Yatuaprendes

MIKROTIK
CERTIFIED TRAINER

Who am I?

Yatuaprendes

I’m Jorge Castellet

I’m a Mikrotik Certified Trainer MTCNA, MTCIPv6E, MTCRE, MTCTCE, MTCWE,MTCUME, MTCINE, MTCSE,MTCSWE, MTCEWE

I’m freelancer

j.castellet@yatuaprendes.com

WireGuard for

~~zonies~~

dummies

MIKROTIK professionals conference

Yatuaprendes

What is WireGuard?

*"WireGuard® is an extremely simple yet fast and modern VPN that utilizes **state-of-the-art cryptography**. It aims to be faster, simpler, leaner, and more useful than IPsec, while avoiding the massive headache. It intends to be considerably more performant than OpenVPN. WireGuard is designed as a general-purpose VPN for running on embedded interfaces and super computers alike, fit for many different circumstances. "*

"…but already it might be regarded as the most secure, easiest to use, and simplest VPN solution in the industry."

Source WireGuard website https://www.wireguard.com/

What is WireGuard?

Symple & Easy to use

🔑 Crytpographically Sound

</> Minimal Attach Surface

⚡ High Performance

🎓 Well Defined & Thoroughly Considered

Source WireGuard website https://www.wireguard.com/

Symple & Easy to use

- ✓Aims to be as easy to configure and deploy as SSH.
- ✓There is no need to manage connections.
- ✓Presents an extremely basic yet powerful interface.

Source WireGuard website https://www.wireguard.com/

Crytpographically Sound

- ✓Uses state-of-the-art cryptography
  - Noise protocol framework, Curve25519, ChaCha20, Poly1305, BLAKE2
  - SipHash24
  - HKDF
- ✓Conservative and reasonable choices
- ✓Reviewed by cryptographers

Source WireGuard website https://www.wireguard.com/

</> Minimal Attach Surface

- ✓Designed with ease-of-implementation and simplicity in mind.
- ✓Implemented in very few lines of code.
- ✓Easily auditable for security vulnerabilities.
- ✓Comprehensively reviewable by single individuals.

Source WireGuard website https://www.wireguard.com/

High Performance

- ✓High-speed cryptographic primitives.
- ✓Lives inside the Linux kernel.

Source WireGuard website https://www.wireguard.com/

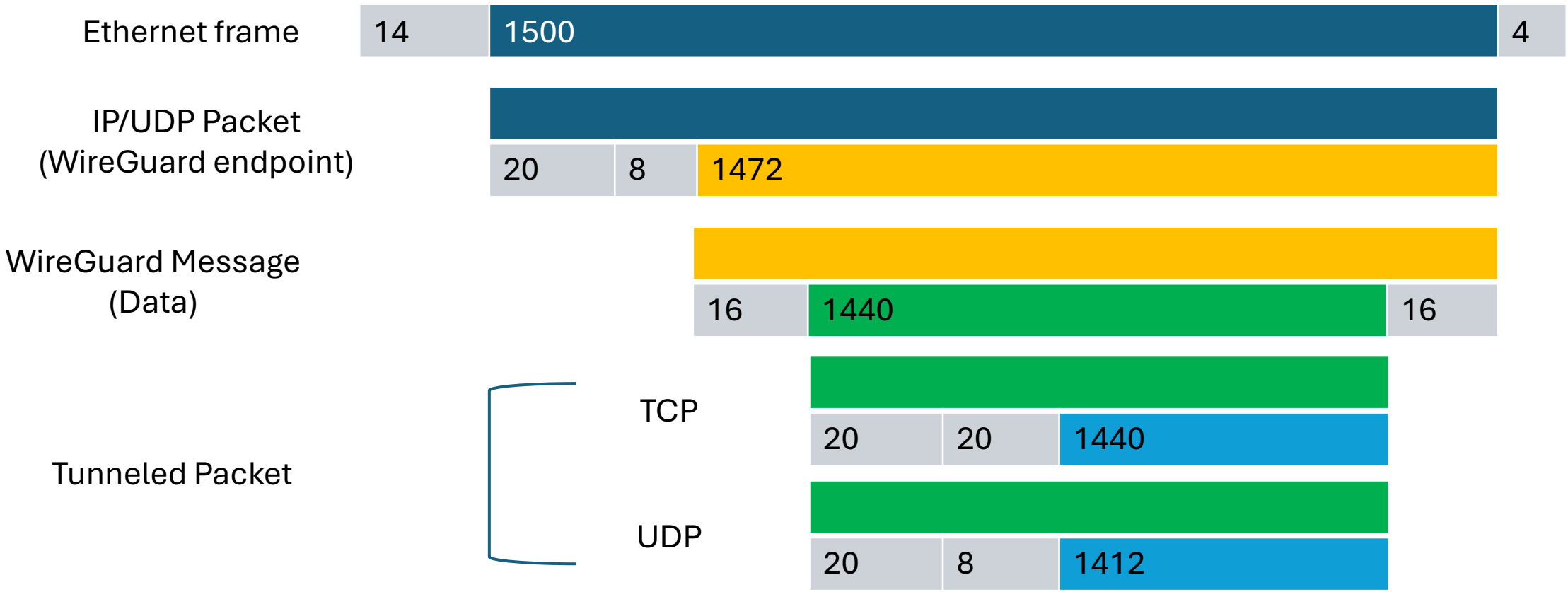
Well Defined & Thoroughly Considered

Is the result of a lengthy and thoroughly considered academic process.

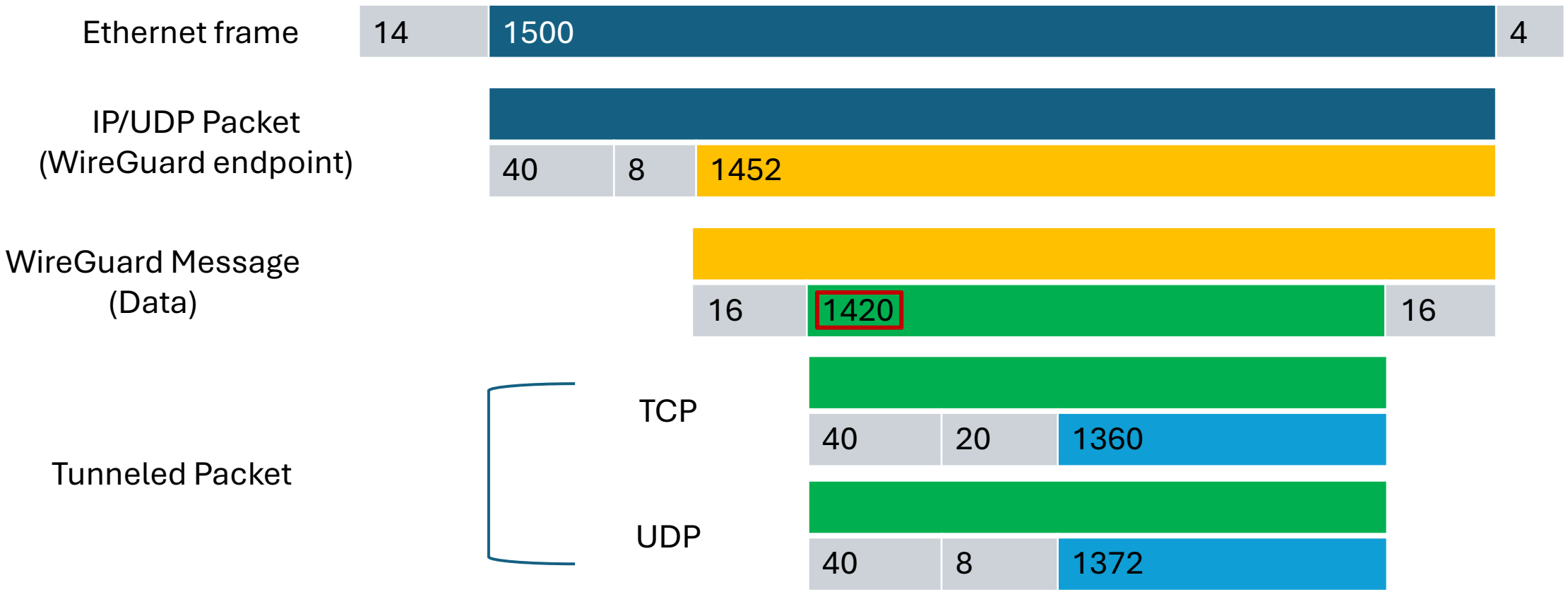
Resulting in the technical whitepaper which clearly defines the protocol and the intense considerations that went into each decision.

Source WireGuard website https://www.wireguard.com/

IPv4

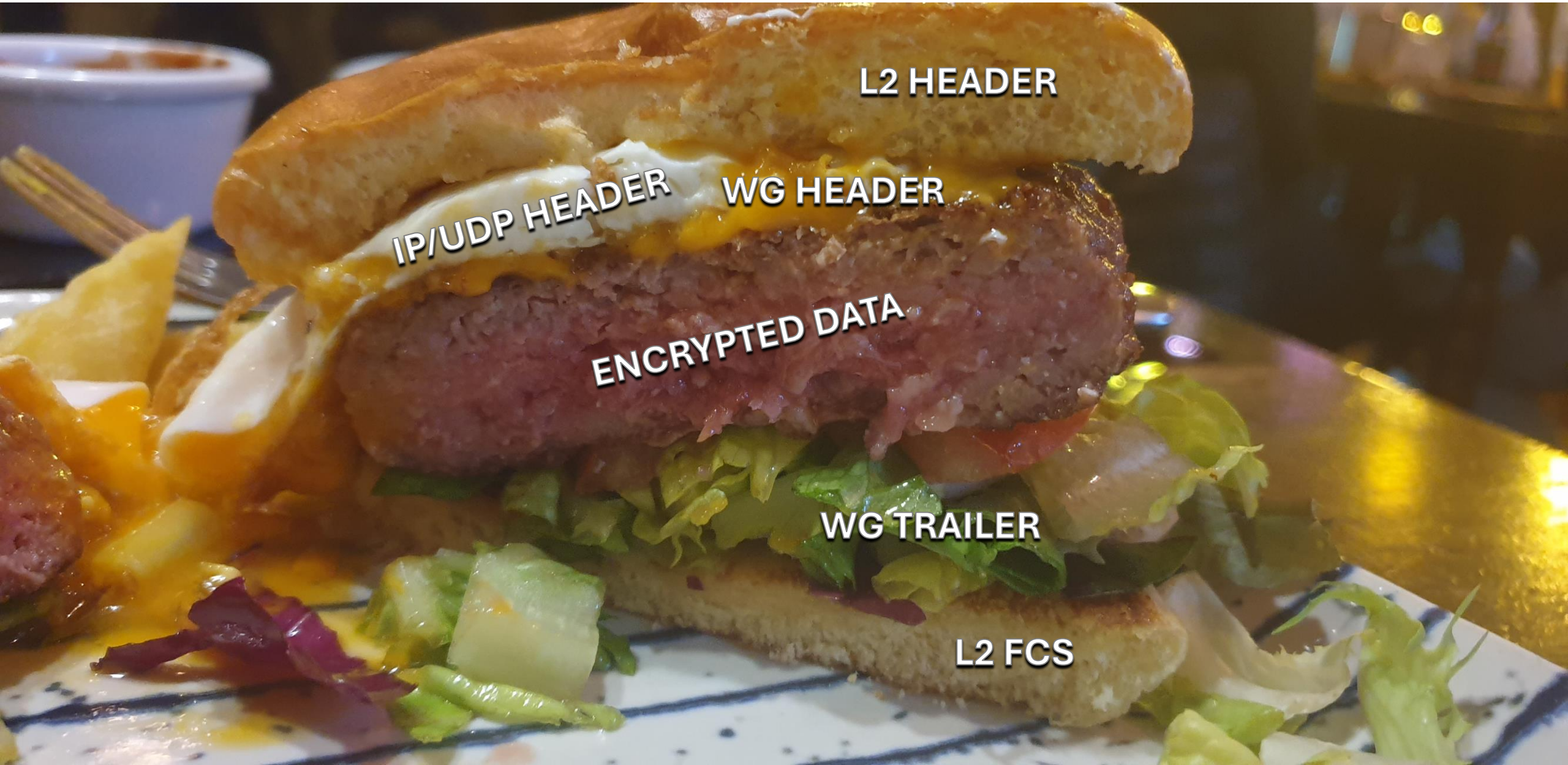

Ethernet frame
14
1500
4
IP/UDP Packet
(WireGuard endpoint)
20
8
1472
WireGuard Message
(Data)
16
1440
16
Tunneled Packet
TCP
20
20
1440
UDP
20
8
1412

IPv6

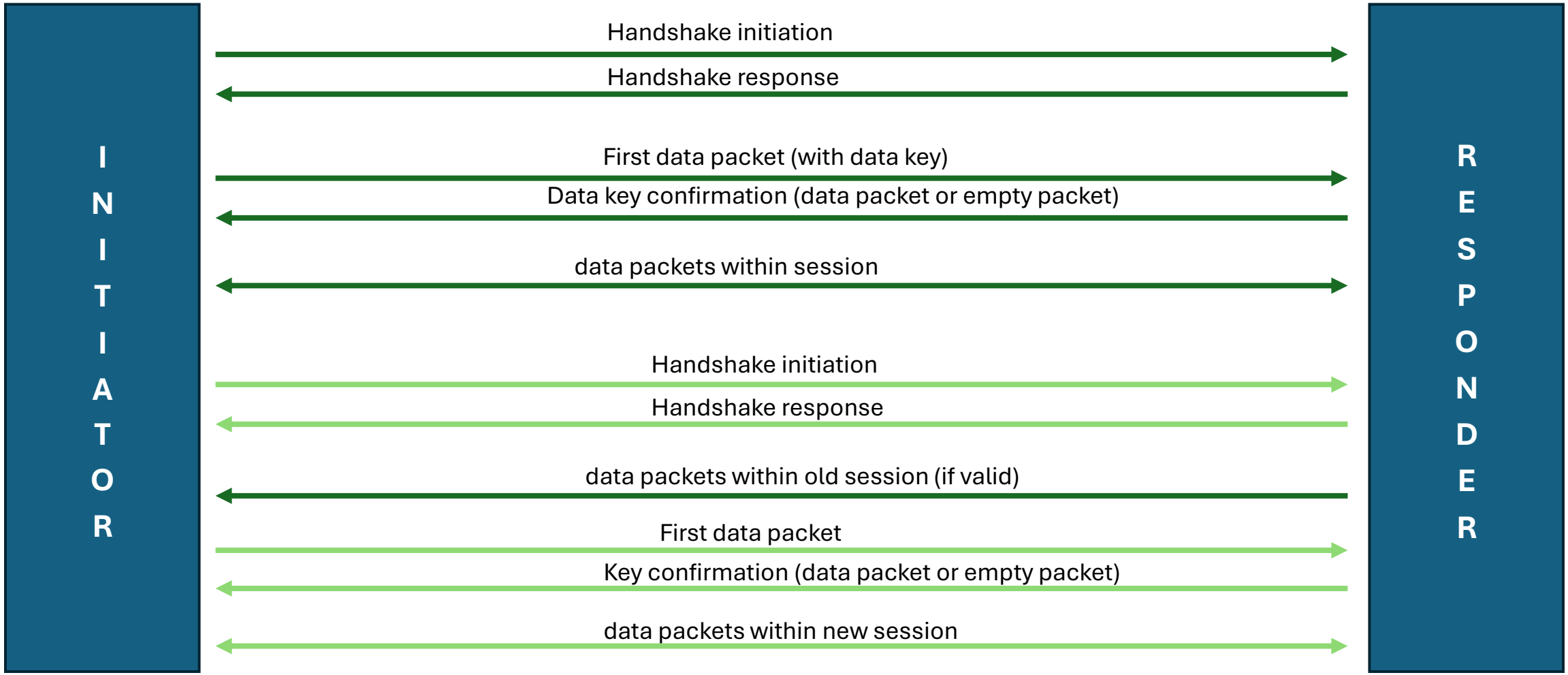


Mikrotik default MTU for Wireguard is 1420.
Therefore, we can tunnel IPv6 packets over IPv6 endpoint as well IPv4 over IPv4 or mix them.

WireGuard real life example



Packet exchange

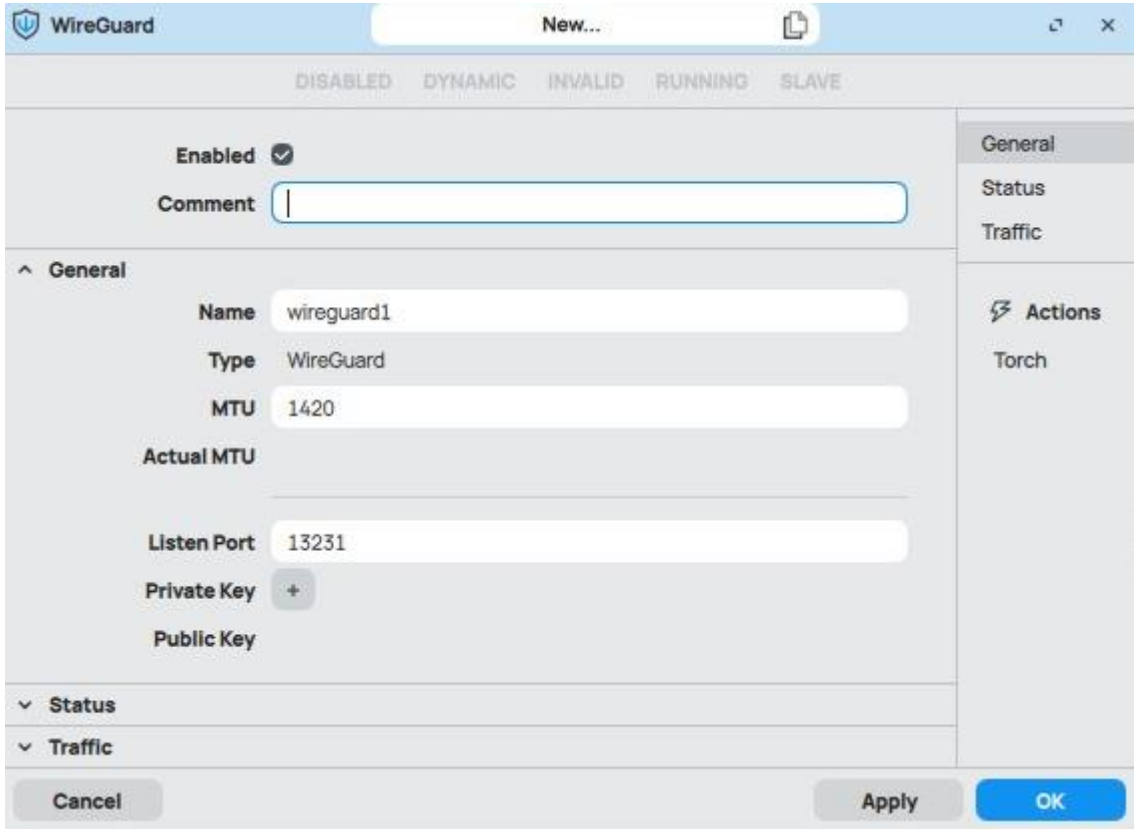


Key Exchange occurs Every few minutes. If response got a previous session key and it's still valid, can use it until receives a packet (with the new session) and then starts using the new session key

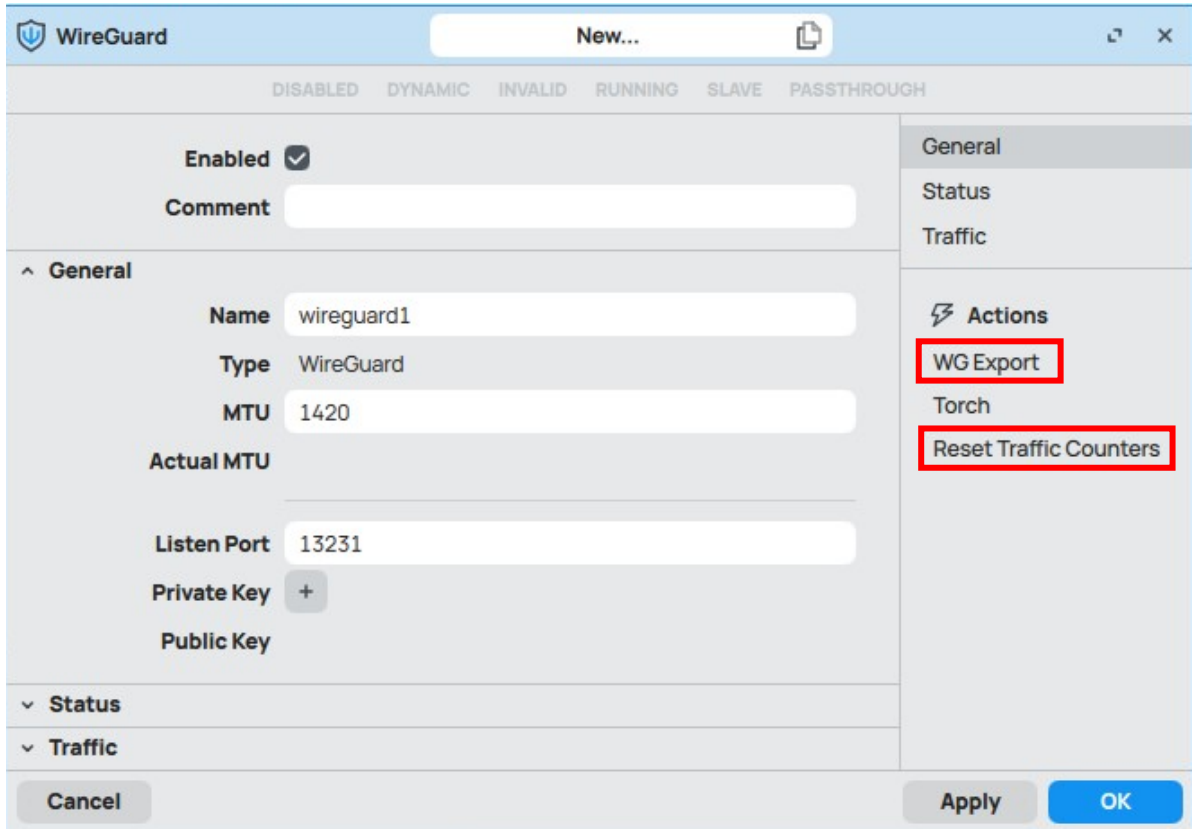
WireGuard on Mikrotik

- *Not implemented in ROS6.*
- *Implemented in ROS7.*
- *Interface changes across versions.*
- *It is in the WireGuard menu.*
- *No need to install additional packages.*

WireGuard menu evolution

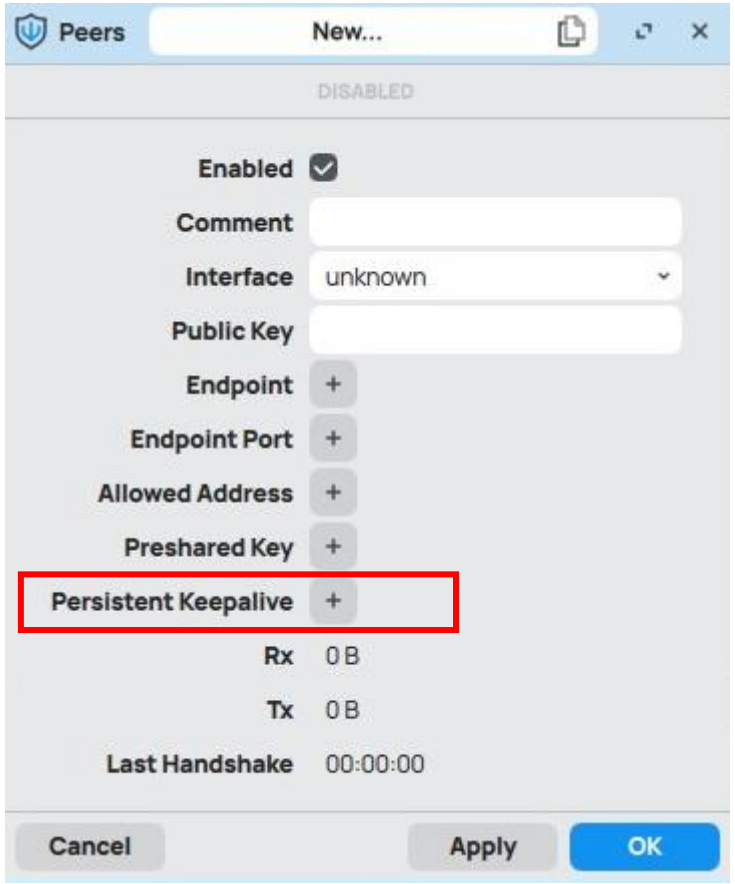


Ros7.1

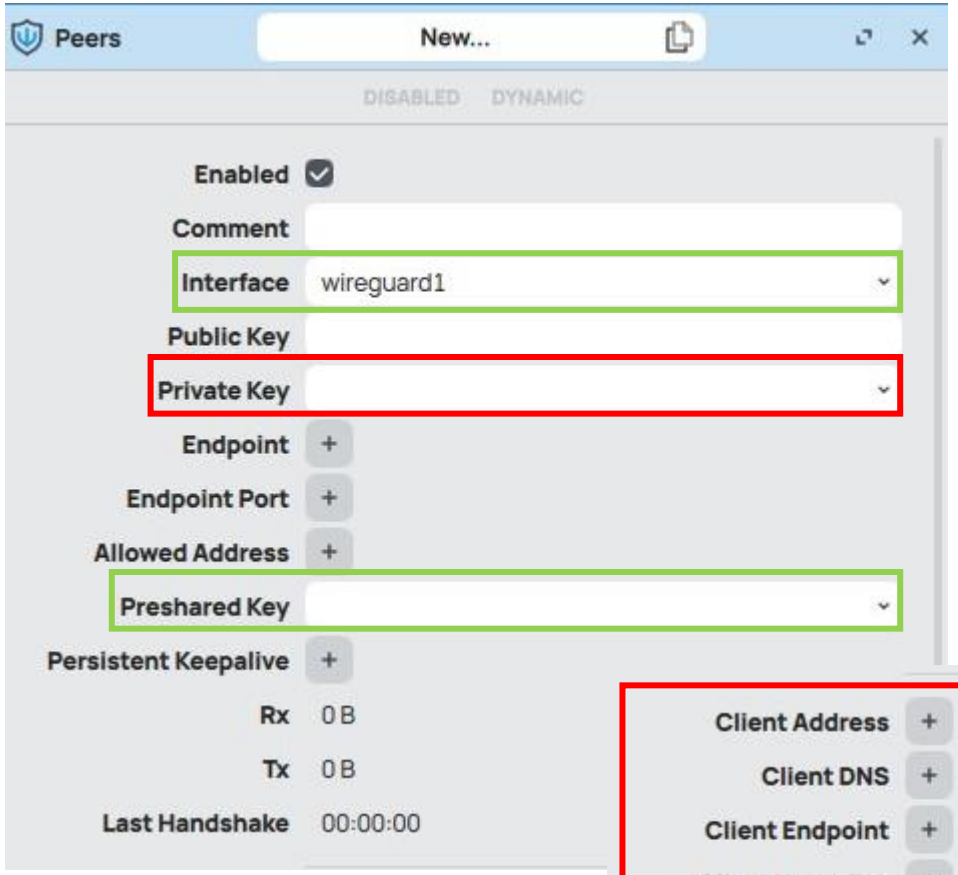


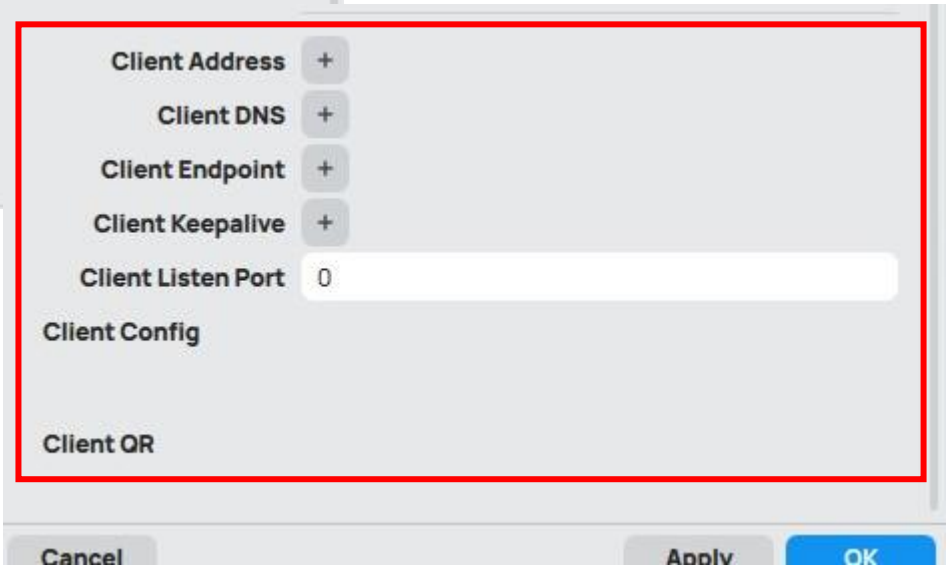
Ros7.12

WireGuard menu evolution

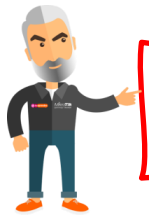


Ros7.1



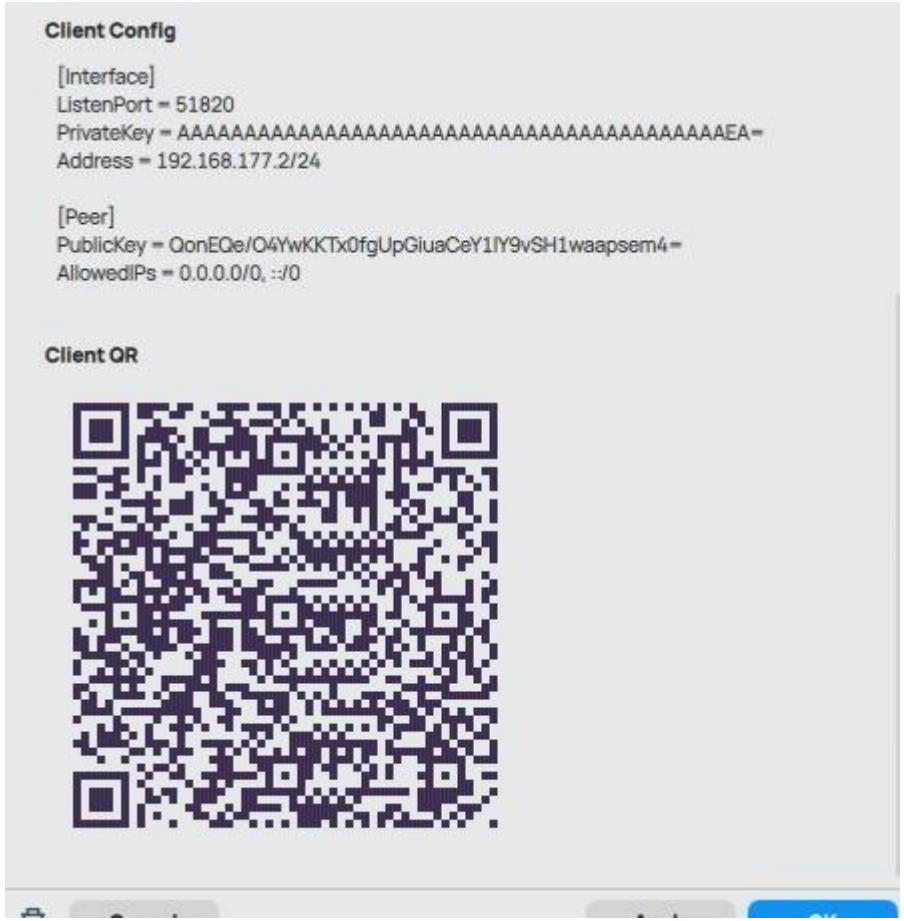


Ros7.12



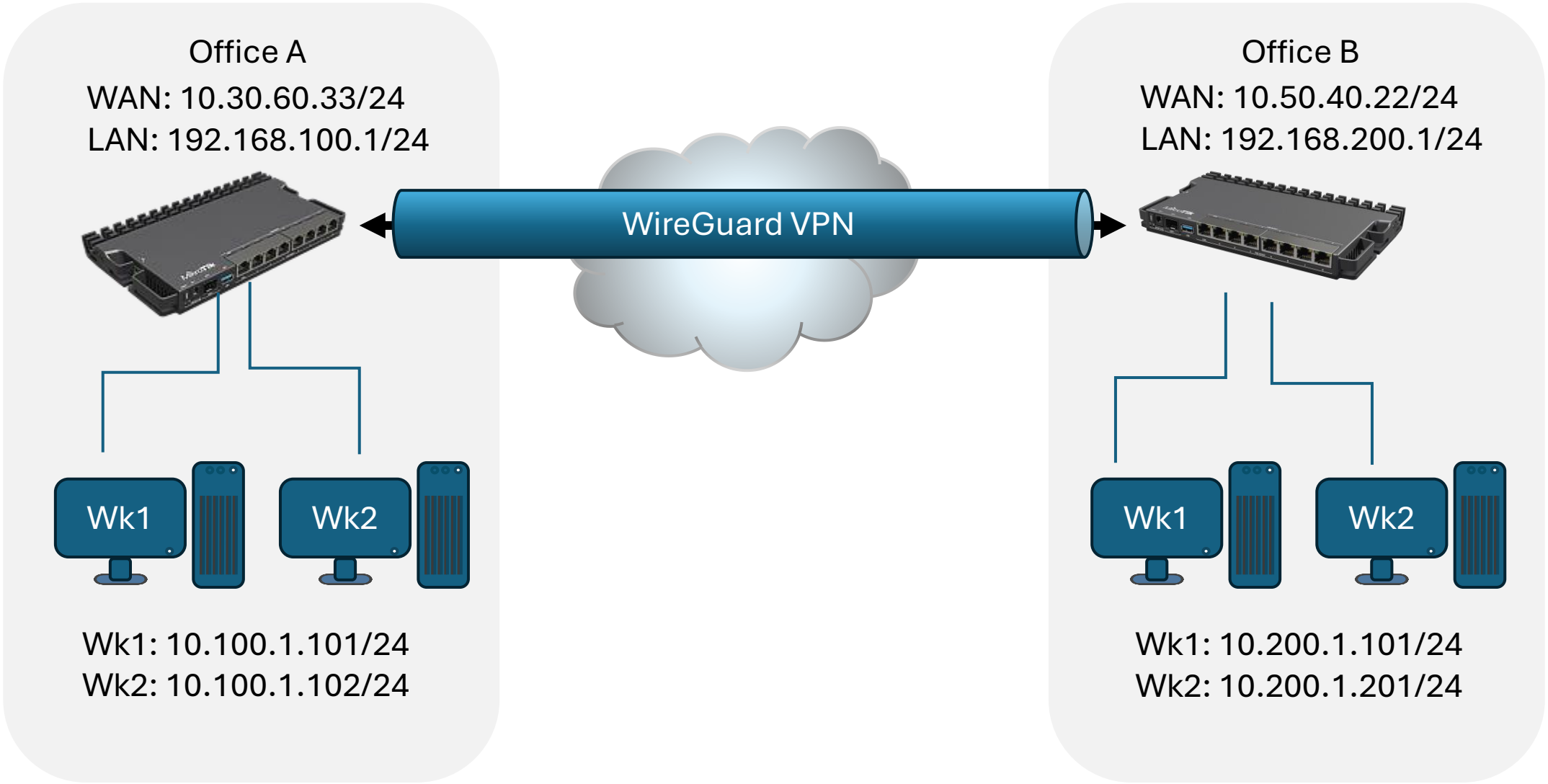
Persistent Keepalive is mandatory in Mikrotik for the peer to get up and running.

WireGuard menu evolution



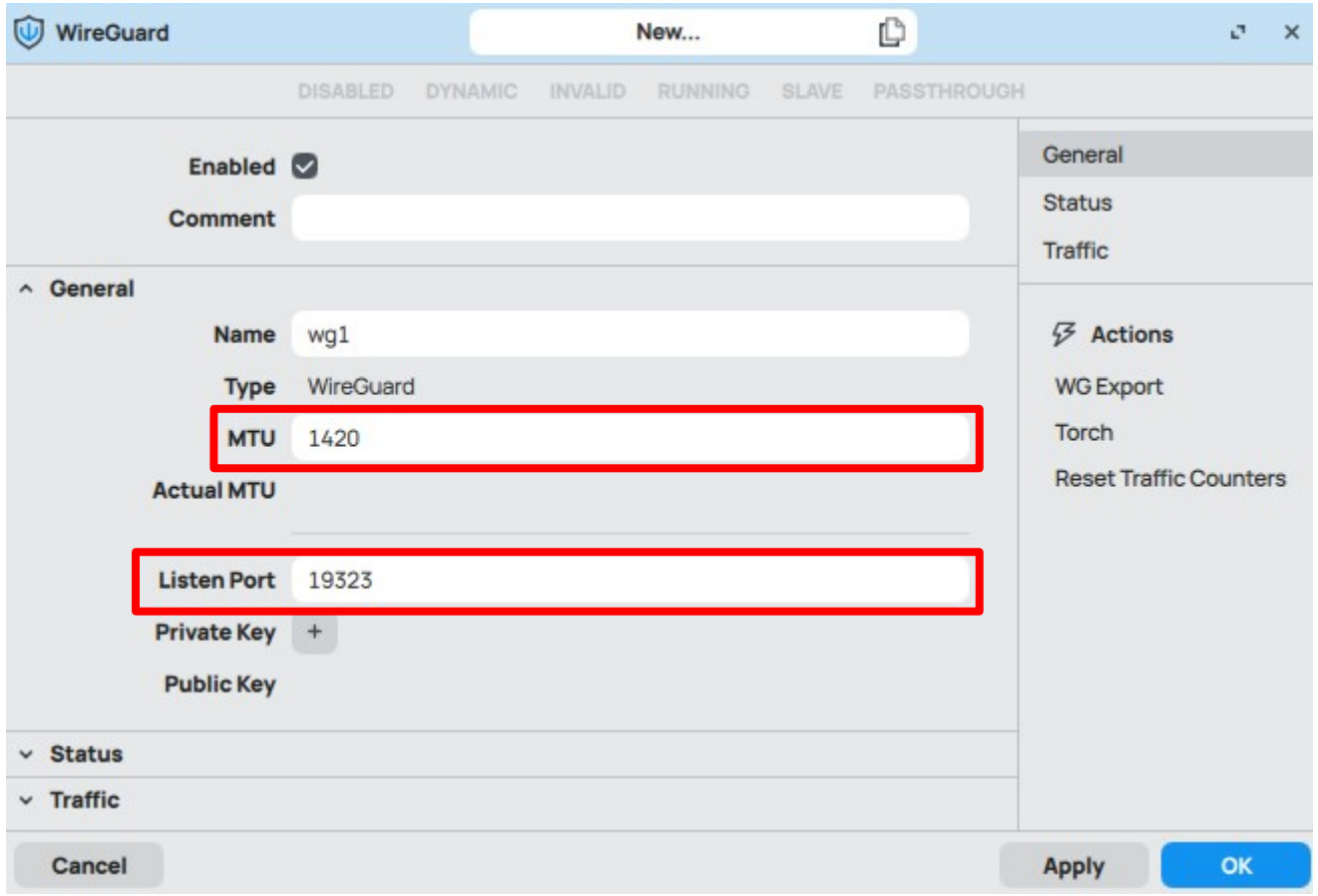
Ros7.12

Site to site configuration



Site to site configuration (wireguard interface)

Create interface

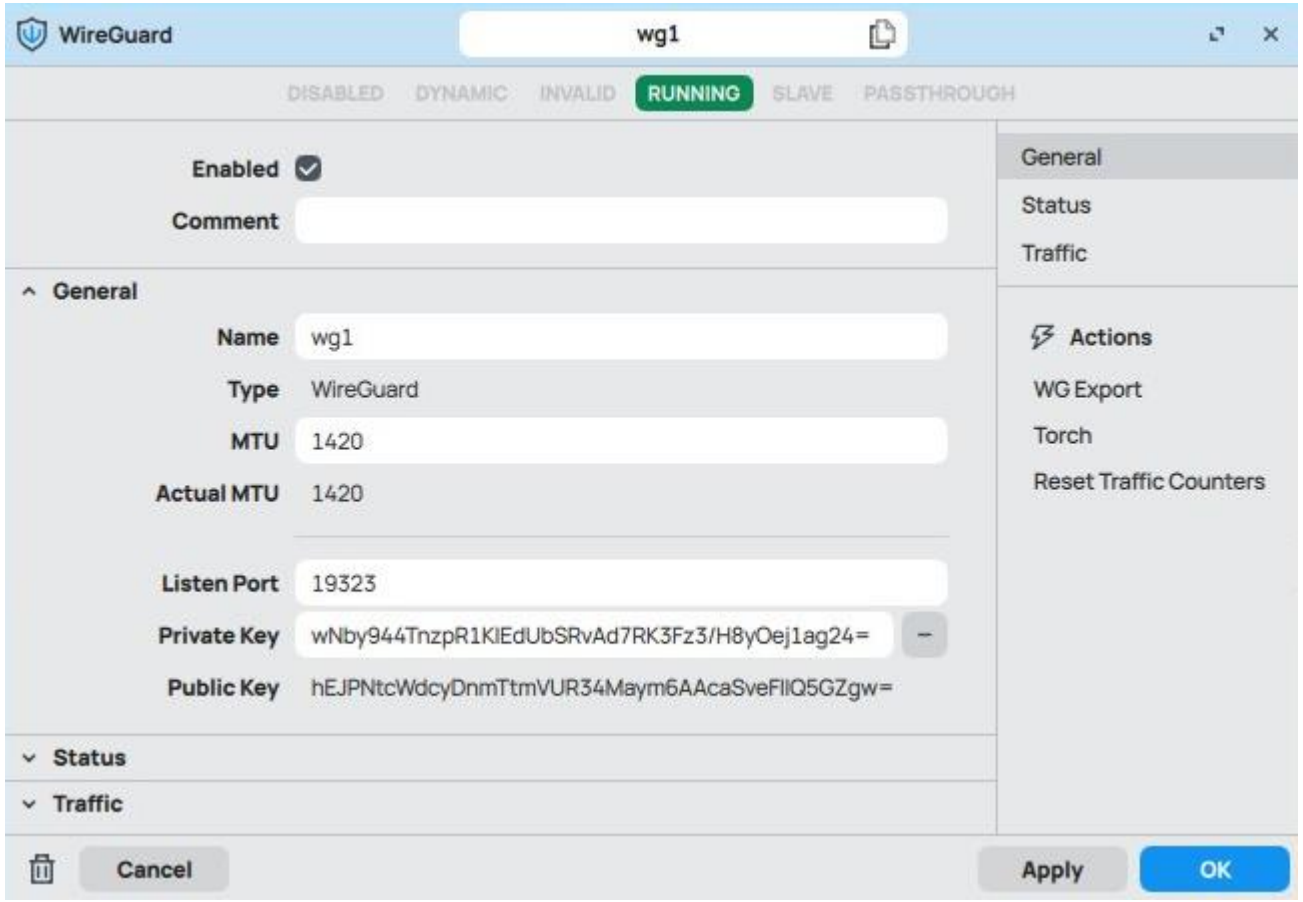


Optional fields:
mtu default value is 1420
listen-port default value is 13231

```
/int wireguard/add name=wg1 mtu=1420 listen-port=19323
```

Site to site configuration (wireguard interface)

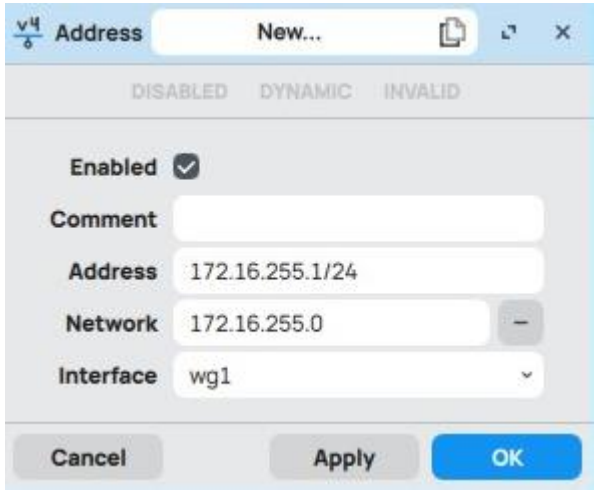
Interface info




```
/int wireguard/pr
Flags: X - disabled; R - running
 0  R name="wg1" mtu=1420 listen-port=19323
      private-key="wNby944TnzpR1KIEdUbSRvAd7RK3Fz3/H8yOej1ag24="
      public-key="hEJPNtcWdcyDnmTtmVUR34Maym6AAcaSveFIIQ5GZgw="
```

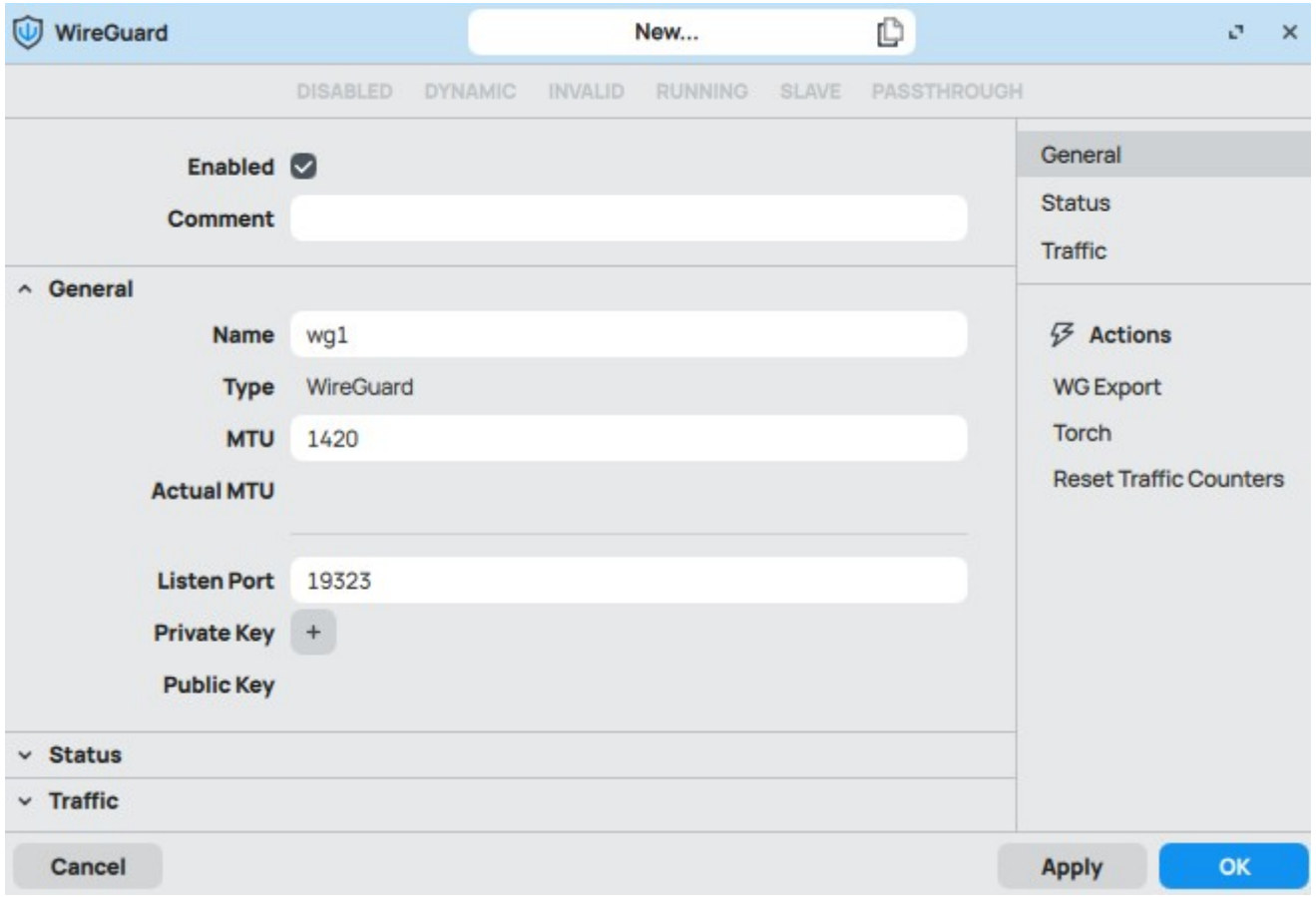
Site to site configuration (wireguard interface)

Assign IPv4 address




```
/ip address/add interface=wg1 address=172.16.255.1/24
```

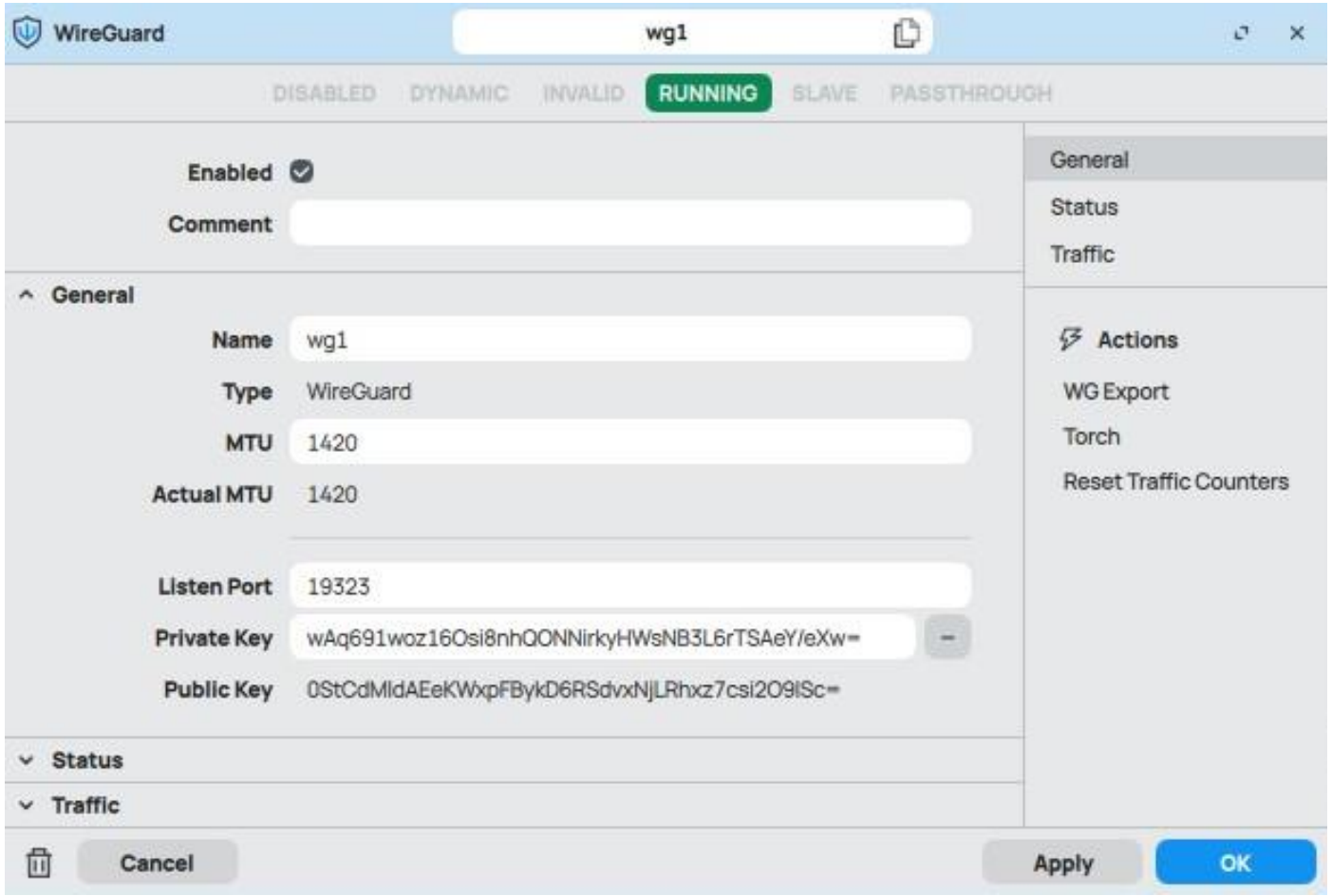
Site to site configuration (winbox version)



Create interface

```
/int wireguard/add name=wg1 mtu=1420 listen-port=19323
```

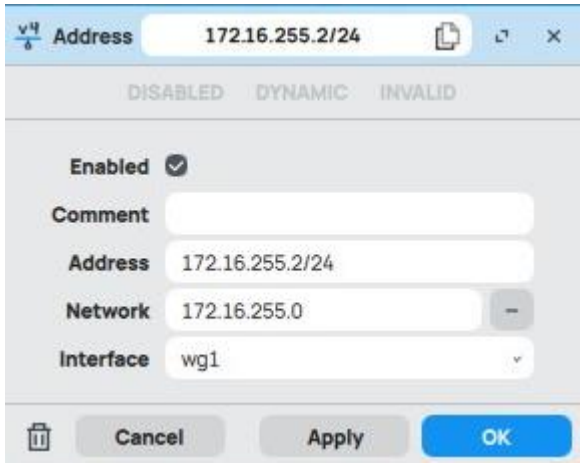
Site to site configuration (winbox version)



Interface info

```
/int wireguard/pr
Flags: X - disabled; R - running
 0   R name="wg1" mtu=1420 listen-port=19323
       private-key="wAq691woz16Osi8nhQONNirkyHWsNB3L6rTSAeY/eXw="
       public-key="0StCdMldAEeKWxpFBykD6RSdvxNjLRhxz7csi2O9ISc="
```

Site to site configuration (winbox version)



Assign IPv4 address

```
/ip address/add interface=wg1 address=172.16.255.2/24
```

Site to site configuration

2) Add peer (Office B)

```
/int wireguard/peers/add interface=wg1 \
endpoint-address=10.50.40.22 \
endpoint-port=19323 \
public-key="0StCdMld....csi2O9ISc=" \
allowed-address=172.16.255.2/32,192.168.200.0/24 \
persistent-keepalive=1m
```

1) Show Wireguard interface info

```
/int wireguard/pr
Flags: X - disabled; R - running
 0   R name="wg1" mtu=1420 listen-port=19323
       private-key="wAq691wo....TSAeY/eXw="
       public-key="0StCdMld....csi2O9ISc="
```

```
/ip address/pr
Columns: ADDRESS, NETWORK, INTERFACE
# ADDRESS            NETWORK         INTERFACE
0 172.16.255.2/24    172.16.255.0    wg1
1 10.50.40.22/24     10.50.40.0      ether1
2 192.168.200.1/24   192.168.200.0   bridgeLAN
```

Site to site configuration

1) Show Wireguard interface info

```
/int wireguard/pr
Flags: X - disabled; R - running
 0   R name="wg1" mtu=1420 listen-port=19323
       private-key="wNby944Tnz....yOej1ag24="
       public-key="hEJPNtcWd....FIIQ5GZgw="
```

```
/ip address/pr
Columns: ADDRESS, NETWORK, INTERFACE
# ADDRESS            NETWORK         INTERFACE
0 172.16.255.1/24    172.16.255.0    wg1
1 10.30.60.33/24     10.30.60.0      ether1
2 192.168.100.1/24   192.168.100.0   bridgeLAN
```

2) Add peer (Office A)

```
/int wireguard/peers/add interface=wg1 \
endpoint-address=10.30.60.33 \
endpoint-port=19323 \
public-key="hEJPNtcWd....FIIQ5GZgw=" \
allowed-address=172.16.255.1/32,192.168.100.0/24 \
persistent-keepalive=1m
```

Site to site configuration

WireGuard is a layer 3 VPN

➢We need to manually add routes on both sides.


```
/ip route/pr
Flags: D - DYNAMIC; A - ACTIVE; c - CONNECT, s - STATIC
Columns: DST-ADDRESS, GATEWAY, DISTANCE
#      DST-ADDRESS        GATEWAY        DISTANCE
0  As  0.0.0.0/0          10.30.60.1            2
   DAc 10.30.60.0/24      ether1                0
   DAc 172.16.255.0/24    wg1                   0
   DAc 192.168.100.0/24   bridgeLAN             0
1  As  192.168.200.0/24   172.16.255.2          1
```


```
/ip route/pr
Flags: D - DYNAMIC; A - ACTIVE; c - CONNECT, s - STATIC
Columns: DST-ADDRESS, GATEWAY, DISTANCE
#      DST-ADDRESS        GATEWAY        DISTANCE
0  As  0.0.0.0/0          10.50.40.1            1
   DAc 10.50.40.0/24      ether1                0
   DAc 172.16.255.0/24    wg1                   0
1  As  192.168.100.0/24   172.16.255.1          1
   DAc 192.168.200.0/24   bridgeLAN             0
```

TCP MSS

WireGuard do not perform clamp TCP MSS.

- We need to do it manually on mangle using action change-mss.

```
/int pr
Flags: R - RUNNING
Columns: NAME, TYPE, ACTUAL-MTU, L2MTU, MAC-ADDRESS
#    NAME         TYPE        ACTUAL-MTU  L2MTU  MAC-ADDRESS
0 R ether1       ether             1500          08:00:27:69:24:38
1 R ether2       ether             1500          08:00:27:59:62:66
2 R bridgeLAN    bridge            1500  65535   92:B0:7A:C9:E6:73
4 R wg1          wg                1420
```

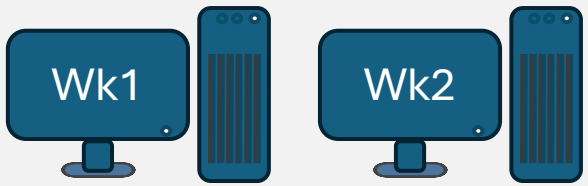
1420 – 20 (ip header) - 20 (tcp) = 1380

```
/ip firewall mangle
add action=change-mss chain=forward \
new-mss=1380 out-interface=wg1 protocol=tcp \
tcp-flags=syn tcp-mss=1381-65535
```

Office A

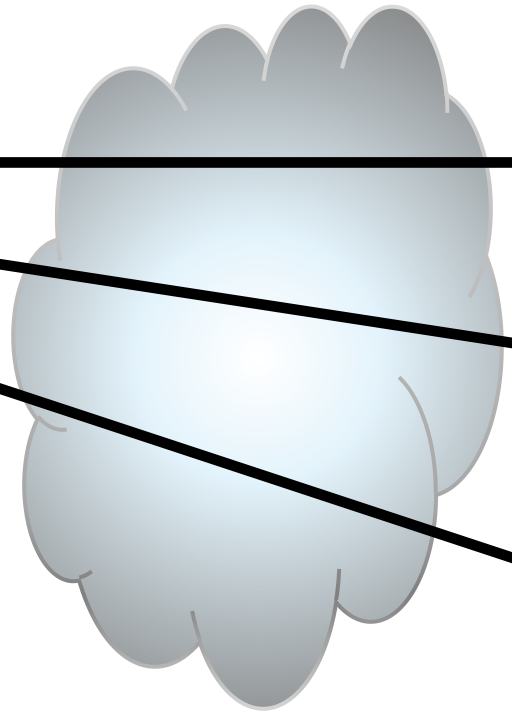
WAN: 10.30.60.33/24

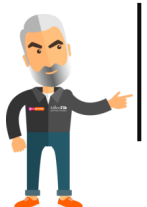
LAN: 192.168.100.1/24



Wk1: 10.100.1.101/24

Wk2: 10.100.1.102/24





In all cases, the MikroTik at Office A does not have the endpoint address or port information of the peer.

ROADWARRIOR

Roadwarrior, originally refers to sales agents who spend their workday traveling from one place to another and do not have a fixed location where they can be reached.

In a VPN, a roadwarrior is a device that does not have a known public IP and is therefore only capable of starting the VPN.

We will classify them into:

- Mikrotik devices.
- Phone devices.
- Laptops.

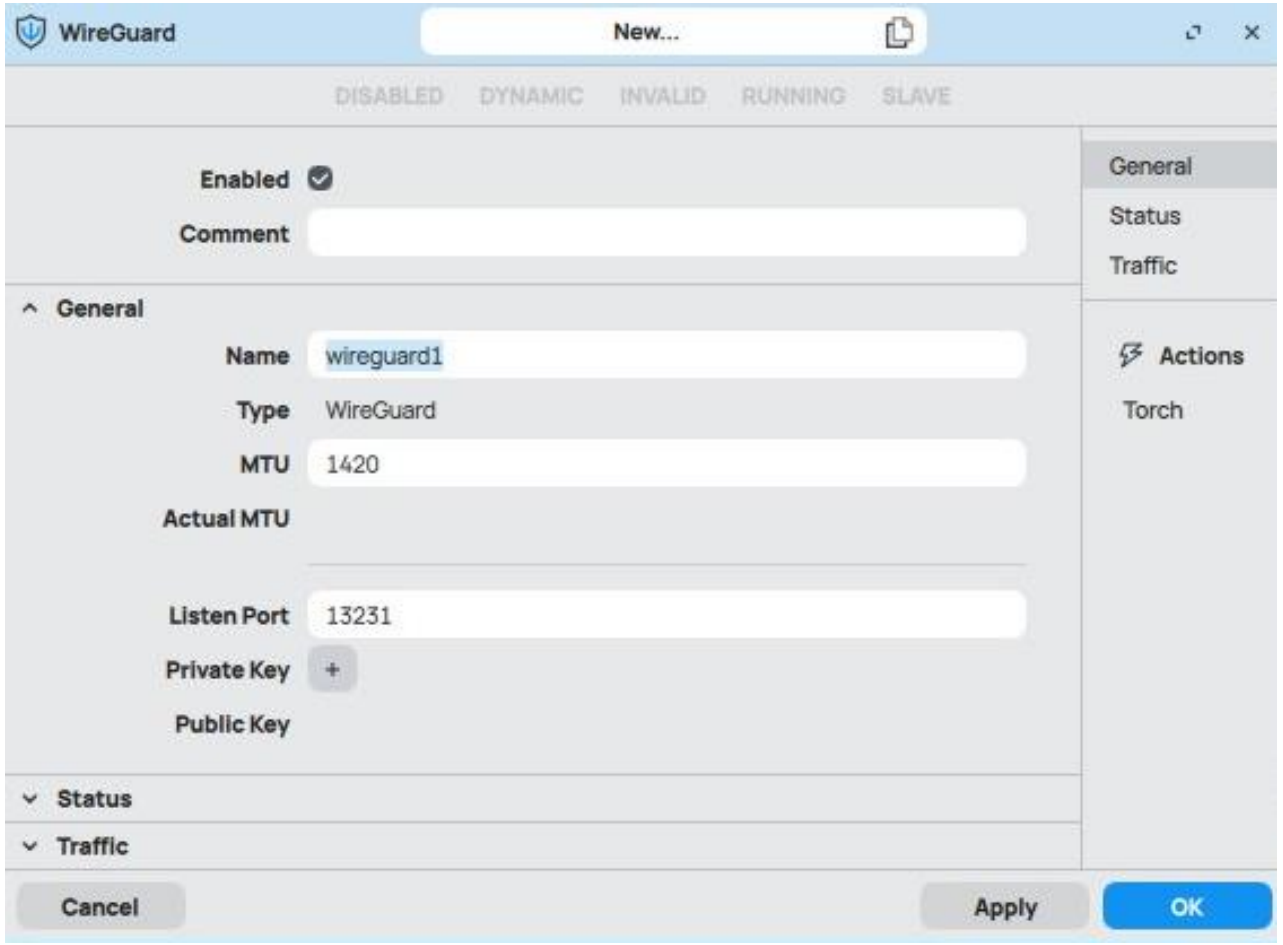
All we need to configure for each peer on the mikrotik at Office A is:

✓The wireguard interface.

✓The remote's public key.

✓The allowed addresses.

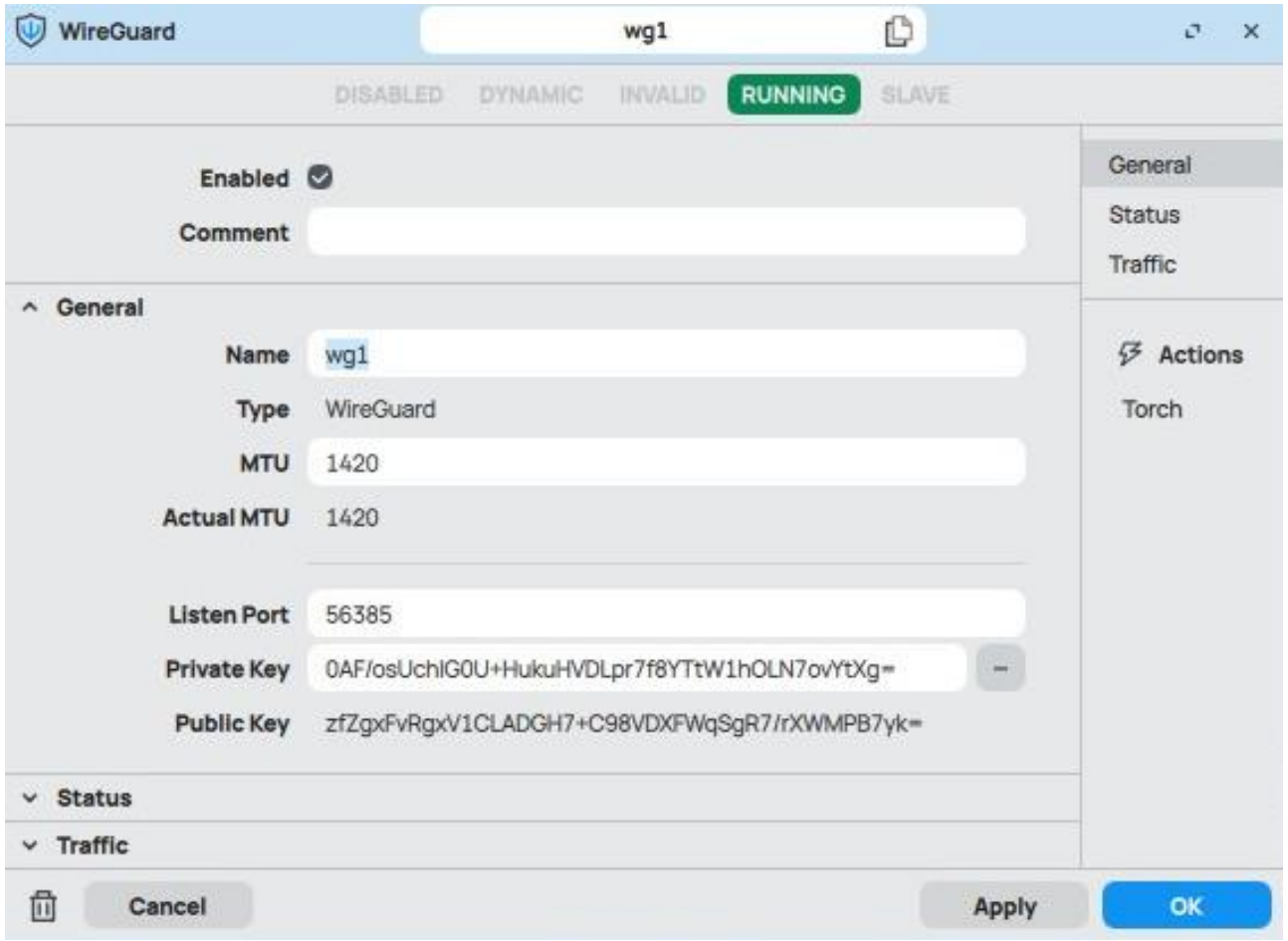
Mikrotik as ROADWARRIOR



Create interface

```
/int wireguard/add name=wg1 mtu=1420 listen-port=13231
```

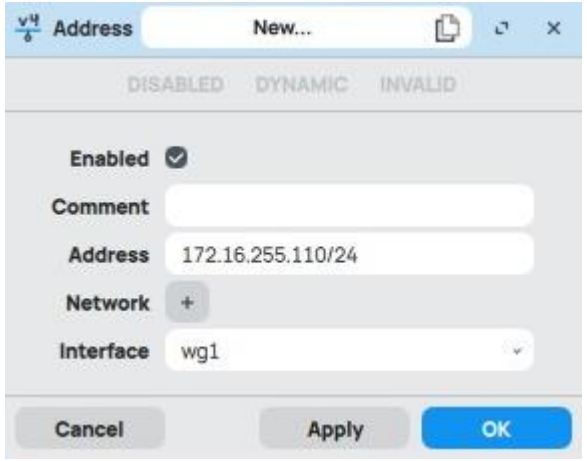
Mikrotik as ROADWARRIOR



Interface info

```
/int wireguard/pr
Flags: X - disabled; R - running
 0  R name="wg1" mtu=1420 listen-port=19323
     private-key="0AF/osUchIG0U+HukuHVDLpr7f8YTtW1hOLN7ovYtXg="
     public-key="zfZgxFvRgxV1CLADGH7+C98VDXFWqSgR7/rXWMPB7yk="
```

Mikrotik as ROADWARRIOR



Assign IPv4 address

```
/ip address/add interface=wg1 address=172.16.255.110/24
```

Mikrotik as ROADWARRIOR

1) Show Wireguard interface info

```
/int wireguard/pr
Flags: X - disabled; R - running
 0   R name="wg1" mtu=1420 listen-port=19323
       private-key="0AF/osUc....LN7ovYtXg="
       public-key="zfZgxFvR....rXWMPB7yk="
```

2) Add peer (RoadWarrior)

```
/int wireguard/peers/add interface=wg1 \
public-key="zfZgxFvR....rXWMPB7yk=" \
allowed-address=172.16.255.110/32 \
responder=true
```

since 7.15

No endpoint-address or endpoint-port
No Persistent keepalive (optional here)

```
/ip address/pr
Columns: ADDRESS, NETWORK, INTERFACE
# ADDRESS             NETWORK         INTERFACE
0 172.16.255.110/24   172.16.255.0    wg1
1 10.70.80.100/24     10.70.80.0      ether1
```

Mikrotik as ROADWARRIOR

1) Show Wireguard interface info

```
/int wireguard/pr
Flags: X - disabled; R - running
 0   R name="wg1" mtu=1420 listen-port=19323
       private-key="wNby944Tnz....yOej1ag24="
       public-key="hEJPNtcWd....FIIQ5GZgw="
```

```
/ip address/pr
Columns: ADDRESS, NETWORK, INTERFACE
# ADDRESS            NETWORK         INTERFACE
0 172.16.255.1/24    172.16.255.0    wg1
1 10.30.60.33/24     10.30.60.0      ether1
2 192.168.100.1/24   192.168.100.0   bridgeLAN
```

2) Add peer (Office A)

```
/int wireguard/peers/add interface=wg1 \
endpoint-address=10.30.60.33 \
endpoint-port=19323 \
public-key="hEJPNtcWd....FIIQ5GZgw=" \
allowed-address=0.0.0.0/0 \
persistent-keepalive=1m
```

Set up Persistent keepalive to get up the tunnel

Mikrotik as ROADWARRIOR

- **STATIC**

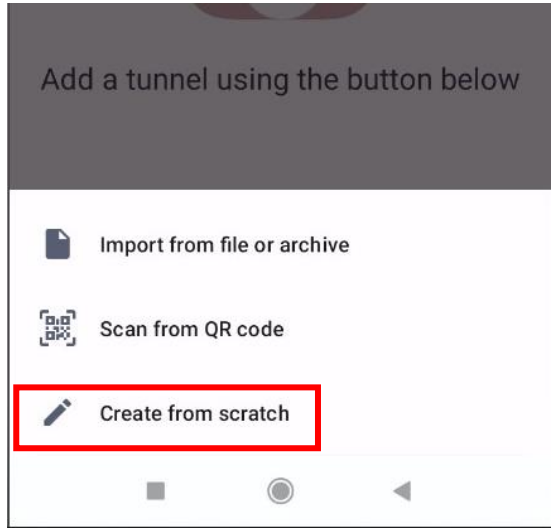
3) Add default gateway

/ip ro add dst-address=0.0.0.0/0 gateway=10.50.40.1

/ip ro add add dst-address=10.30.60.33 gateway=10.50.40.1

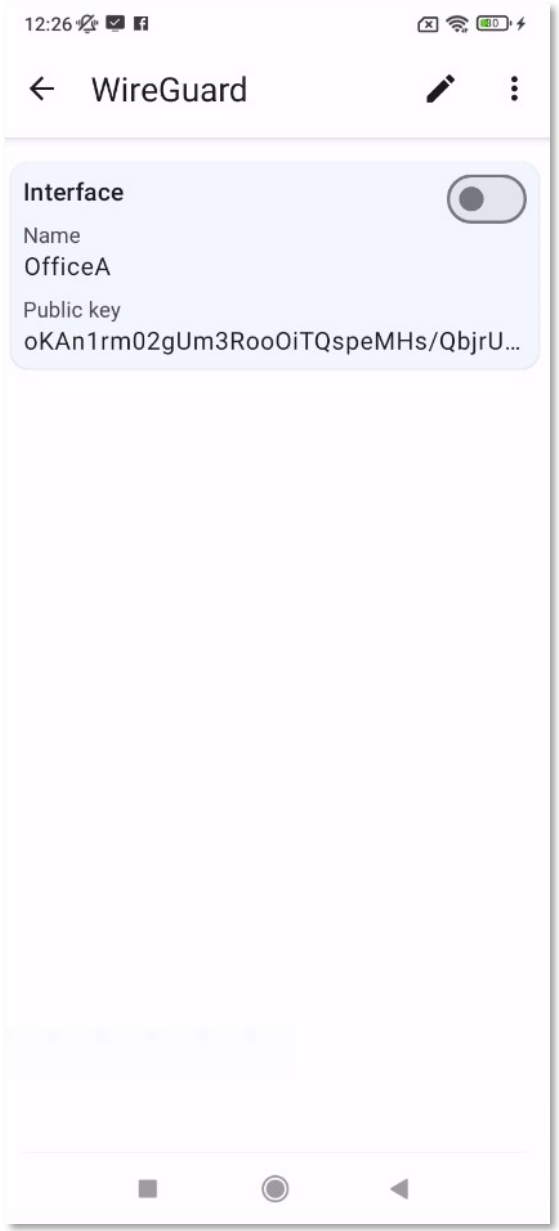
Mobile as ROADWARRIOR

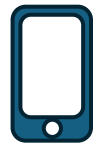
- Download and install WireGuard app.
- Create new from scratch
  - ➢Add interface information



Mobile as ROADWARRIOR

Press the rollup button to generate a key pair



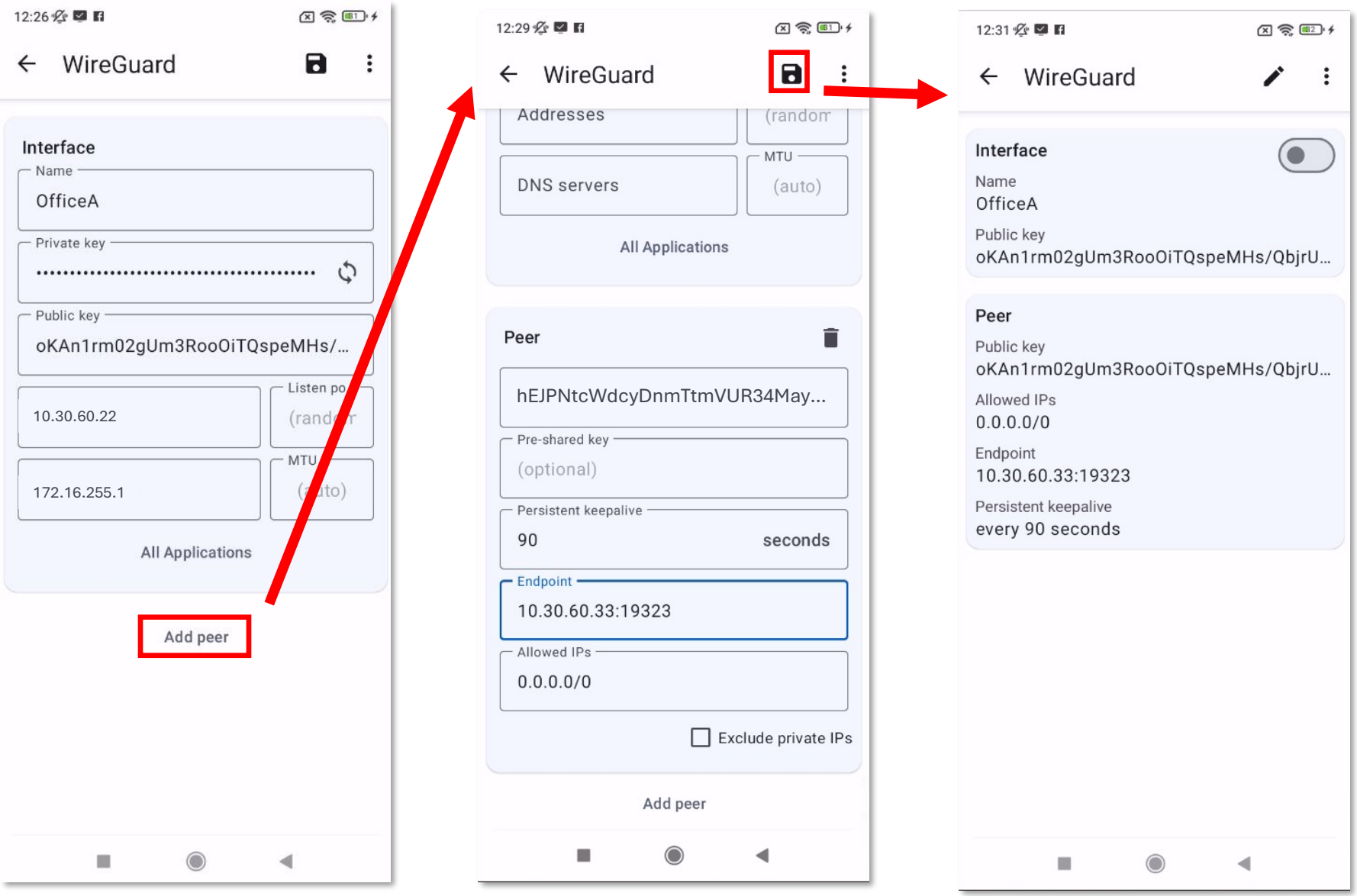


Create interface & Interface info

Not compliant with Mitchell's talk

Copy the public key to the clipboard to paste it later into Mikrotik Office A.

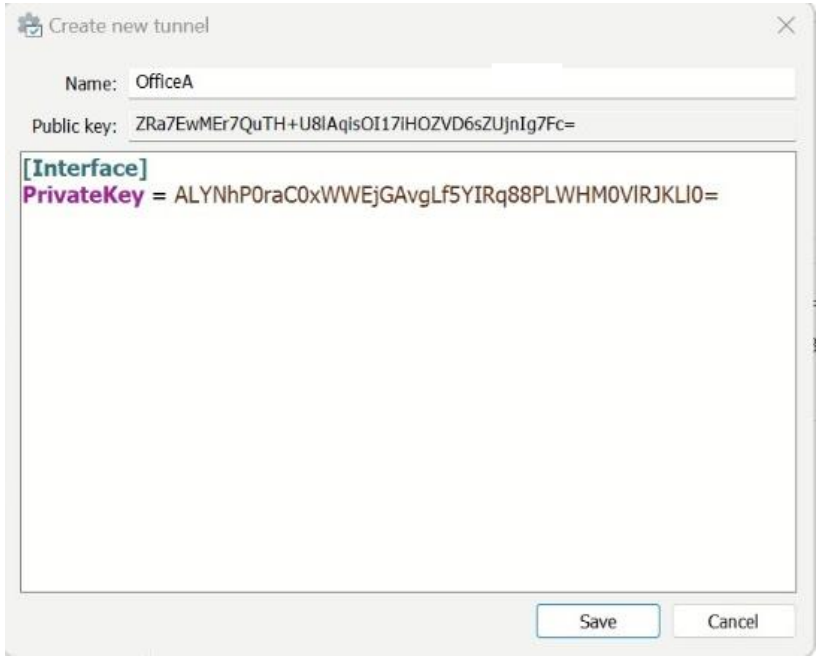
Mobile as RoadWarrior

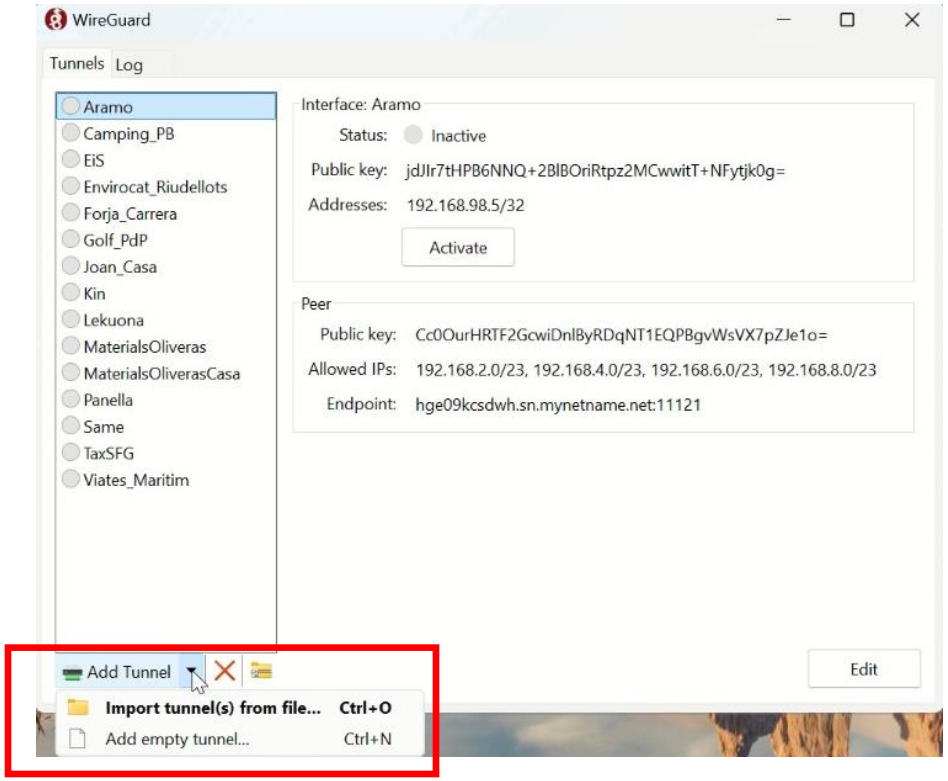


Add peer (Office A)

Windows Laptop as ROADWARRIOR

- Download Wireguard installer from WireGuard website.
- Run as administrator
- Press Ctrl+n to add new empty tunnel



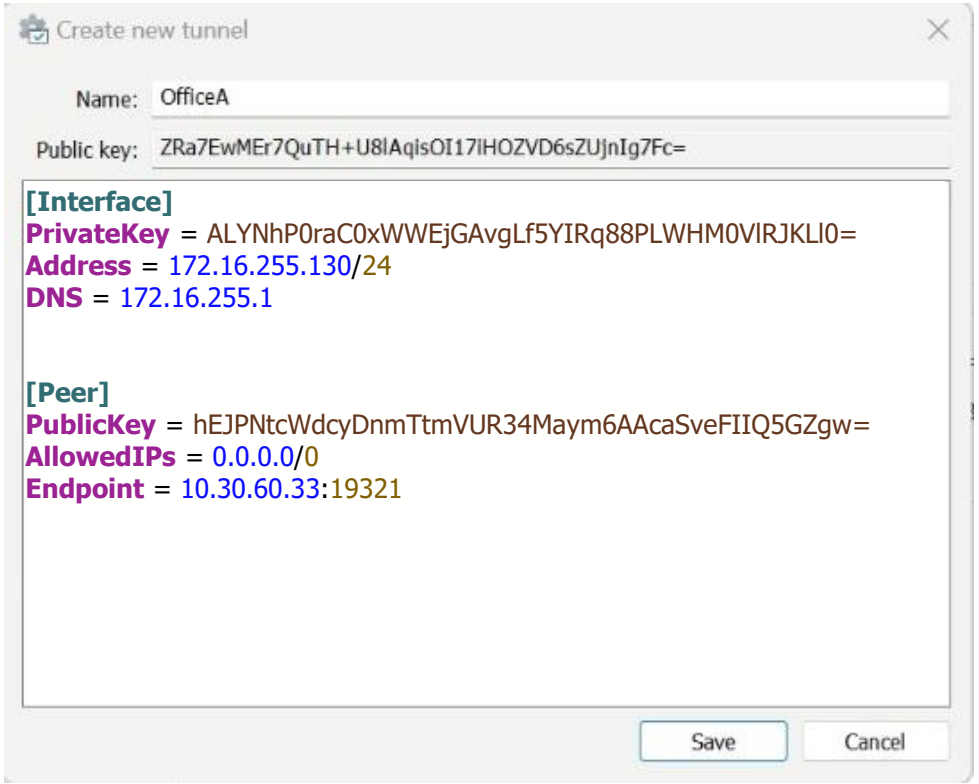


Windows Laptop as ROADWARRIOR

- Add a name to the new tunnel
- Add to interface section:

  `Address, DNS`
- Add the peer section:

  `PublicKey, AllowedIPs, Endpoint`



ROADWARRIOR endpoint

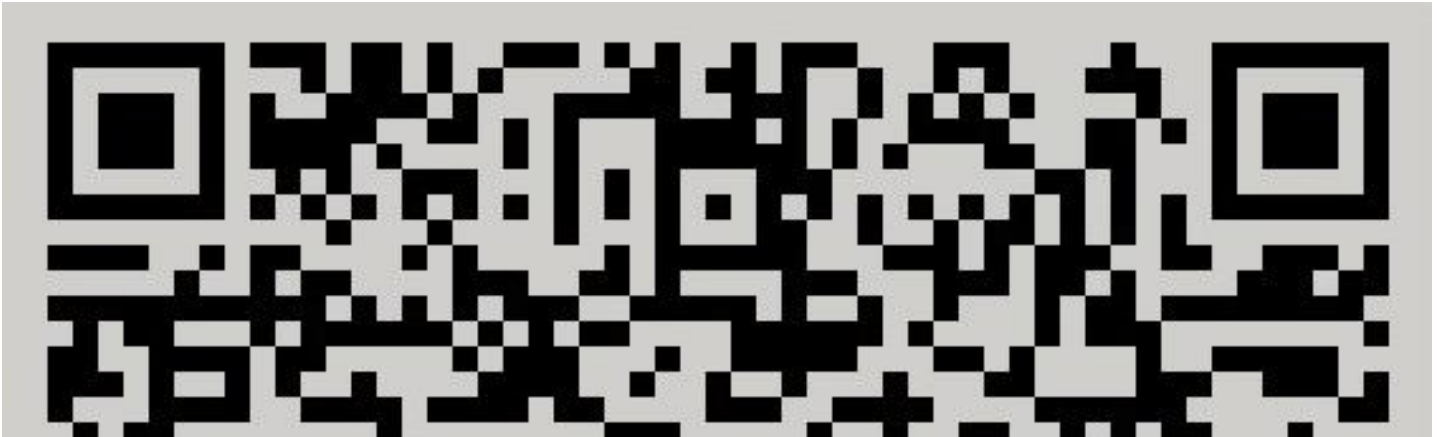
1) Add peer (Office A)

```
/int wireguard/peers/add interface=wg1 \
public-key="Qaq..." \
allowed-address=172.16.255.100/32
responder=yes
client-address=172.16.255.120/32 \
Client-endpoint=10.30.40.33
Client-dns=172.16.255.1 (optional)
```

```
/int wireguard/peer/show-client-config number=0

    [Interface]
    ListenPort = 51820
    PrivateKey = AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAEA=
    Address = 172.16.255.120/32

    [Peer]
    PublicKey = hEJPNtcWdcyDnmTtmVUR34Maym6AAcaSveFIIQ5GZgw=
    AllowedIPs = 0.0.0.0/0, ::/0
```



We can create the RoadWarrior private key from the Mikrotik in office A.

```
/int wireguard/peers/add comment="RoadWarrior" \
private-key="....." allowed-address=172.16.255.111/32
```

This way we do not need to first create the key pair on the client device and then copy the public key to the Mikrotik.

The only problem is that it becomes persistent and even if the information is deleted, it reappears. Therefore, the client's private key is always visible on the Mikrotik router in Office A and a malicious administrator could "clone" the client to impersonate the client's connection.

MTU

Ping from Office A to Office B:

```
/ping 172.16.255.2 size=1420 do-not-fragment count=1
Columns: SEQ, HOST, SIZE, TTL, TIME
SEQ  HOST            SIZE  TTL  TIME
  0  172.16.255.2    1420   64  4ms559us
```

14 + 1480

There are 20 bytes unused

No.	Time	Source	Destination	Protocol	Lengtl	Info
1	0.000000	10.30.60.33	10.50.40.22	WireGu…	1494	Transport Data, receiver=0xBFB27AD5, counter=1, datalen=1420
2	0.006133	10.50.40.22	10.30.60.33	WireGu…	1494	Transport Data, receiver=0xC03CF302, counter=0, datalen=1420

```
> Frame 1: 1494 bytes on wire (11952 bits), 1494 bytes captured (11952 bits)
> Ethernet II, Src: PCSSystemtec_69:24:38 (08:00:27:69:24:38), Dst: PCSSystemtec_eb:79:b2 (08:00:27:eb:79:b2)
> Internet Protocol Version 4, Src: 10.30.60.33, Dst: 10.50.40.22
v User Datagram Protocol, Src Port: 19323, Dst Port: 19323
    Source Port: 19323
    Destination Port: 19323
    Length: 1460
    Checksum: 0x7e4c [unverified]
    [Checksum Status: Unverified]
    [Stream index: 0]
    [Stream Packet Number: 1]
  > [Timestamps]
    UDP payload (1452 bytes)
v WireGuard Protocol
    Type: Transport Data (4)
    Reserved: 000000
    Receiver: 0xbfb27ad5
    Counter: 1
    Encrypted Packet
```

```
0000  08
0010  05
0020  28
0030  b2
0040  c7
0050  83
```

1452 – 32 = 1420

32 bytes:
- 16 wireguard header
- 16 wireguard trailer

14 bytes:
- 6 dst mac address
- 6 dst mac address
- 2 protocol (ip)

MTU

We only are using IPv4 so…

As we saw on slide 6, we can use an MTU of 1440 without exceeding 1500 bytes (underlying layer 3 MTU).

Let’s configure (office A & office B)!

```
/int wireguard/set mtu=1440 numbers=0
```

MTU

Ping from Office A to Office B:

```
/ping 172.16.255.2 size=1440 do-not-fragment count=1
Columns: SEQ, HOST, SIZE, TTL, TIME
SEQ  HOST            SIZE  TTL  TIME
  0  172.16.255.2    1440   64  4ms559us
```

14 + 1500 ✓

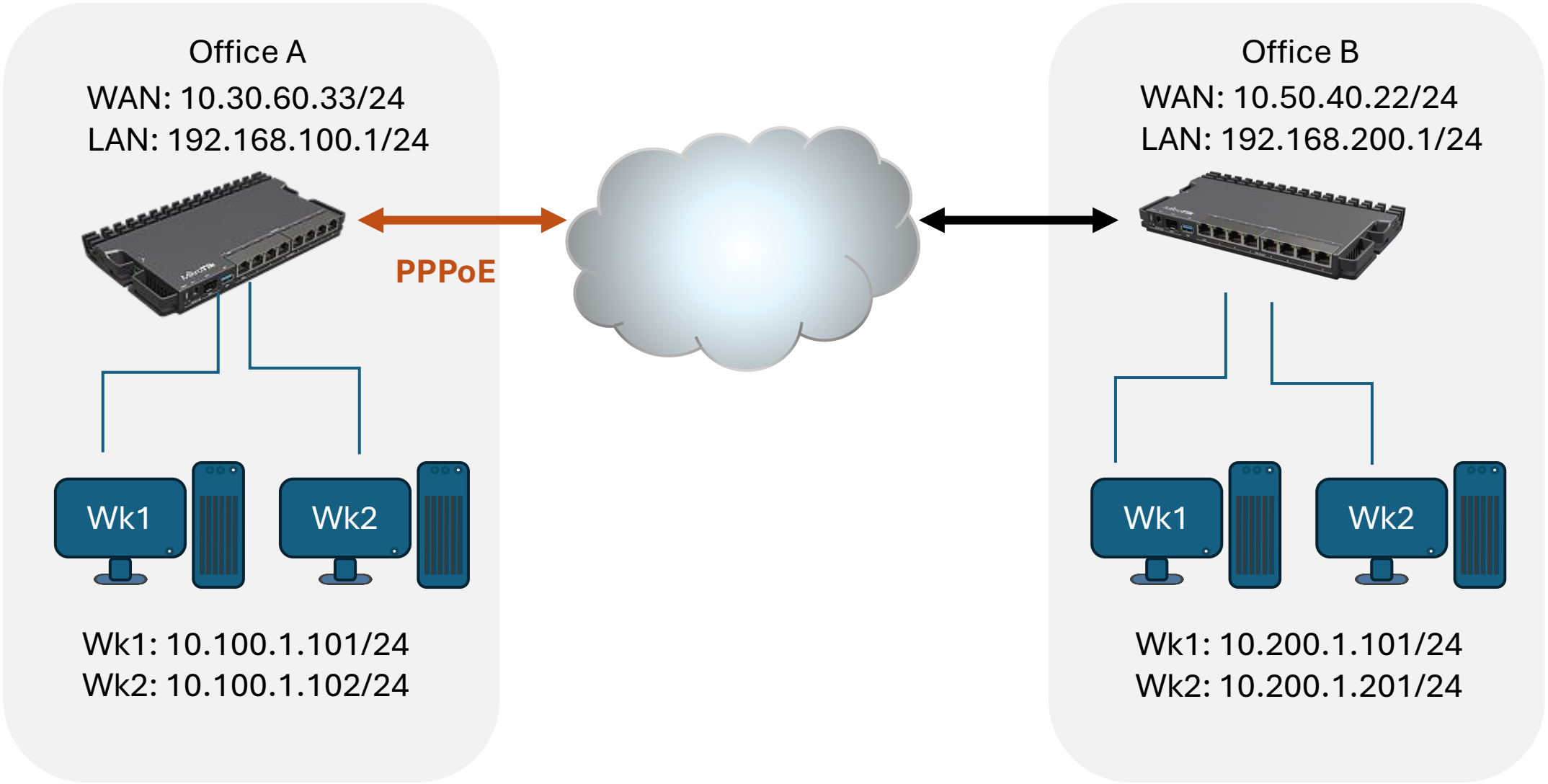
No.	Time	Source	Destination	Protocol	Lengtl	Info
1	0.000000	10.30.60.33	10.50.40.22	WireGu…	1514	Transport Data, receiver=0x5C953FE1, counter=3, datalen=1440
2	0.005056	10.50.40.22	10.30.60.33	WireGu…	1514	Transport Data, receiver=0x5B46D3AA, counter=1, datalen=1440

```
> Frame 1: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits)
> Ethernet II, Src: PCSSystemtec_69:24:38 (08:00:27:69:24:38), Dst: PCSSystemtec_eb:79:b2 (08:00:27:eb:79:b2)
> Internet Protocol Version 4, Src: 10.30.60.33, Dst: 10.50.40.22
v User Datagram Protocol, Src Port: 19323, Dst Port: 19323
    Source Port: 19323
    Destination Port: 19323
    Length: 1480
    Checksum: 0x7e60 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 0]
    [Stream Packet Number: 1]
  > [Timestamps]
    UDP payload (1472 bytes)
v WireGuard Protocol
    Type: Transport Data (4)
    Reserved: 000000
    Receiver: 0x5c953fe1
    Counter: 3
    Encrypted Packet
```

1472 – 32 = 1440

```
0000  08 0
0010  05 d
0020  28 1
0030  95 5
0040  d8 c
0050  11 1
0060  28 1
0070  ba f
0080  f8 a
0090  44 b
00a0  64 1
00b0  2f 1
00c0  6b 2
00d0  0f 8
00e0  b5 3
00f0  81 1
0100  af 6
0110  64 5
0120  17 b
0130  02 2
0140  48 5
0150  74 3
```

MTU


Office A
WAN: 10.30.60.33/24
LAN: 192.168.100.1/24
PPPoE
Wk1
Wk2
Wk1: 10.100.1.101/24
Wk2: 10.100.1.102/24
Office B
WAN: 10.50.40.22/24
LAN: 192.168.200.1/24
Wk1
Wk2
Wk1: 10.200.1.101/24
Wk2: 10.200.1.201/24

MTU


```
/int pppoe-client/add interface=ether1 user=myuser password=mypass \
name=pppoe-out1 add-default-route=yes disabled=no

 /int pr
 Flags: R - RUNNING
 Columns: NAME, TYPE, ACTUAL-MTU, L2MTU, MAC-ADDRESS
 #    NAME         TYPE        ACTUAL-MTU  L2MTU  MAC-ADDRESS
 0  R ether1       ether             1500         08:00:27:69:24:38
 1  R ether2       ether             1500         08:00:27:59:62:66
 2  R bridgeLAN    bridge            1500  65535  92:B0:7A:C9:E6:73
 3  R pppoe-out1   pppoe-out         1480
 4  R wg1          wg                1420
```

Maximum MTU for PPPoE connection is 1492 bytes.
Mikrotik by default uses a 1480 as maximum MTU.

MTU

If I use IPv4 as endpoint address, I can use 1440 for Wireguard MTU

```
/int wireguard/set mtu=1440
```

```
/int pr
Flags: R - RUNNING
Columns: NAME, TYPE, ACTUAL-MTU, L2MTU, MAC-ADDRESS
#    NAME         TYPE        ACTUAL-MTU  L2MTU  MAC-ADDRESS
0 R ether1       ether             1500         08:00:27:69:24:38
1 R ether2       ether             1500         08:00:27:59:62:66
2 R bridgeLAN    bridge            1500  65535  92:B0:7A:C9:E6:73
3 R pppoe-out1   pppoe-out         1480
4 R wg1          wg                1440
```

MTU

No.	Time	Source	Destination	Protocol	Lengtl	Info
5	2.738188	10.100.0.2	10.50.40.22	IPv4	1498	Fragmented IP protocol (proto=UDP 17, off=0, ID=40ec) [Reassembled in #6]
6	2.739198	10.100.0.2	10.50.40.22	WireGu…	66	Transport Data, receiver=0x627DFEE0, counter=4, datalen=1440
7	2.742562	10.100.0.2	10.50.40.22	WireGu…	198	Handshake Initiation, sender=0x4C23DB46
8	2.745950	10.50.40.22	10.100.0.2	IPv4	1498	Fragmented IP protocol (proto=UDP 17, off=0, ID=8eb7) [Reassembled in #9]
9	2.745962	10.50.40.22	10.100.0.2	WireGu…	66	Transport Data, receiver=0xDB20FF66, counter=1, datalen=1440

```
> Frame 6: 66 bytes on wire (528 bits), 66 bytes captured (528 bits)
> Ethernet II, Src: PCSSystemtec_69:24:38 (08:00:27:69:24:38), Dst: PCSSystemtec_eb:79:b2 (08:00:27:eb:79:b2)
v PPP-over-Ethernet Session
    0001 .... = Version: 1
    .... 0001 = Type: 1
    Code: Session Data (0x00)
    Session ID: 0x0002
    Payload Length: 46
v Point-to-Point Protocol
    Protocol: Internet Protocol version 4 (0x0021)
v Internet Protocol Version 4, Src: 10.100.0.2, Dst: 10.50.40.22
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 44
    Identification: 0x40ec (16620)
  > 000. .... = Flags: 0x0
    ...0 0000 1011 0110 = Fragment Offset: 1456
    Time to Live: 64
    Protocol: UDP (17)
    Header Checksum: 0xfc71 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 10.100.0.2
    Destination Address: 10.50.40.22
  v [2 IPv4 Fragments (1480 bytes): #5(1456), #6(24)]
      [Frame: 5, payload: 0-1455 (1456 bytes)]
      [Frame: 6, payload: 1456-1479 (24 bytes)]
      [Fragment count: 2]
      [Reassembled IPv4 length: 1480]
      [Reassembled IPv4 data […]: 4b7b4b7b05c839cf04000000e0fe7d6204000000000000016e326f0f739d84366f9942b77b4ca066df4703c4c4c19ae5a4c2a8e8798775d65850bb67d
    [Stream index: 0]
v User Datagram Protocol, Src Port: 19323, Dst Port: 19323
    Source Port: 19323
    Destination Port: 19323
    Length: 1480
    Checksum: 0x39cf [unverified]
    [Checksum Status: Unverified]
    [Stream index: 0]
    [Stream Packet Number: 1]
  > [Timestamps]
    UDP payload (1472 bytes)
> WireGuard Protocol
```

We have 8 bytes overflow

1440 + 32 + 8 + 20 = 1500

1500 + 2 + 6 = 1508

Ping from Office A to Office B:

```
/ping 172.16.255.2 size=1440 do-not-fragment count=1
Columns: SEQ, HOST, SIZE, TTL, TIME
SEQ  HOST           SIZE  TTL  TIME
  0  172.16.255.2  1440   64  4ms559us
```

MTU

Ping from Office B to Office A

```
/ping 172.16.255.1 size=1440 do-not-fragment count=1
Columns: SEQ, HOST, SIZE, TTL, TIME
SEQ  HOST           SIZE  TTL  TIME
  0  172.16.255.1  1440   64  4ms559us
```

Office B

5	1.438920	10.50.40.22	10.100.0.2	WireGu…	1514 Transport Data, receiver=0x174DBBED, counter=0, datalen=1440
6	1.445025	10.100.0.2	10.50.40.22	IPv4	1490 Fragmented IP protocol (proto=UDP 17, off=0, ID=e8be) [Reassembled in #7]
7	1.445183	10.100.0.2	10.50.40.22	WireGu…	60 Transport Data, receiver=0xC306A3FF, counter=1, datalen=1440

Office A

1	0.000000	10.50.40.22	10.100.0.2	IPv4	1498 Fragmented IP protocol (proto=UDP 17, off=0, ID=2930) [Reassembled in #2]
2	0.000012	10.50.40.22	10.100.0.2	WireGu…	66 Transport Data, receiver=0x174DBBED, counter=0, datalen=1440
3	0.000269	10.100.0.2	10.50.40.22	IPv4	1498 Fragmented IP protocol (proto=UDP 17, off=0, ID=e8be) [Reassembled in #4]
4	0.000860	10.100.0.2	10.50.40.22	WireGu…	66 Transport Data, receiver=0xC306A3FF, counter=1, datalen=1440

worse from Office B to Office A

MTU

I do even worst the math and put 1472 for wireguard MTU

1500 – 20 (ip header) – 8 (udp)

```
/int wireguard/set mtu=1472
```

```
/int pr
Flags: R - RUNNING
Columns: NAME, TYPE, ACTUAL-MTU, L2MTU, MAC-ADDRESS
#   NAME         TYPE        ACTUAL-MTU  L2MTU  MAC-ADDRESS
0 R ether1       ether             1500         08:00:27:69:24:38
1 R ether2       ether             1500         08:00:27:59:62:66
2 R bridgeLAN    bridge            1500  65535  92:B0:7A:C9:E6:73
3 R pppoe-out1   pppoe-out         1480
4 R wg1          wg                1472
```

Ping from Office A to Office B

```
/ping 172.16.255.2 size=1472 do-not-fragment count=1
Columns: SEQ, HOST, SIZE, TTL, TIME
SEQ  HOST           SIZE  TTL  TIME
  0  172.16.255.2  1472   64  2ms13us
```

Office A

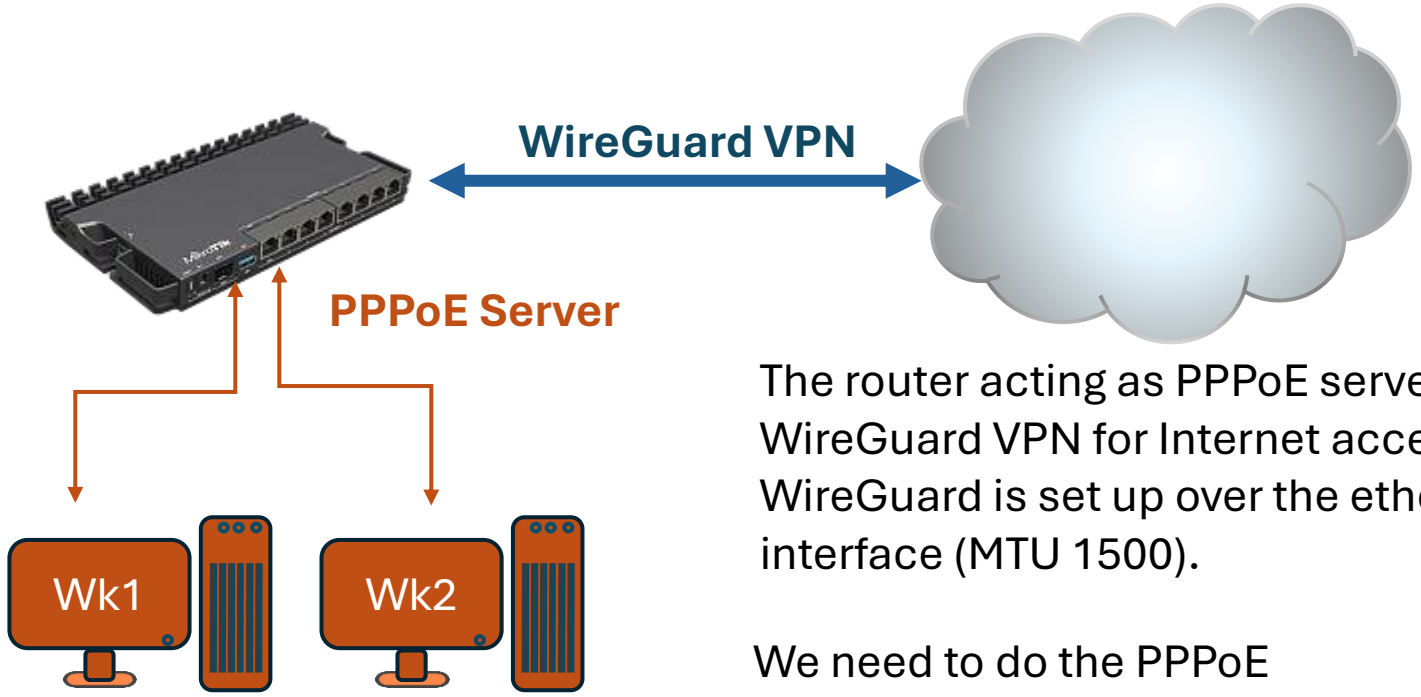
```
5 0.042129   10.100.0.2    10.50.40.22   IPv4     1498 Fragmented IP protocol (proto=UDP 17, off=0, ID=3f6d) [Reassembled in #6]
6 0.043275   10.100.0.2    10.50.40.22   WireGu…    98 Transport Data, receiver=0xFA65C372, counter=2, datalen=1472
7 0.057703   10.50.40.22   10.100.0.2    IPv4     1498 Fragmented IP protocol (proto=UDP 17, off=0, ID=b18b) [Reassembled in #9]
8 0.057717   10.50.40.22   10.100.0.2    IPv4       66 Fragmented IP protocol (proto=UDP 17, off=1456, ID=b18b) [Reassembled in #9]
9 0.064047   10.50.40.22   10.100.0.2    WireGu…    74 Transport Data, receiver=0x2FC6AB61, counter=0, datalen=1472
```

Office B

```
1 0.000000   10.100.0.2    10.50.40.22   IPv4     1490 Fragmented IP protocol (proto=UDP 17, off=0, ID=3f6d) [Reassembled in #2]
2 0.000027   10.100.0.2    10.50.40.22   WireGu…    90 Transport Data, receiver=0xFA65C372, counter=2, datalen=1472
3 0.000218   10.50.40.22   10.100.0.2    IPv4     1514 Fragmented IP protocol (proto=UDP 17, off=0, ID=b18b) [Reassembled in #4]
4 0.005352   10.50.40.22   10.100.0.2    WireGu…    66 Transport Data, receiver=0x2FC6AB61, counter=0, datalen=1472
```

Now even worse from Office B to Office A

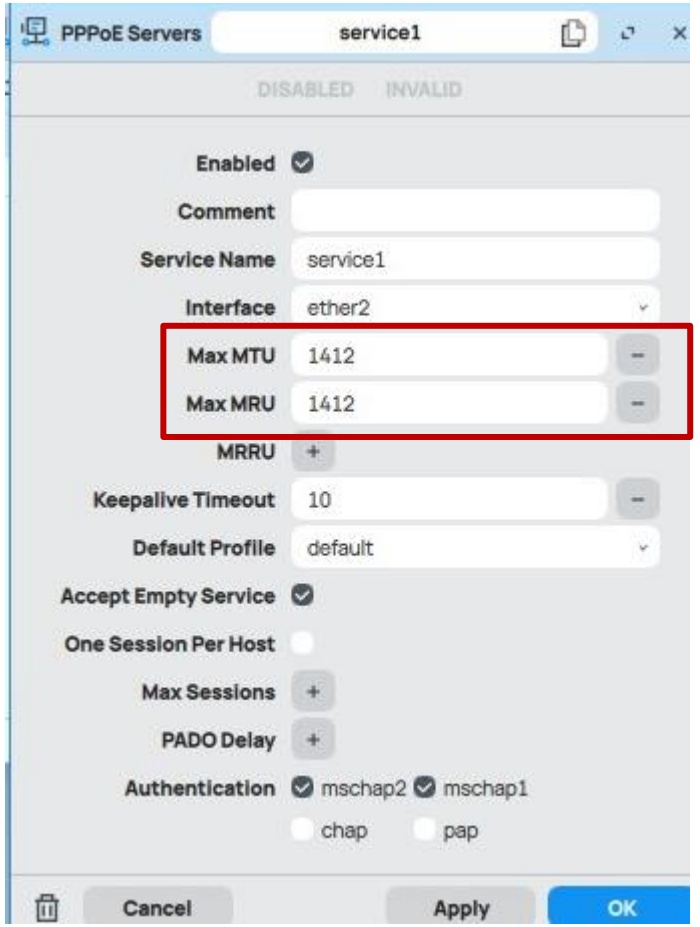
PPPoE for customers



The router acting as PPPoE server has a WireGuard VPN for Internet access. WireGuard is set up over the ethernet interface (MTU 1500).

We need to do the PPPoE server MTU calculations for the clients.

```
1420  WireGuard MTU
  -8  PPPoE header
1412
```



Be carefull about MTU

WireGuard VPN is established over a PPPoE Client.

- ✓WireGuard default MTU is 1420.
- ✓PPPoE Client on Mikrotik is 1480.

If we use IPv4 for endpoints the math is perfect

1420 + 32 + 28 = 1480 ✓

But if we use IPv6 for endpoints the maths fails

1420 + 32 + 48 = 1500 ✗ (exceeds PPPoE Client MTU on 20)

Subtract 20 from WireGuard MTU to fit. new WireGuard MTU **1400**

More information?

If you want more information about this presentation, please email me at

j.castellet@yatuaprendes.com

And i'll reply you as soon as possible.

Thank for your time!