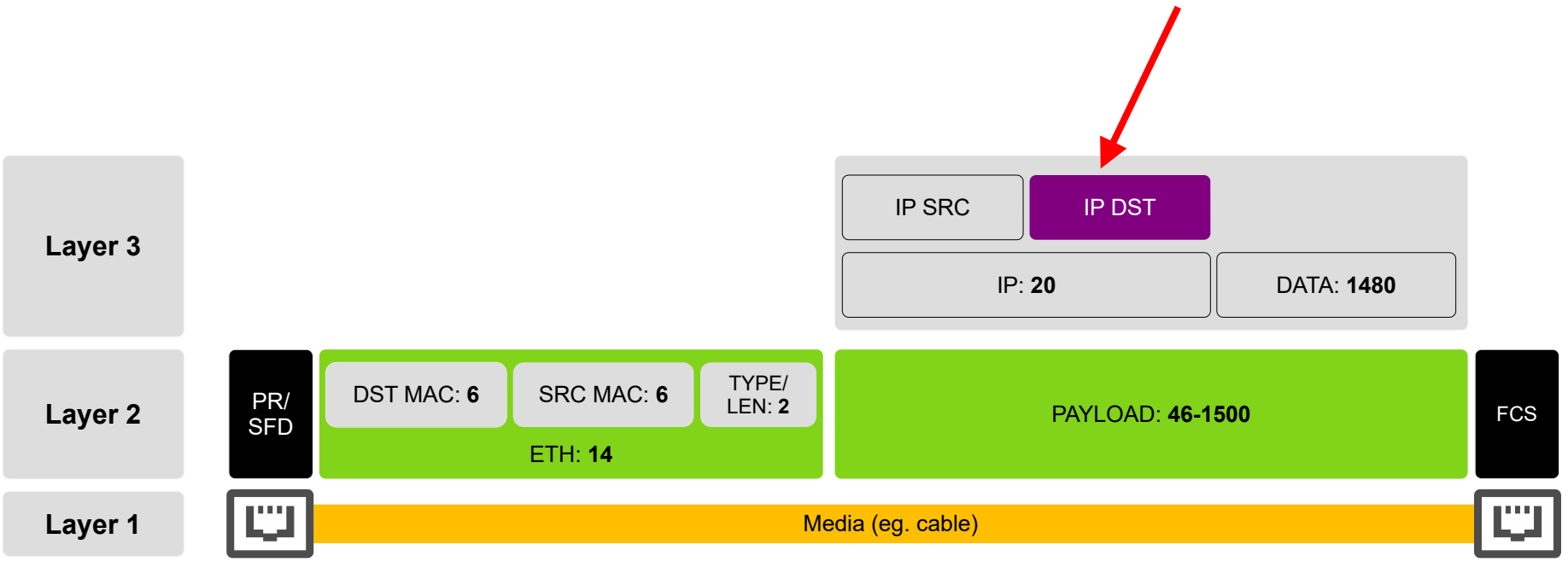
RouterOS as a VPN Server

Corsi MikroTik

- Available with ALL architecture
- You can build a VPN Server with RouterOS
- With Physical device
  (any device with RouterOS)
- With Virtual device
  (CHR)
- The options available are the same for every devices!

Basic routing concepts

A router forwards packets by looking at the **destination IP** address



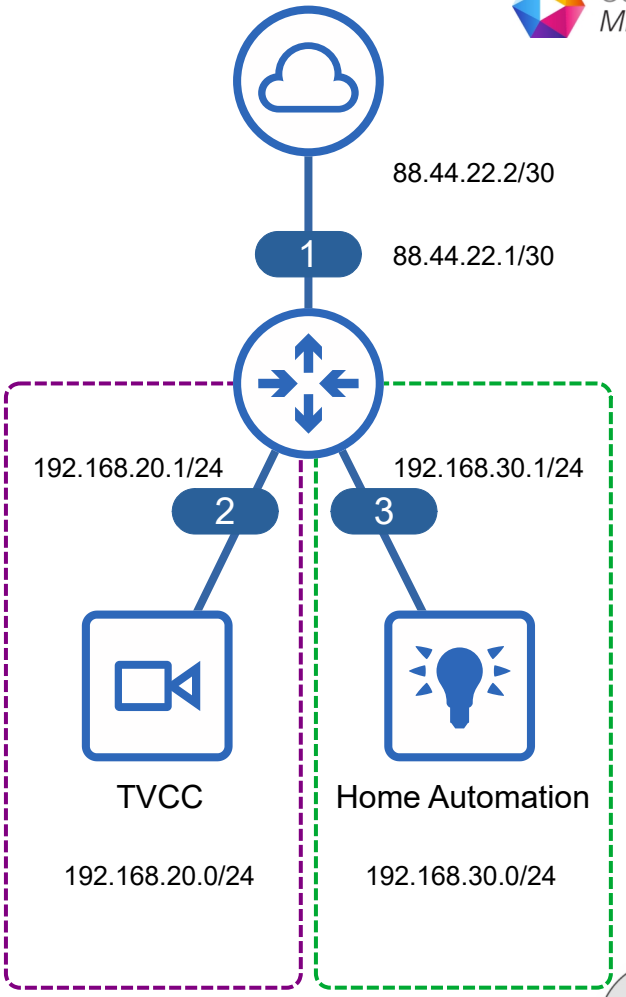
(field size not in scale)

Basic routing concepts

A router searches for the interface where to forward out the packets **inside his routing table**, selecting the best path.

Terminal R1

```
[admin@R1] /ip/route> print

#      DST-ADDRESS       GATEWAY     DISTANCE
0  As  0.0.0.0/0         88.44.22.2         1
   DAc 88.44.22.0/30     ether1             0
   DAc 192.168.20.0/24   ether2             0
   DAc 192.168.30.0/24   ether3             0
```



Basic routing concepts

- If a home automation device (192.168.30.30) send a request to TVCC device (eg. 192.168.20.20) the router receive an IP packets with this IP address fields inside the IP header:

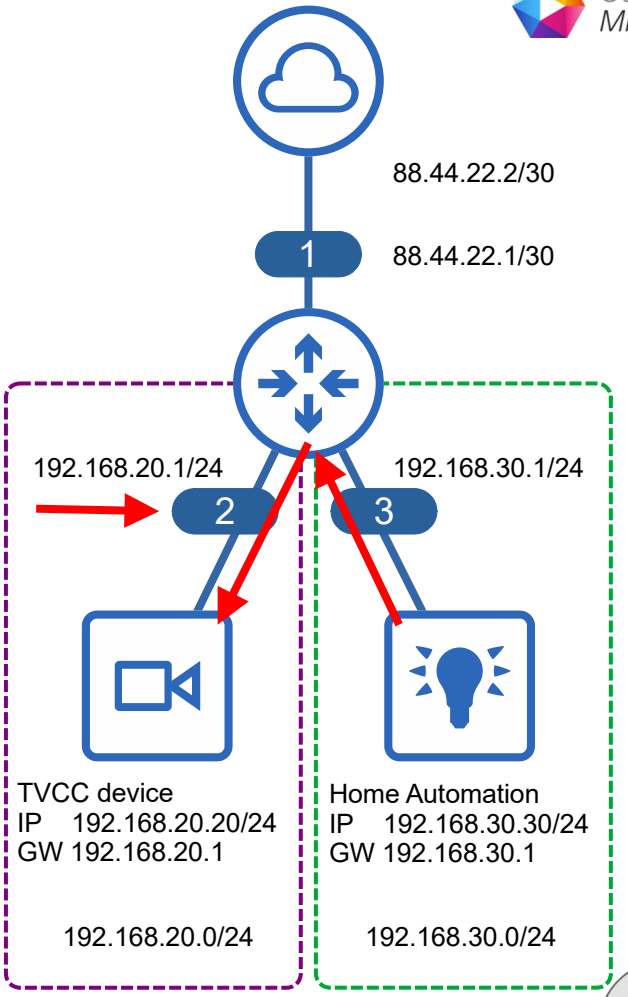
- Router looks at the destination IP address and select where to forward the outgoing packet by selecting the interface that leads to the most accurate destination network

Basic routing concepts

- Router select to forward packets out to ether2

Terminal R1

```
[admin@R1] /ip/route> print

#      DST-ADDRESS       GATEWAY     DISTANCE
0  As  0.0.0.0/0         88.44.22.2         1
   DAc 88.44.22.0/30     ether1             0
   DAc 192.168.20.0/24   ether2             0
   DAc 192.168.30.0/24   ether3             0
```



Basic routing concepts

- When **TVCC** device (192.168.30.30) **respond back** to the sender home automation device (192.168.20.20) the router receives IP packets with these IP address fields inside the IP header:

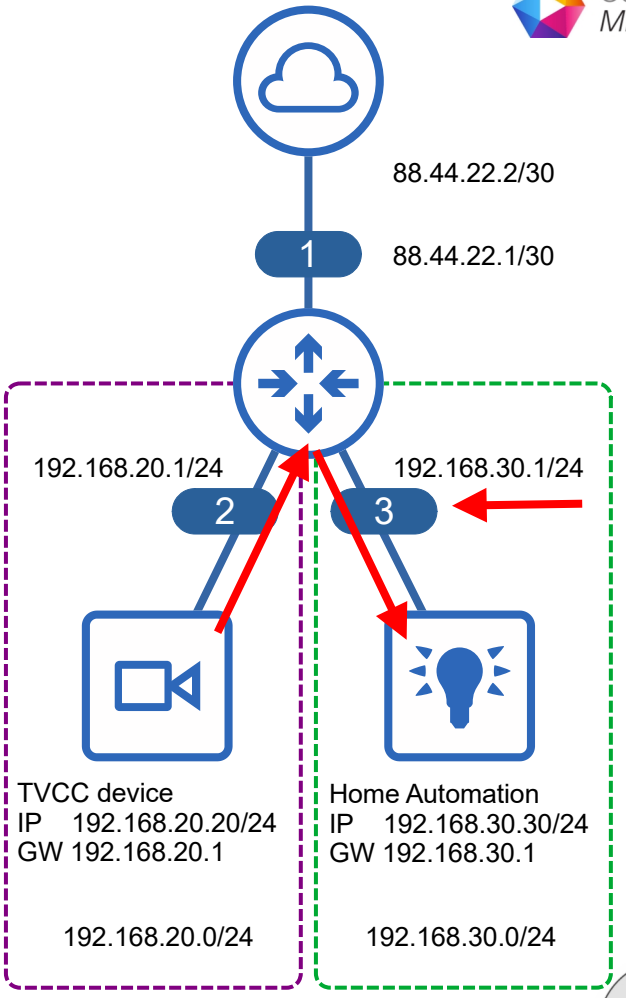
- Router looks at the destination IP address and select where to forward the outgoing packet by selecting the interface that leads to the most accurate destination network

Basic routing concepts

- Router select to forward packets out to ether3

Terminal R1

```
[admin@R1] /ip/route> print

#      DST-ADDRESS       GATEWAY     DISTANCE
0  As  0.0.0.0/0         88.44.22.2         1
   DAc 88.44.22.0/30     ether1             0
   DAc 192.168.20.0/24   ether2             0
   DAc 192.168.30.0/24   ether3             0
```

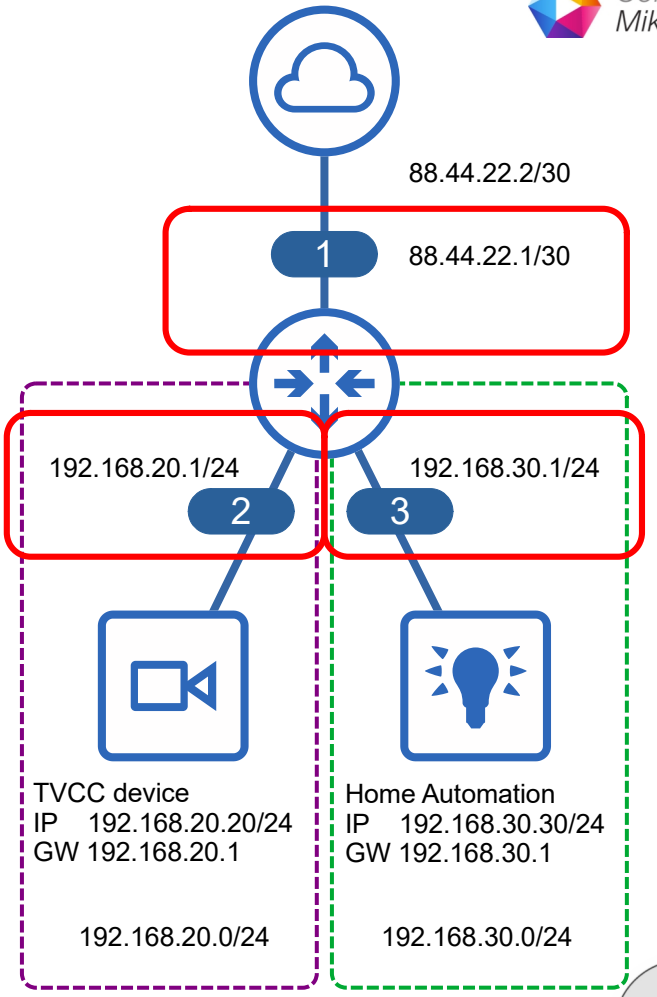


Basic routing concepts

- These Routes are **Connected**, because the router has an interface with an address that belongs to that network. There is no distance.

Terminal R1

```
[admin@R1] /ip/route> print

#      DST-ADDRESS       GATEWAY     DISTANCE
0  As  0.0.0.0/0         88.44.22.2         1
   DAc 88.44.22.0/30     ether1             0
   DAc 192.168.20.0/24   ether2             0
   DAc 192.168.30.0/24   ether3             0
```

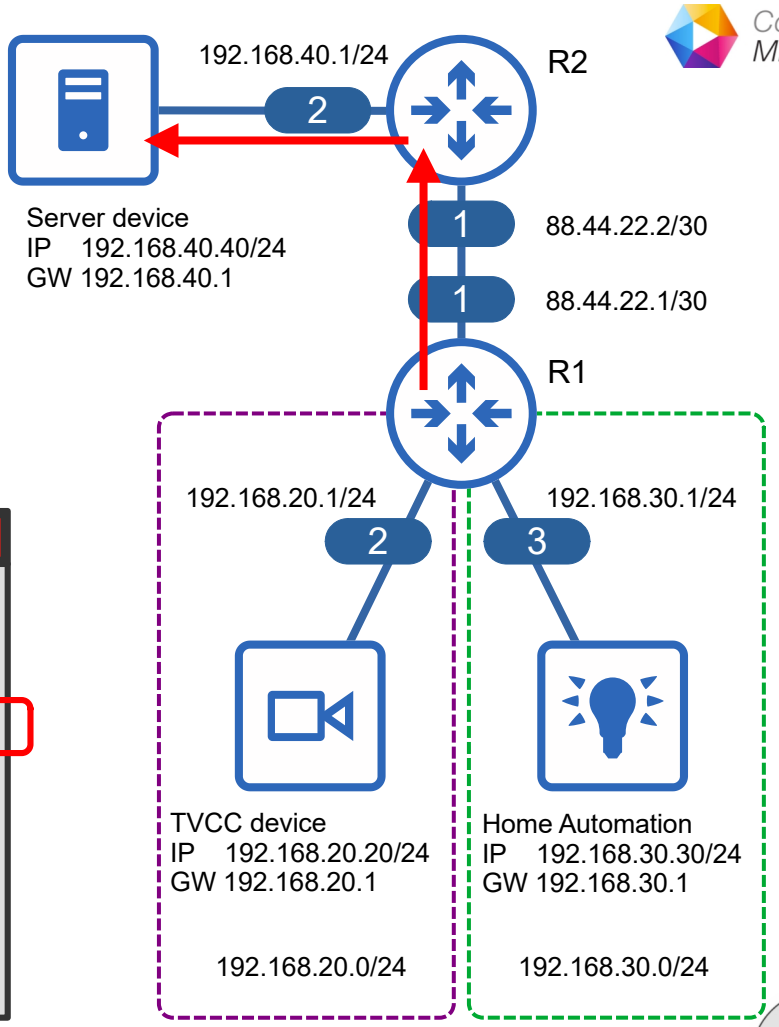


Basic routing concepts

- This route **is not connected**, because it require to send the packets to another router (next-hop) to reach the destination. **There's a distance more than 0**.

Terminal R1

```
[admin@R1] /ip/route> print

#     DST-ADDRESS       GATEWAY     DISTANCE
0  As 0.0.0.0/0         88.44.22.2         1
  DAc 88.44.22.0/30     ether1             0
  DAc 192.168.20.0/24   ether2             0
  DAc 192.168.30.0/24   ether3             0
```

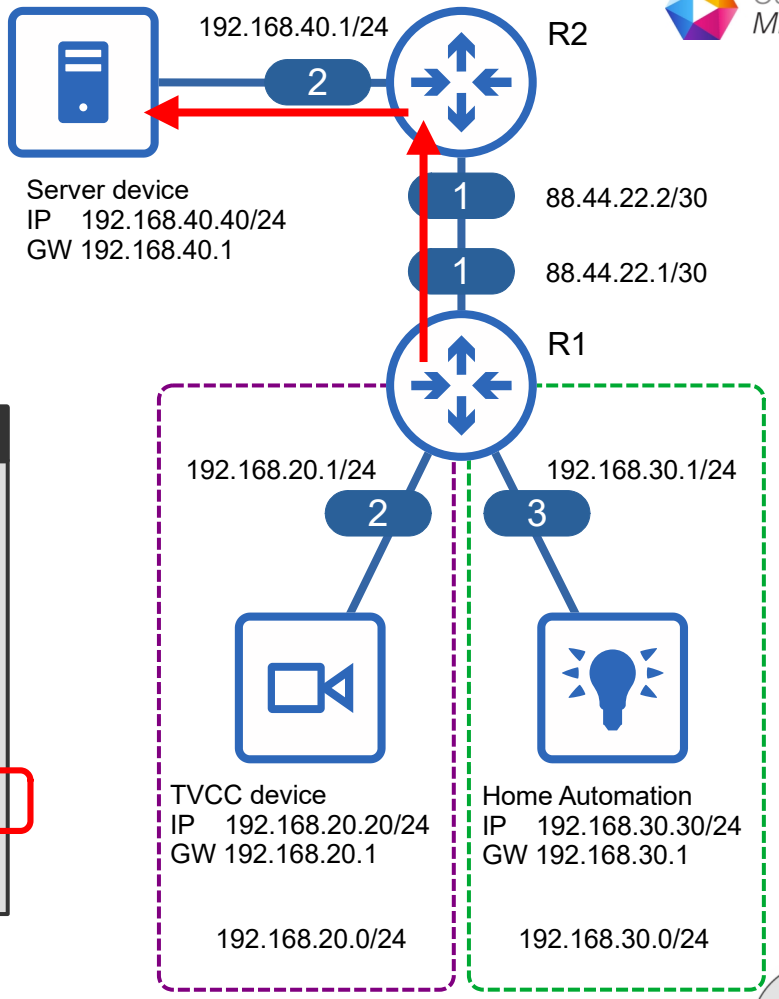


Basic routing concepts

- I can set a more specific route

Terminal R1

```
[admin@R1] /ip/route> print

#      DST-ADDRESS       GATEWAY      DISTANCE
0  As  0.0.0.0/0         88.44.22.2          1
   DAc 88.44.22.0/30     ether1              0
   DAc 192.168.20.0/24   ether2              0
   DAc 192.168.30.0/24   ether3              0
1  As  192.168.40.0/24   88.44.22.2          1
```

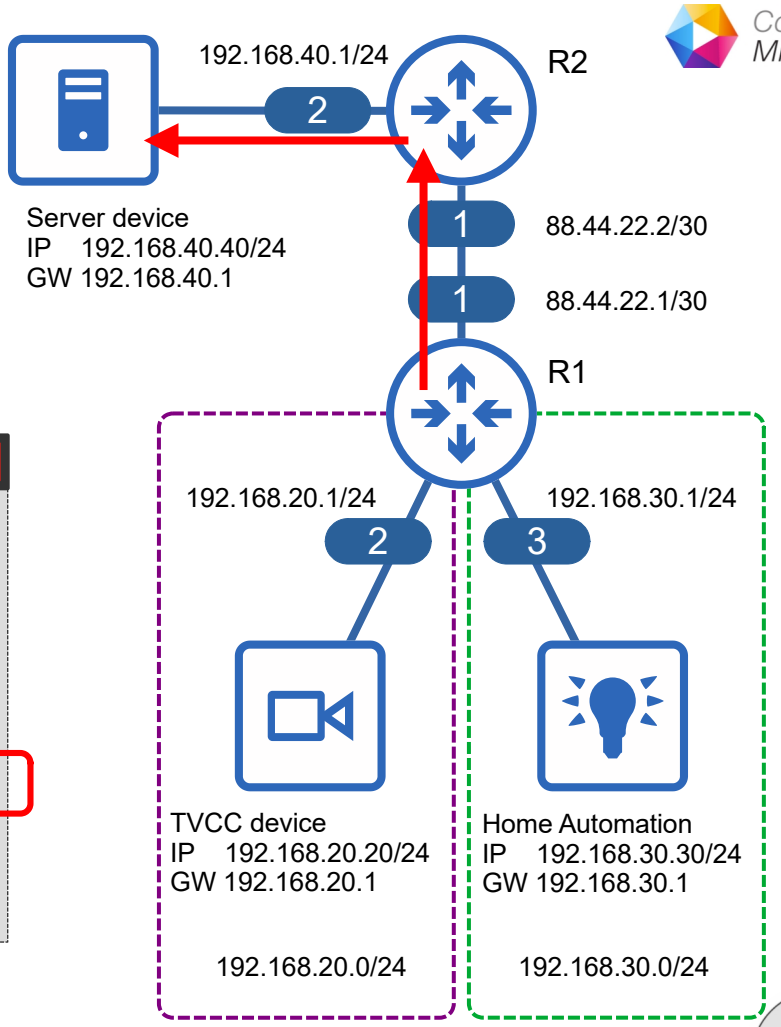


Basic routing concepts

- Or only specific route, without default route.

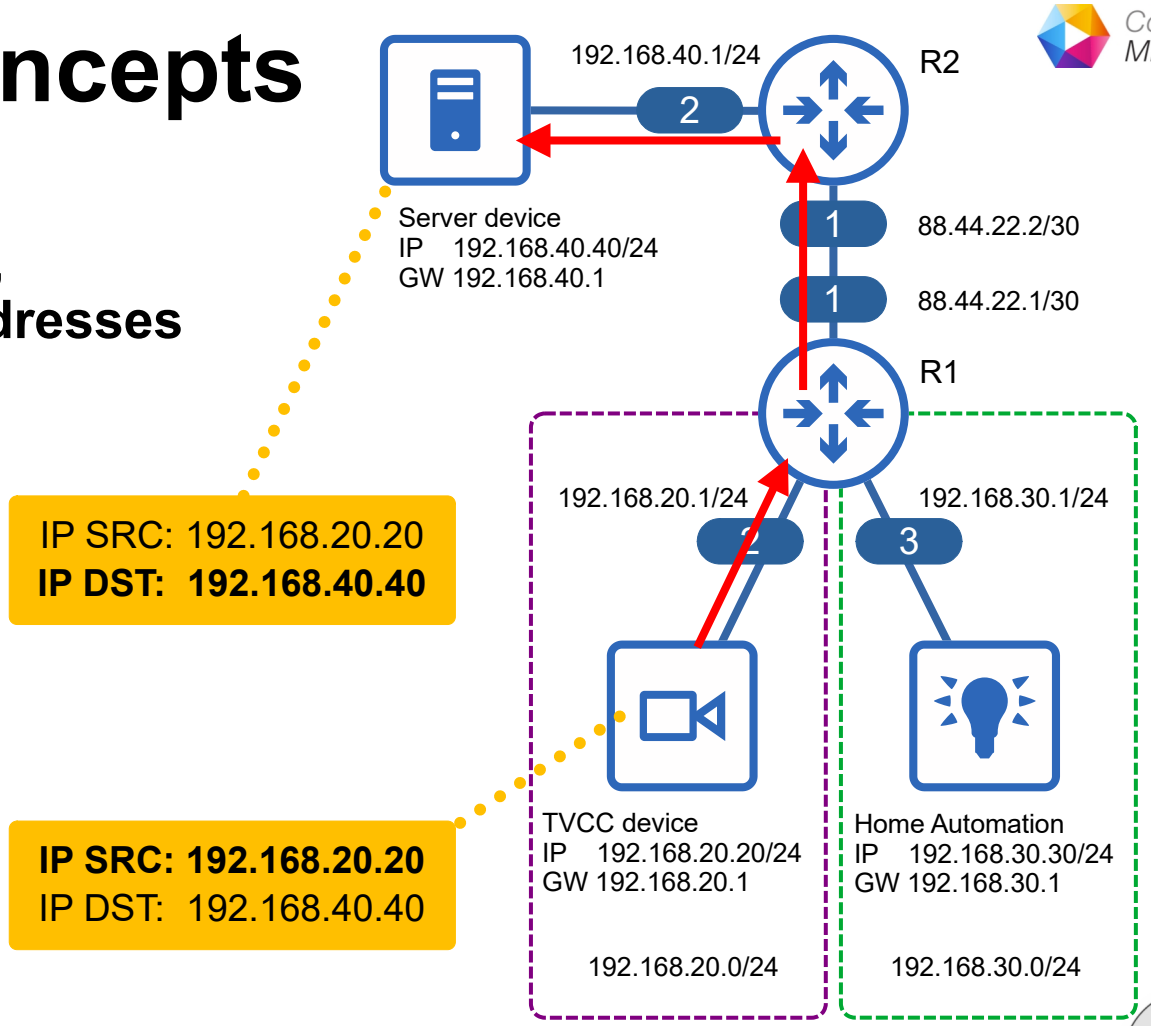
Terminal R1

```
[admin@R1] /ip/route> print

#      DST-ADDRESS        GATEWAY      DISTANCE
  DAc  88.44.22.0/30      ether1              0
  DAc  192.168.20.0/24    ether2              0
  DAc  192.168.30.0/24    ether3              0
1  As  192.168.40.0/24    88.44.22.2          1
```



Basic routing concepts

During the forwarding process,
routers do not change IP addresses
inside the IP packets.

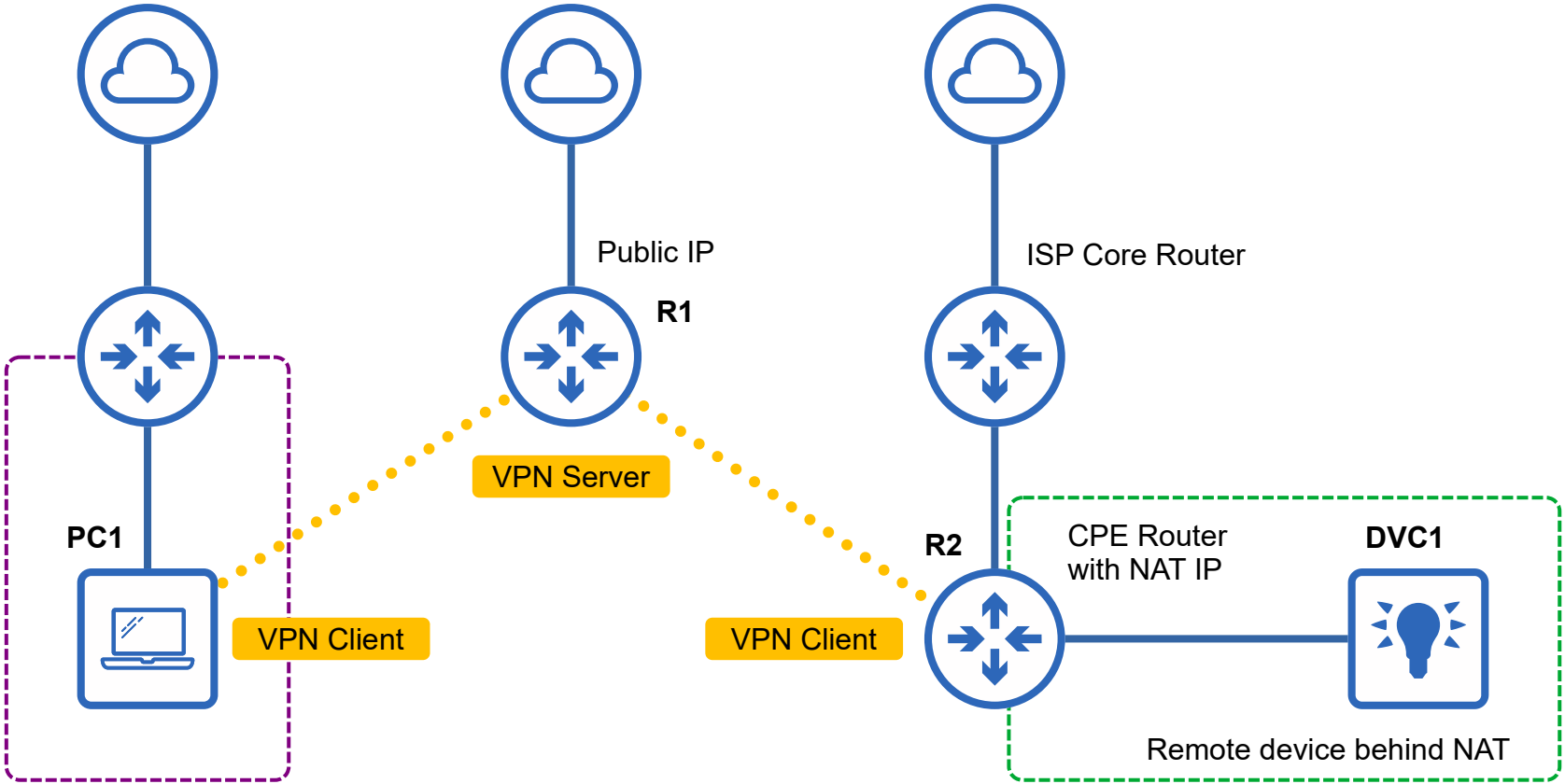


MikroTik as a VPN Client (main router)

Corsi MikroTik

- The router is the main router of the network
- Every device is reachable because the router is the default gateway
- To install the VPN Client it is required to change/add the main router of the network
- The installation can be difficult

RouterOS as VPN concentrator

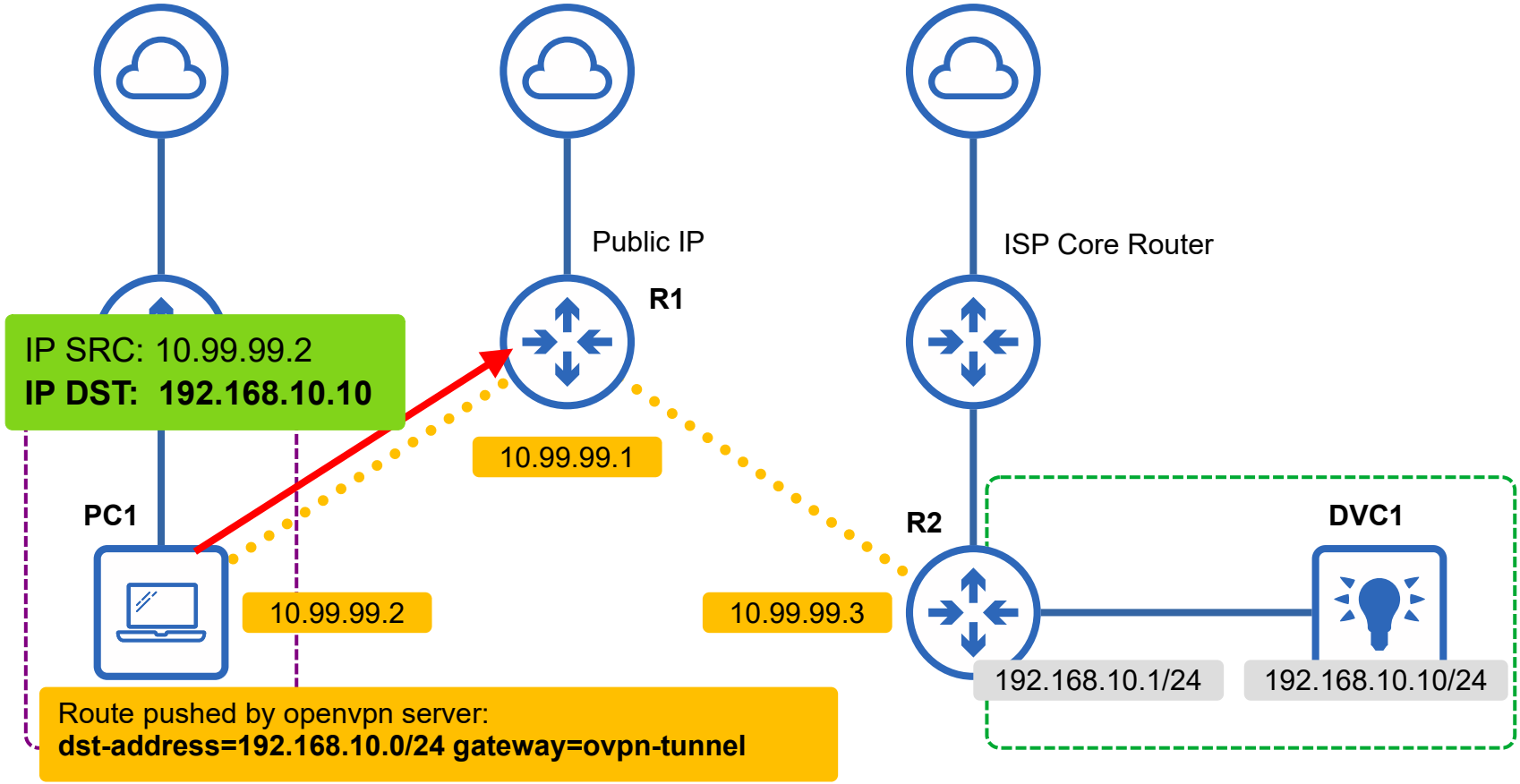


OpenVPN Client RouterOS
(without certificate)

Terminal R2

```
/interface ovpn-client

add name=ovpn-R1 \
connect-to=88.44.22.2
protocol=udp\
user=R2 password=my,Strong,Secret,88 \
route-nopull=yes \
profile=default-encryption
```

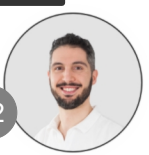
RouterOS as VPN concentrator



Static route

Terminal R1

```
[marco@R1] /ip route
add dst-address=192.168.10.0/24 gateway=10.99.99.3

[marco@R1] /ip route print
#      DST-ADDRESS        GATEWAY                    DISTANCE
0    As 0.0.0.0/0          ether1                            1
   DAc 88.44.22.0/24       ether1                            0
1    As 192.168.10.0/24    10.99.99.3                        1
   DAc 10.99.99.2/32       <ovpn-marco.boschini>             0
   DAc 10.99.99.3/32       <ovpn-R2>                         0
```

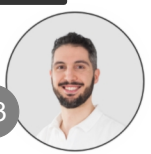


Dynamic Route from /ppp secret

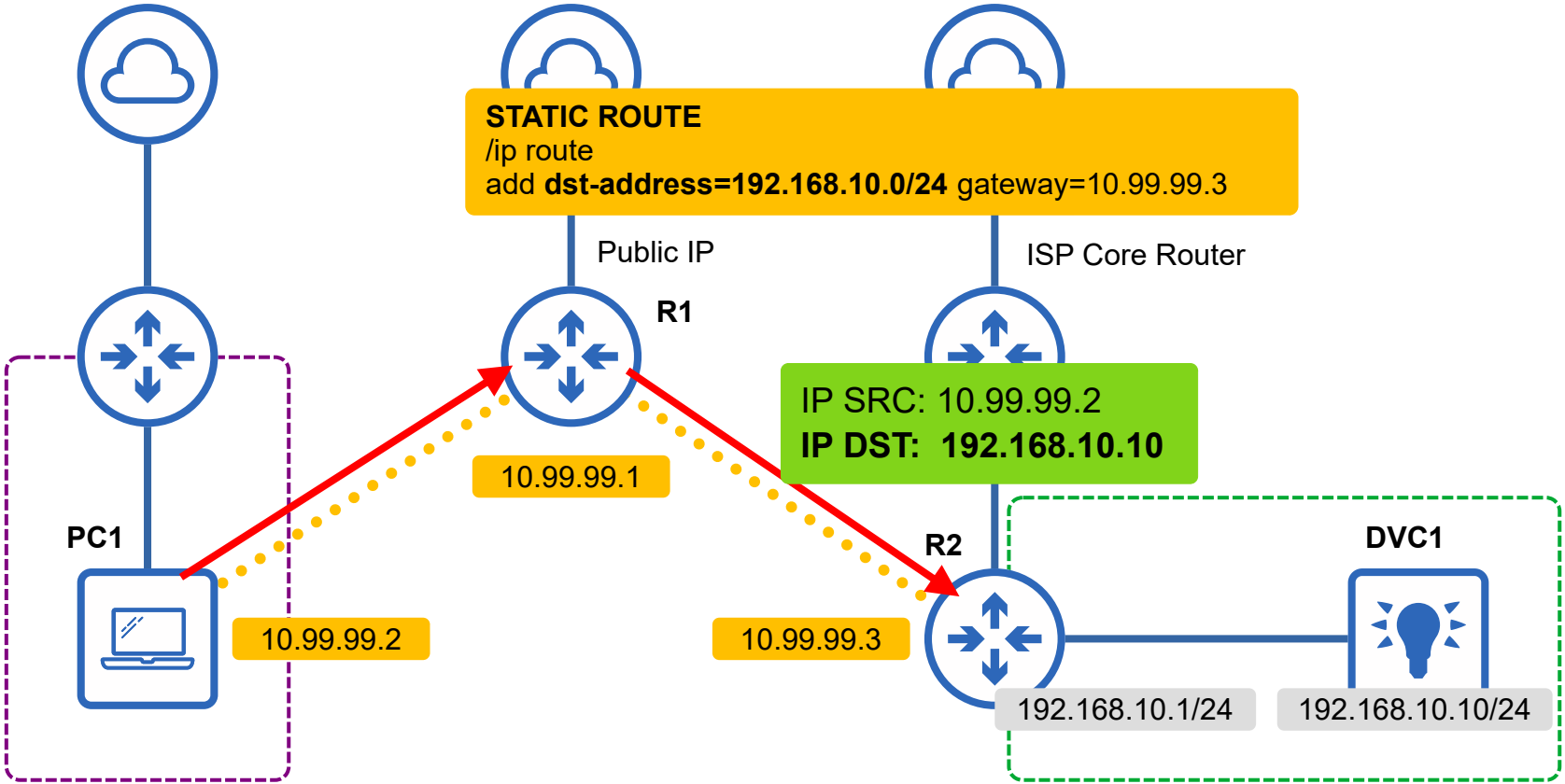
Corsi MikroTik

Terminal R1

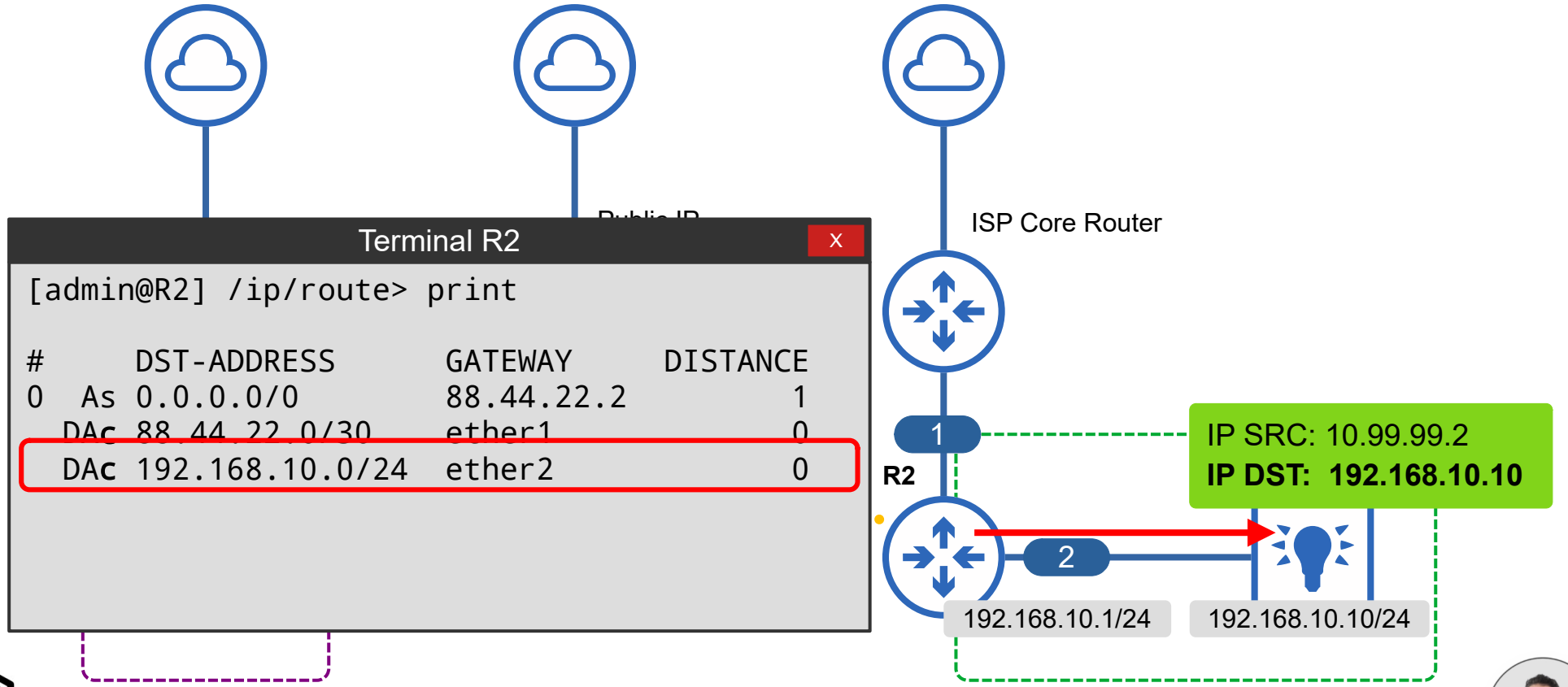
```
[marco@R1] /ppp secret
add name=R2 password=my,Strong,Secret,88 routes=192.168.10.0/24

[marco@R1] /ip route print
#      DST-ADDRESS        GATEWAY                    DISTANCE
0    As 0.0.0.0/0          ether1                           1
   DAc 88.44.22.0/24       ether1                           0
   DAv 192.168.10.0/24     <ovpn-R2>                        1
   DAc 10.99.99.2/32       <ovpn-marco.boschini>            0
   DAc 10.99.99.3/32       <ovpn-R2>                        0
```

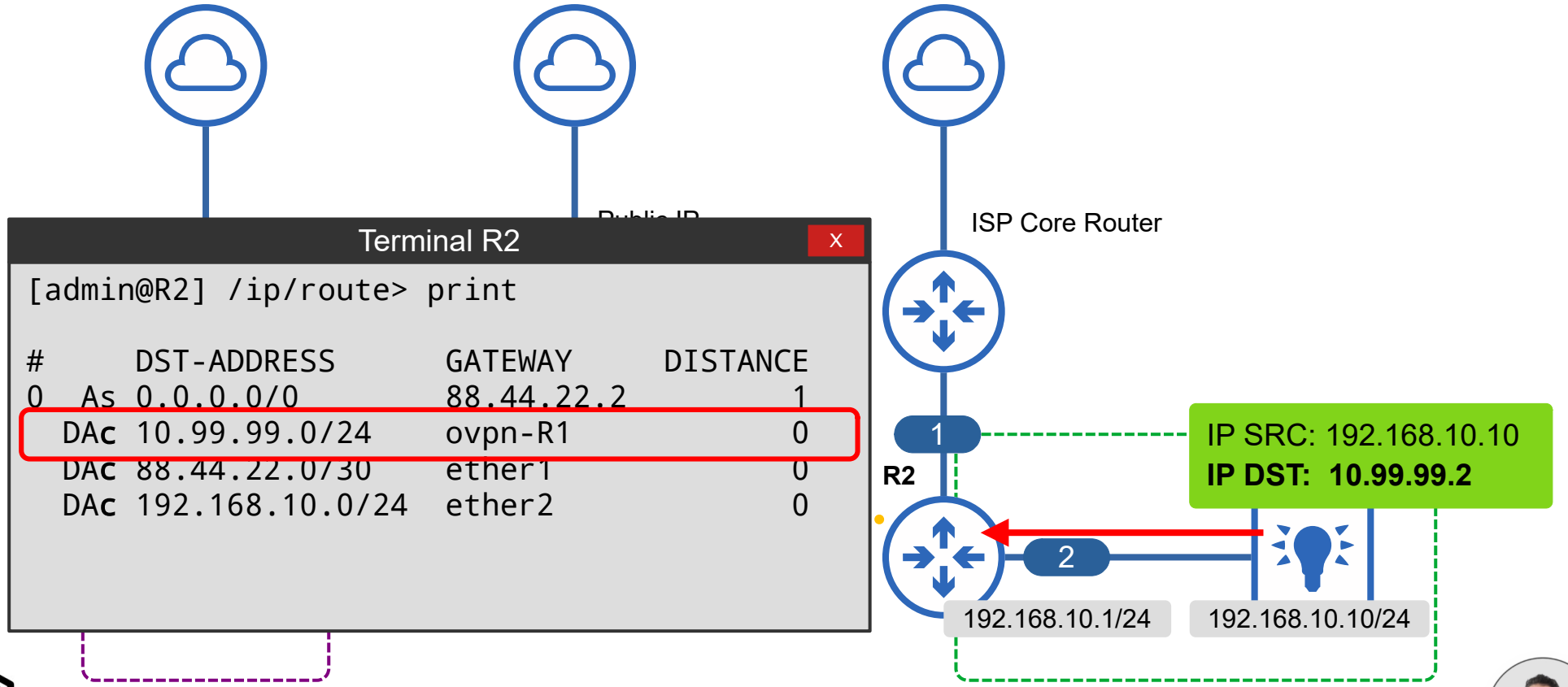
RouterOS as VPN concentrator



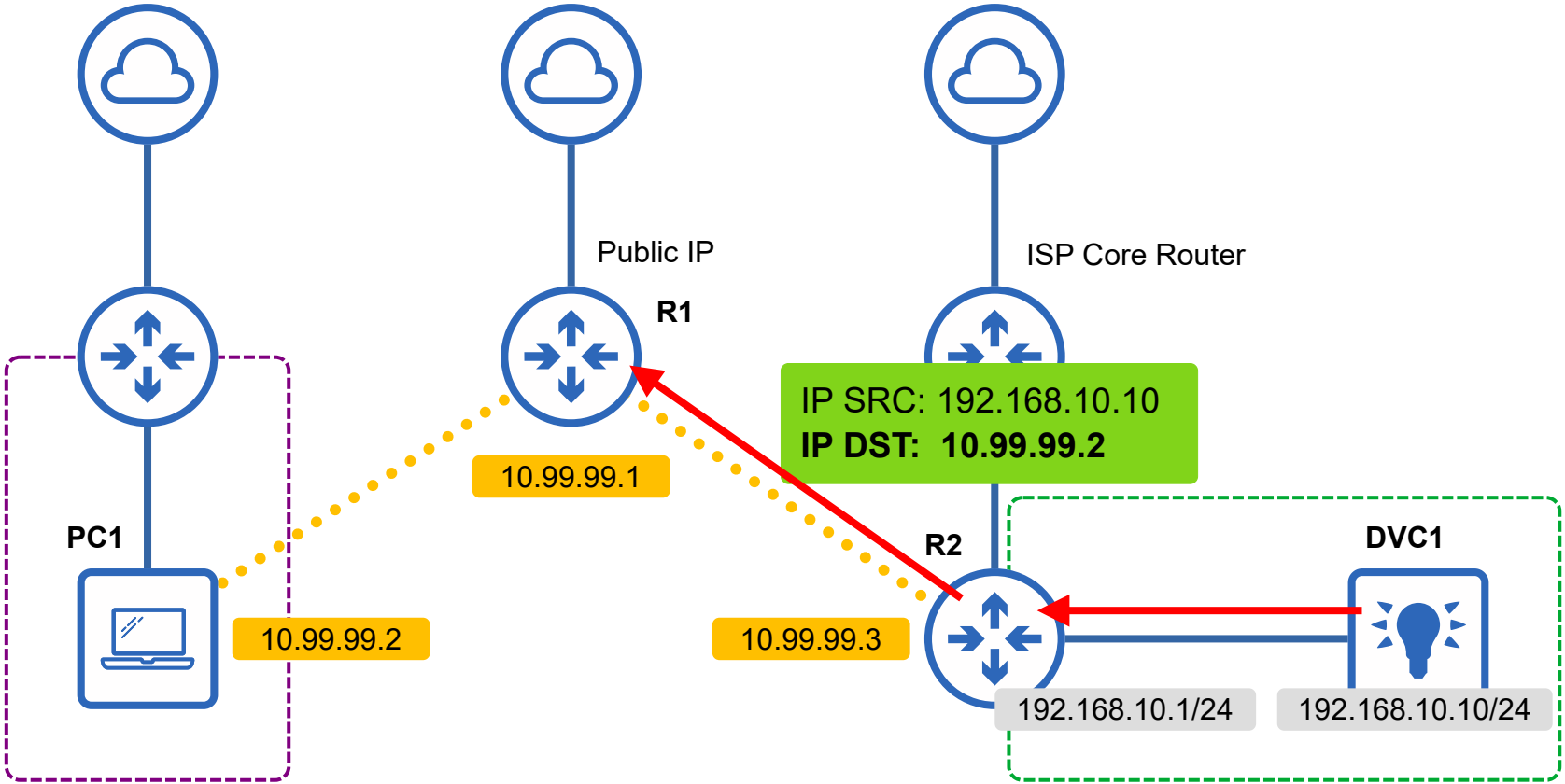
RouterOS as VPN concentrator



RouterOS as VPN concentrator

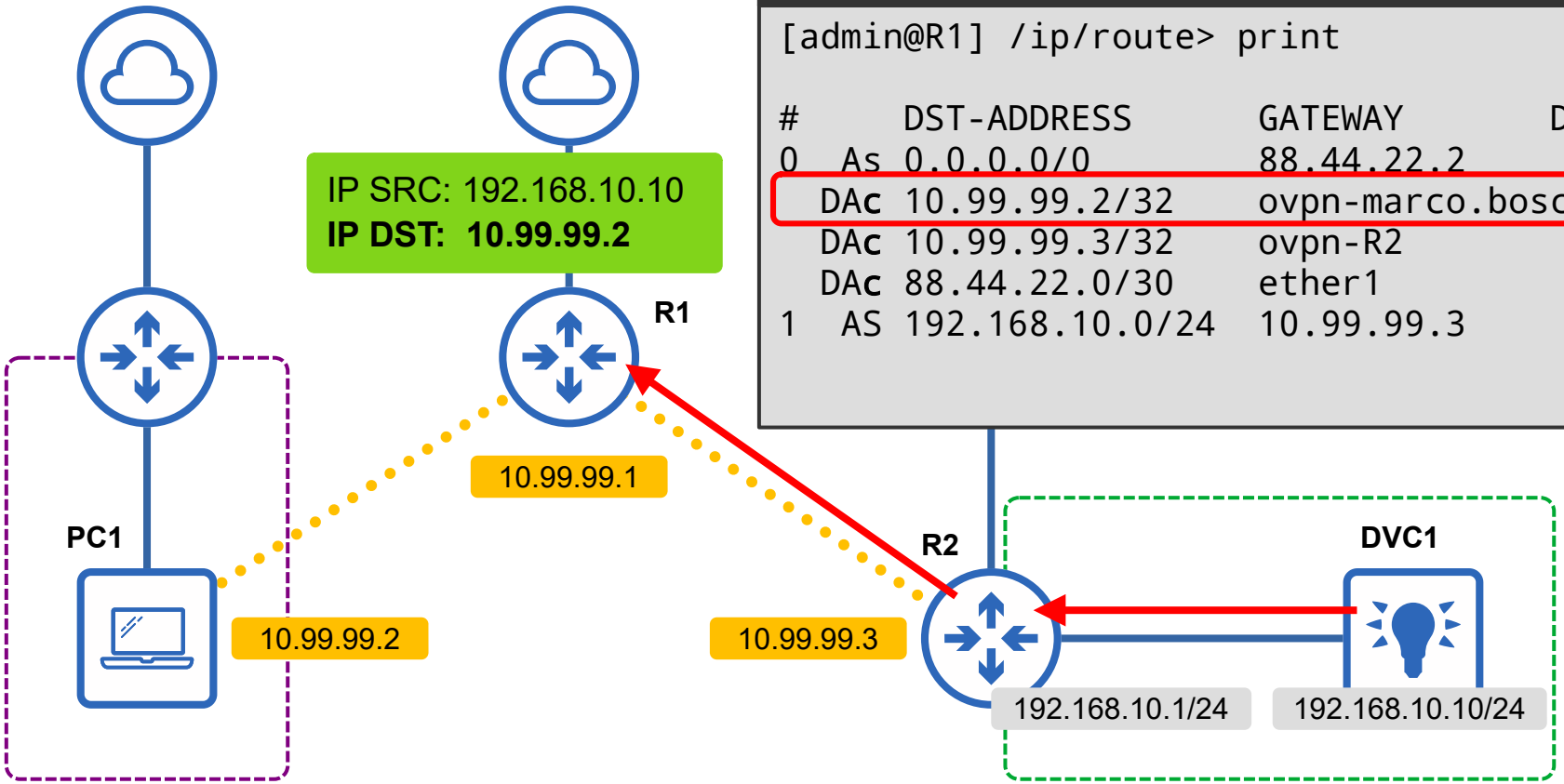


RouterOS as VPN concentrator

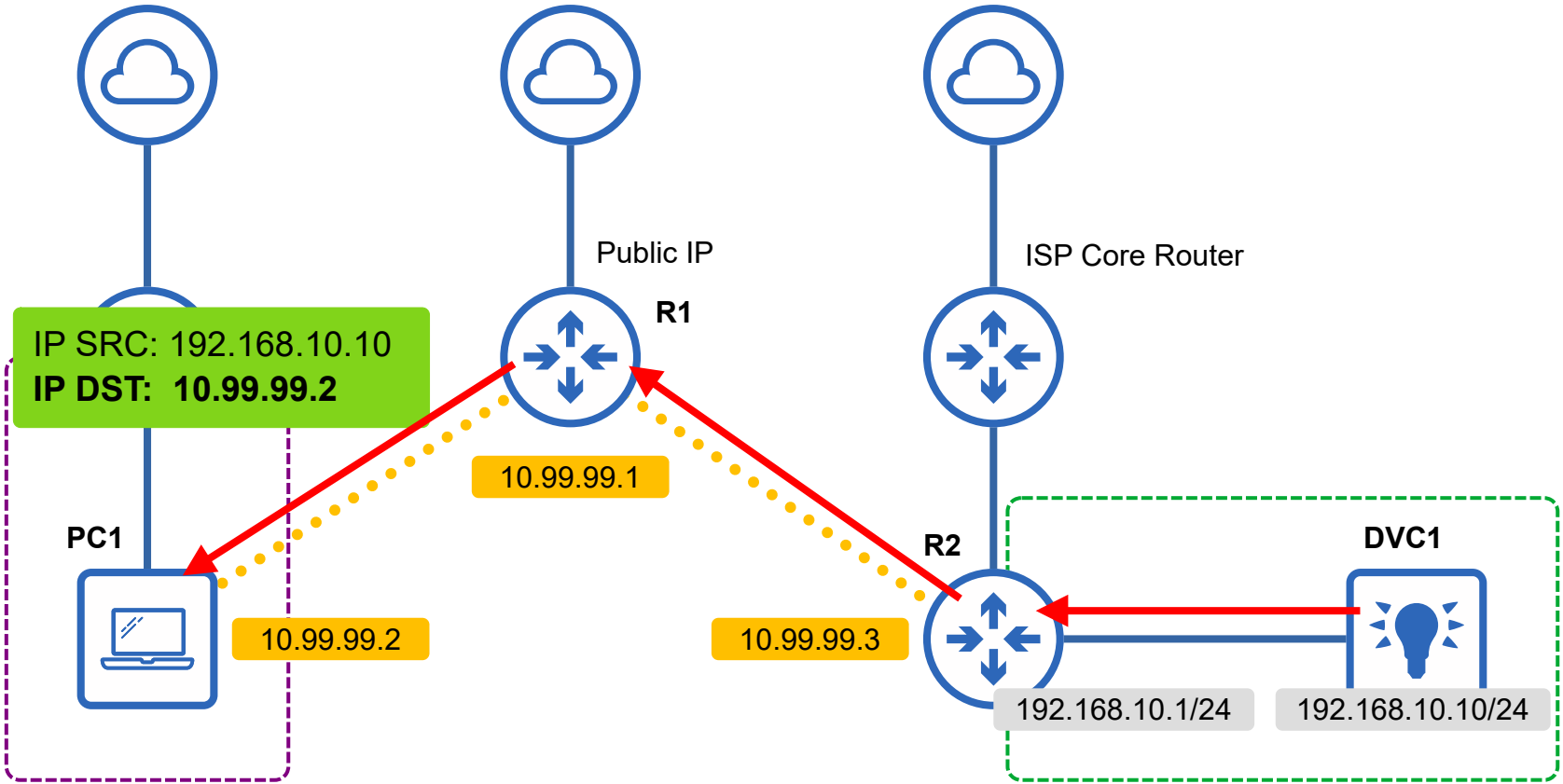


RouterOS as VPN concentrator

Corsi MikroTik



RouterOS as VPN concentrator



It works!

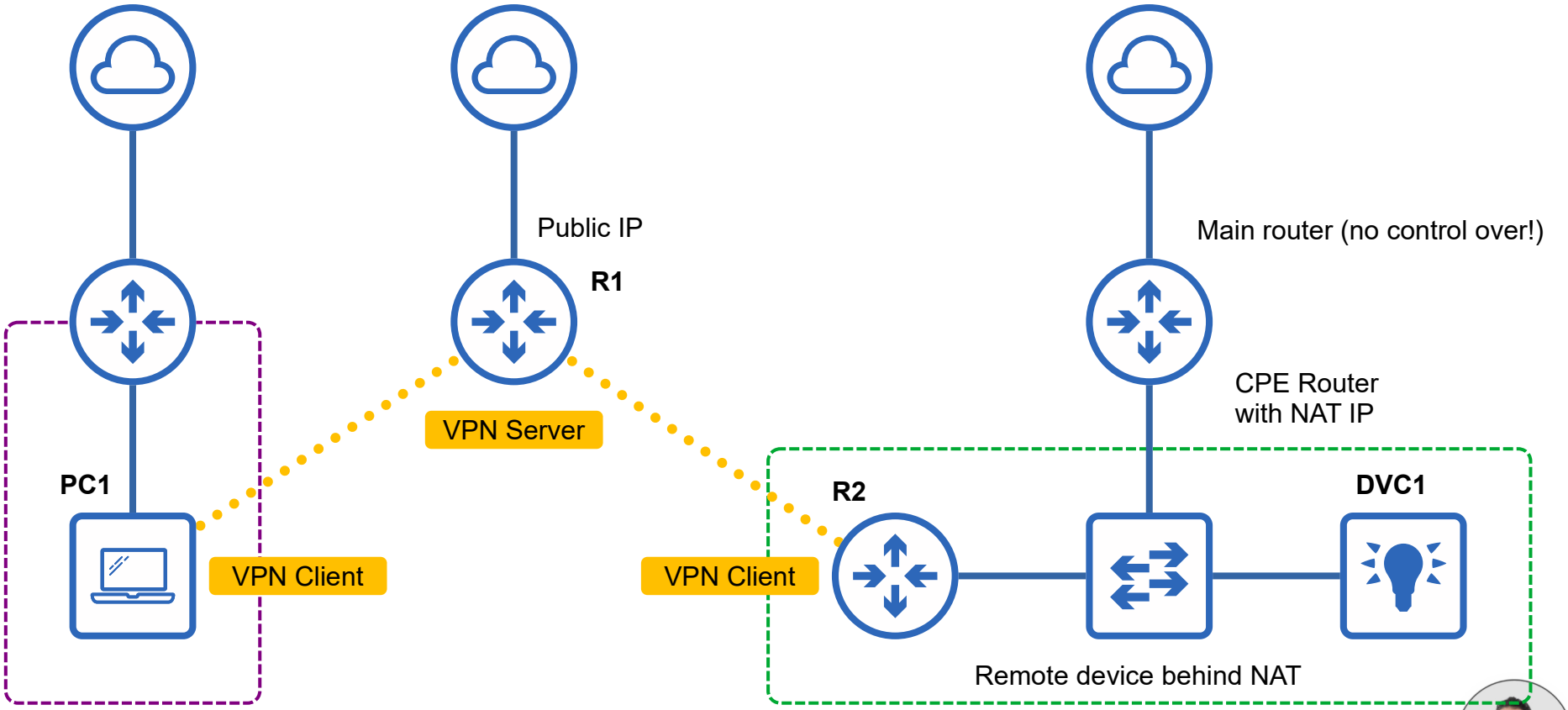
Terminal PC1

```
[admin@PC1] > ping 192.168.10.10
  SEQ HOST                     SIZE TTL TIME       STATUS
    0 192.168.10.10             56  62 9ms630us
    1 192.168.10.10             56  62 10ms261us
    2 192.168.10.10             56  62 9ms172us
    sent=3 received=3 packet-loss=0% min-rtt=9ms172us
avg-rtt=9ms687us max-rtt=10ms261us
```

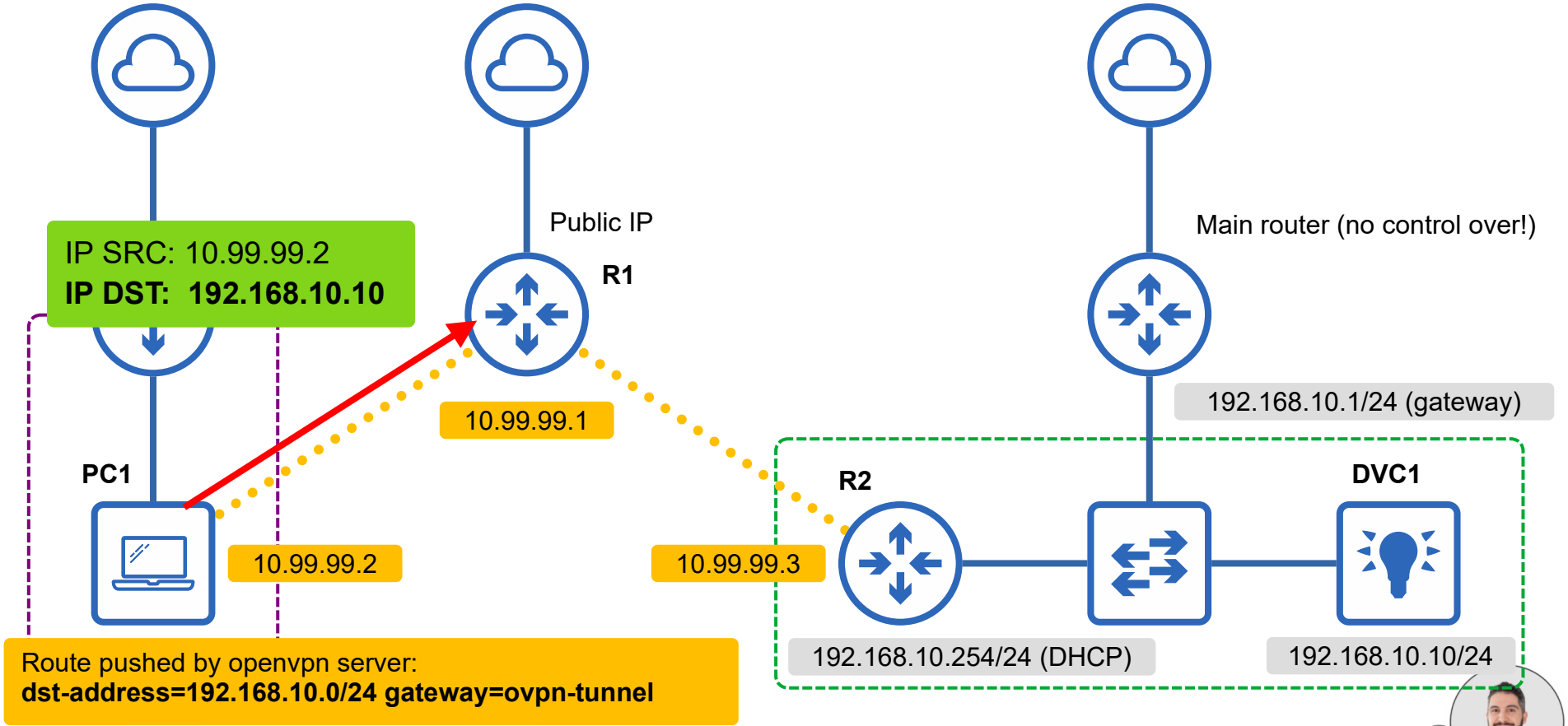
MikroTik as a VPN Client (not main router)

- In certain situations, adding a device as a main router is not possible
- In this case, the router is a host in the network (like a printer or a smartphone)
- The router is NOT the default gateway of the device
- The VPN Client can be installed easily with DHCP client configuration

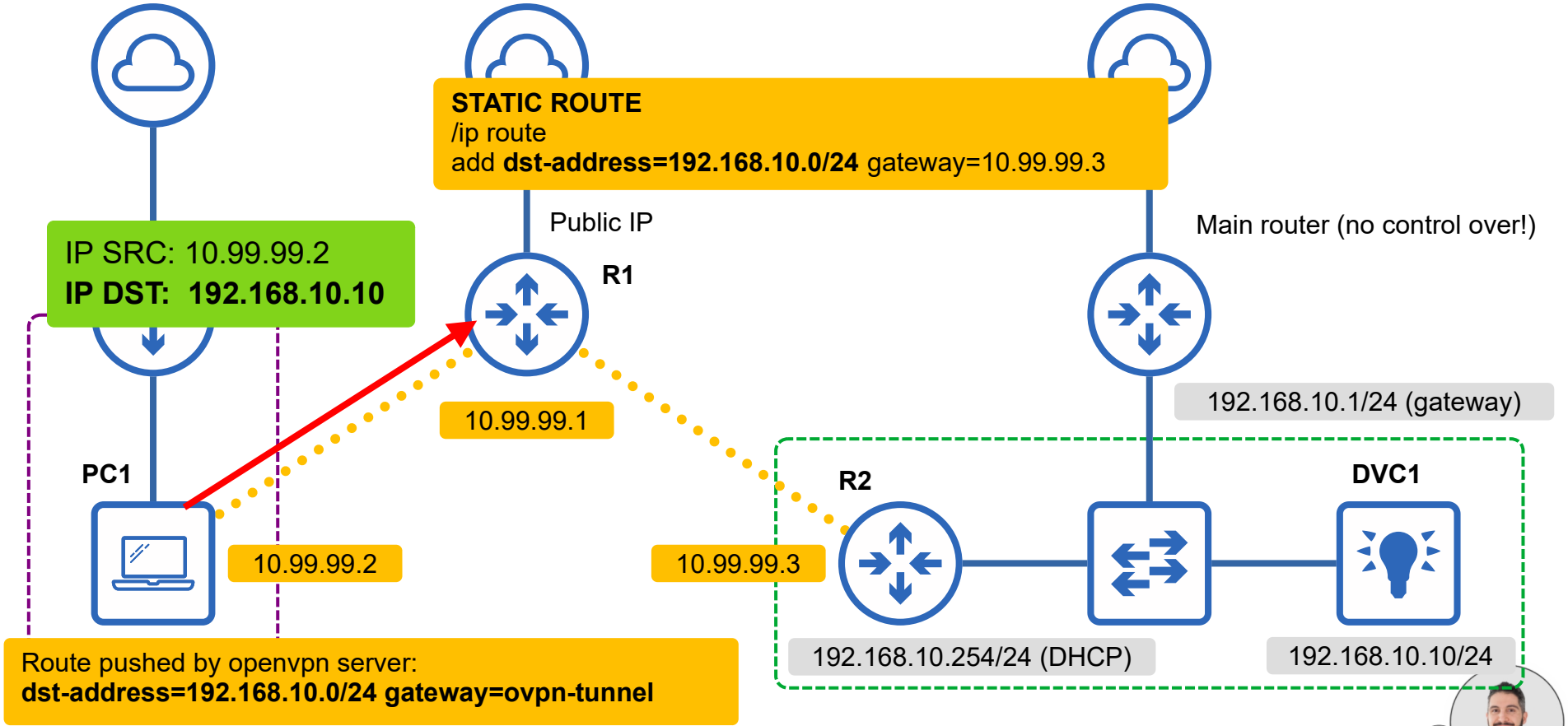
RouterOS as VPN concentrator



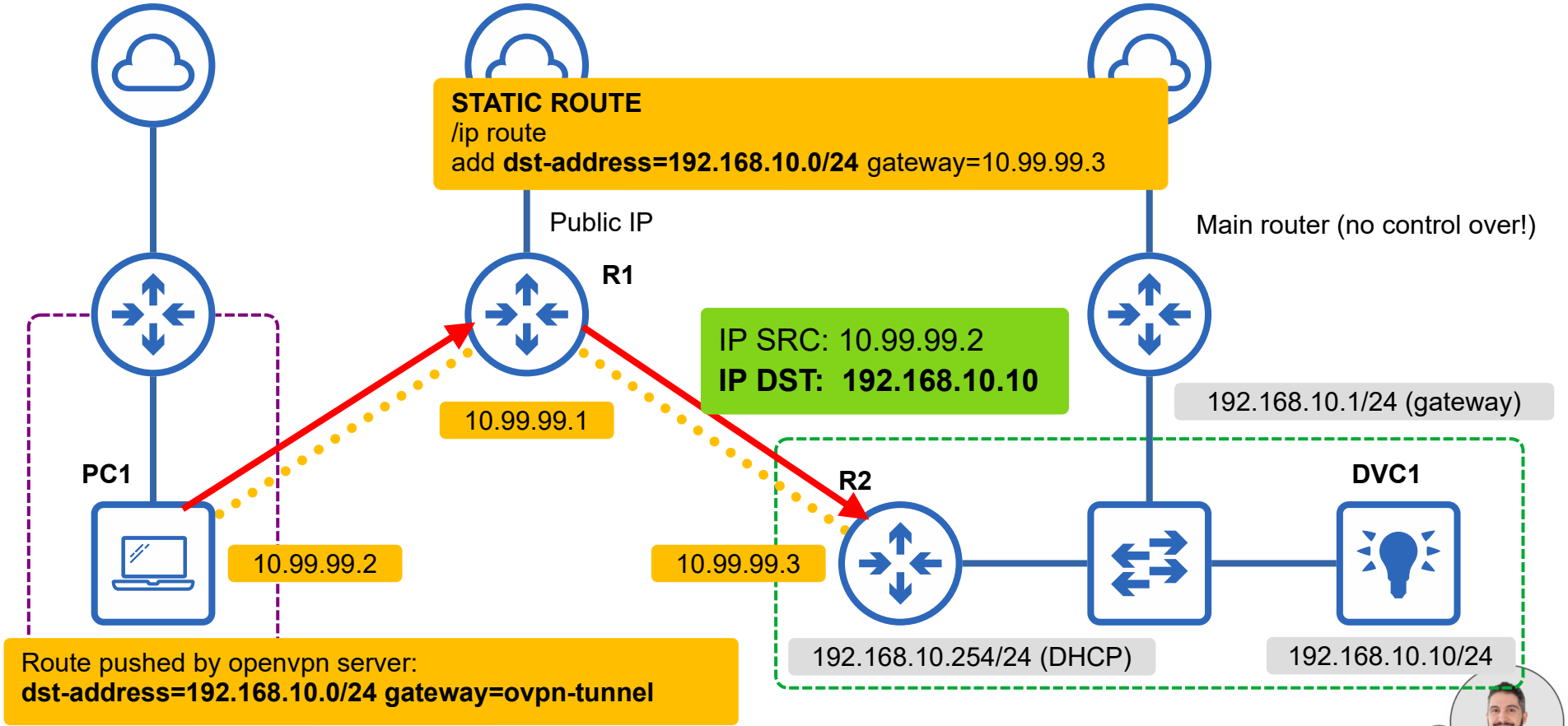
RouterOS as VPN concentrator



RouterOS as VPN concentrator



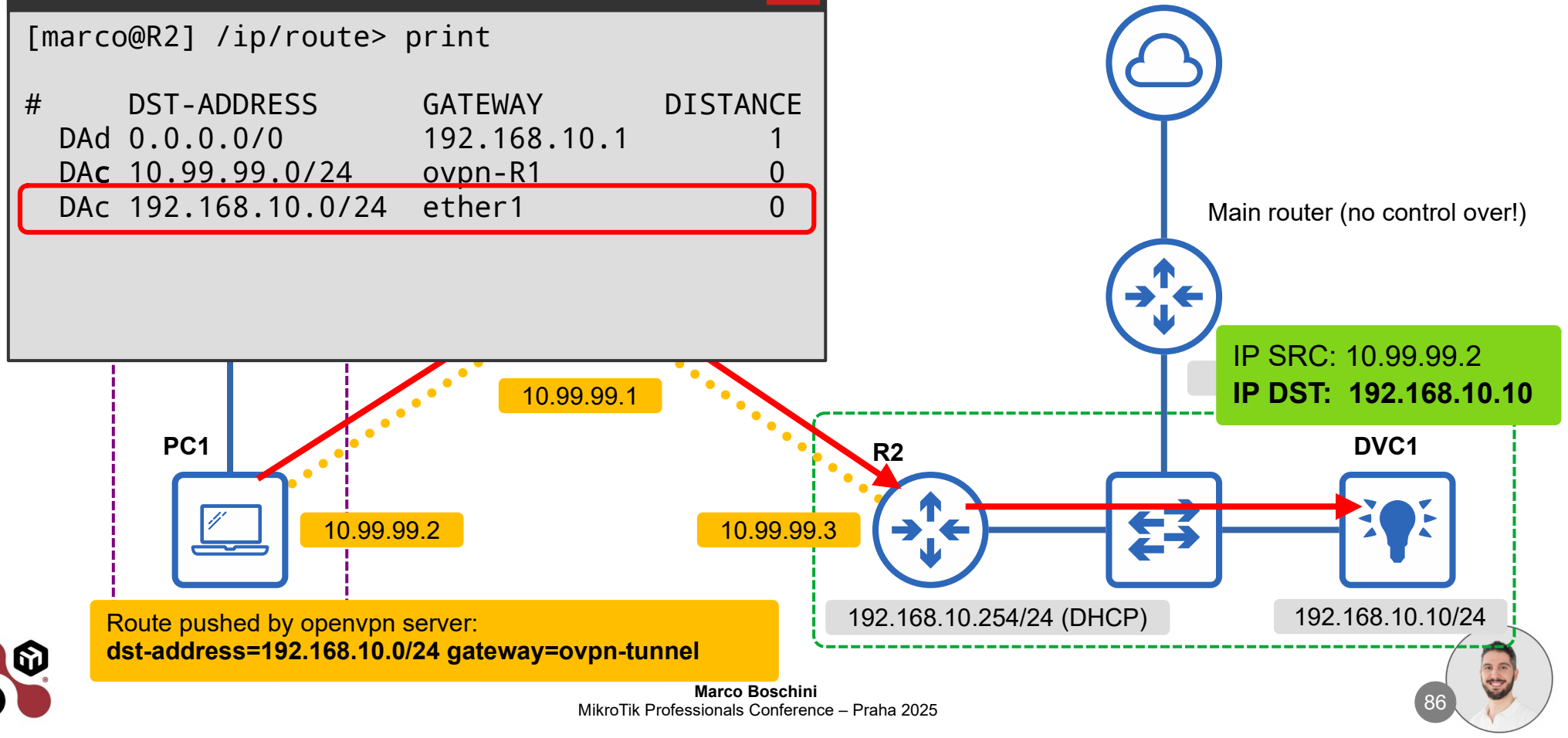
RouterOS as VPN concentrator



RouterOS as VPN concentrator

Terminal R2

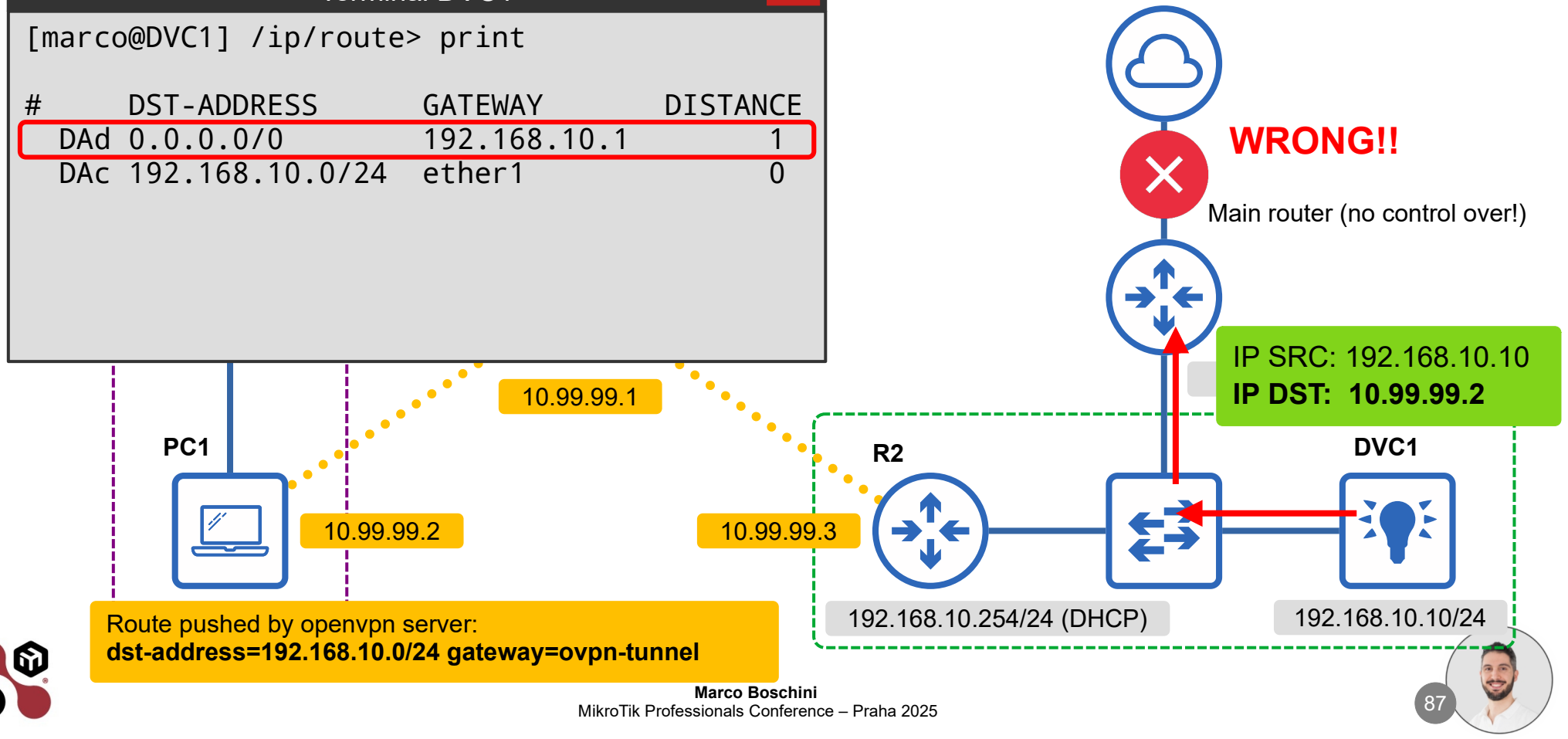
```
[marco@R2] /ip/route> print

#      DST-ADDRESS       GATEWAY         DISTANCE
  DAd  0.0.0.0/0         192.168.10.1           1
  DAc  10.99.99.0/24     ovpn-R1                0
  DAc  192.168.10.0/24   ether1                 0
```



RouterOS as VPN concentrator

Corsi MikroTik

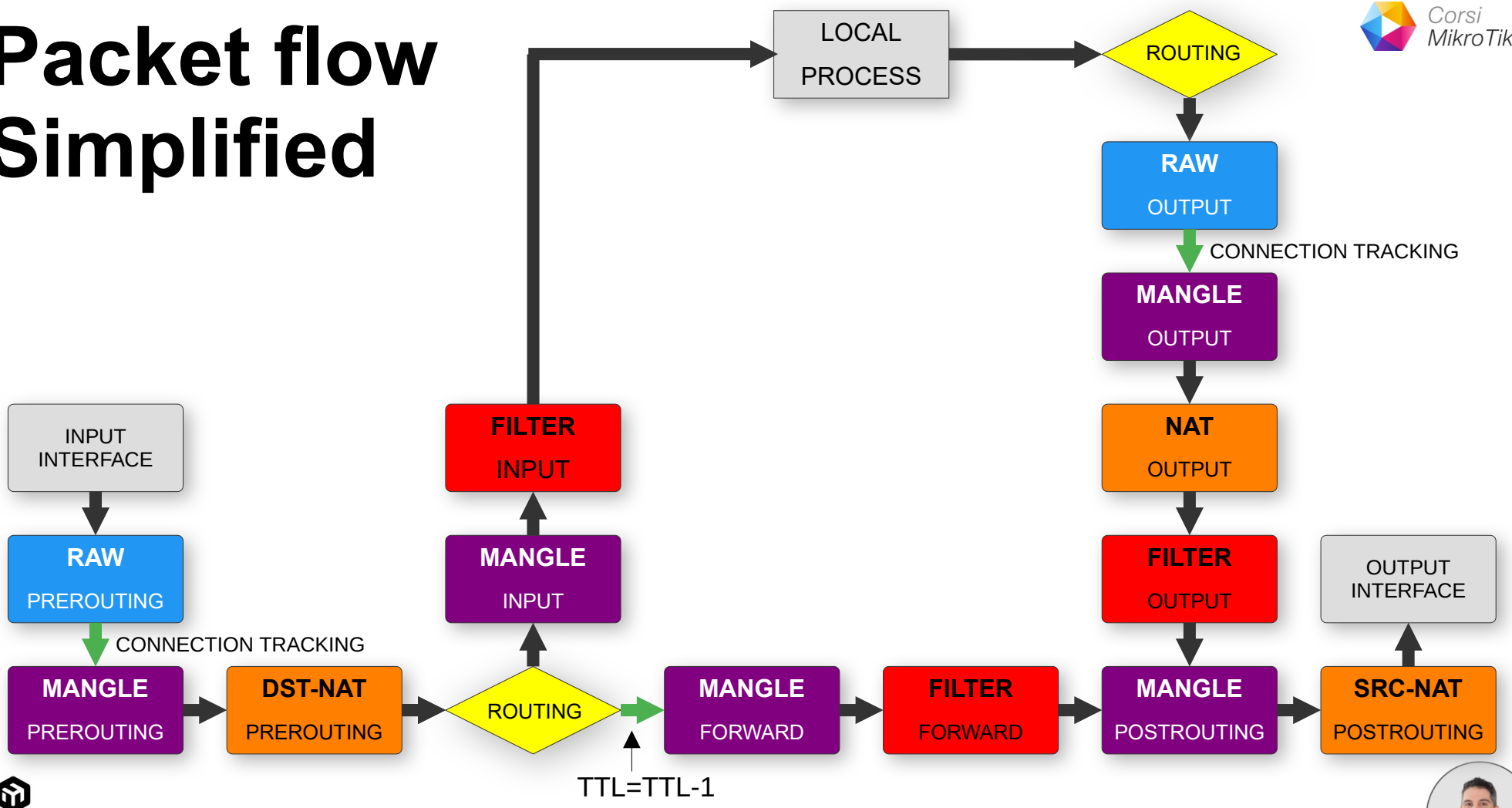
```
Terminal DVC1                                                    x
[marco@DVC1] /ip/route> print

#     DST-ADDRESS       GATEWAY         DISTANCE
  DAd 0.0.0.0/0         192.168.10.1           1
  DAc 192.168.10.0/24   ether1                 0
```



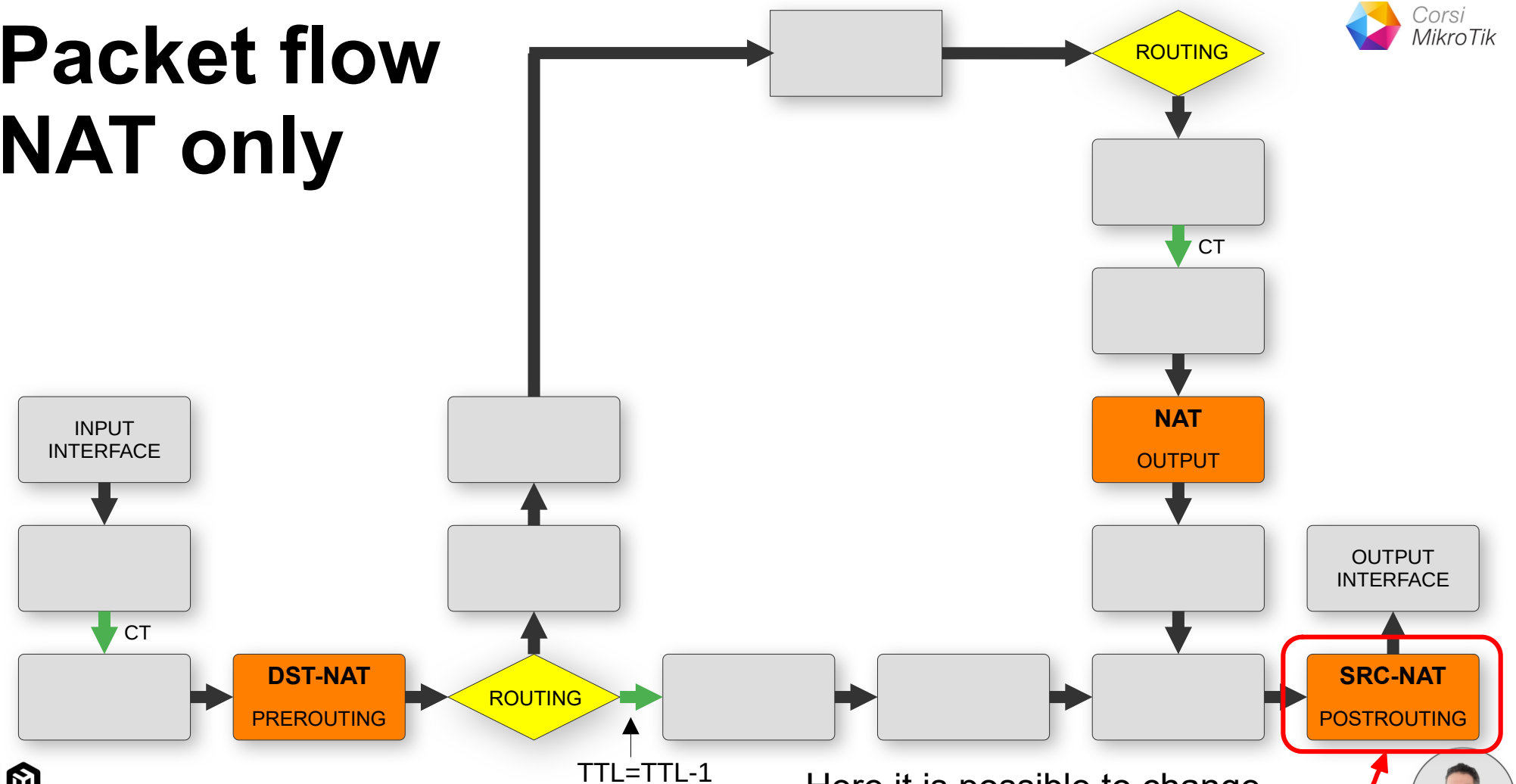
MikroTik as a VPN Client (not main router)

- DVC1 try to response to the original sender, and select the best route to reach them.
- Because 10.99.99.0/24 is a subnet that DVC1 doesn’t know, it send packets to his gateway.
- Can I change the source IP address of the packets arrived from the VPN with the IP address of the router itself?

Packet flow Simplified



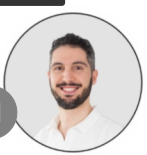
Packet flow
NAT only



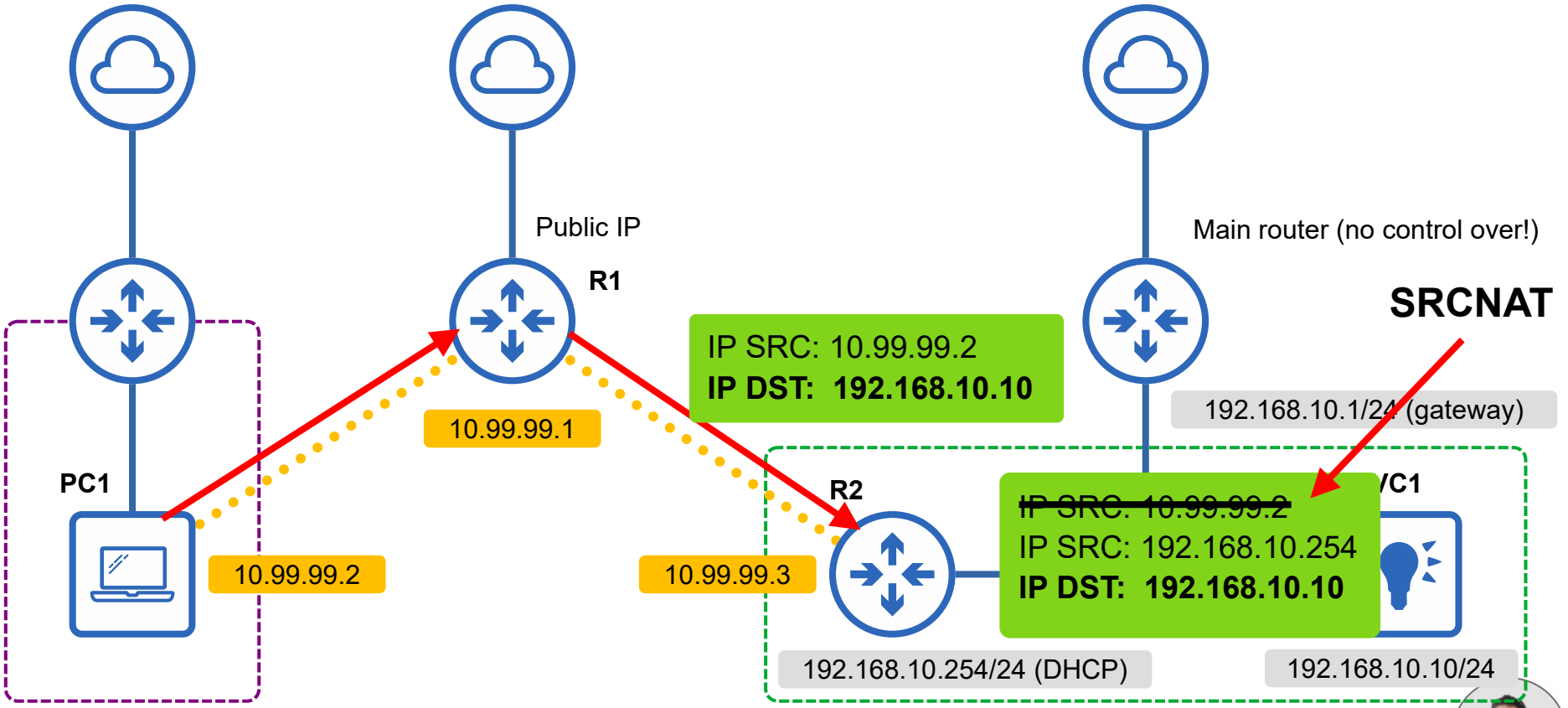
IP Firewall NAT

Terminal R2

```
[marco@R2] /ip firewall nat
add chain=srcnat src-address=10.99.99.0/24 \
out-interface=ether1 action=masquerade
```



RouterOS as VPN concentrator

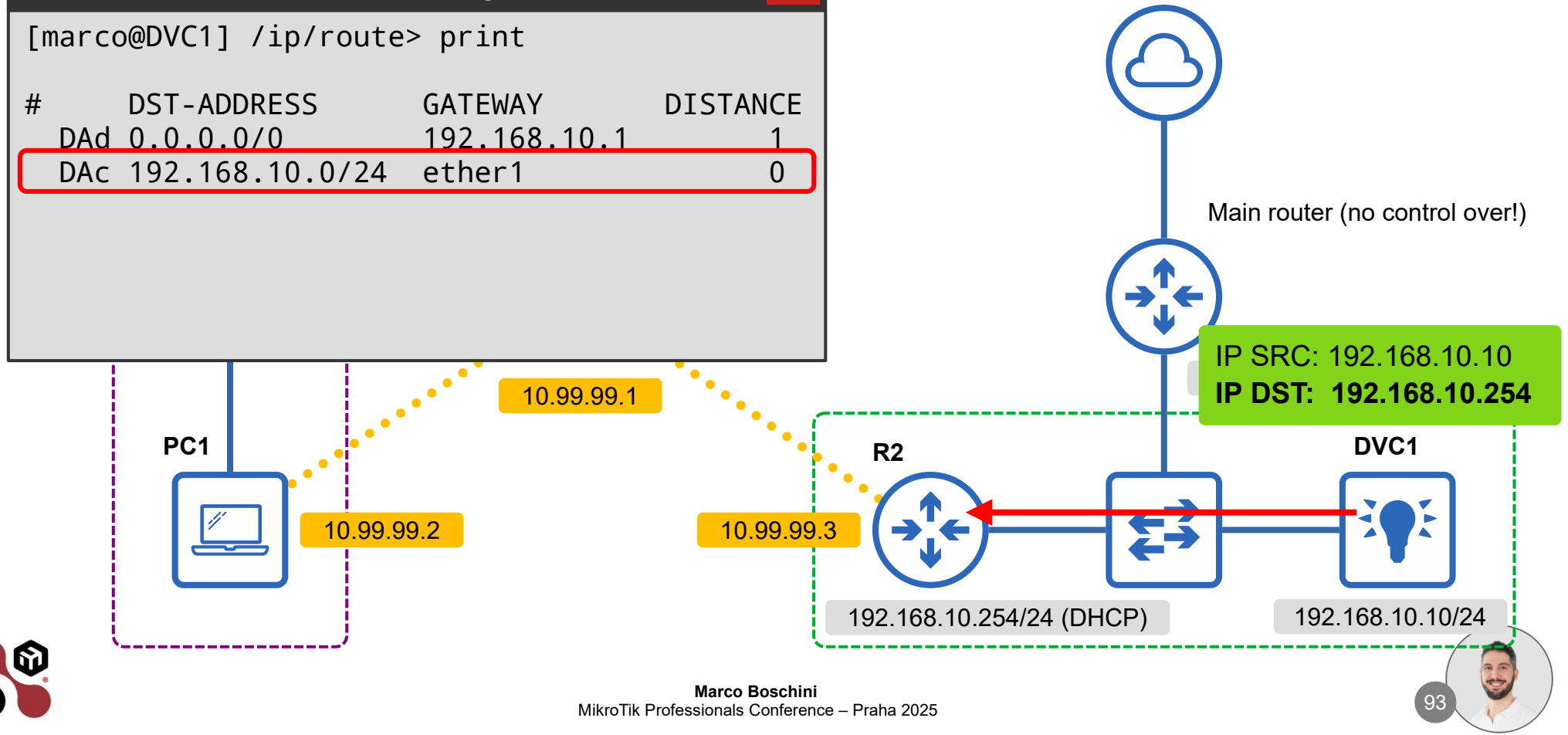


RouterOS as VPN concentrator

Corsi MikroTik

Terminal DVC1

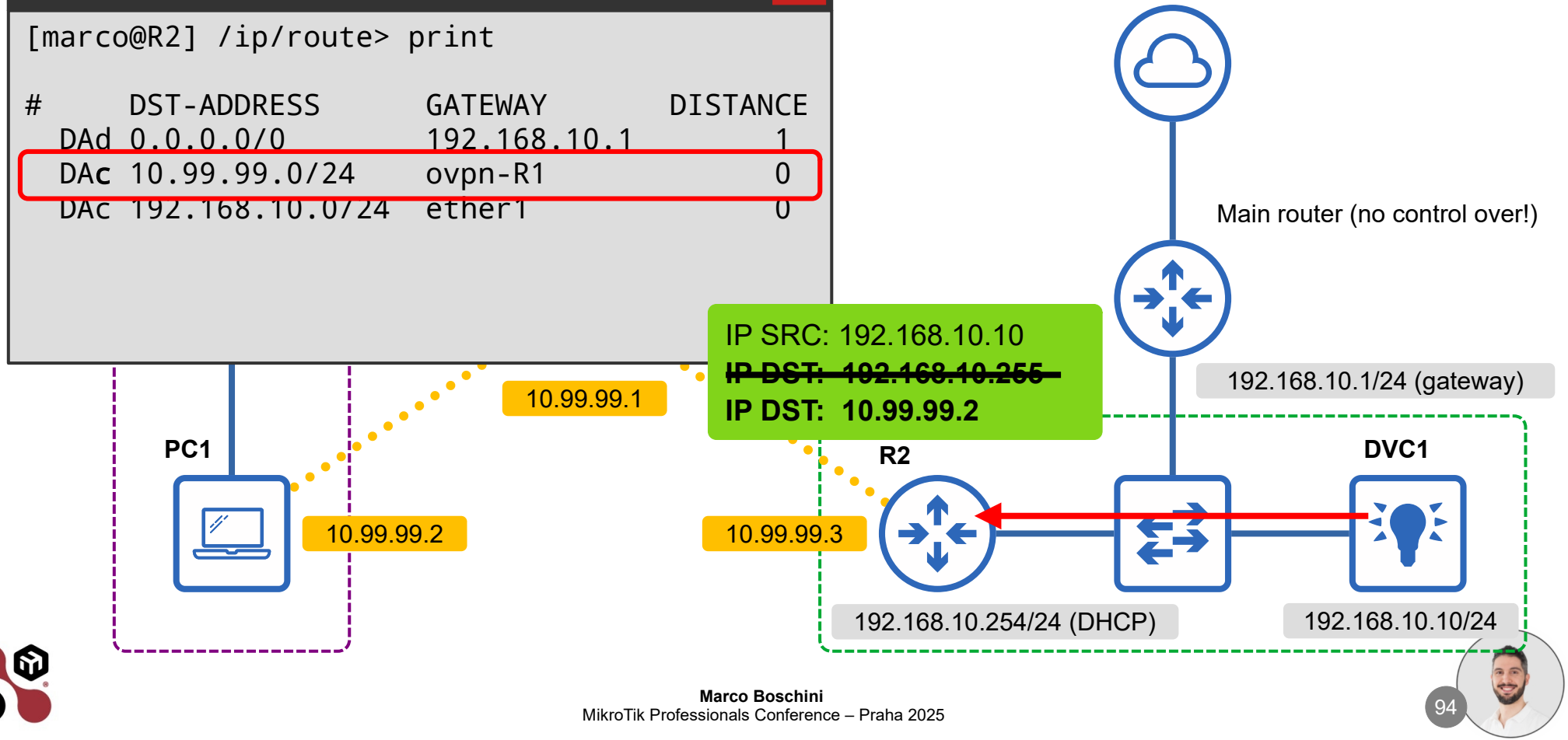
```
[marco@DVC1] /ip/route> print

#      DST-ADDRESS       GATEWAY         DISTANCE
  DAd  0.0.0.0/0         192.168.10.1           1
  DAc  192.168.10.0/24   ether1                 0
```



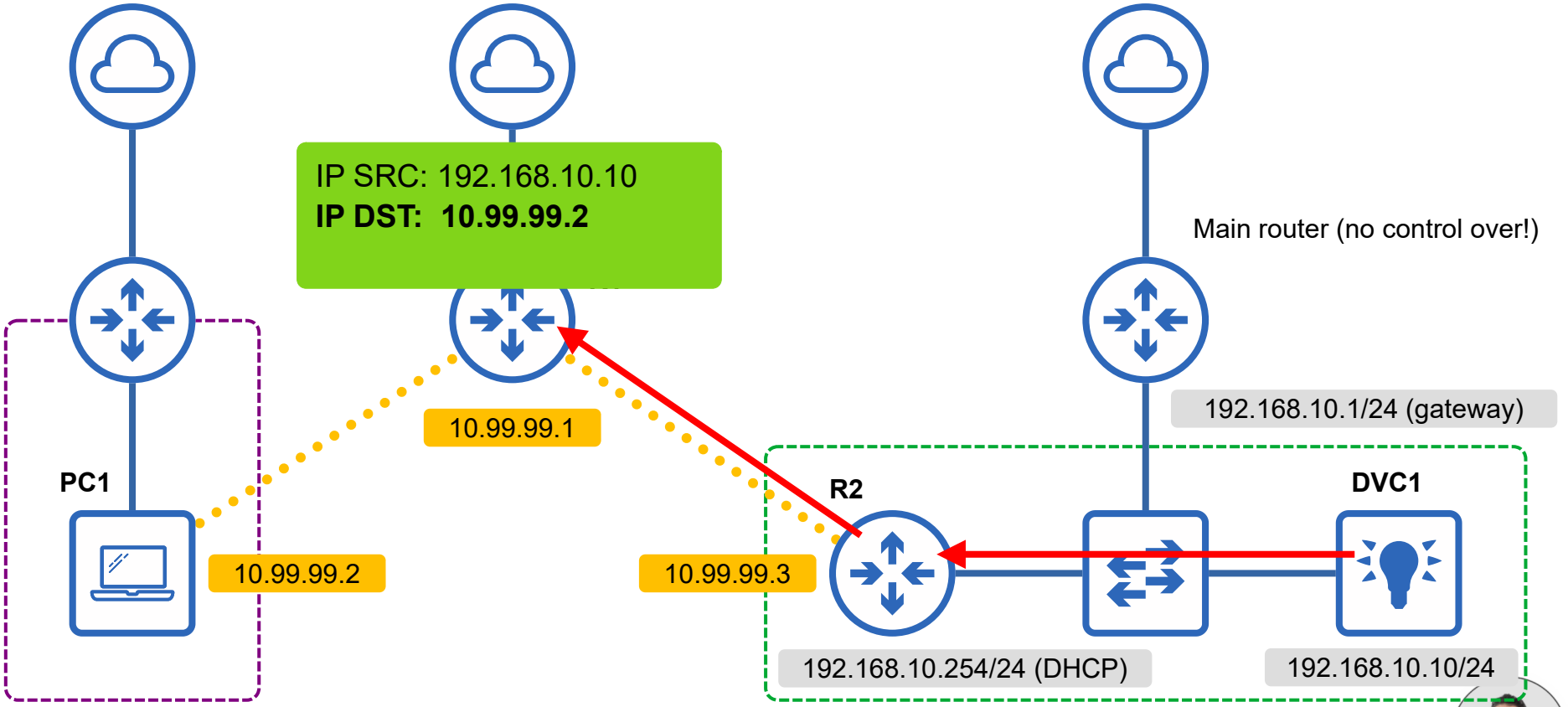
RouterOS as VPN concentrator

Terminal R2

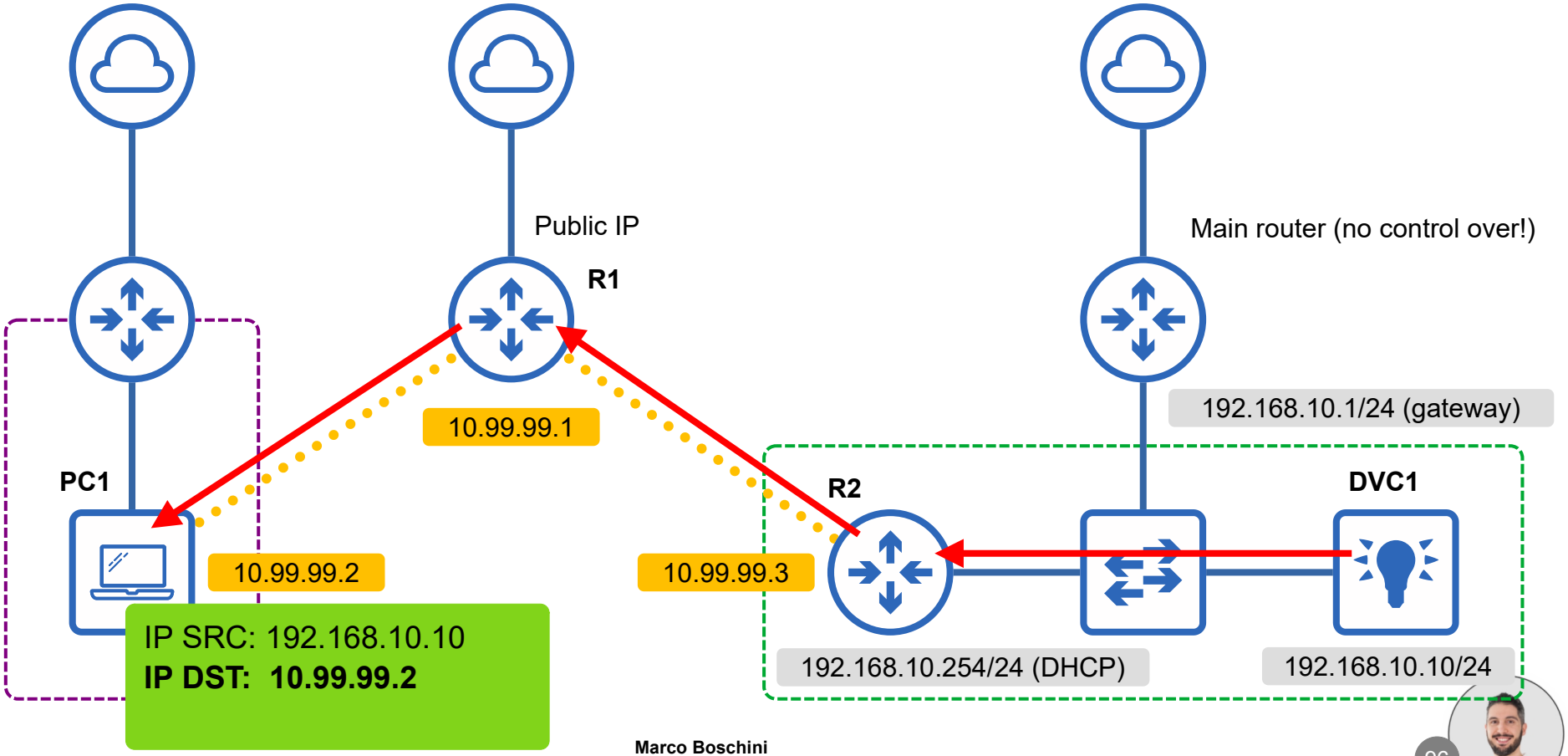
```
[marco@R2] /ip/route> print

#      DST-ADDRESS       GATEWAY         DISTANCE
  DAd  0.0.0.0/0         192.168.10.1           1
  DAc  10.99.99.0/24     ovpn-R1                0
  DAc  192.168.10.0/24   ether1                 0
```



RouterOS as VPN concentrator



RouterOS as VPN concentrator



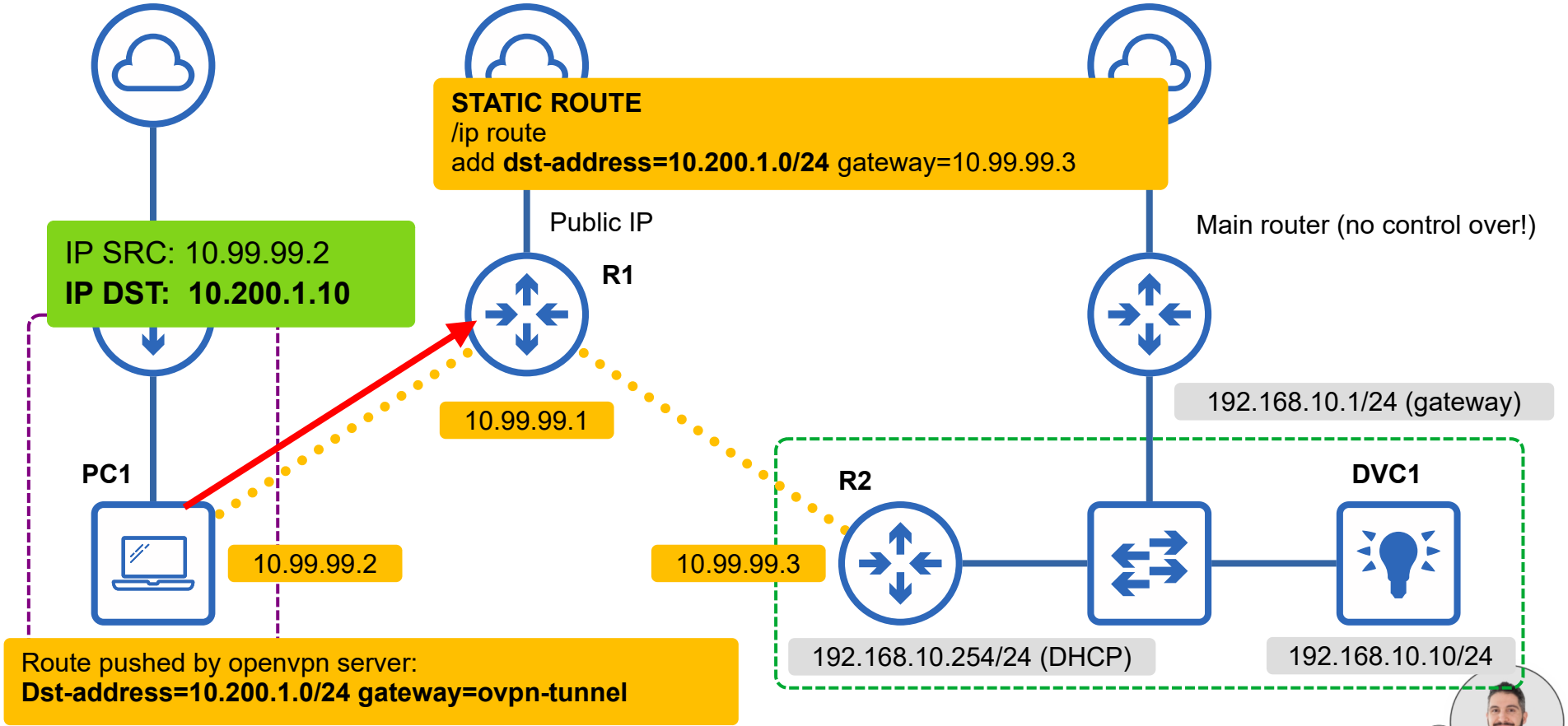
How about if two or more customers have the same network subnet?

- In my implementation I found a lot of customers with the same subnet network:
  192.168.1.0/24
- If I would reach different network with the same subnet address, what can I do?

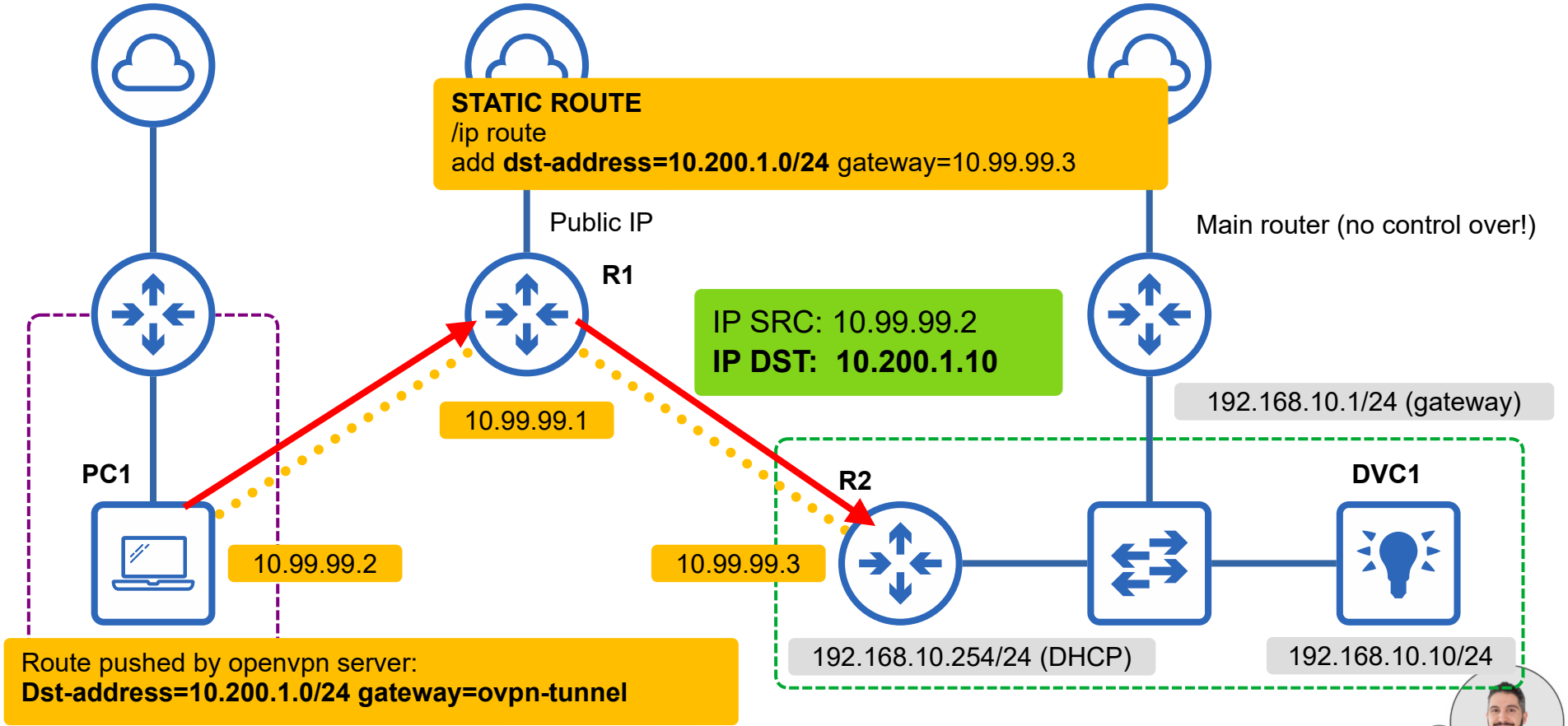
How about if two or more customers have the same network subnet?

- You can use what I call **“the “f*ke route”** (means “Not real!”)
- The **router forward packets by destination address**.
- You can use a route that points to a network that is not really present in the destination, only to forward packet at the destionation
- The destination router can modify the destionation with a NAT rule like dst-nat or netmap with the real destionation subnet.

RouterOS as VPN concentrator

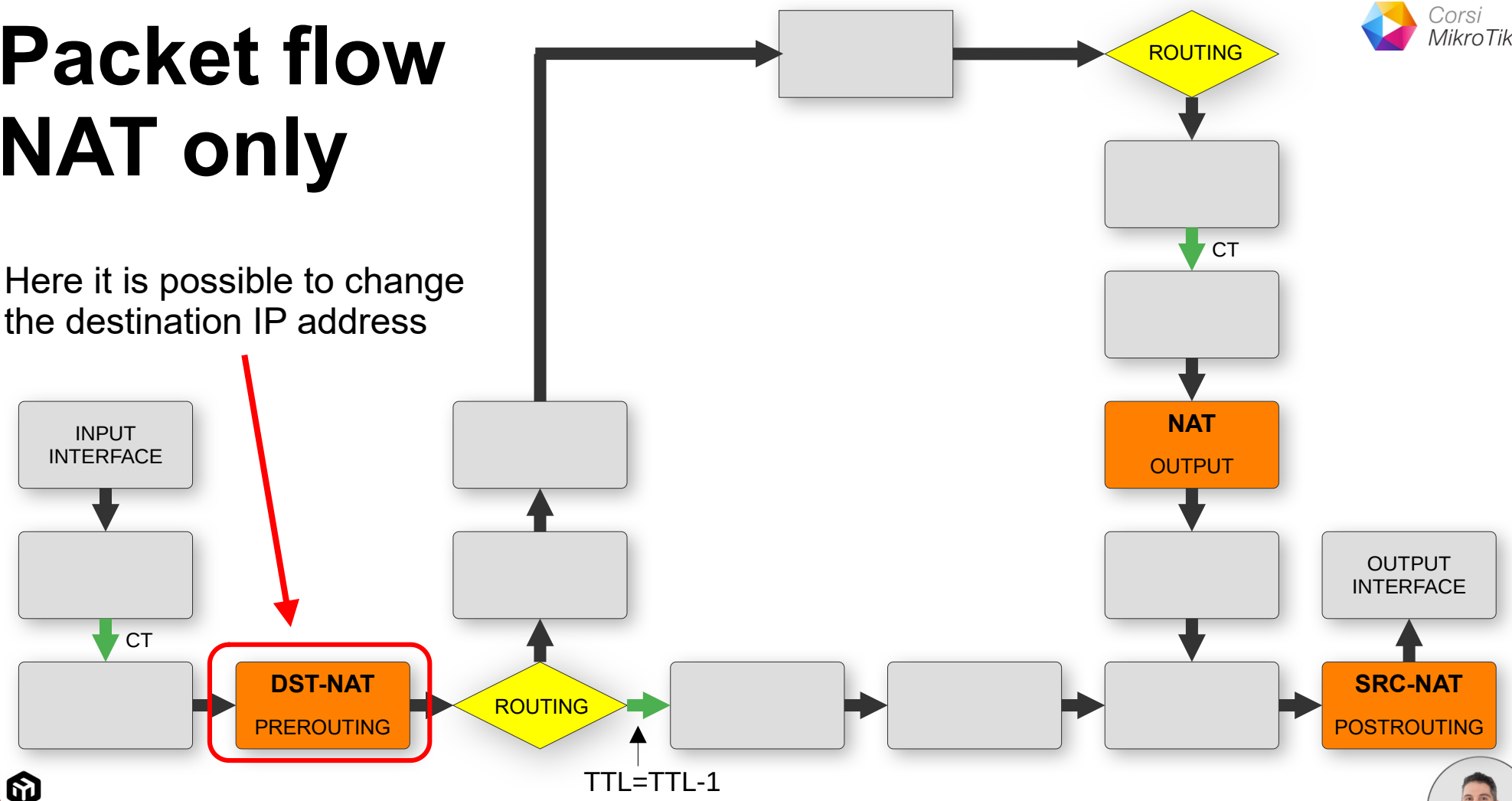


RouterOS as VPN concentrator



Packet flow NAT only

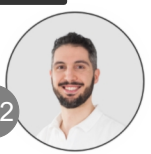
Here it is possible to change the destination IP address



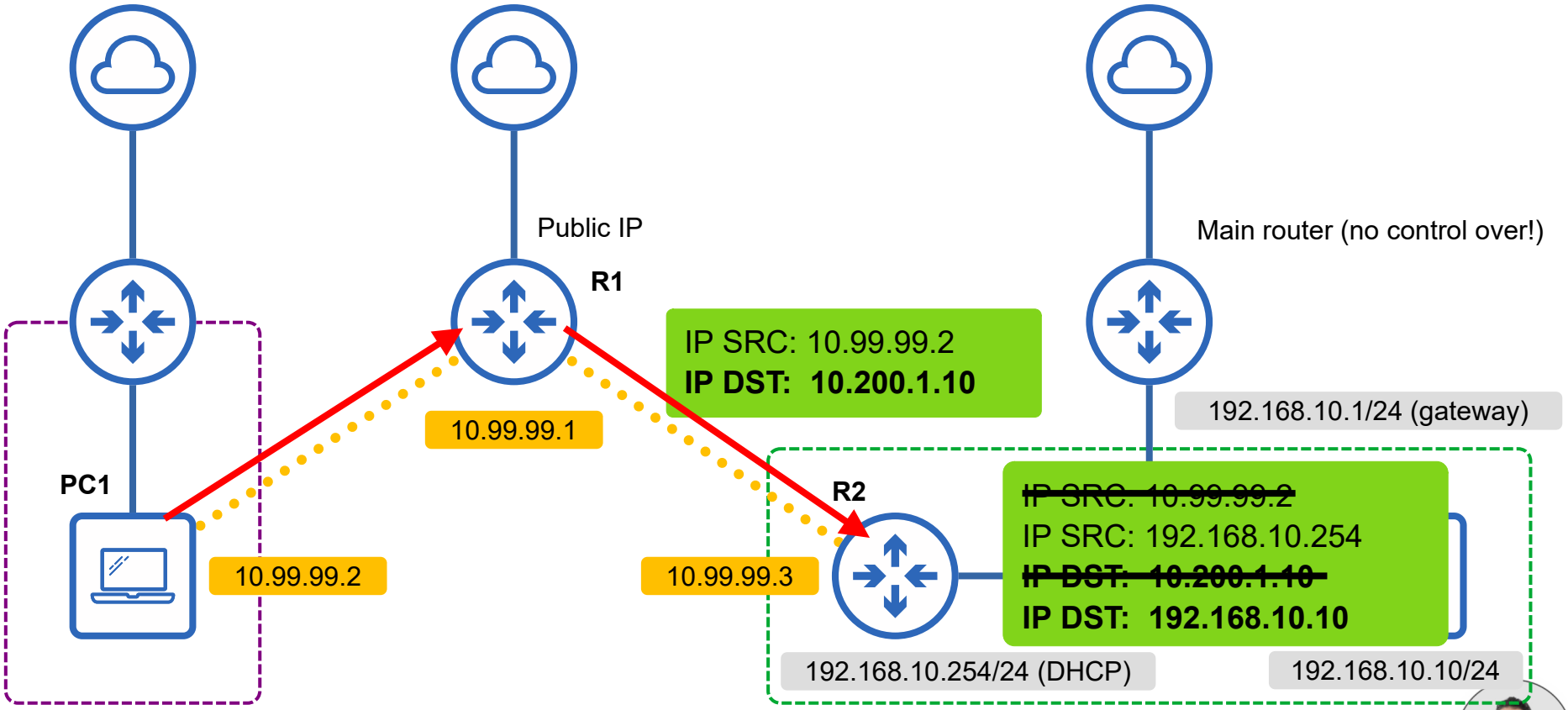
IP Firewall NAT

Terminal R2

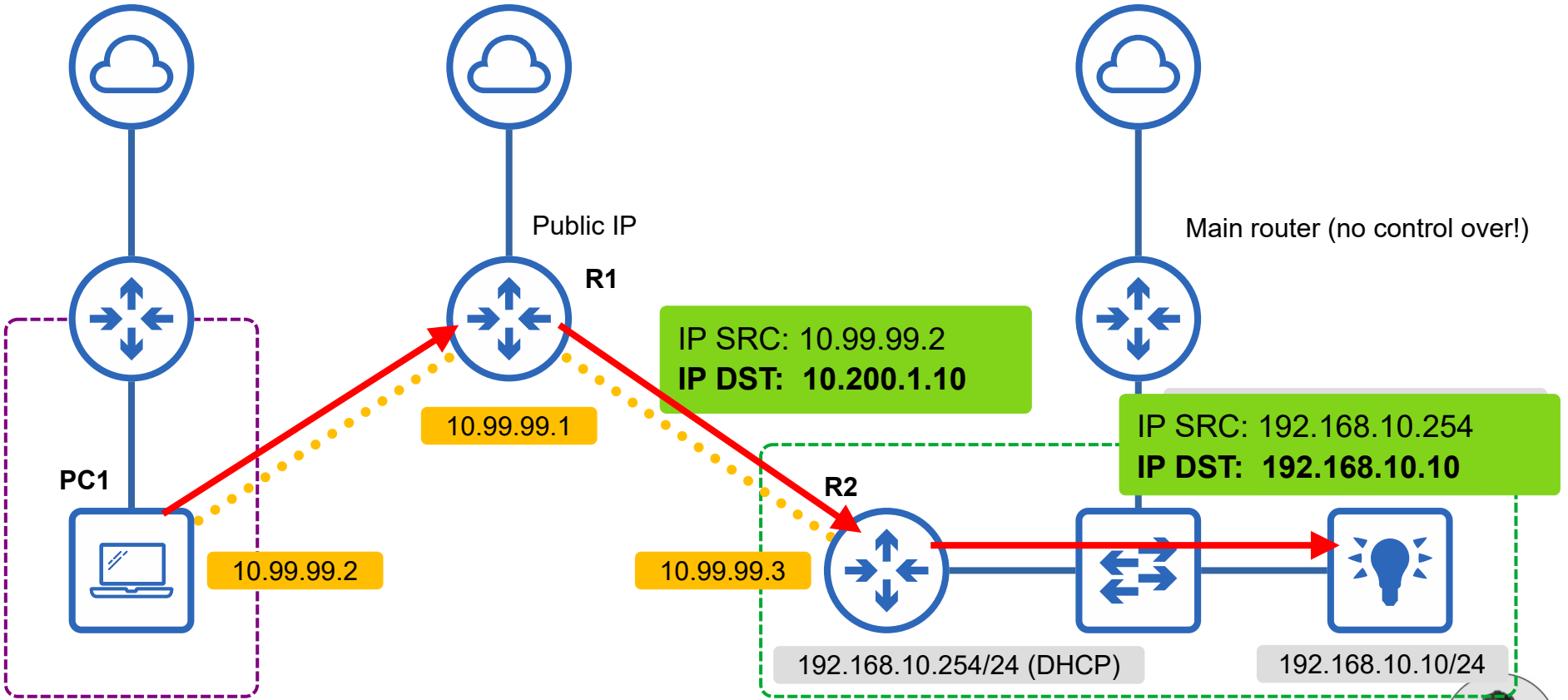
```
[marco@R2] /ip firewall nat
add chain=dstnat \
src-address=10.99.99.0/24 dst-address=10.200.1.0/24 \
action=netmap to-addresses=192.168.10.0/24
```



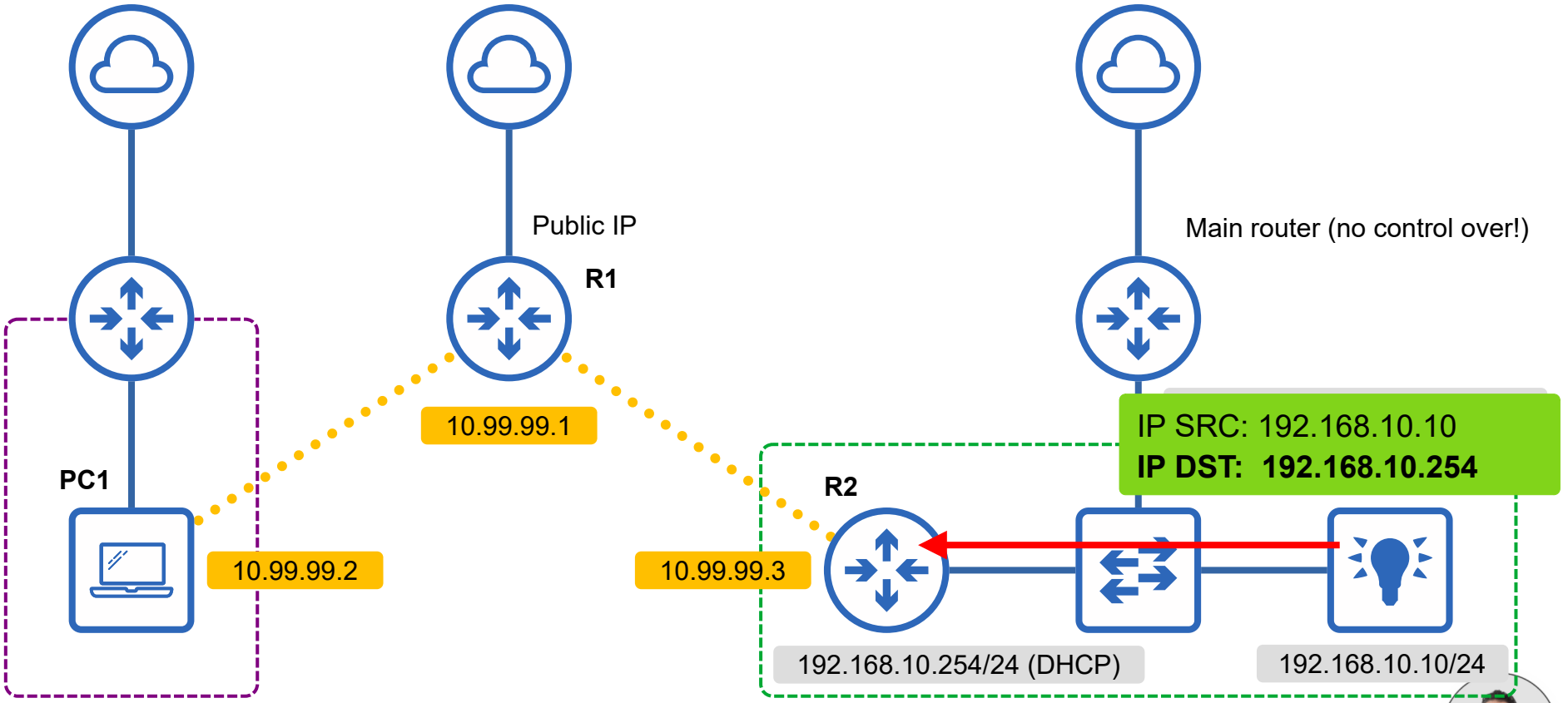
RouterOS as VPN concentrator



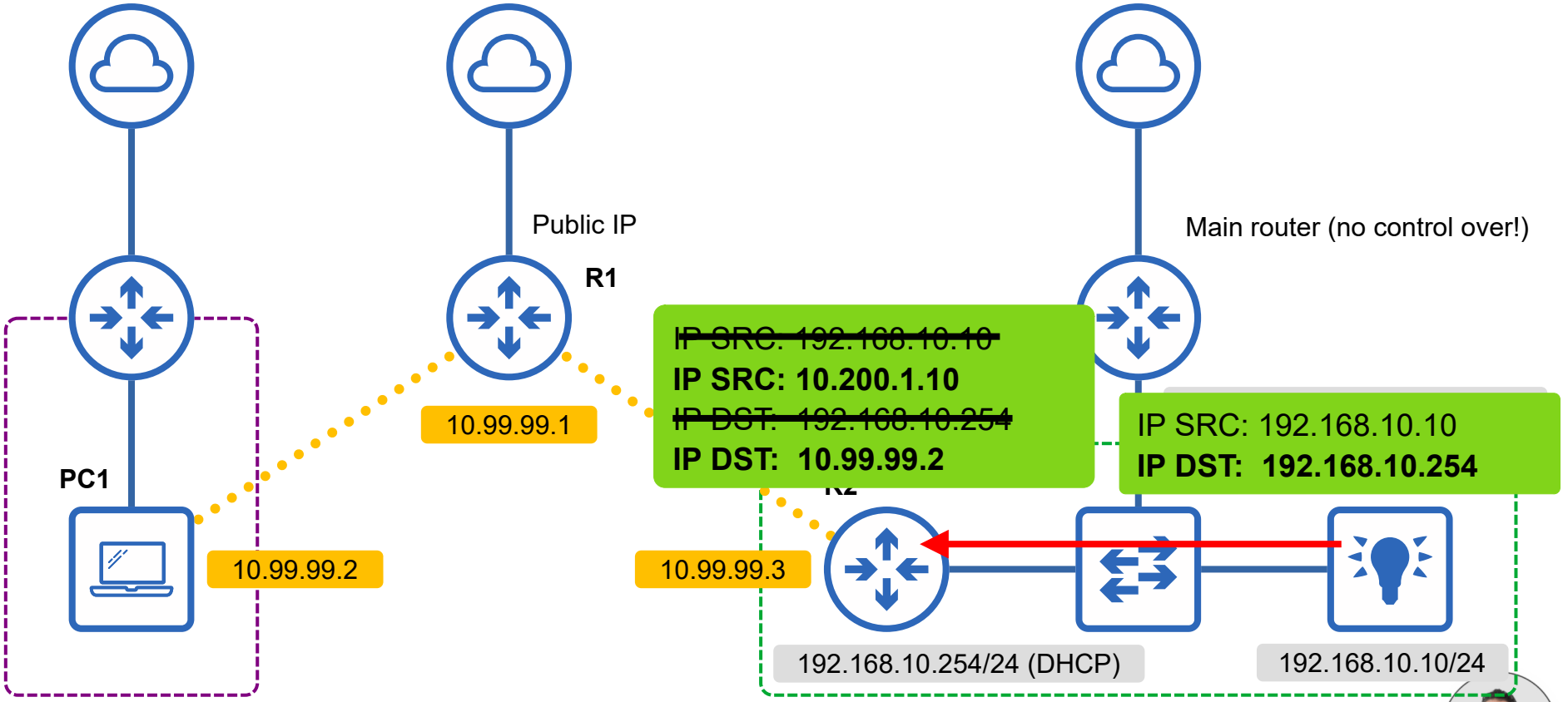
RouterOS as VPN concentrator



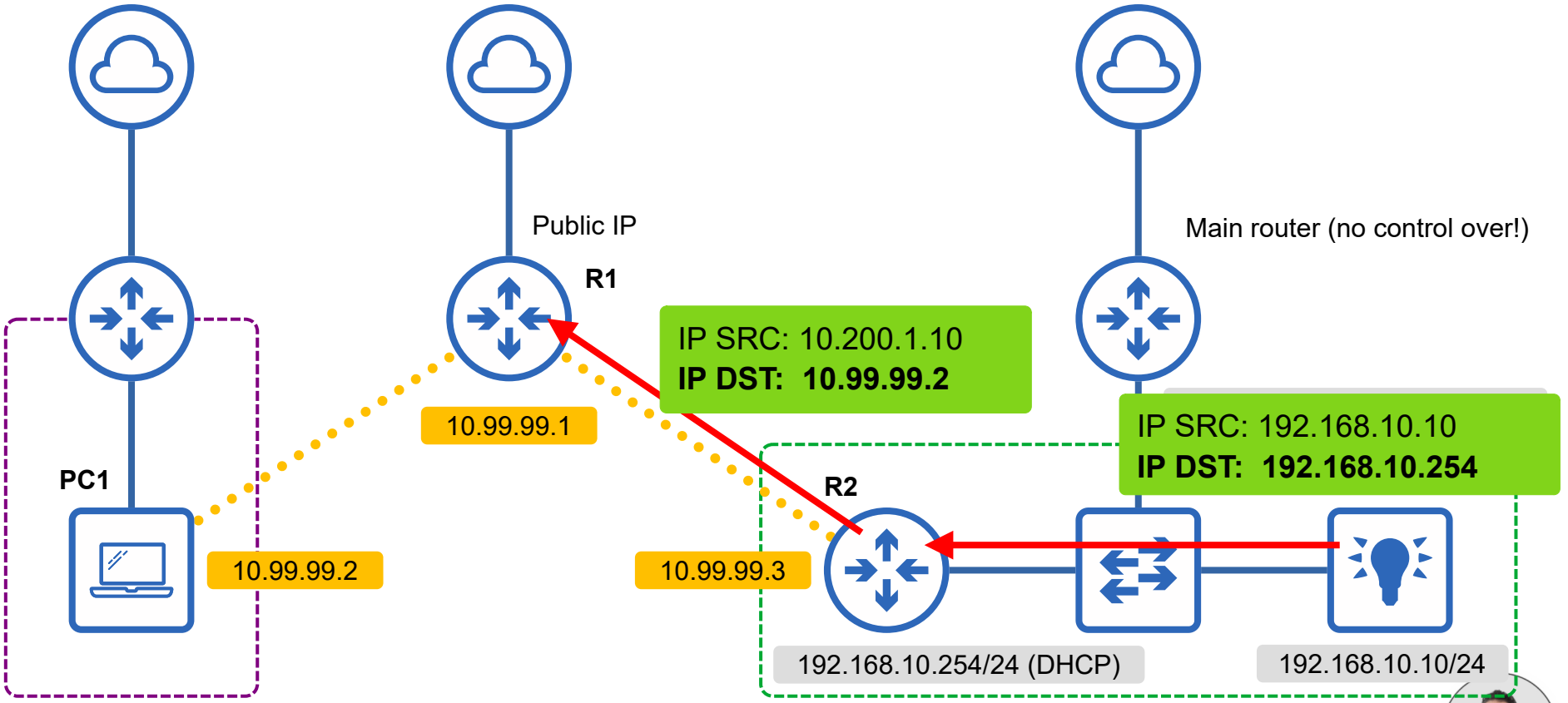
RouterOS as VPN concentrator



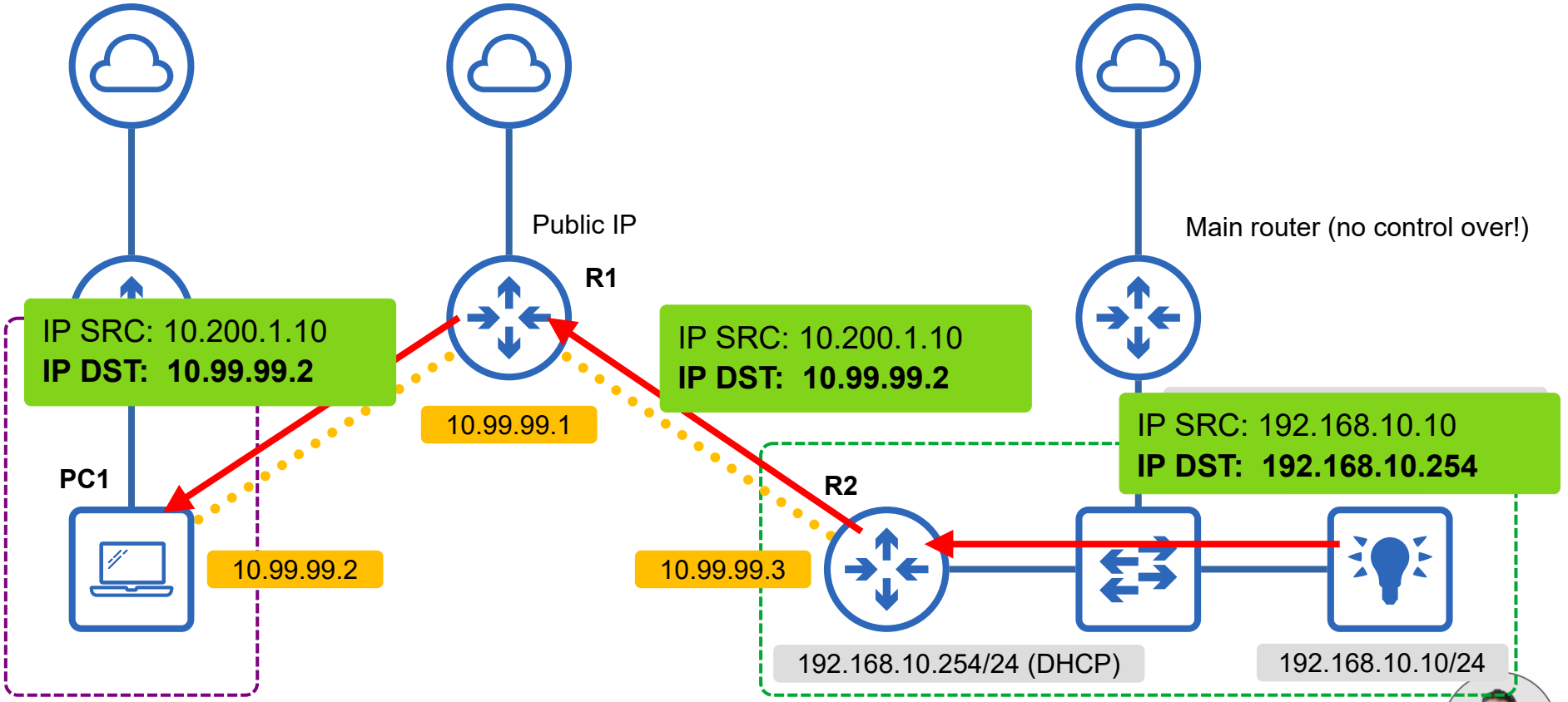
RouterOS as VPN concentrator



RouterOS as VPN concentrator



RouterOS as VPN concentrator



It works!

Terminal PC1

```
[admin@PC1] > ping 10.200.1.10
  SEQ HOST                     SIZE TTL TIME       STATUS
    0 10.200.1.10               56  62 9ms630us
    1 10.200.1.10               56  62 10ms261us
    2 10.200.1.10               56  62 9ms172us
    sent=3 received=3 packet-loss=0% min-rtt=9ms172us
avg-rtt=9ms687us max-rtt=10ms261us
```

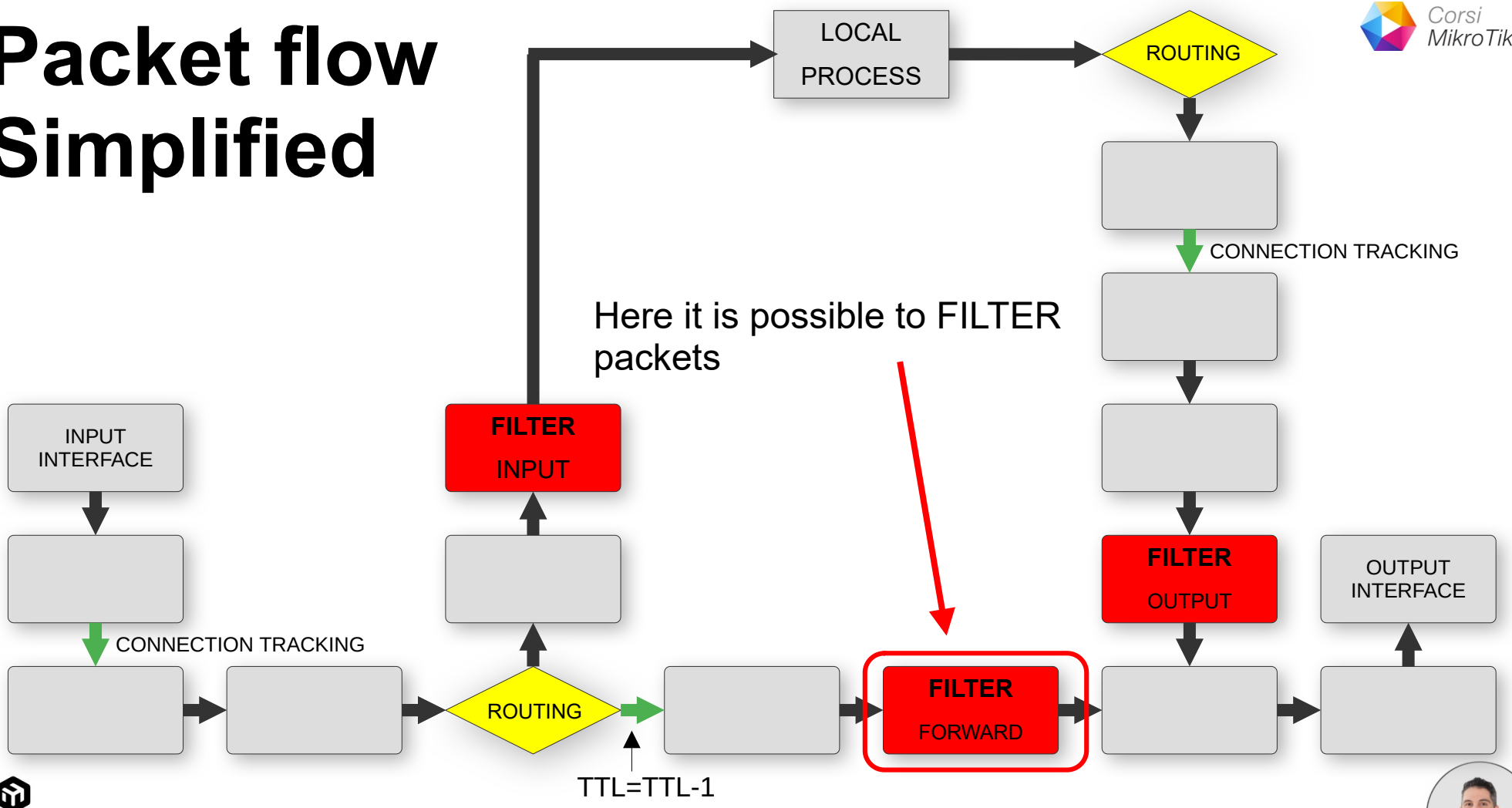
How about security?

- **Authentication**:
  - VPN tunnel require authentication
    - user/password
    - certificate
    - 2FA

How about security?

- **Authorization:**
  - RouterOS implement a statefull firewall
  - RouterOS implement a Address List function
  - You can assign specific address at every VPN Client connected (ppp/secret)
  - You can build lists with src-address and dst-address allowed so that the client can reach the destionation, and drop everything else

Packet flow Simplified



How about security?

- **Encryption**
  - VPN tunnel can encrypt the communication

Best VPN Client to connect customers

Corsi MikroTik

- OpenVPN client is available in every platform (newer and older)
- Is considered secure
- Strong encryption (SHA512/AES256)
- Use of certificate is possible
- Very easy to configure in the server side
- You can send configuration file to the customers via email/Whatsapp/Telegram

Best VPN Client to connect customers

Corsi MikroTik

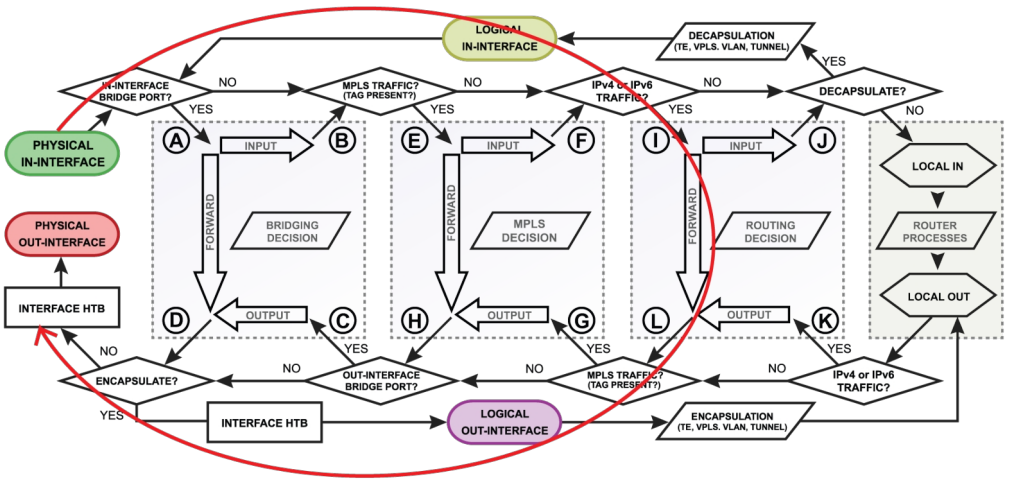
- Is not necessary to set routes in the configuration file
- You can PUSH ROUTE to the client from the server
- VPN connection could not be the default gateway
- Client reach internet via his connection
- Client reach only the device via VPN

WireGuard

- WireGuard is a great choice as VPN, secure and FAST
- BTW is available only from RouterOS v7
- It is not possibile to PUSH ROUTES to the clients, you must manually edit the config on the client devices
- You can easly implememt the same concept with WireGuard

Conclusion

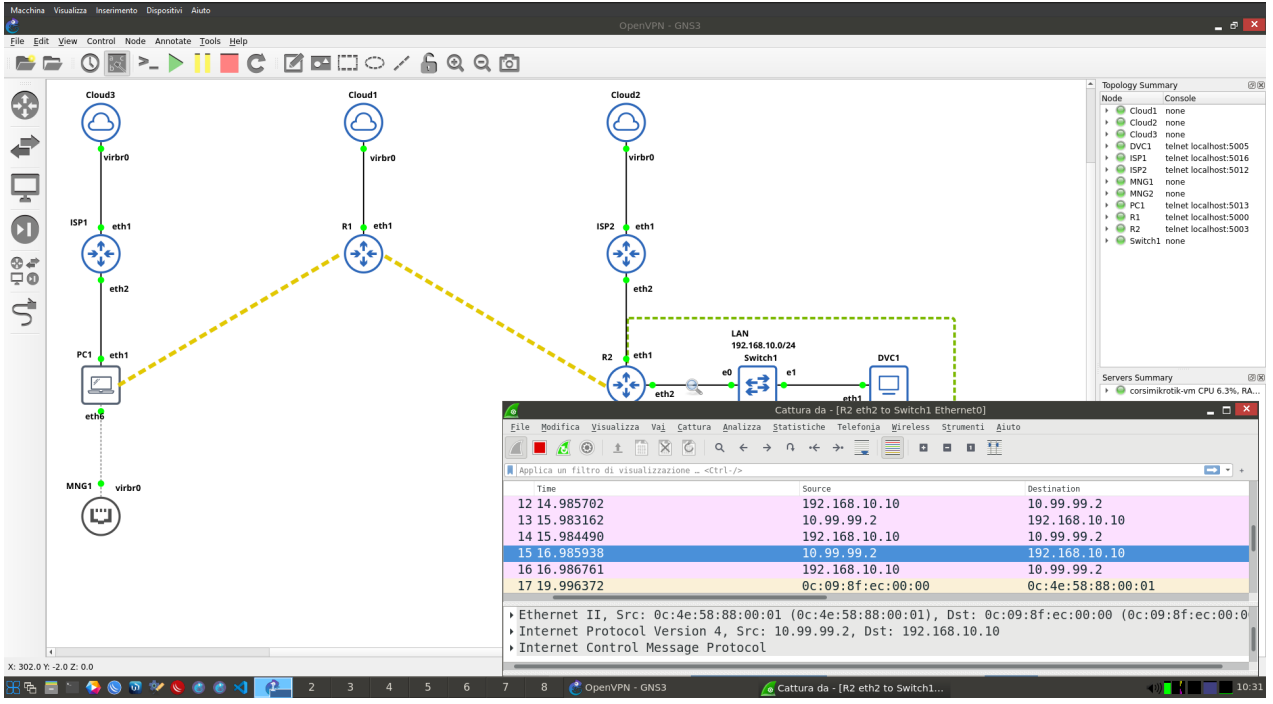
- In this presentation I have touched on a lot of points
- Understanding **routing process** and **packet flow** is essential



https://help.mikrotik.com/docs/spaces/ROS/pages/328227/Packet+Flow+in+RouterOS

Conclusion

To deeply understand this presentation, I suggest trying it out on some labs and analyzing packets with WireShark



Conclusion

- Last year, at MTPC 2024, I presented a solution on how to create these virtual labs with GNS3
- If you don’t know how to create labs or where to start, watch this presentation: https://www.youtube.com/watch?v=IdFTgPLXqKs

SCAN ME

“Dreams, without goals, are just dreams.

And ultimately they fuel disappointment.

On the road, to achieving your dreams,
you **MUST** apply discipline.

But more importantly
Consistency

Because without commitment, you never start
but **without consistency, you never finish.”**

Denzel Washinton

Thank you for your time.

Thanks to
Ron, Lorenzo, Jaromir
and all the staff
for organizing this fantastic meeting.

Linkedin
Marco Boschini