RADIO DATA NETWORKS

Application notes

RipEX2/M!DGE3 OpenVPN

fw 2.2.1.0
2024-12-11
version 1.2

RACOM s.r.o. | Mirova 1283 | 592 31 Nove Mesto na Morave | Czech Republic
Tel.: +420 722 937 522 | E-mail: racom@racom.eu

www.racom.eu

Table of Contents

OpenVPN

OpenVPN is a virtual private network (VPN) system that allows to create secure encrypted point-to-multipoint connections in routed (TUN) or bridged (TAP) modes. Up to four instances (clients and/or servers) can be used simultaneously in one unit. Each server is capable of establishing connections with several tens of clients.

OpenVPN allows peers to authenticate to each other using pre-shared secret keys and certificates. An OpenVPN server is capable to release an authentication certificate for every client, using signatures and certificate authority (certificates can be generated / uploaded in the SETTINGS > Security > Credentials menu).

A time synchronization of individual units is required for proper OpenVPN function.

All the configuration parameters are explained within the manual. The application notes will describe you several use-cases and step-by-step configurations, including screenshots and basic explanations. Eventually, OpenVPN debugging is explained.

Note

All the M!DGE3 or RipEX2 units run 2.1.1.0 firmware or newer.

1. M!DGE3/RipEX2 OpenVPN examples

1.1. Routed (TUN) OpenVPN topology

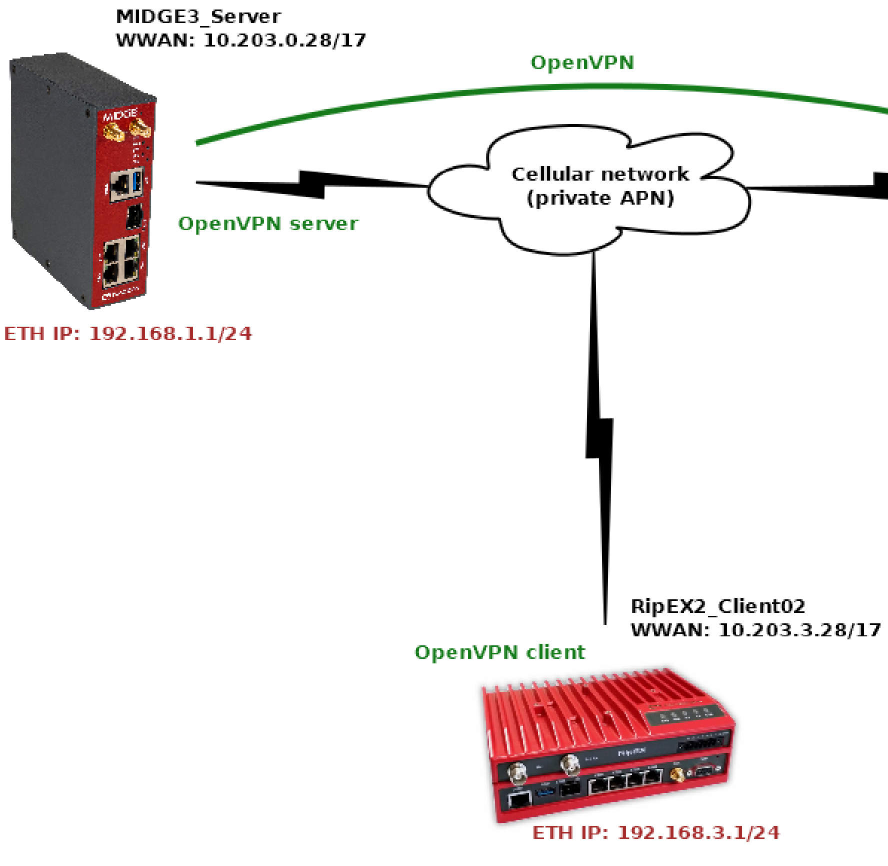


Fig. 1: Routed (TUN) OpenVPN topology

The topology depicts one OpenVPN server and two OpenVPN clients. M!DGE3 and RipEX2 share the same GUI and are configured exactly the same way. RipEX2 in the diagram is equipped with the LTE extension so it can be connected both to the cellular and radio networks.

The 1st example shows a configuration in which all the units utilize different LAN subnets, i.e., interconnecting them via one shared and secured OpenVPN network to build one secure end-to-end routed network.

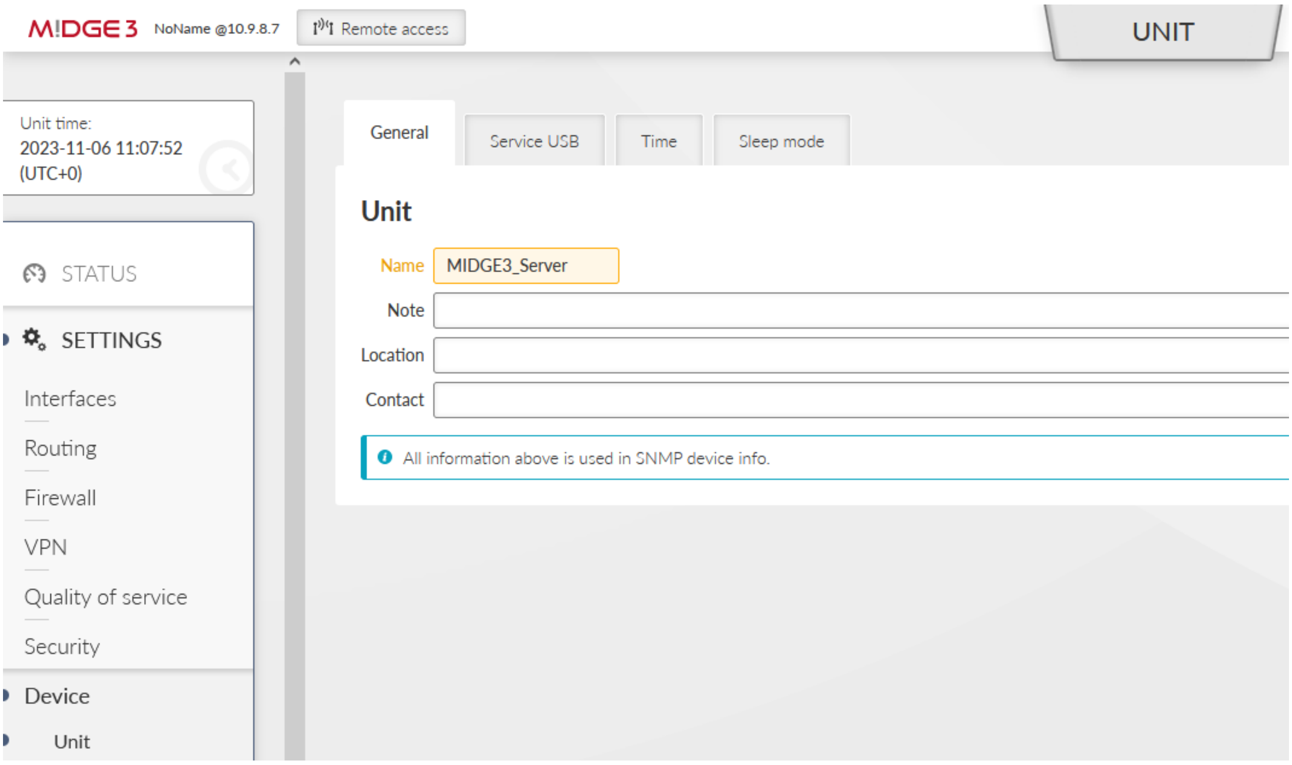
The cellular APN is a private APN with specified IP addresses within 10.203.0.0/17. Each device can “see” (ping) each other, but cannot access public Internet. While testing or configuring your scenario, you need to edit all the 10.203.0.0/17 IP addresses to suit your topology.

Note

Keep in mind the connections via the cellular network require some kind of VPN or at least NAPT so that packets can go LAN2LAN. Adding just static routing rules would end by discarding such traffic being discarded in operator’s network.

1.1.1. MIDGE3_Server

MIDGE3_Server will be set as the OpenVPN server. Set the Unit name “MIDGE3_Server” in the SETTINGS > Device > Unit menu.



Unit

Name MIDGE3_Server

Note

Location

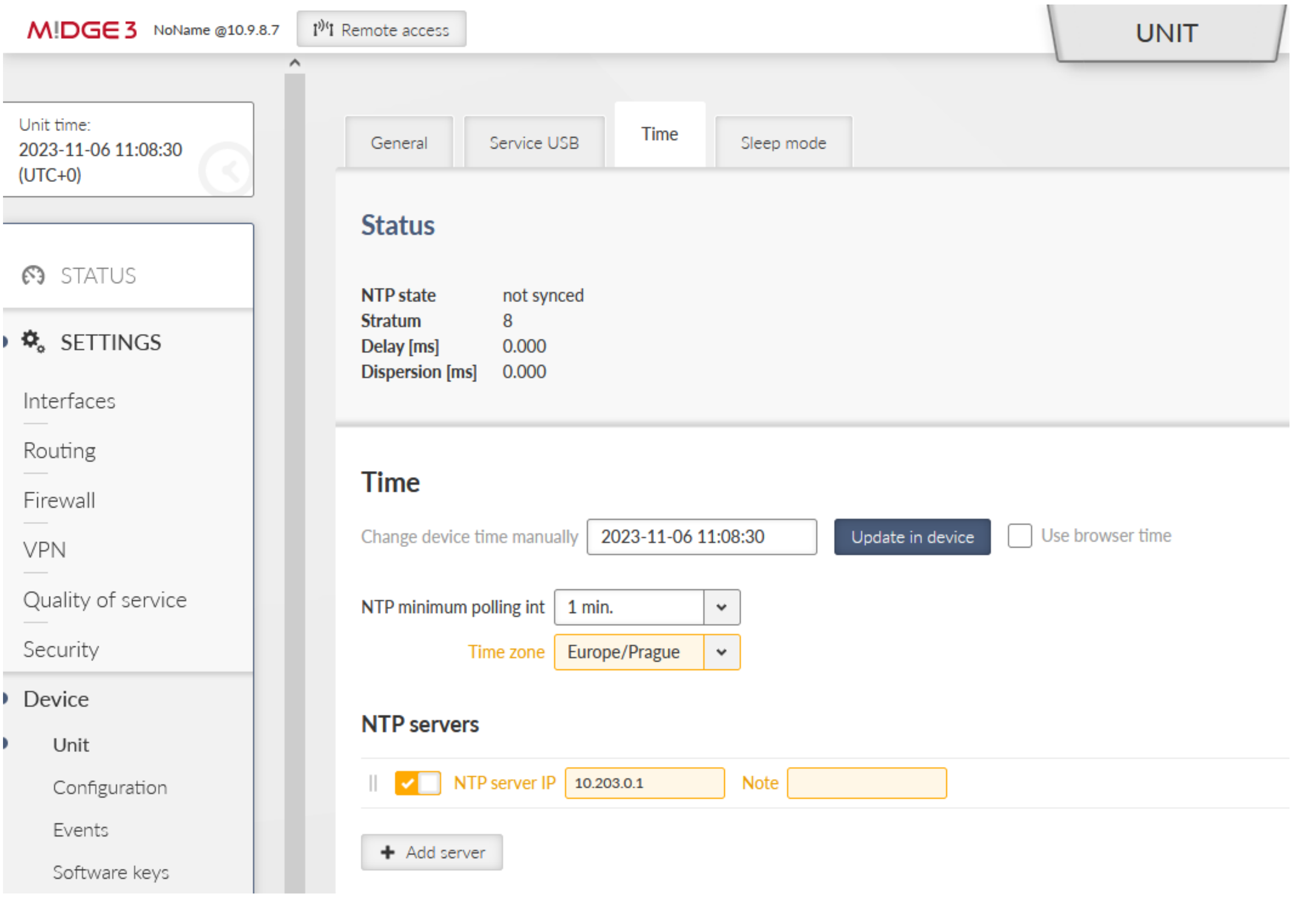
Contact

All information above is used in SNMP device info.

Fig. 2: MIDGE3_Server Unit name

Make sure to have correct and the same time in all units. OpenVPN works with certificates and their validation times so the correct time sync is required.

Set a correct Time zone (Europe/Prague) and NTP server (10.203.0.1) in the SETTINGS > Device > Unit > Time menu. Even if you do not have a working NTP server in your network, at least manually set a correct time, because each M!DGE3 and RipEX2 are equipped with the RTC and should be able to keep the proper time over the years even without the NTP server. NTP server is a recommended solution though – consider one unit within the network to be equipped with GPS – such unit would provide precise time to the rest of the network.



Unit time:
2023-11-06 11:08:30
(UTC+0)

Status

NTP state	not synced
Stratum	8
Delay [ms]	0.000
Dispersion [ms]	0.000

Time

Change device time manually: 2023-11-06 11:08:30

□ Use browser time

NTP minimum polling int: 1 min.

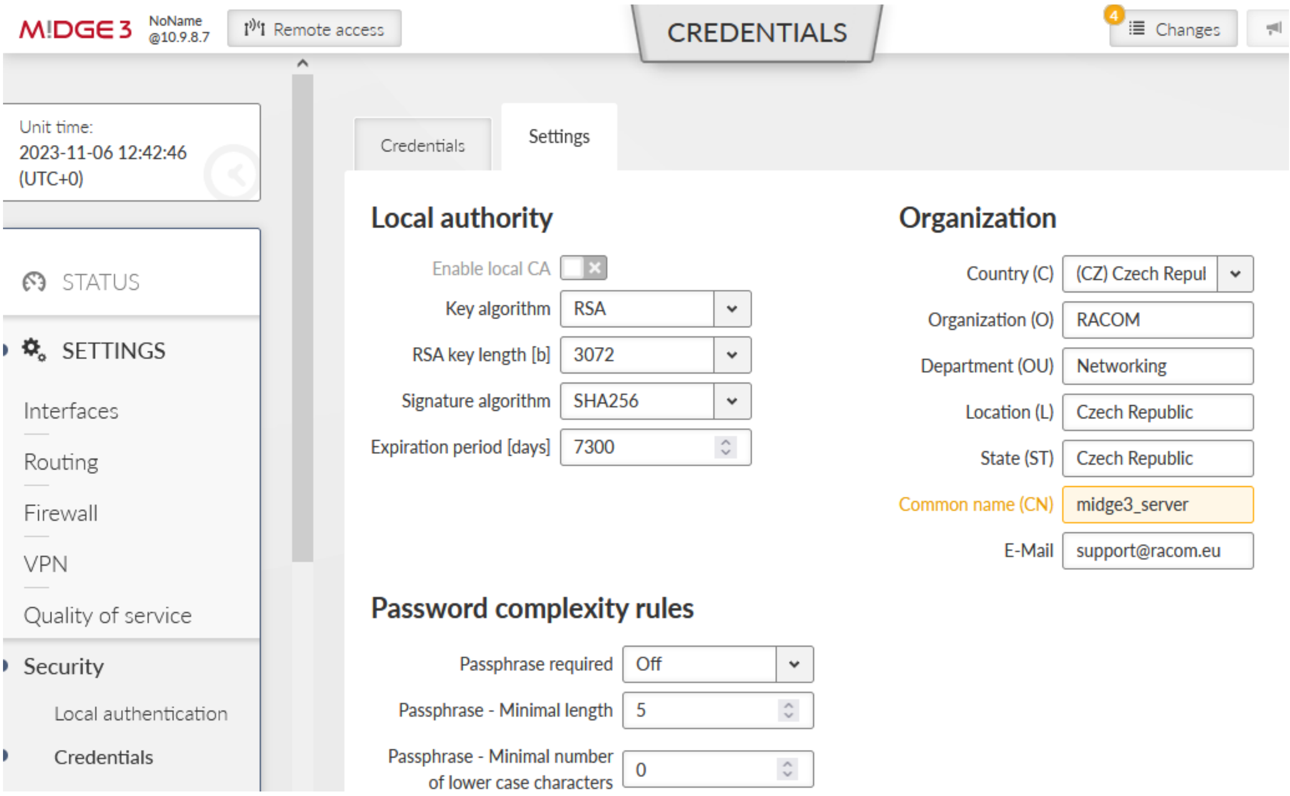
Time zone: Europe/Prague

NTP servers

☑ NTP server IP: 10.203.0.1 Note:

Fig. 3: MIDGE3_Server Time settings

Go to the SETTINGS > Security > Credentials > Settings menu and configure a Common Name for our MIDGE3_Server – “midge3_server”. CNs are very important for proper OpenVPN certificates.



Local authority

Enable local CA ☒

Key algorithm: RSA

RSA key length [b]: 3072

Signature algorithm: SHA256

Expiration period [days]: 7300

Organization

Country (C): (CZ) Czech Repul

Organization (O): RACOM

Department (OU): Networking

Location (L): Czech Republic

State (ST): Czech Republic

Common name (CN): midge3_server

E-Mail: support@racom.eu

Password complexity rules

Passphrase required: Off

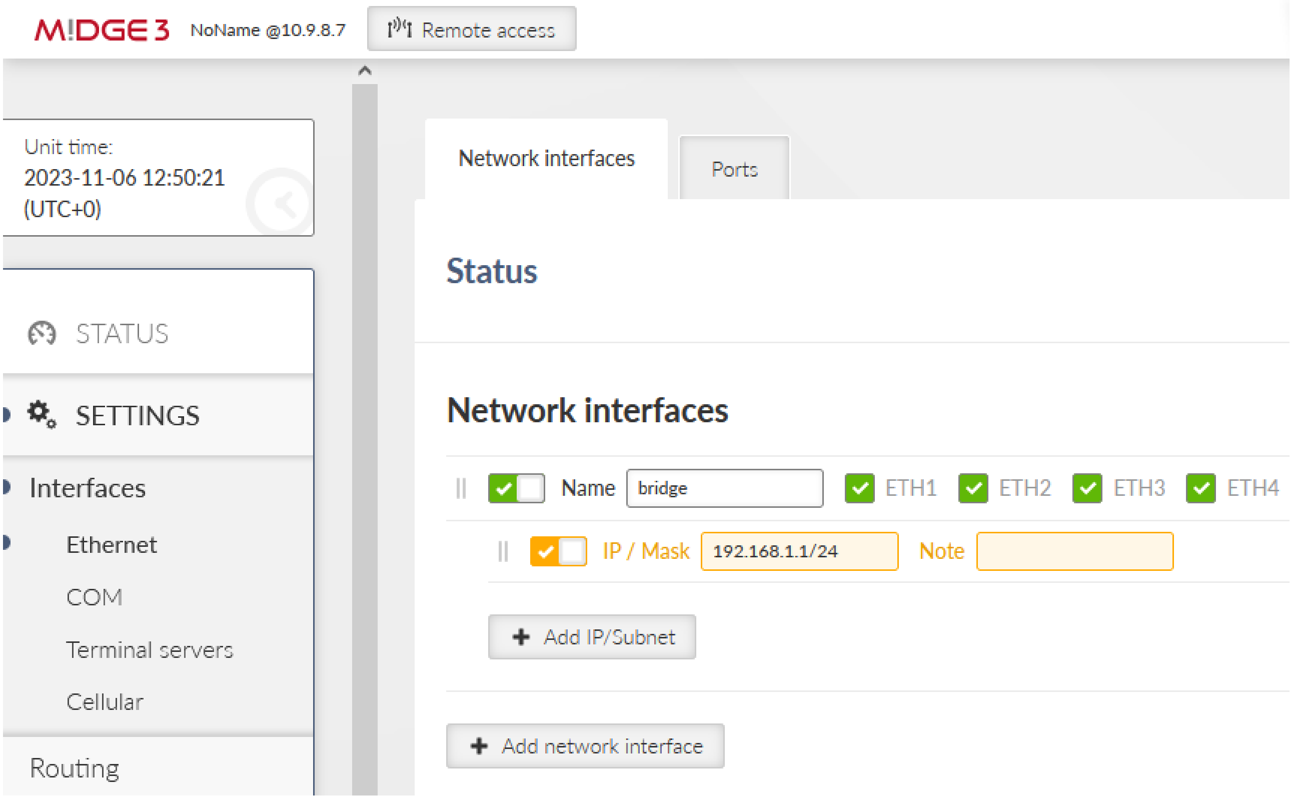
Passphrase - Minimal length: 5

Passphrase - Minimal number of lower case characters: 0

Fig. 4: MIDGE3_Server Common Name (CN)

All the units are in the Factory settings so we need to configure proper Ethernet and Cellular interfaces and also Routing.

Go to the SETTINGS > Interfaces > Ethernet > Network interfaces menu. Set the IP of the ‘bridge’ interface to be 192.168.1.1/24.



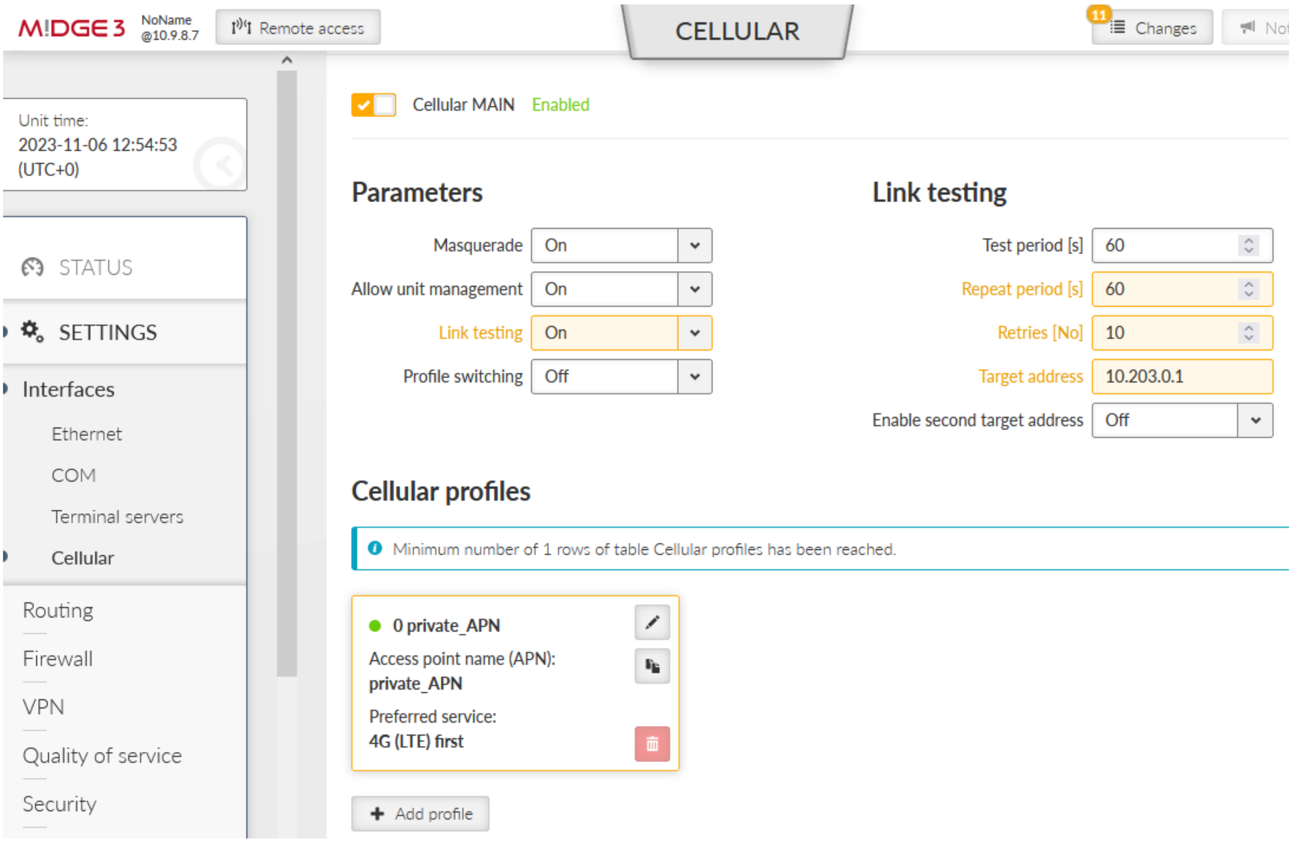
Unit time: 2023-11-06 12:50:21 (UTC+0)

Name: bridge ☑ ETH1 ☑ ETH2 ☑ ETH3 ☑ ETH4

IP / Mask: 192.168.1.1/24 Note:

Fig. 5: MIDGE3_Server Ethernet IP

Go to the SETTINGS > Interfaces > Cellular menu. Set the interface to suit your APN setup (APN name, credentials, MTU, …). We also suggest setting the “Link testing” option so that M!DGE3 periodically pings a defined IP address via the cellular interface. If the ping is not working correctly, the cellular connection is restarted (which may help in particular situations).



☑ Cellular MAIN Enabled

Parameters

Masquerade	On
Allow unit management	On
Link testing	On
Profile switching	Off

Link testing

Test period [s]	60
Repeat period [s]	60
Retries [No]	10
Target address	10.203.0.1
Enable second target address	Off

Cellular profiles

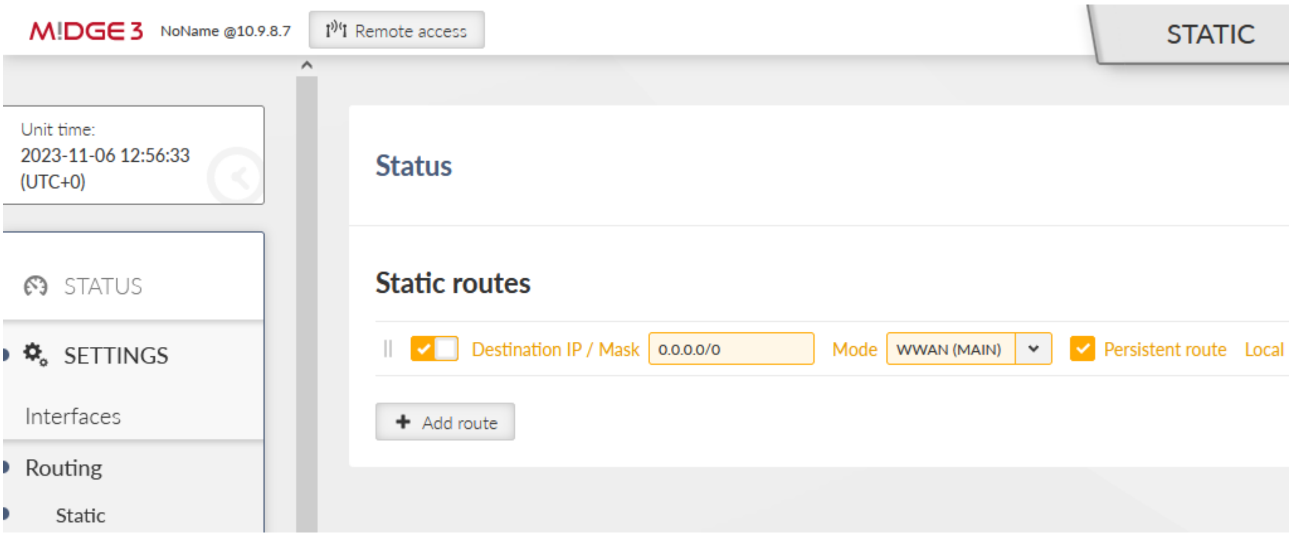
Minimum number of 1 rows of table Cellular profiles has been reached.

● 0 private_APN
Access point name (APN):
private_APN
Preferred service:
4G (LTE) first

+ Add profile

Fig. 6: MIDGE3_Server Cellular interface

Go to the SETTINGS > Routing > Static menu and add one static route. The Destination should either be the APN subnet, or a complete default gateway (0.0.0.0/0). Set the Mode to “WWAN (MAIN)”.



Status

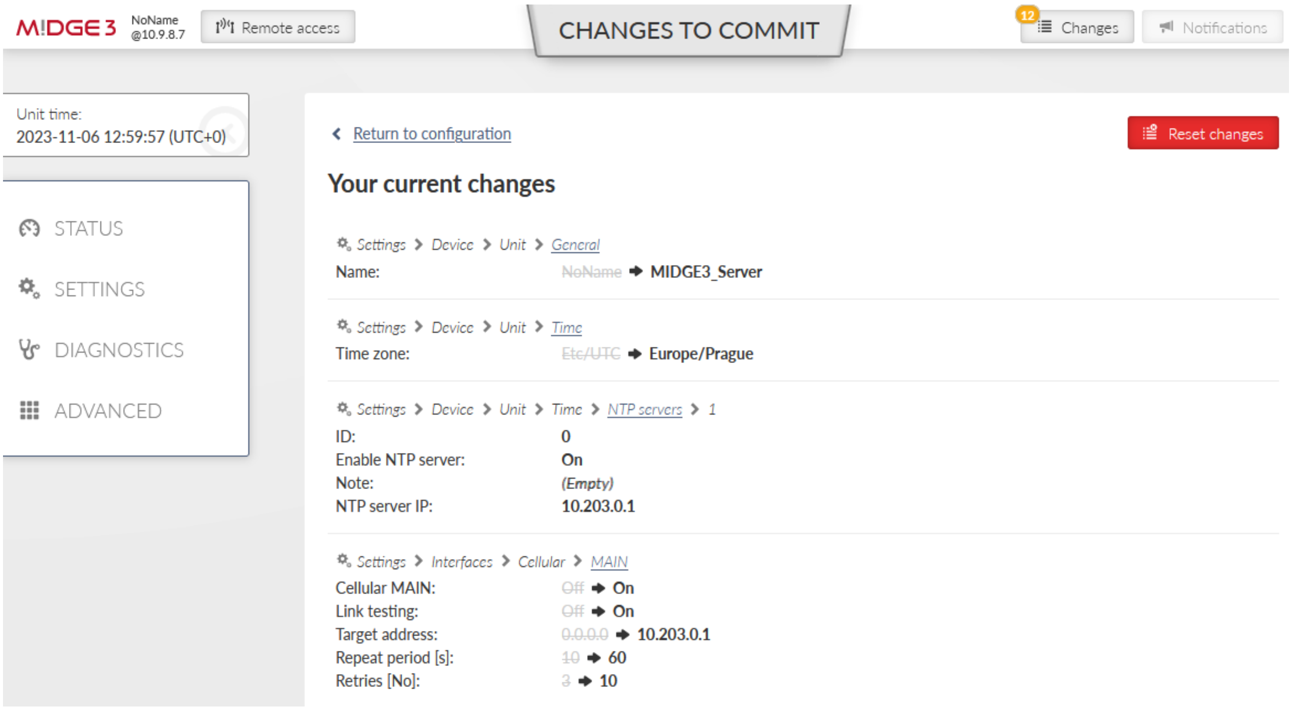
Static routes

☑ Destination IP / Mask 0.0.0.0/0 Mode WWAN (MAIN) ☑ Persistent route Local

+ Add route

Fig. 7: MIDGE3_Server Static routing

Save the changes now. OpenVPN configuration follows.



Unit time:
2023-11-06 12:59:57 (UTC+0)

Your current changes

Settings > Device > Unit > General

Name: ~~NoName~~ → MIDGE3_Server

Settings > Device > Unit > Time

Time zone: ~~Etc/UTC~~ → Europe/Prague

Settings > Device > Unit > Time > NTP servers > 1

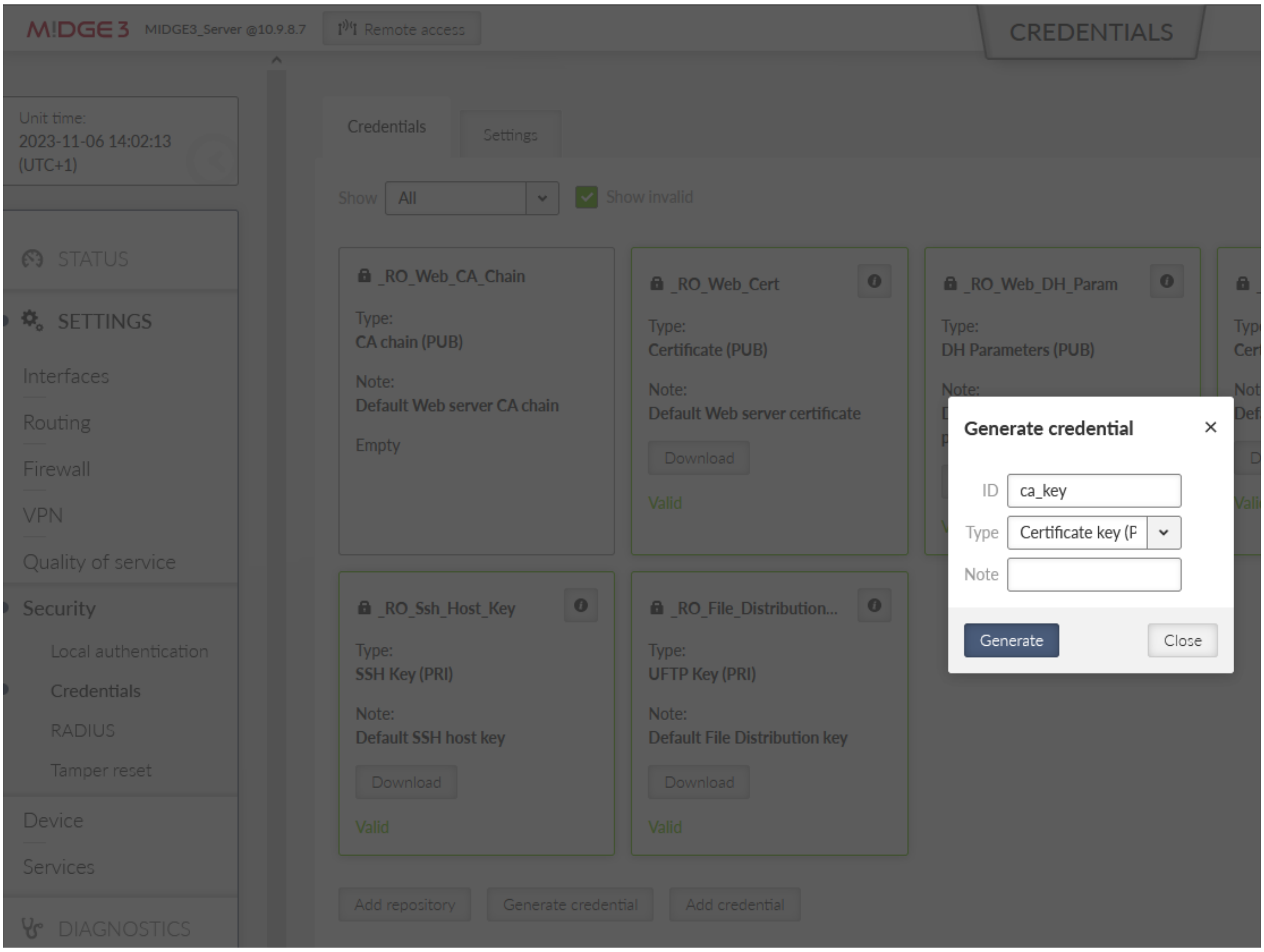
ID: 0
Enable NTP server: On
Note: *(Empty)*
NTP server IP: 10.203.0.1

Settings > Interfaces > Cellular > MAIN

Cellular MAIN: ~~Off~~ → On
Link testing: ~~Off~~ → On
Target address: ~~0.0.0.0~~ → 10.203.0.1
Repeat period [s]: ~~10~~ → 60
Retries [No]: ~~3~~ → 10

Fig. 8: MIDGE3_Server Changes to commit

Go back to the SETTINGS > Security > Credentials menu. Click on the “Generate credential” button. Select “Certificate key (PRI)” option to generate a private key. This private key is going to be a private key of our Local CA (Certification Authority). Let’s name it “ca_key”.



Unit time:
2023-11-06 14:02:13
(UTC+1)

Show: All

_RO_Web_CA_Chain
Type:
CA chain (PUB)
Note:
Default Web server CA chain
Empty

_RO_Web_Cert
Type:
Certificate (PUB)
Note:
Default Web server certificate
Valid

_RO_Web_DH_Param
Type:
DH Parameters (PUB)
Note:

_RO_Ssh_Host_Key
Type:
SSH Key (PRI)
Note:
Default SSH host key
Valid

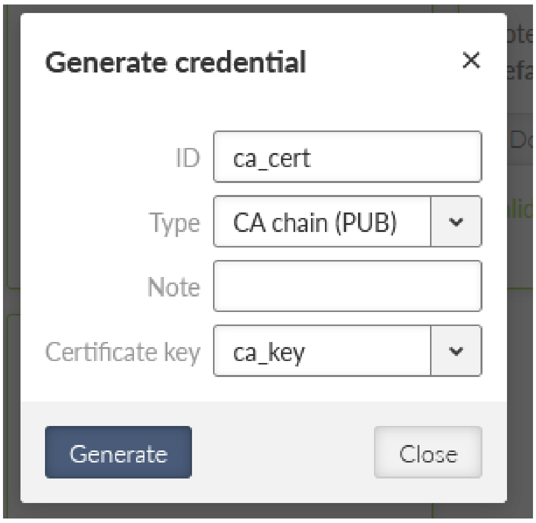
_RO_File_Distribution...
Type:
UFTP Key (PRI)
Note:
Default File Distribution key
Valid

Generate credential

ID: ca_key
Type: Certificate key (P
Note:

Fig. 9: MIDGE3_Server Generating the CA key

Once completed, create the CA's public certificate by clicking on the Generate credential button again. Set the ID to be "ca_cert" and the Type "CA chain (PUB)".



Generate credential

ID: ca_cert

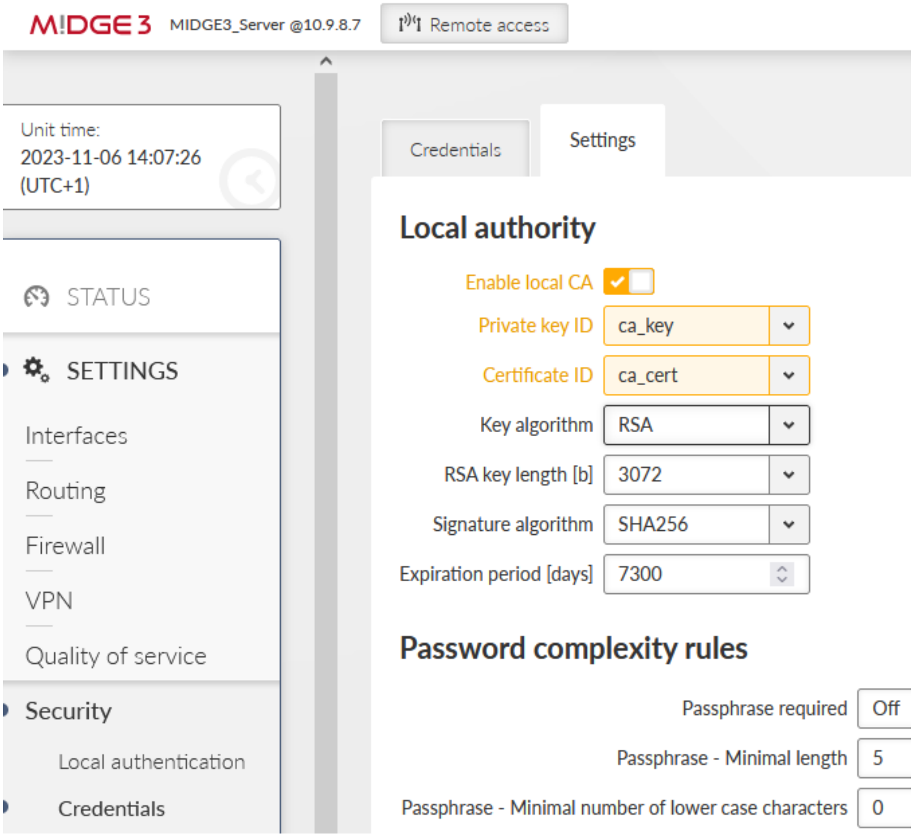
Type: CA chain (PUB)

Note:

Certificate key: ca_key

Fig. 10: MIDGE3_Server, CA certificate

Go to the "Settings" tab within the same menu and enable this MIDGE3_Server to be a local CA and specify created key & certificate.



Local authority

Enable local CA: ☑

Private key ID: ca_key

Certificate ID: ca_cert

Key algorithm: RSA

RSA key length [b]: 3072

Signature algorithm: SHA256

Expiration period [days]: 7300

Password complexity rules

Passphrase required: Off

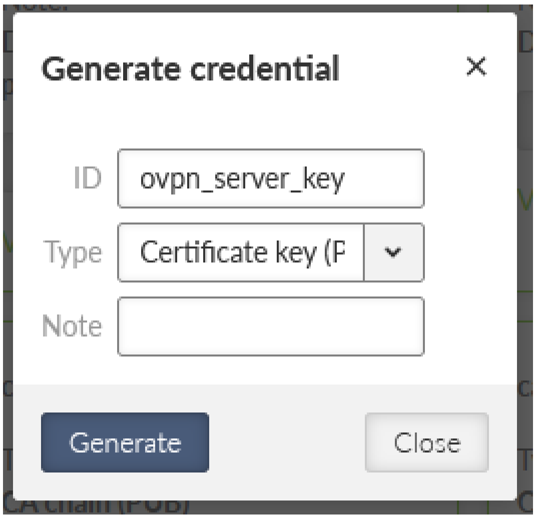
Passphrase - Minimal length: 5

Passphrase - Minimal number of lower case characters: 0

Fig. 11: MIDGE3_Server Enabling local CA

Save the changes.

From now on, we can use MIDGE3_Server as a trusted source for generating and signing keys for OpenVPN server and clients. Go back to the “Credentials” tab and generate another private key, now for the MIDGE3_Master again, but for the OpenVPN server itself (not the CA).



Generate credential

ID: ovpn_server_key

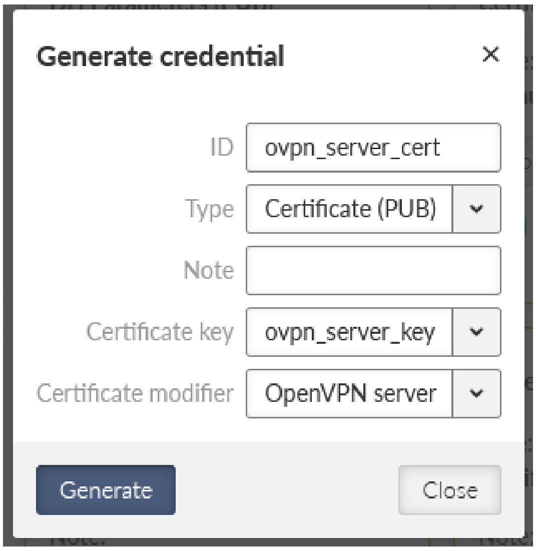
Type: Certificate key (P

Note:

Fig. 12: MIDGE3_Server OpenVPN server private key

Generate a new OpenVPN server certificate. Set

- ID to “ovpn_server_cert”
- Type to “Certificate (PUB)”
- Certificate key to our generated key “ovpn_server_key”
- and Certificate modifier to “OpenVPN server” – this is not mandatory, but it improves the VPN tunnel security – primarily it protects any client to mimic it is a server (strict roles for each certificate usage – server, or client).



Generate credential

ID: ovpn_server_cert

Type: Certificate (PUB)

Note:

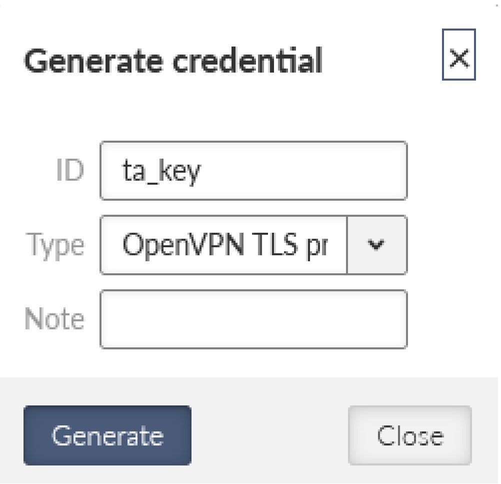
Certificate key: ovpn_server_key

Certificate modifier: OpenVPN server

Fig. 13: MIDGE3_Server, OpenVPN server public certificate

These two files with the “ca_cert” are enough for the OpenVPN server. We can still improve a security by additional TLS Protection – either “TLS auth” or “TLS Crypt”. TLS Auth helps against port scanning, UDP floods or DoS. TLS Crypt does the same, and adds some additional protection against sniffing certificate attributes or OpenVPN presence.

We will configure the TLS-Auth option. Generate a new shared key with ID equal to “ta_key”. Select a type to be “OpenVPN TLS Protection key (PRI)”.



Generate credential

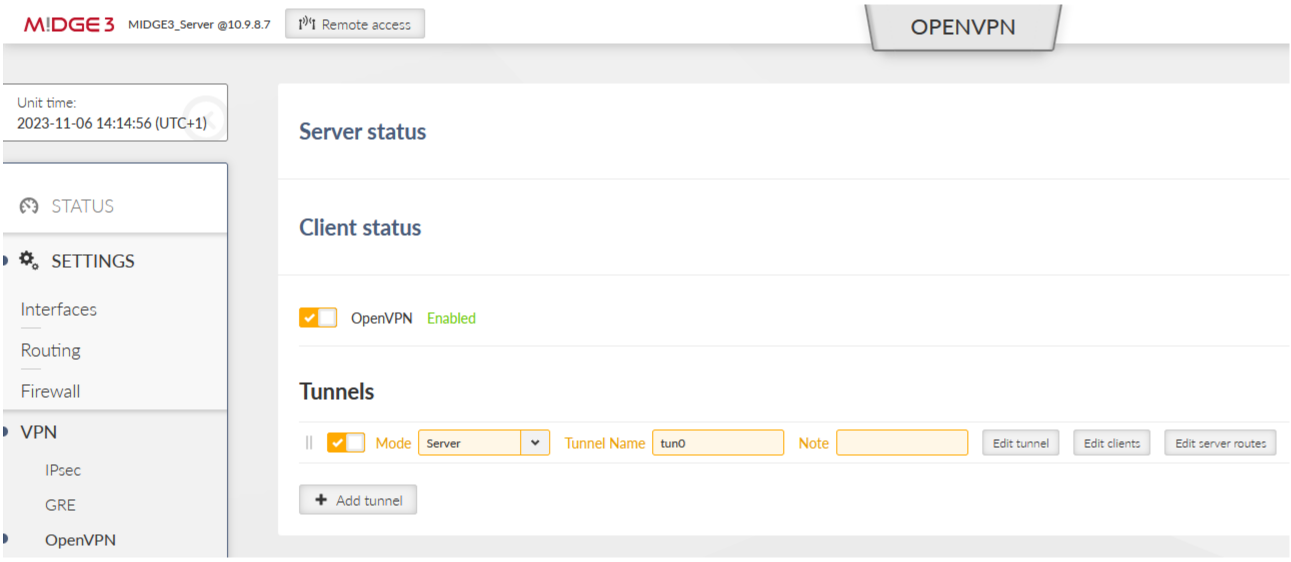
ID: ta_key

Type: OpenVPN TLS pr

Note:

Fig. 14: MIDGE3_Server, OpenVPN TLS Protection key ta_key (TLS-Auth)

All the files for the OpenVPN server are ready. Go to the SETTINGS > VPN > OpenVPN menu. Enable OpenVPN service and click on the “Add tunnel” button. Select a mode to be a “server”.



Unit time: 2023-11-06 14:14:56 (UTC+1)

☑ OpenVPN Enabled

☑ Mode: Server | Tunnel Name: tun0 | Note:

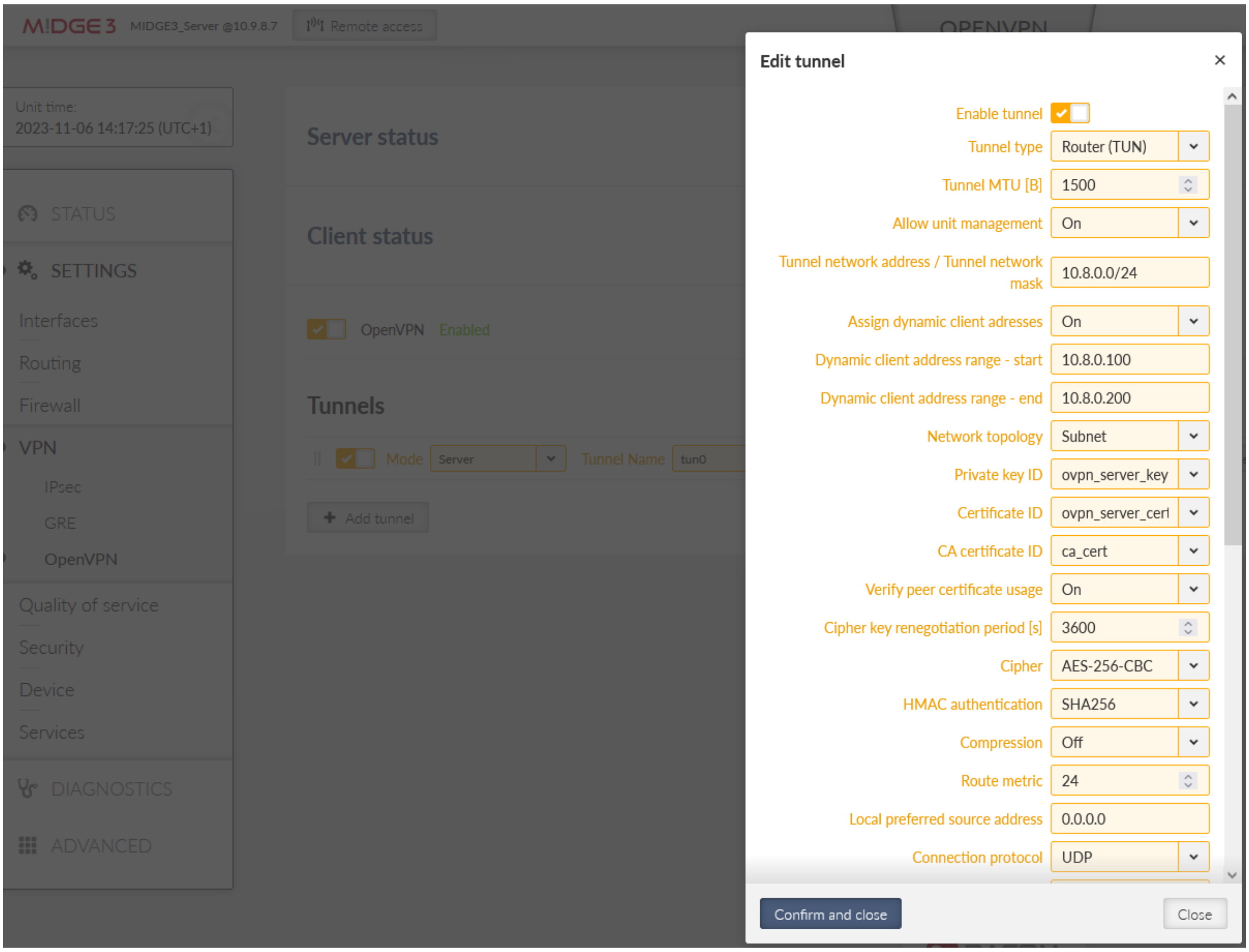
Fig. 15: MIDGE3_Server, OpenVPN server settings

Open the “Edit tunnel” menu. Select the required tunnel parameters. We leave all in defaults, except:

Assign dynamic client addresses	“On”
Private key ID	“ovpn_server_key”
Certificate ID	“ovpn_server_cert”
CA certificate ID	“ca_cert”
Cipher	“AES-256-CBC”

HMAC authentication	“SHA256”
DH parameters ID	“_RO_Web_DH_Param”
Enhanced TLS protection	“On”
TLS protection shared key ID	“ta_key”

Confirm the settings.



Edit tunnel ×

Enable tunnel	☑
Tunnel type	Router (TUN)
Tunnel MTU [B]	1500
Allow unit management	On
Tunnel network address / Tunnel network mask	10.8.0.0/24
Assign dynamic client adresses	On
Dynamic client address range - start	10.8.0.100
Dynamic client address range - end	10.8.0.200
Network topology	Subnet
Private key ID	ovpn_server_key
Certificate ID	ovpn_server_cert
CA certificate ID	ca_cert
Verify peer certificate usage	On
Cipher key renegotiation period [s]	3600
Cipher	AES-256-CBC
HMAC authentication	SHA256
Compression	Off
Route metric	24
Local preferred source address	0.0.0.0
Connection protocol	UDP

Fig. 16: MIDGE3_Server, OpenVPN server settings

Click on the “Edit clients” button. Add two clients, set the 1st client:

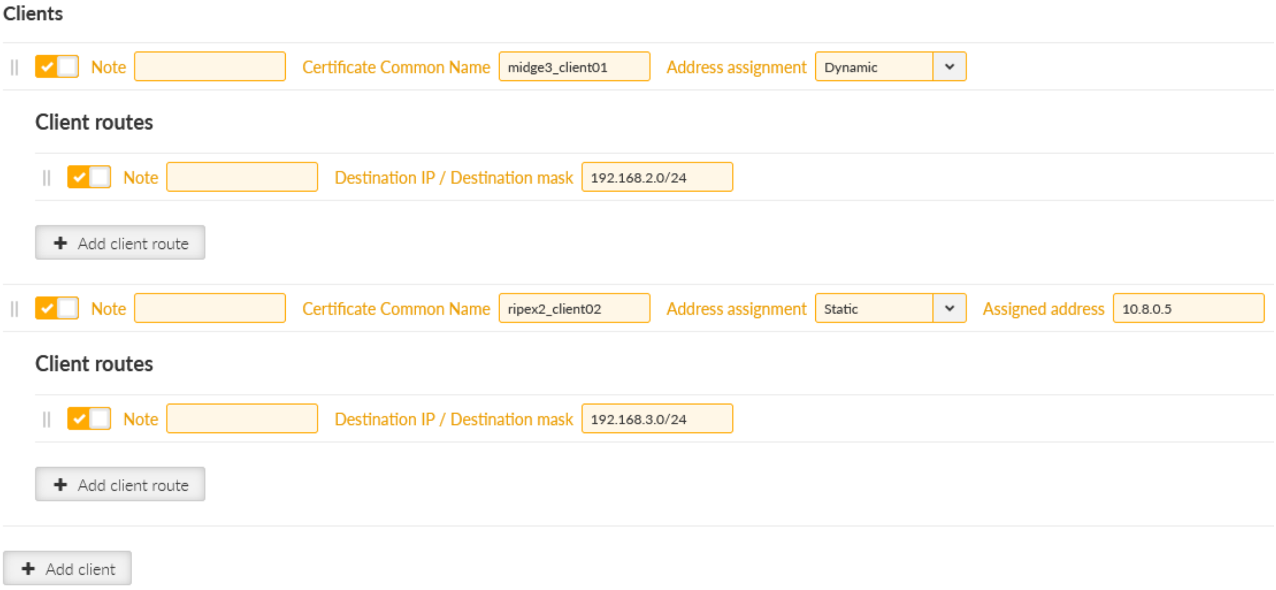
Certificate Common Name	midge3_client01
Address assignment	Dynamic (within 10.8.0.100 – 200 address range)
Client routes	192.168.2.0/24

And 2nd client:

Certificate Common Name	ripex2_client02
Address assignment	10.8.0.5

Client routes	192.168.3.0/24

The configured routes are particular client's LANs which are then propagated to the Server and to each Client for direct client-to-client communication.



Clients

☑ Note: — Certificate Common Name: midge3_client01 — Address assignment: Dynamic

Client routes

☑ Note: — Destination IP / Destination mask: 192.168.2.0/24

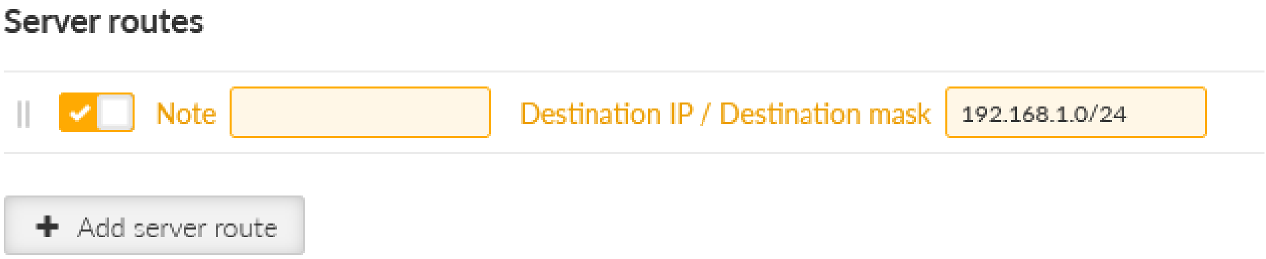
☑ Note: — Certificate Common Name: ripex2_client02 — Address assignment: Static — Assigned address: 10.8.0.5

Client routes

☑ Note: — Destination IP / Destination mask: 192.168.3.0/24

Fig. 17: MIDGE3_Server List of clients

Close the window and click on the "Edit server routes" button. Configure a network 192.168.1.0/24 to be pushed to connected clients. This range is the Server's LAN segment.



Server routes

☑ Note: — Destination IP / Destination mask: 192.168.1.0/24

Fig. 18: MIDGE3_Server, OpenVPN server route

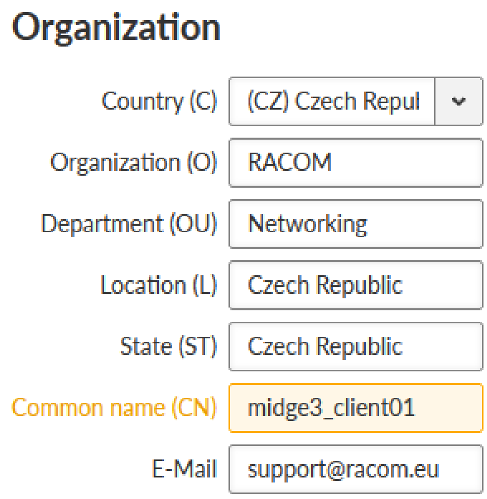
Apply all the changes.

Once completed, the OpenVPN server is ready, but we still need to configure both clients and generate/upload keys&certificates.

First of all, we need correct credentials (keys and certificates) for both the clients. We generate them via two different ways.

- For the 1st client (midge3_client01), we will generate all in the server. Then, we will download them to our PC and upload them to the MIDGE3_Client01's credentials.
- For the 2nd client (ripex2_client02), we will generate them in the client itself, and we use our MIDGE3_Server CA to sign the generated CSR (Certificate Signing Request) - so it creates a valid certificate without a ripex_client02's private key being shared/exposed. This procedure is more secure.

For now, stay in the MIDGE3_Server menu and go to the SETTINGS > Security > Credentials > Settings menu. Change the Organization Common Name (CN) to “midge3_client01”. Apply the changes.

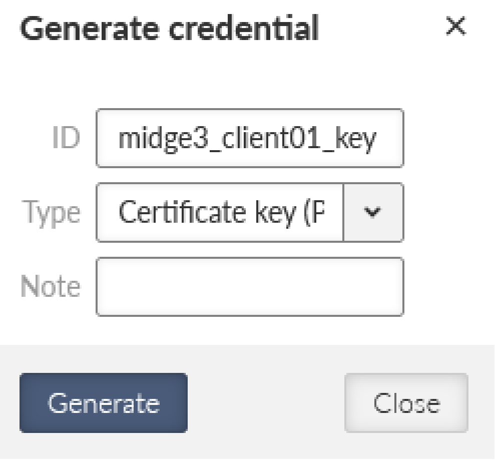


Organization

Country (C)	(CZ) Czech Repul
Organization (O)	RACOM
Department (OU)	Networking
Location (L)	Czech Republic
State (ST)	Czech Republic
Common name (CN)	midge3_client01
E-Mail	support@racom.eu

Fig. 19: MIDGE3_Server Common Name (CN) for the 1st client

Now, we can generate correct files for MIDGE3_Client01. Go back to the Credentials menu and generate a private key for MIDGE3_Client01’s OpenVPN. The ID is “midge3_client01_key”.

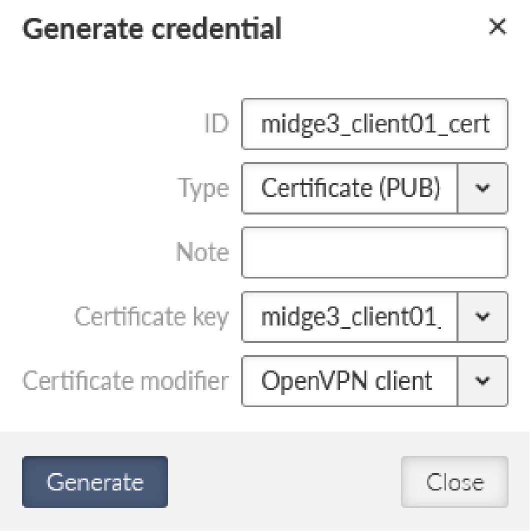


Generate credential

ID	midge3_client01_key
Type	Certificate key (F
Note	

Fig. 20: MIDGE3_Server OpenVPN Private key for MIDGE3_Client01

Generate a valid MIDGE3_Client01’s certificate for OpenVPN using the newly generated key. The ID is “midge3_client01_cert”, Type is “Certificate (PUB)”, Certificate key must be our “midge3_client01_key” and because we check the Extended Key Usage of the certificate, select the “OpenVPN client” modifier.



Generate credential

ID	midge3_client01_cert
Type	Certificate (PUB)
Note	
Certificate key	midge3_client01_
Certificate modifier	OpenVPN client

Fig. 21: MIDGE3_Server Generating MIDGE3_Client01's public OpenVPN certificate (client)

Download required files for the 1st client (MIDGE3_Client01):

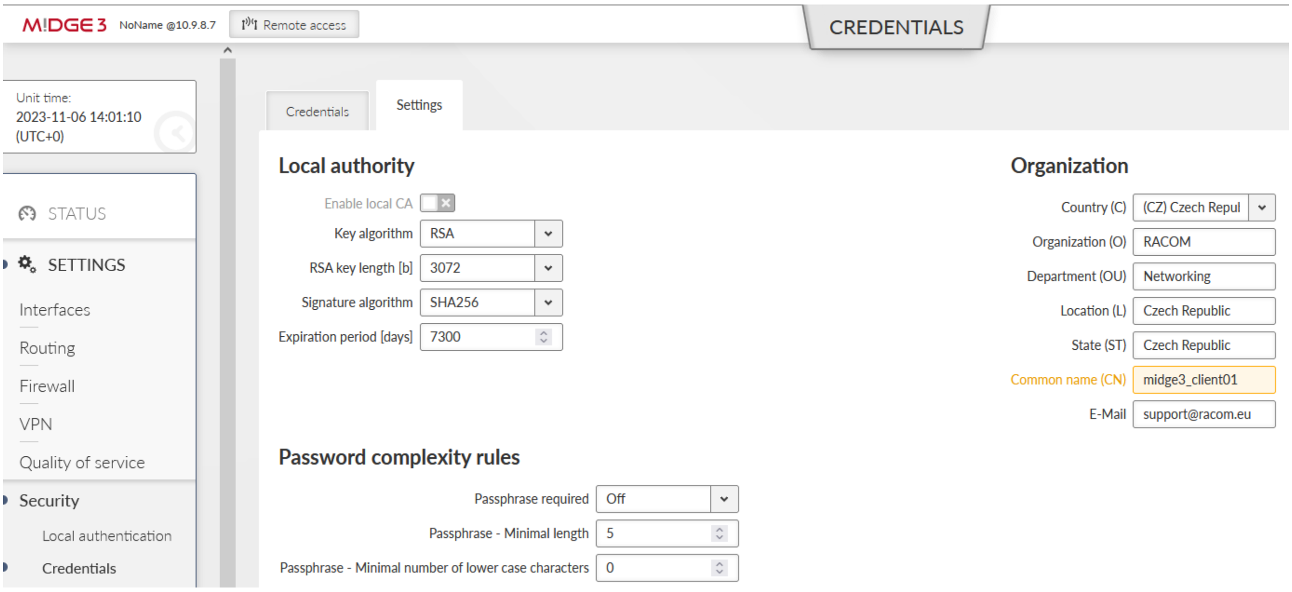
ca_cert	CA certificate
midge3_client01_cert	client's certificate
midge3_client01_key	client's private key
ta_key	TLS Protection key (TLS-Auth)

You can either download them encrypted (using a strong password) or unencrypted. Select proper names if not fully satisfied with automatic file names.

Change the MIDGE3_Server Common name back to “midge3_server” and save the changes.

1.1.2. MIDGE3_Client01

Login to the MIDGE3_Client01 unit and go to the SETTINGS > Security > Credentials > Settings menu. You can change the Common name to “midge3_client01”, even though it is not necessary.



Local authority

Enable local CA ☒
Key algorithm: RSA
RSA key length [b]: 3072
Signature algorithm: SHA256
Expiration period [days]: 7300

Organization

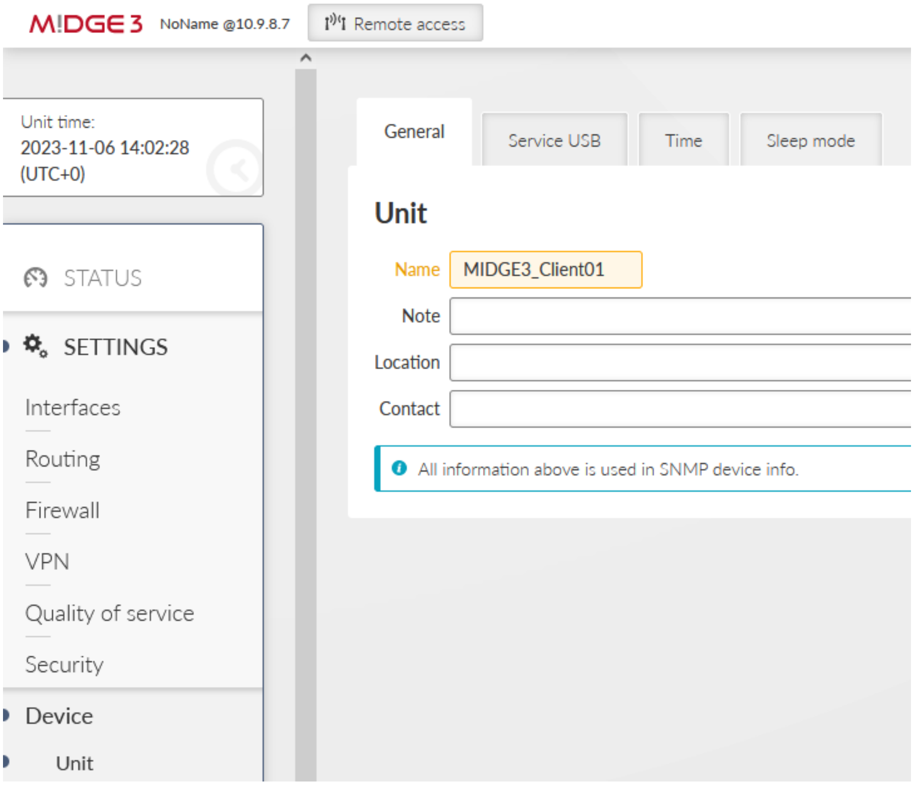
Country (C): (CZ) Czech Repul
Organization (O): RACOM
Department (OU): Networking
Location (L): Czech Republic
State (ST): Czech Republic
Common name (CN): midge3_client01
E-Mail: support@racom.eu

Password complexity rules

Passphrase required: Off
Passphrase - Minimal length: 5
Passphrase - Minimal number of lower case characters: 0

Fig. 22: MIDGE3_Client01 Common name

Set the unit name to MIDGE3_Client01.



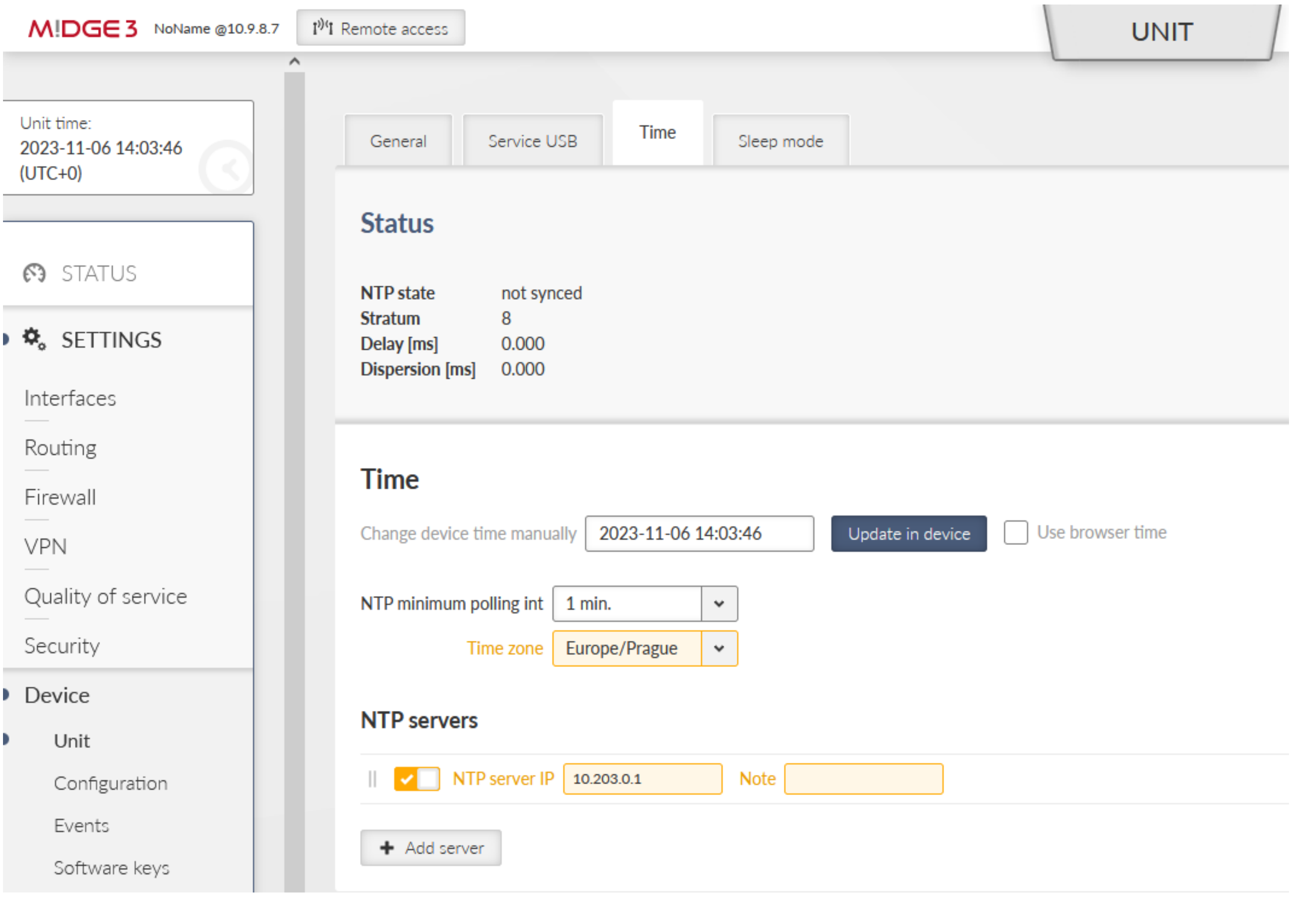
Unit

Name: MIDGE3_Client01
Note:
Location:
Contact:

All information above is used in SNMP device info.

Fig. 23: MIDGE3_Client01 Unit name

Configure the Time zone and NTP server to suit your environment.



Unit time:
2023-11-06 14:03:46
(UTC+0)

Status

NTP state not synced
Stratum 8
Delay [ms] 0.000
Dispersion [ms] 0.000

Time

Change device time manually 2023-11-06 14:03:46

□ Use browser time

NTP minimum polling int 1 min.

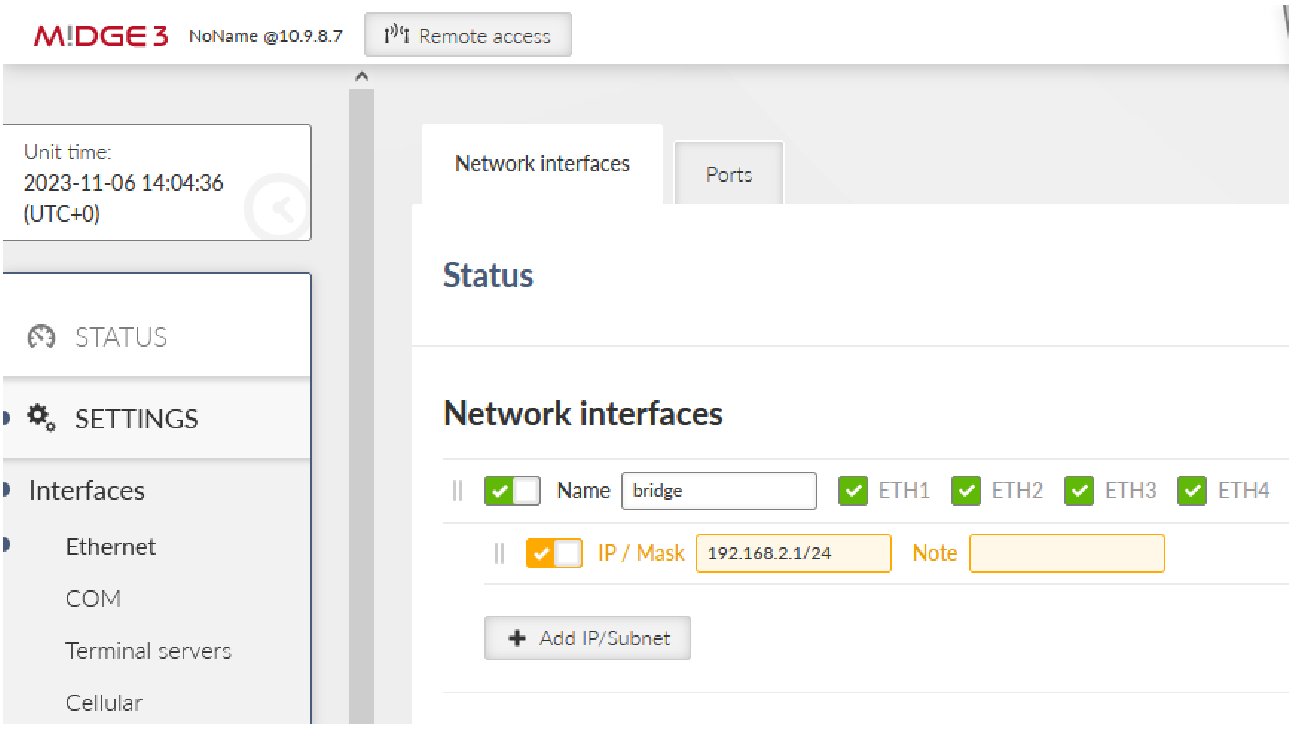
Time zone Europe/Prague

NTP servers

☑ NTP server IP 10.203.0.1 Note

Fig. 24: MIDGE3_Client01 Time settings

Go to the SETTINGS > Interfaces > Ethernet and set the LAN IP to 192.168.2.1/24.



Unit time:
2023-11-06 14:04:36
(UTC+0)

Status

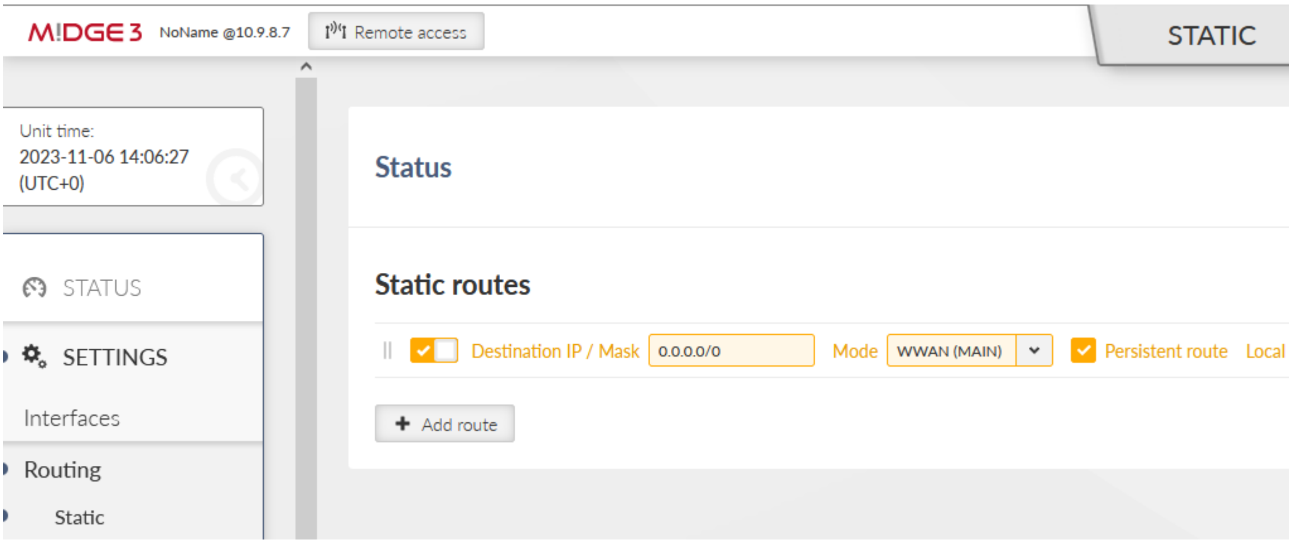
Network interfaces

☑ Name bridge ☑ ETH1 ☑ ETH2 ☑ ETH3 ☑ ETH4

☑ IP / Mask 192.168.2.1/24 Note

Fig. 25: MIDGE3_Client01 Ethernet IP address

Enable and configure the SETTINGS > Interfaces > Cellular interface to suit your APN. Continue with adding the default GW via WWAN (MAIN).



Destination IP / Mask 0.0.0.0/0 Mode WWAN (MAIN) ☑ Persistent route Local

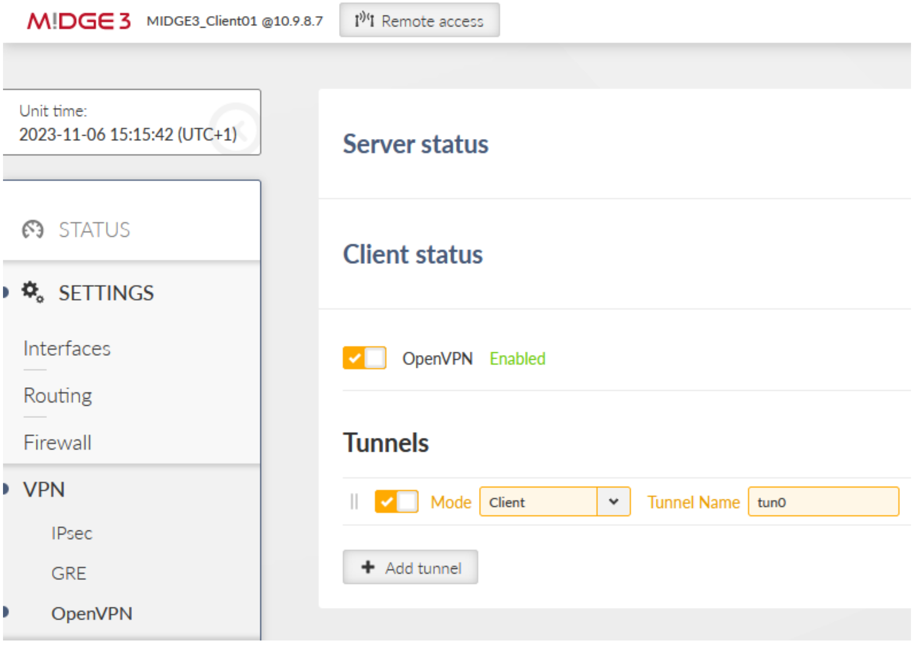
Fig. 26: MIDGE3_Client01 Static route

Save and apply all the changes.

Go to the SETTINGS > Security > Credentials menu and add all four downloaded keys/certificates. You need to import each with a correct "Type".

CA cert ID	ca_cert, Type: CA chain (PUB)
Client's certificate ID	midge3_client01_cert, Type: Certificate (PUB)
Client's key ID	midge3_client01_key, Type: Certificate key (PRI)
TLS-Auth key ID	ta_key, Type: OpenVPN TLS Protection key (PRI)

Go to the SETTINGS > VPN > OpenVPN menu. Enable the service and add the tunnel (Mode: Client).



Unit time:
2023-11-06 15:15:42 (UTC+1)

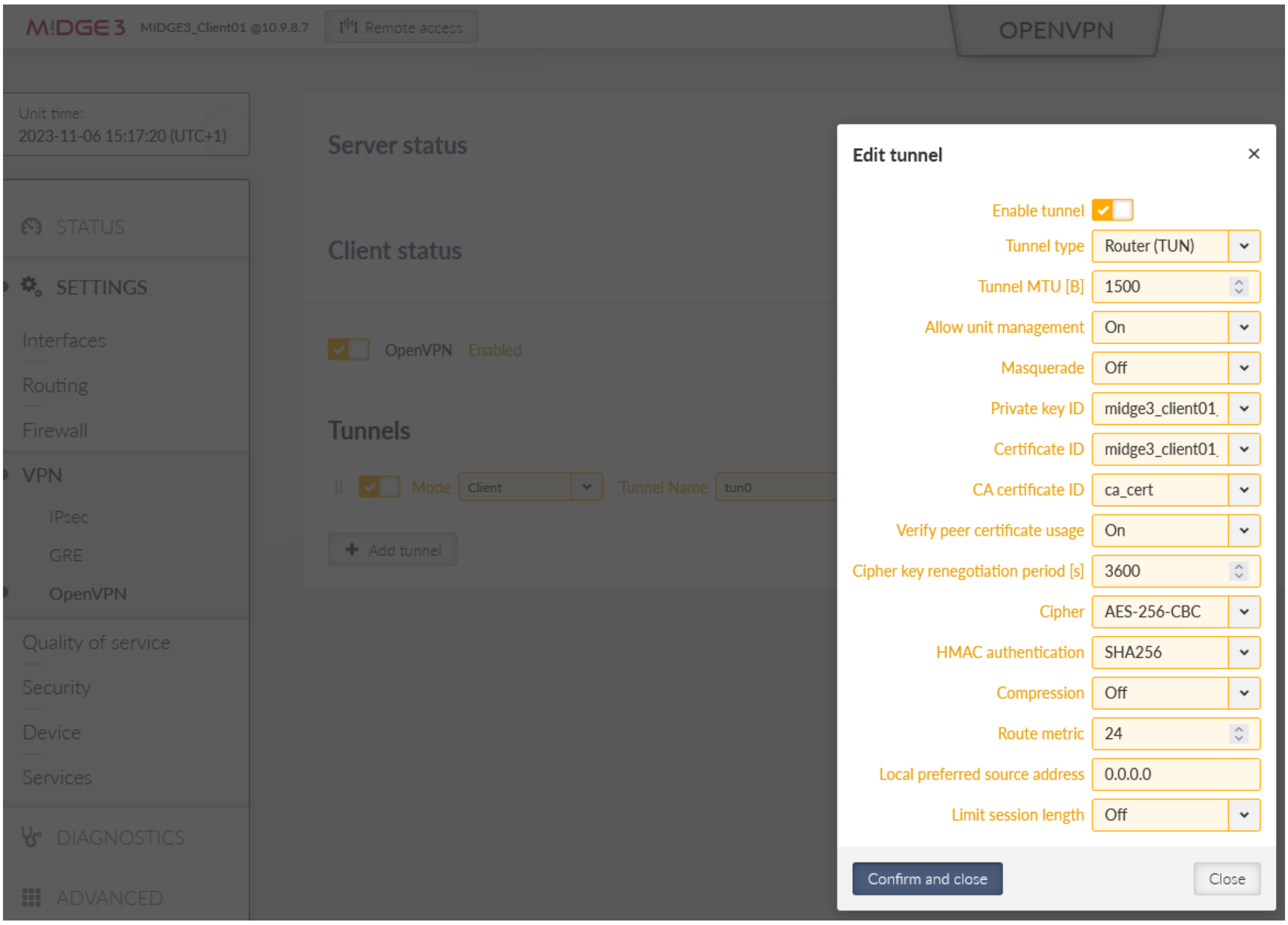
☑ OpenVPN Enabled

☑ Mode Client Tunnel Name tun0

Fig. 27: MIDGE3_Client01 OpenVPN settings

Edit the tunnel to match the Server’s settings. Keep all parameters in defaults, except:

Private key ID	midge3_client01_key
Certificate ID	midge3_client01_cert
CA certificate ID	ca_cert
Cipher	AES_256_CBC
HMAC authentication	SHA256



Edit tunnel

Enable tunnel	☑
Tunnel type	Router (TUN)
Tunnel MTU [B]	1500
Allow unit management	On
Masquerade	Off
Private key ID	midge3_client01_
Certificate ID	midge3_client01_
CA certificate ID	ca_cert
Verify peer certificate usage	On
Cipher key renegotiation period [s]	3600
Cipher	AES-256-CBC
HMAC authentication	SHA256
Compression	Off
Route metric	24
Local preferred source address	0.0.0.0
Limit session length	Off

Fig. 28: MIDGE3_Client01 OpenVPN tunnel parameters

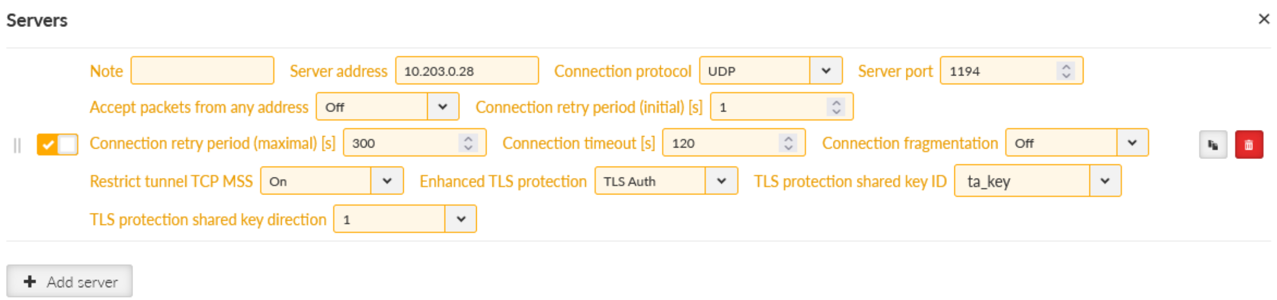
Confirm the changes.

Click on the “Edit servers” button and add one server. Fill in

Server address	10.203.0.28
Enhanced TLS Protection	TLS Auth
TLS Protection shared key ID	ta_key

Note

Your server address can be different, suit the setting to your APN.

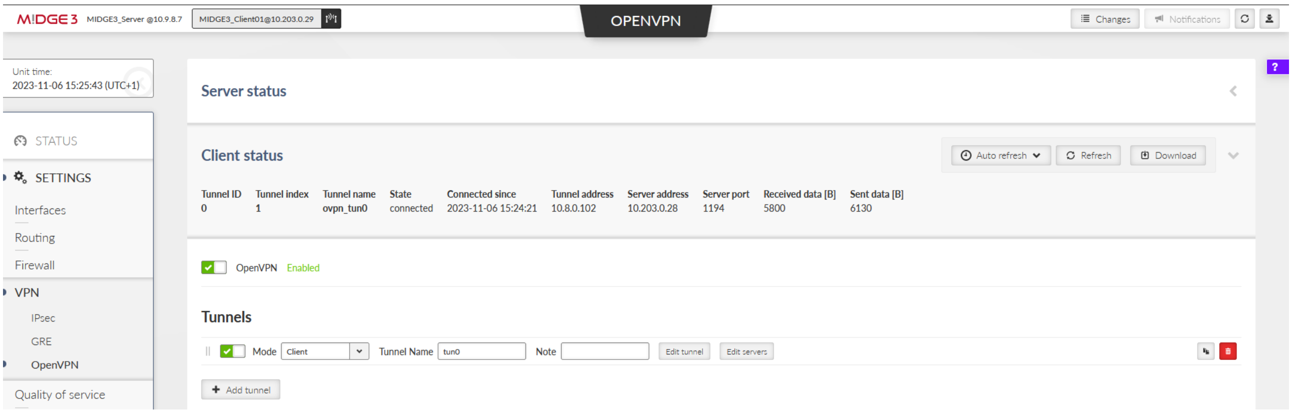


Note		Server address	10.203.0.28	Connection protocol	UDP	Server port	1194
Accept packets from any address	Off	Connection retry period (initial) [s]	1				
Connection retry period (maximal) [s]	300	Connection timeout [s]	120	Connection fragmentation	Off		
Restrict tunnel TCP MSS	On	Enhanced TLS protection	TLS Auth	TLS protection shared key ID	ta_key		
TLS protection shared key direction	1						

Fig. 29: MIDGE3_Client01 OpenVPN Servers

Apply the changes.

You can check if this OpenVPN client has connected successfully, or not. If not, download a Diagnostic package and go through the OpenVPN logs to find possible reasons. You should also check all the OpenVPN parameters so that they match.



Unit time:
2023-11-06 15:25:43 (UTC+1)

Server status

Client status

Tunnel ID	Tunnel index	Tunnel name	State	Connected since	Tunnel address	Server address	Server port	Received data [B]	Sent data [B]
0	1	ovpn_tun0	connected	2023-11-06 15:24:21	10.8.0.102	10.203.0.28	1194	5800	6130

☑ OpenVPN Enabled

Tunnels

☑ Mode Client Tunnel Name tun0 Note

Fig. 30: MIDGE3_Client01, successfully connected via OpenVPN

1.1.3. RipEX2_Client02

Login to the RipEX2_Client02 unit and go to the SETTINGS > Security > Credentials > Settings menu and set the Common name to be “ripex2_client02”. It is mandatory this time, because we will generate the client’s key and CSR on this unit, not in the server.

Note

Due to multiple same steps, some of the configuration screens are not provided.

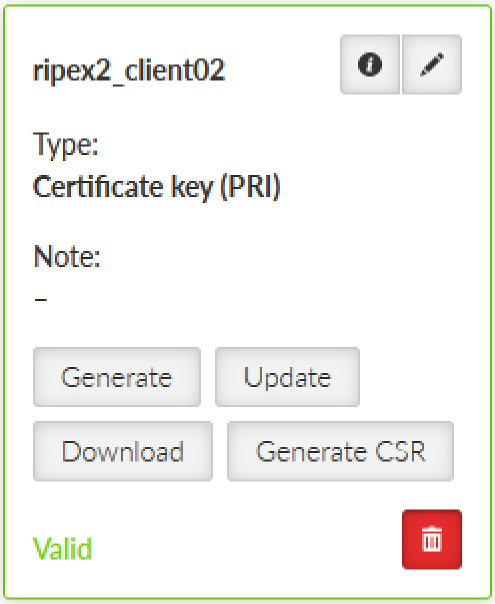
Repeat most of the steps from the M!DGE3 unit as well.

- Unit name to be RipEX2_Client02
- Correct Time zone and NTP server
- Ethernet bridge IP 192.168.3.1/24
- Cellular settings to suit your APN with Link testing
- Default (static) route, Mode: WWAN (EXT)
    - EXT is a must now, because setting RipEX2, not M!DGE3 device

Apply the changes.

Go back to the SETTINGS > Security > Credentials menu. Generate a private key. ID is “ripex2_client02_key”. Type is “Certificate key (PRI)”.

Generate CSR for this particular key. Click on the “Generate CSR” button located at this particular Credential window.



ripex2_client02

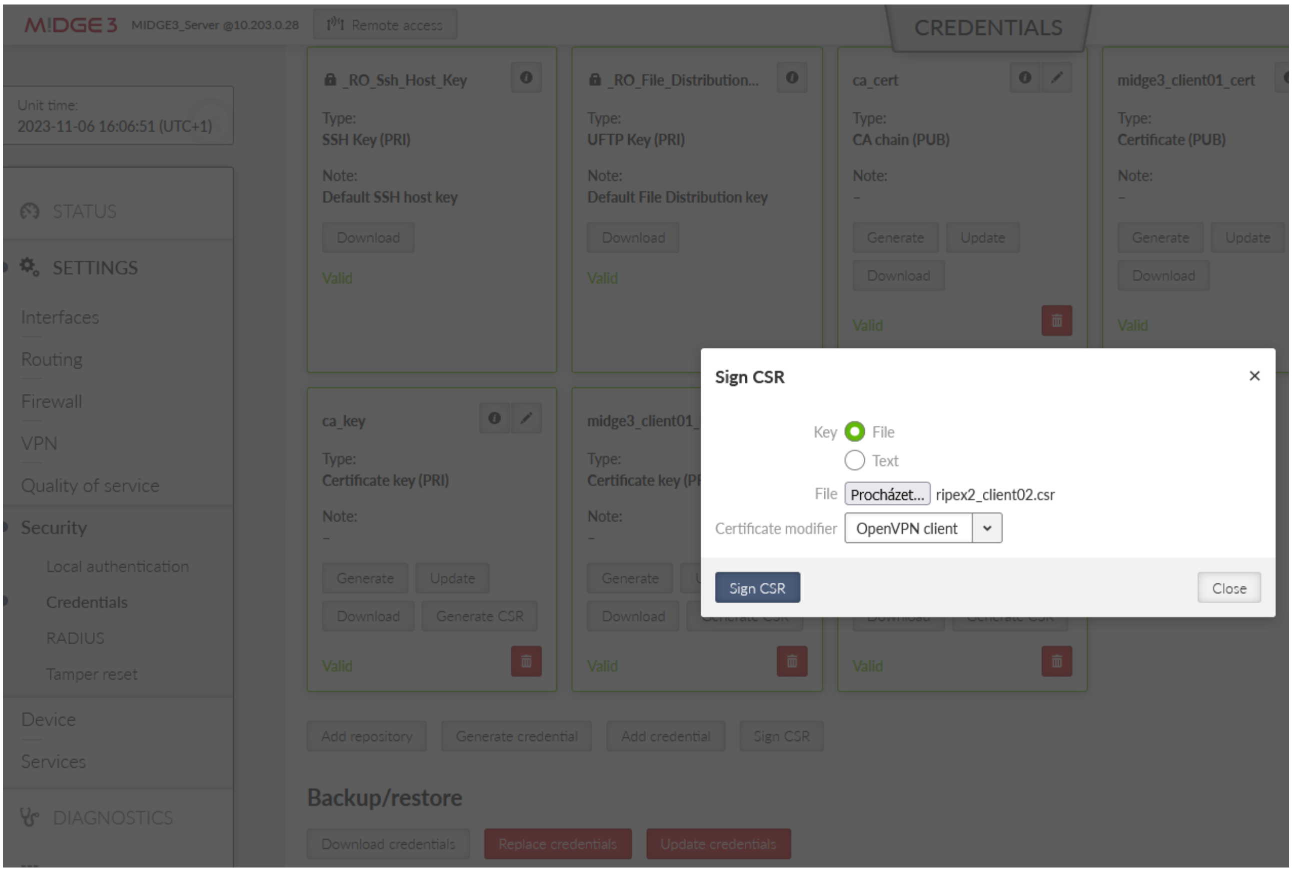
Type:
Certificate key (PRI)

Note:
–

Fig. 31: RipEX2_Client02 Generate CSR button

It downloads the CSR file into your PC. Login (locally or remotely) to MIDGE3_Server and go to the SETTINGS > Security > Credentials menu. Find the button “Sign CSR” next to the buttons with generating/uploading credentials.

Click on it and select the CSR file located in your PC. Select the Certificate modifier to be the OpenVPN client. Click on the Sign CSR button.



Sign CSR

Key ● File ○ Text

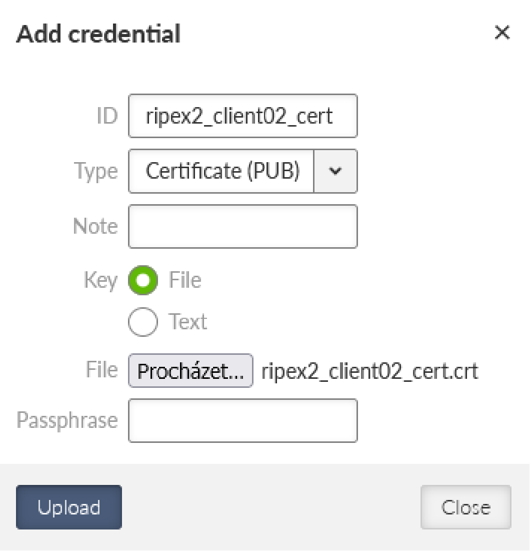
File Procházet... ripex2_client02.csr

Certificate modifier OpenVPN client

Fig. 32: MIDGE3_Server Signing 2nd client’s CSR

This downloads a valid certificate into your PC. Name it accordingly, e.g., “ripex2_client02_cert.crt”.

Go back to RipEX2_Client02’s web interface and add/upload this certificate. Set the ID to “ripex2_client02_cert” and Type to “Certificate (PUB)”.



Add credential

ID: ripex2_client02_cert

Type: Certificate (PUB)

Note:

Key: ● File ○ Text

File: Procházet... ripex2_client02_cert.crt

Passphrase:

Fig. 33: RipEX2_client02 Uploading a client's certificate (signed CSR)

Upload the CA certificate and the TLS Protection key the same way as in MIDGE3_Client01.

Go to the SETTINGS > VPN > OpenVPN menu, enable it and add one tunnel.

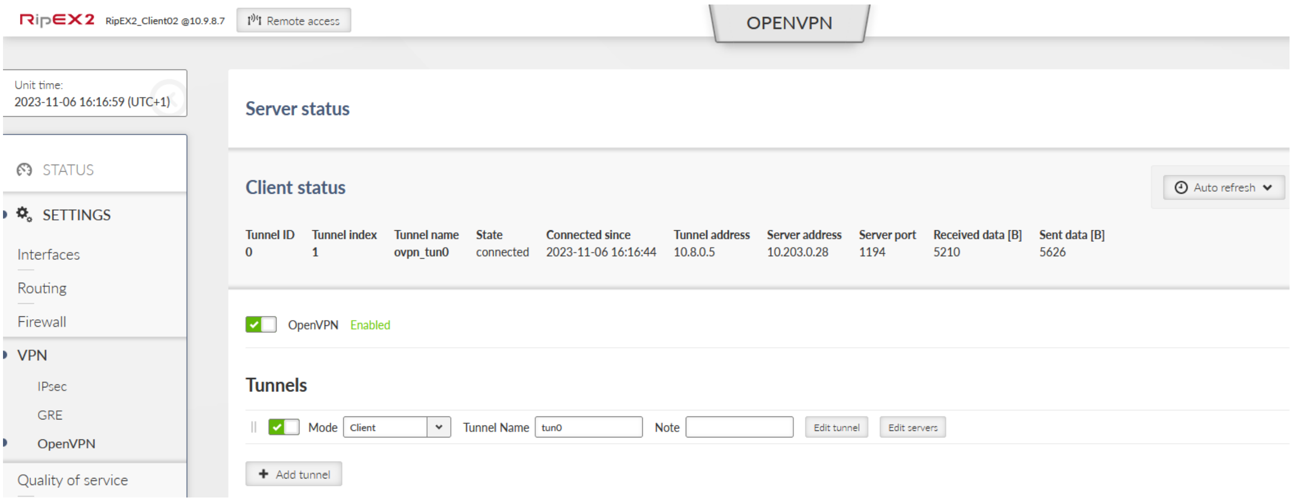
Edit the tunnel's parameters:

Private key ID	ripex2_client02_key
Certificate ID	ripex2_client02_cert
CA certificate ID	ca_cert
Cipher	AES_256_CBC
HMAC authentication	SHA256

Confirm the changes.

Edit the server settings. Add one server and set its address to 10.203.0.28 (it may differ for your APN). Enable Enhanced TLS protection with the ta_key.

Close the window and apply all the changes. Check if the client got connected and the assigned IP address for the tunnel is 10.8.0.5.



Server status

Client status

Tunnel ID	Tunnel index	Tunnel name	State	Connected since	Tunnel address	Server address	Server port	Received data [B]	Sent data [B]
0	1	ovpn_tun0	connected	2023-11-06 16:16:44	10.8.0.5	10.203.0.28	1194	5210	5626

☑ OpenVPN Enabled

Tunnels

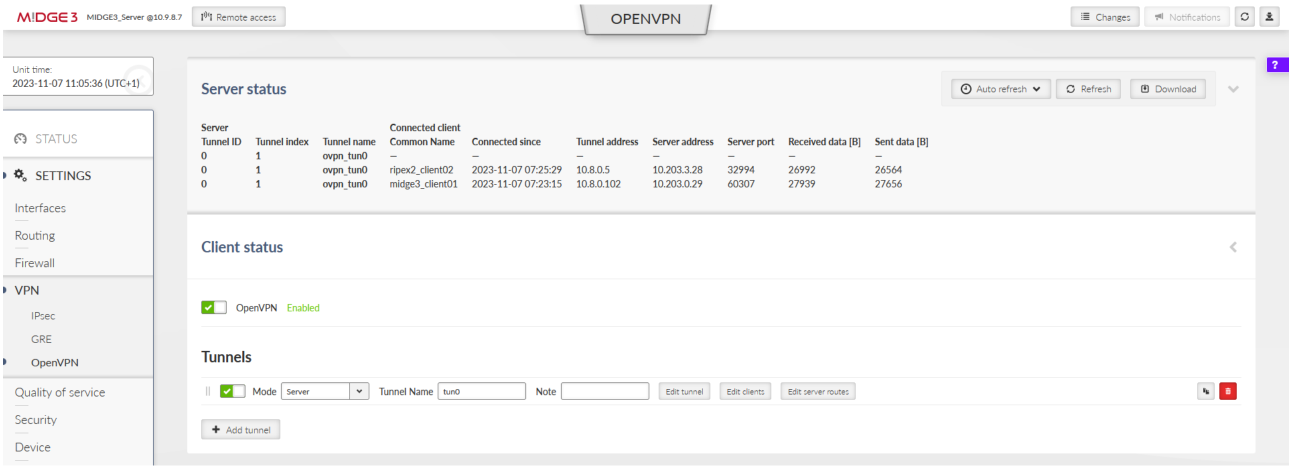
☑ Mode: Client; Tunnel Name: tun0; Note:

Fig. 34: RipEX2_Client02, successfully connected

1.1.4. Diagnostics

Within the SETTINGS > VPN > OpenVPN menu, open either the Server or Client status to see the details about currently connected units.

The MIDGE3_Server Status while both the clients are connected:



Server status

Server Tunnel ID	Tunnel index	Tunnel name	Connected client Common Name	Connected since	Tunnel address	Server address	Server port	Received data [B]	Sent data [B]
0	1	ovpn_tun0	–	–	–	–	–	–	–
0	1	ovpn_tun0	ripex2_client02	2023-11-07 07:25:29	10.8.0.5	10.203.3.28	32994	26992	26564
0	1	ovpn_tun0	midge3_client01	2023-11-07 07:23:15	10.8.0.102	10.203.0.29	60307	27939	27656

Client status

☑ OpenVPN Enabled

Tunnels

☑ Mode: Server; Tunnel Name: tun0; Note:

Fig. 35: MIDGE3_Server OpenVPN Status

Important fields

Tunnel name	Linux interface name (useful for firewalls, NAPT, …)
Common name	CN of the client. Especially in case only know CNs are allowed to be connected.
Connected since	Since what date&time the client is connected.
Tunnel address	Assigned OpenVPN tunnel IP address
Server address	WAN IP address of the client

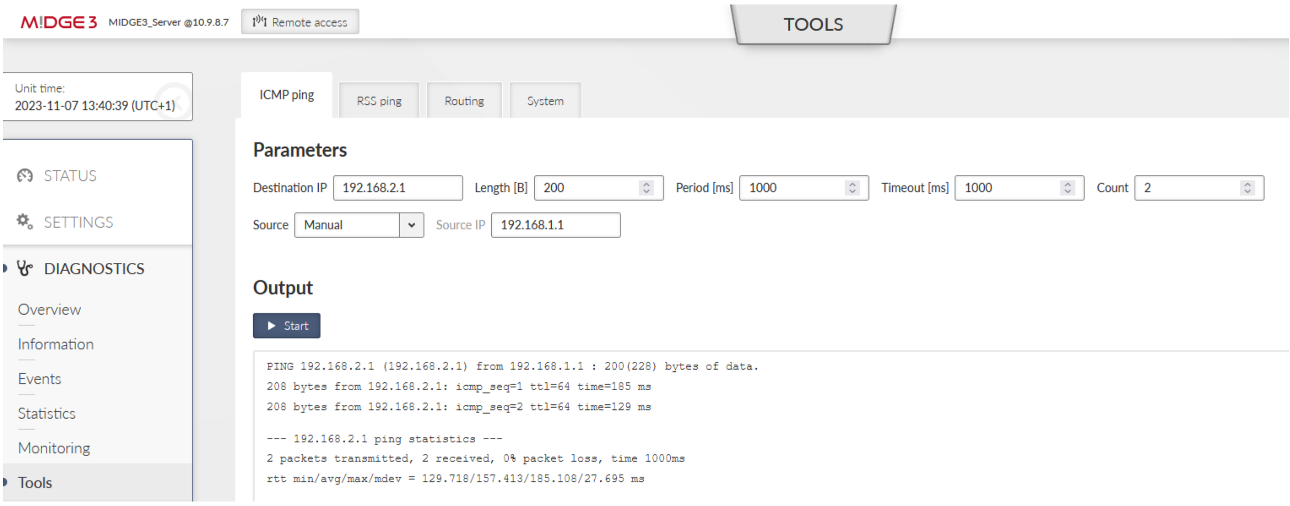
In case of the Client instance, there is “State” column – it can either be connected or disconnected.

Example of a client which cannot connect

A particular client is displayed in the Status, but has no Tunnel address assigned. And it also happens that such client is deleted from the Status within several seconds and may appear again within a while, …

- The particular client's Common name is not allowed to connect to the server – add it or fix a possible typo in the list of clients.

You should also try the DIAGNOSTICS > Tools > ICMP ping to check the accessibility of all LAN subnets. Either from the server to the client and back, or from client to client (it is allowed by default).



Unit time: 2023-11-07 13:40:39 (UTC+1)

Parameters

Destination IP: 192.168.2.1 | Length [B]: 200 | Period [ms]: 1000 | Timeout [ms]: 1000 | Count: 2

Source: Manual | Source IP: 192.168.1.1

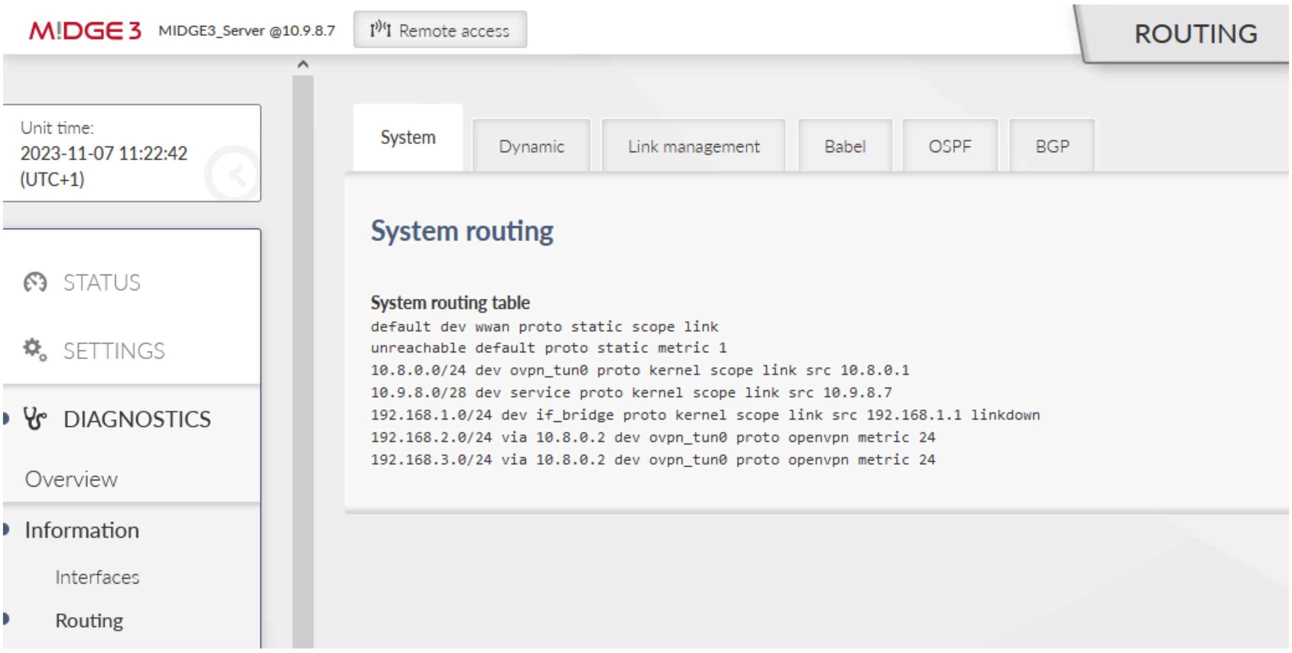
Output

```
PING 192.168.2.1 (192.168.2.1) from 192.168.1.1 : 200(228) bytes of data.
208 bytes from 192.168.2.1: icmp_seq=1 ttl=64 time=185 ms
208 bytes from 192.168.2.1: icmp_seq=2 ttl=64 time=129 ms

--- 192.168.2.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1000ms
rtt min/avg/max/mdev = 129.718/157.413/185.108/27.695 ms
```

Fig. 36: MIDGE3_Server ICMP ping

Another place to check a current state is the DIAGNOSTICS > Information > Routing menu. Within the System routing tab, you should also see the particular VPN routes – via some "ovpn_" interface.



Unit time: 2023-11-07 11:22:42 (UTC+1)

System routing

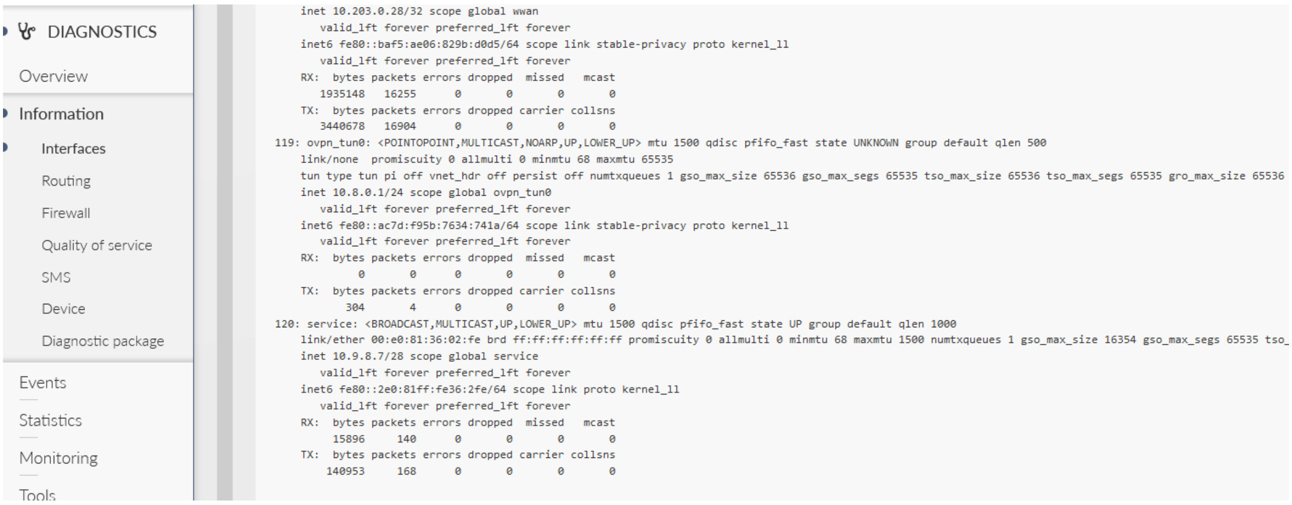
System routing table

```
default dev wwan proto static scope link
unreachable default proto static metric 1
10.8.0.0/24 dev ovpn_tun0 proto kernel scope link src 10.8.0.1
10.9.8.0/28 dev service proto kernel scope link src 10.9.8.7
192.168.1.0/24 dev if_bridge proto kernel scope link src 192.168.1.1 linkdown
192.168.2.0/24 via 10.8.0.2 dev ovpn_tun0 proto openvpn metric 24
192.168.3.0/24 via 10.8.0.2 dev ovpn_tun0 proto openvpn metric 24
```

Fig. 37: MIDGE3_Server Routing information

If some routing is missing, or is wrong, double check your server configuration – especially the Client and Server routes.

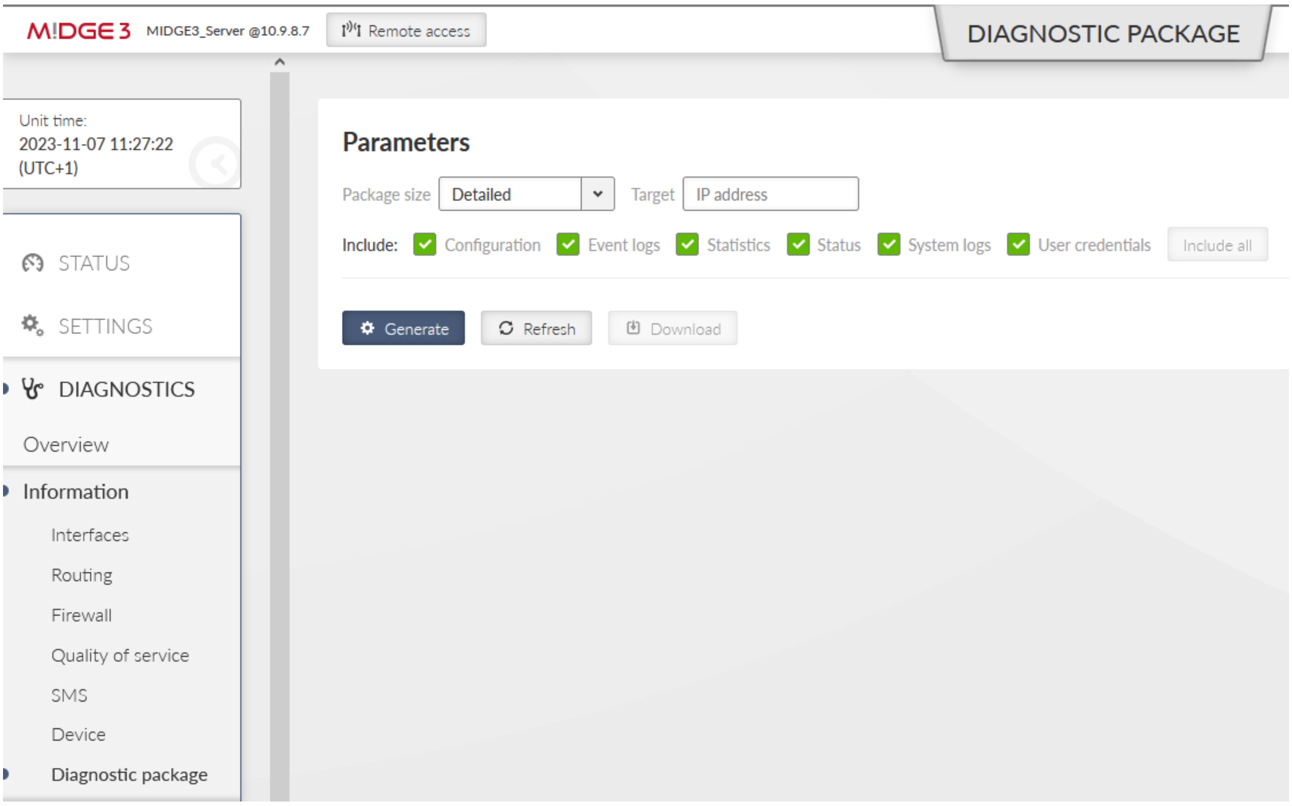
List of all interfaces can be displayed within the DIAGNOSTICS > Information > Interfaces menu. Try to find interfaces with “ovpn_” prefixes.



```
    inet 10.203.0.28/32 scope global wwan
       valid_lft forever preferred_lft forever
    inet6 fe80::baf5:ae06:829b:d0d5/64 scope link stable-privacy proto kernel_ll
       valid_lft forever preferred_lft forever
    RX:  bytes packets errors dropped  missed   mcast
       1935148   16255      0       0       0       0
    TX:  bytes packets errors dropped carrier collsns
       3440678   16904      0       0       0       0
119: ovpn_tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UNKNOWN group default qlen 500
    link/none  promiscuity 0 allmulti 0 minmtu 68 maxmtu 65535
    tun type tun pi off vnet_hdr off persist off numtxqueues 1 gso_max_size 65536 gso_max_segs 65535 tso_max_size 65536 tso_max_segs 65535 gro_max_size 65536
    inet 10.8.0.1/24 scope global ovpn_tun0
       valid_lft forever preferred_lft forever
    inet6 fe80::ac7d:f95b:7634:741a/64 scope link stable-privacy proto kernel_ll
       valid_lft forever preferred_lft forever
    RX:  bytes packets errors dropped  missed   mcast
             0       0      0       0       0       0
    TX:  bytes packets errors dropped carrier collsns
           304       4      0       0       0       0
120: service: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:e0:81:36:02:fe brd ff:ff:ff:ff:ff:ff promiscuity 0 allmulti 0 minmtu 68 maxmtu 1500 numtxqueues 1 gso_max_size 16354 gso_max_segs 65535 tso_
    inet 10.9.8.7/28 scope global service
       valid_lft forever preferred_lft forever
    inet6 fe80::2e0:81ff:fe36:2fe/64 scope link proto kernel_ll
       valid_lft forever preferred_lft forever
    RX:  bytes packets errors dropped  missed   mcast
         15896     140      0       0       0       0
    TX:  bytes packets errors dropped carrier collsns
        140953     168      0       0       0       0
```

Fig. 38: MIDGE3_Server List of interfaces

If you encounter any advanced connectivity/configuration issues, you can download the **Diagnostic package** within the DIAGNOSTICS > Information > Diagnostic package menu.



Unit time:
2023-11-07 11:27:22
(UTC+1)

Parameters

Package size: Detailed

Target: IP address

Include: ☑ Configuration ☑ Event logs ☑ Statistics ☑ Status ☑ System logs ☑ User credentials

Fig. 39: MIDGE3_Server Diagnostic package

Once downloaded, you need to extract the file and open the directory with logs. Go through the OpenVPN logs and try to find any reason for issues.

More details about possible issues in *Section 2, “Troubleshooting”*.

1.2. Bridged (TAP) OpenVPN topology

One typical network type is that all the end devices are within the same subnet (e.g. 192.168.1.0/24), but require a communication to each other via the RipEX2 Radio network or M!DGE3 cellular network. This is now possible using a secured channel via OpenVPN and its option to bridge particular interfaces together so that it creates a flat L2 topology over the routed scenario.

Note

It is possible to do it for the RipEX2 radio network operating in the Router mode as well.

The following example is just a continuation of *Section 1.1, “Routed (TUN) OpenVPN topology”*. We only do the required changes to match the desired IP topology and connectivity.

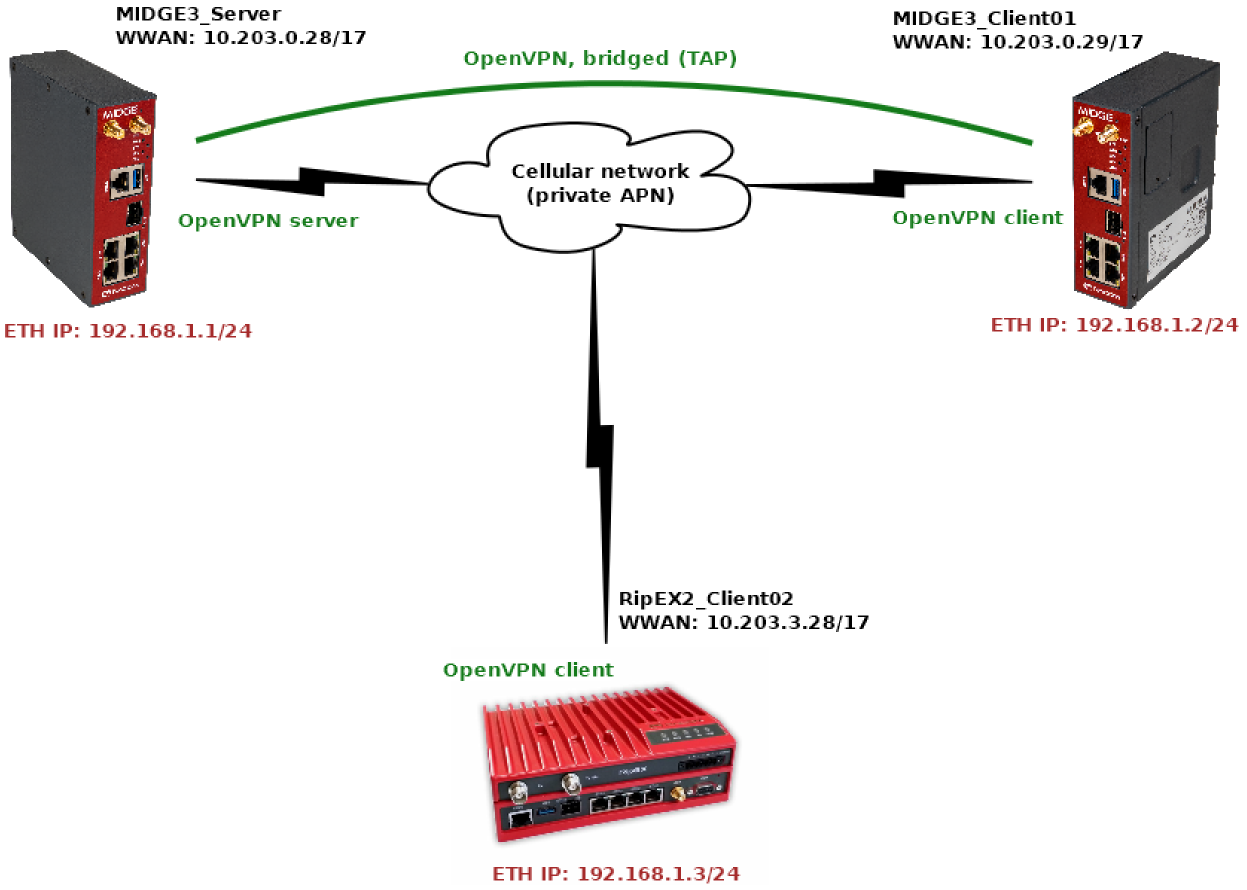


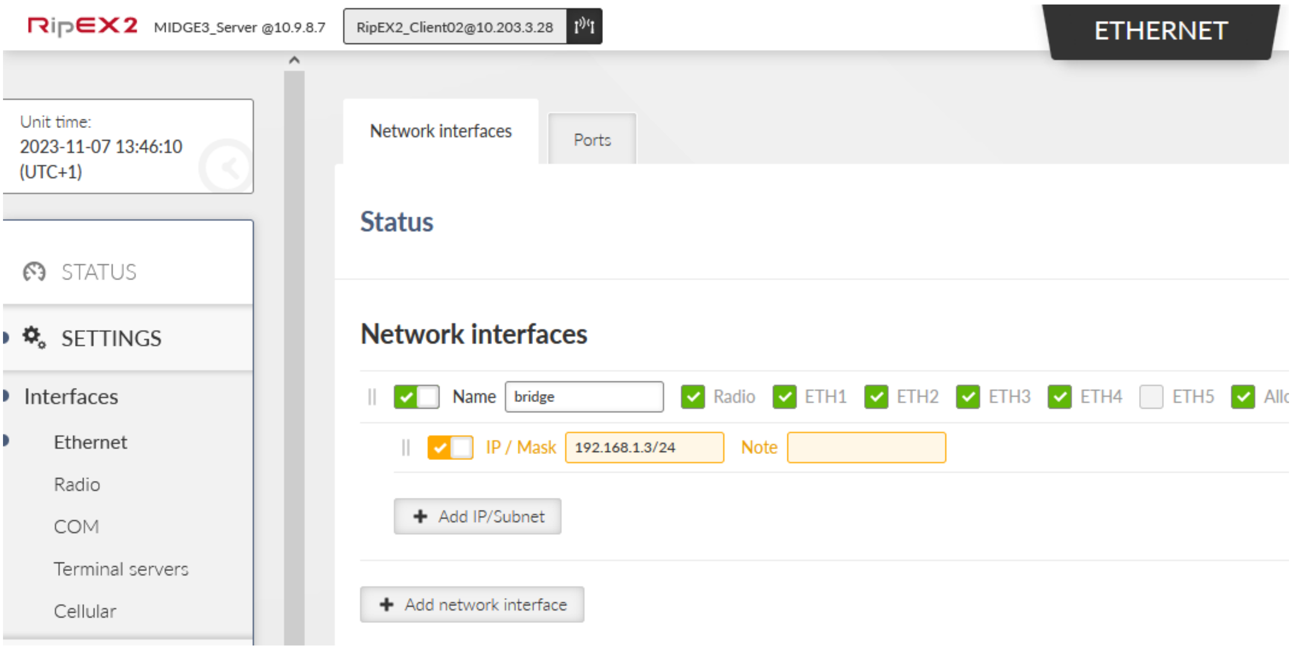
Fig. 40: Bridged (TAP) OpenVPN topology

We will need to

- change the Network interfaces’ addresses in both the clients
- change the OpenVPN configurations in all the units to match the bridged (TAP) topology

1.2.1. RipEX2_Client02

Start with the RipEX2_Client02 unit. Go to the SETTINGS > Interfaces > Ethernet. Change the IP address to 192.168.1.3/24.



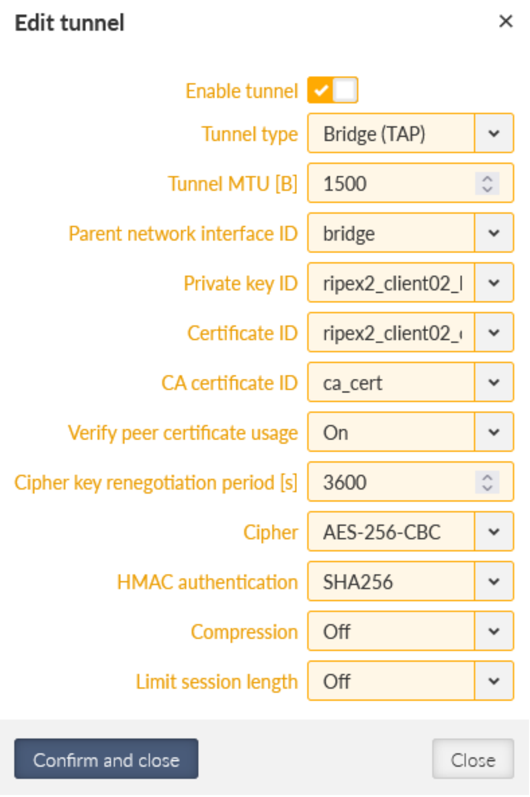
Network interfaces

☑ Name: bridge ☑ Radio ☑ ETH1 ☑ ETH2 ☑ ETH3 ☑ ETH4 □ ETH5 ☑ All

☑ IP / Mask: 192.168.1.3/24 Note:

Fig. 41: RipEX2_Client02 Ethernet if_bridge IP address

Go to the SETTINGS > VPN > OpenVPN menu and change the Type of the tunnel from Router (TUN) to Bridge (TAP). We only have one Network Interface (bridge) so the “Parent network interface ID” is automatically set to “bridge” and there is no other option. In case of multiple interfaces, select a correct one from the given list.

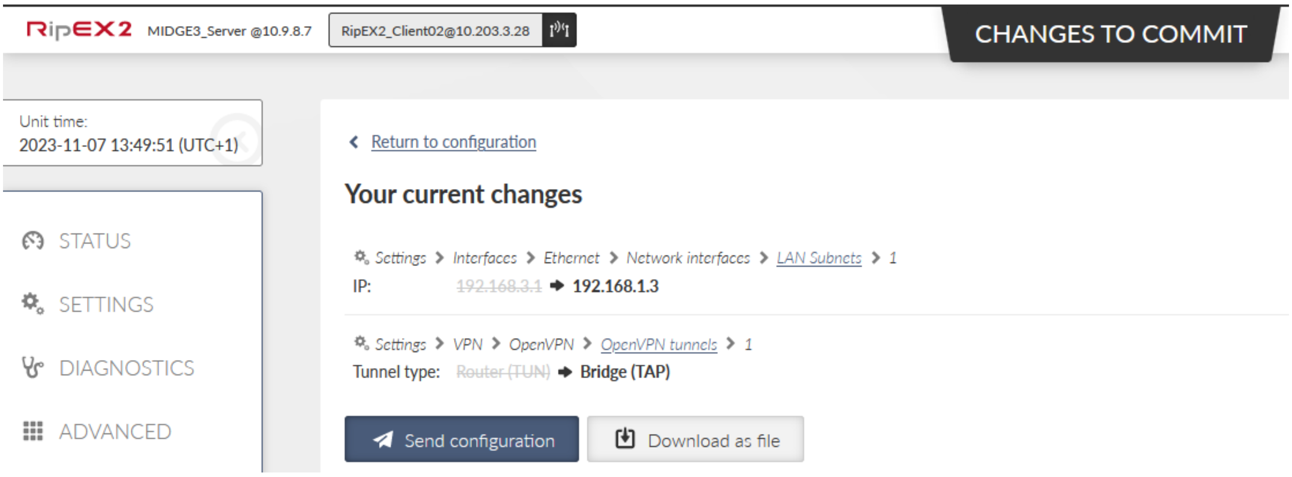


Edit tunnel

Enable tunnel	☑
Tunnel type	Bridge (TAP)
Tunnel MTU [B]	1500
Parent network interface ID	bridge
Private key ID	ripex2_client02_I
Certificate ID	ripex2_client02_(
CA certificate ID	ca_cert
Verify peer certificate usage	On
Cipher key renegotiation period [s]	3600
Cipher	AES-256-CBC
HMAC authentication	SHA256
Compression	Off
Limit session length	Off

Fig. 42: RipEX2_Client02 OpenVPN Tunnel type change

Apply all the changes.



Unit time:
2023-11-07 13:49:51 (UTC+1)

Your current changes

Settings > Interfaces > Ethernet > Network interfaces > LAN Subnets > 1
IP: ~~192.168.3.1~~ → 192.168.1.3

Settings > VPN > OpenVPN > OpenVPN tunnels > 1
Tunnel type: ~~Router (TUN)~~ → Bridge (TAP)

Fig. 43: RipEX2_Client02 Changes to commit

Note

Based on type of connection to the client unit, you may be disconnected now.

1.2.2. MIDGE3_Client01

Go to the MIDGE3_Client01 unit and do the similar changes

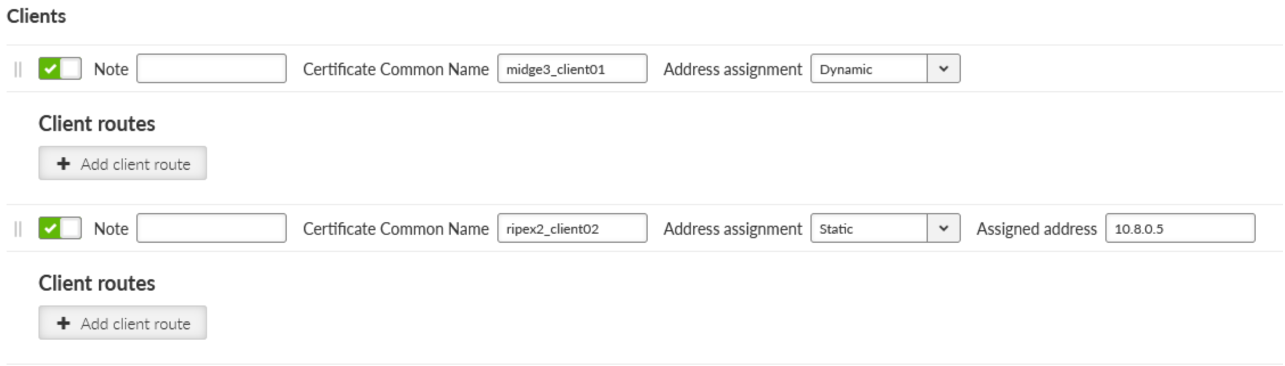
- Change the Ethernet “bridge” Network interface IP to 192.168.1.2/24
- Change the OpenVPN Tunnel type to Bridge (TAP)

Commit the changes as well.

1.2.3. MIDGE3_Server

Eventually, go back to the MIDGE3_Server and go to the SETTINGS > VPN > OpenVPN menu. Change the Tunnel type to Bridge (TAP) as well. Close window.

Click on the “Edit clients” button and delete both Client routes. Close the window.



Clients

☑ Note Certificate Common Name midge3_client01 Address assignment Dynamic

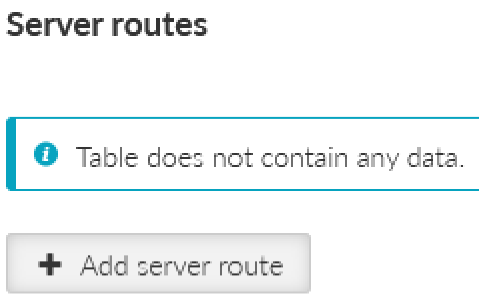
Client routes

☑ Note Certificate Common Name ripex2_client02 Address assignment Static Assigned address 10.8.0.5

Client routes

Fig. 44: MIDGE3_Server Edit clients

Click on the “Edit server routes” button and delete the single line with 192.168.1.0/24.



Server routes

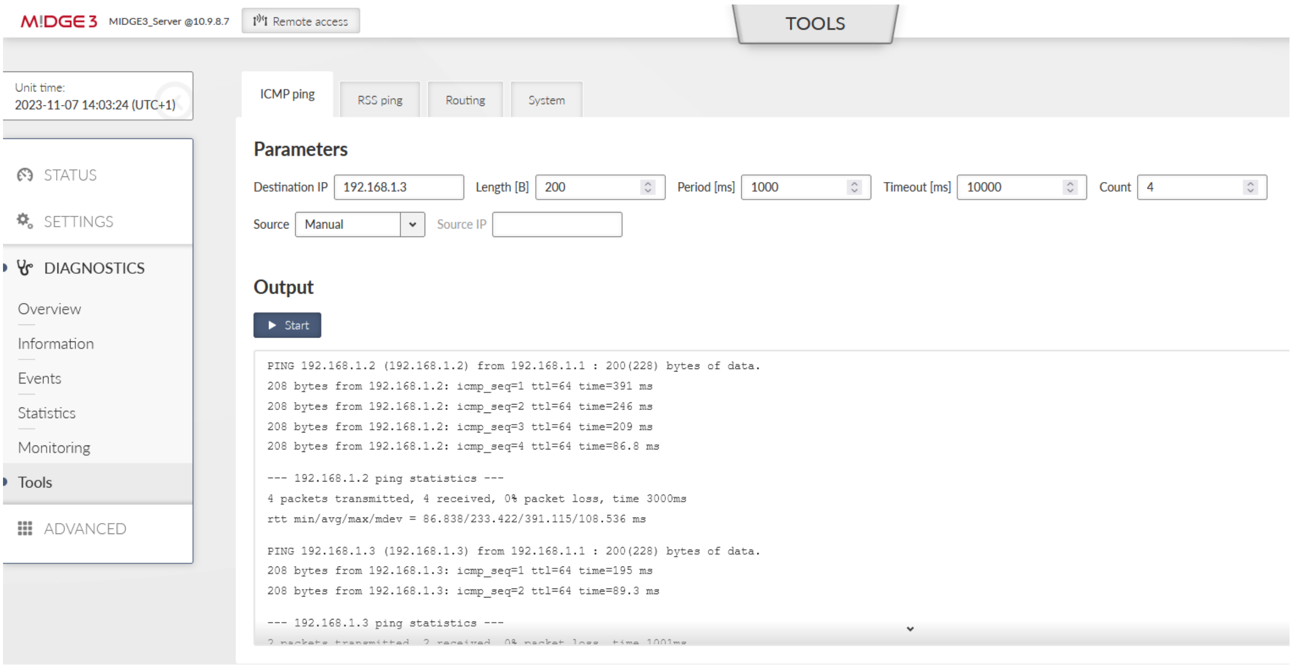
Table does not contain any data.

Fig. 45: MIDGE3_Server Empty Server routes

Commit all the changes.

1.2.4. Diagnostics

Go to the DIAGNOTICS > Tools > ICMP ping and try to ping both the remotes 192.168.1.2 and 192.168.1.3.



Unit time: 2023-11-07 14:03:24 (UTC+1)

Parameters

Destination IP 192.168.1.3 Length [B] 200 Period [ms] 1000 Timeout [ms] 10000 Count 4

Source Manual Source IP

Output

```
PING 192.168.1.2 (192.168.1.2) from 192.168.1.1 : 200(228) bytes of data.
208 bytes from 192.168.1.2: icmp_seq=1 ttl=64 time=391 ms
208 bytes from 192.168.1.2: icmp_seq=2 ttl=64 time=246 ms
208 bytes from 192.168.1.2: icmp_seq=3 ttl=64 time=209 ms
208 bytes from 192.168.1.2: icmp_seq=4 ttl=64 time=86.8 ms

--- 192.168.1.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3000ms
rtt min/avg/max/mdev = 86.838/233.422/391.115/108.536 ms

PING 192.168.1.3 (192.168.1.3) from 192.168.1.1 : 200(228) bytes of data.
208 bytes from 192.168.1.3: icmp_seq=1 ttl=64 time=195 ms
208 bytes from 192.168.1.3: icmp_seq=2 ttl=64 time=89.3 ms

--- 192.168.1.3 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
```

Fig. 46: MIDGE3_Server ICMP ping

Go through the OpenVPN Status, Information and logs in case of any issues.

2. Troubleshooting

We have already explained multiple debugging/troubleshooting options. See below several hints while going through OpenVPN logs and solving connectivity issues.

2.1. Logs

OpenVPN logs are called *openvpn_X* and *openvpn_diag_X* where X is the OpenVPN tunnel ID. The 2nd log does not provide much information, it is mostly our own diagnostics with basic connected/disconnected information.

openvpn_x logs

Server: client got connected

- Ciphers negotiated
- Connection information
- Assigning a client tunnel interface address
- Reporting from the connection event script

```
2023-10-09T05:18:17+00:00 2023-10-09 05:18:17 10.10.1.55:54280 Control Channel: TLSv1.3, ▶
cipher TLSv1.3 TLS_AES_256_GCM_SHA384, peer certificate: 3072 bit RSA, signature: RSA-SHA256
2023-10-09T05:18:17+00:00 2023-10-09 05:18:17 10.10.1.55:54280 [Epsilon] Peer Connection ▶
Initiated with [AF_INET]10.10.1.55:54280 (via [AF_INET]10.10.1. 11%radio)
2023-10-09T05:18:17+00:00 2023-10-09 05:18:17 10.10.1.55:54280 Control Channel: TLSv1.3, ▶
cipher TLSv1.3 TLS_AES_256_GCM_SHA384, peer certificate: 3072 bit RSA, signature: RSA-SHA256
2023-10-09T05:18:17+00:00 2023-10-09 05:18:17 Epsilon/10.10.1.55:54280 MULTI_sva: pool ▶
returned IPv4=10.201.0.100, IPv6=(Not enabled)
2023-10-09T05:18:17+00:00 ===== Client Epsilon (10.10.1.55:54280) connected. =====
```

Server: client got disconnected

```
2023-10-09T11:22:16+00:00 ===== Client Epsilon (10.10.1.55:54280) disconnected. =====
```

Client: connection to the server finished

- Going to the “Connected state”
- Ciphers negotiated

```
2023-10-09T05:18:04+00:00 2023-10-09 05:18:04 Initialization Sequence Completed ▶
2023-10-09T05:18:04+00:00 2023-10-09 05:18:04 MANAGEMENT: ▶
>STATE:1696828684,CONNECTED,SUCCESS,10.201.0.100,10.10.1.11,1194,,
2023-10-09T05:18:04+00:00 2023-10-09 05:18:04 Data Channel: cipher 'AES-256-GCM', peer-id: 0
```

Client: The server does not respond to client’s connection trials

```
2023-10-09T05:12:10+00:00 2023-10-09 05:12:10 read UDPv4 [EHOSTUNREACH]: No route to host ▶
(fd=4,code=148)
2023-10-09T05:12:32+00:00 2023-10-09 05:12:32 TLS Error: TLS key negotiation failed to ▶
occur within 60 seconds (check your network connectivity)
2023-10-09T05:12:32+00:00 2023-10-09 05:12:32 TLS Error: TLS handshake failed
```

Client authentication failure in the server:

- Server:

```
2023-10-09T11:25:49+00:00 2023-10-09 11:25:49 10.10.1.55:40917 VERIFY ERROR: depth=1, ▶
error=self-signed certificate in certificate chain: O=RACOM, OU=Development, L=Bystrice ▶
n. P., ST=CR, C=CZ, CN=Alfa, emailAddress=sectech@racom.eu, ▶
serial=22739517585033819111443524660849504900645973455
2023-10-09T11:25:49+00:00 2023-10-09 11:25:49 10.10.1.55:40917 OpenSSL: error:0A000086:SSL ▶
routines::certificate verify failed
2023-10-09T11:25:49+00:00 2023-10-09 11:25:49 10.10.1.55:40917 TLS_ERROR: BIO read ▶
tls_read_plaintext error
2023-10-09T11:25:49+00:00 2023-10-09 11:25:49 10.10.1.55:40917 TLS Error: TLS object -> ▶
incoming plaintext read error
2023-10-09T11:25:49+00:00 2023-10-09 11:25:49 10.10.1.55:40917 TLS Error: TLS handshake ▶
failed
```

- Client:

```
2023-10-09T11:26:43+00:00 2023-10-09 11:26:43 TLS Error: TLS key negotiation failed to ▶
occur within 60 seconds (check your network connectivity
2023-10-09T11:26:43+00:00 2023-10-09 11:26:43 TLS Error: TLS handshake failed
```

Server authentication failure in the client:

- Server:

```
2023-10-09T11:33:26+00:00 2023-10-09 11:33:26 read UDPv4 [CMSG=8|ECONNREFUSED]: Connection ▶
refused (fd=6,code=146)
```

- Client:

```
2023-10-09T11:33:44+00:00 2023-10-09 11:33:44 VERIFY ERROR: depth=1, error=self-signed ▶
certificate in certificate chain: O=RACOM, OU=Development, L=Bystrice n. P., ST=CR, C=CZ, ▶
CN=Alfa, emailAddress=sectech@racom.eu, ▶
serial=22739517585033819111443524660849504900645973455
2023-10-09T11:33:44+00:00 2023-10-09 11:33:44 OpenSSL: error:0A000086:SSL ▶
routines::certificate verify failed
2023-10-09T11:33:44+00:00 2023-10-09 11:33:44 TLS_ERROR: BIO read tls_read_plaintext error
2023-10-09T11:33:44+00:00 2023-10-09 11:33:44 TLS Error: TLS object -> incoming plaintext ▶
read error
2023-10-09T11:33:44+00:00 2023-10-09 11:33:44 TLS Error: TLS handshake failed
```

Server: Maximum number of connected clients exceeded:

```
2023-10-02T09:36:38+00:00 2023-10-02 09:36:38 10.10.1.55:32786 MULTI: new incoming ▶
connection would exceed maximum number of clients (1)
```

Server: The client’s connection is refused, because it is not listed within the OpenVPN clients:

```
2023-10-09T11:39:21+00:00 2023-10-09 11:39:21 10.10.1.55:54272 TLS Auth Error: ▶
--client-config-dir authentication failed for common name 'Epsilon' ▶
file='/var/run/openvpn/1/ccd/Epsilon'
2023-10-09T11:39:21+00:00 2023-10-09 11:39:21 10.10.1.55:54272 TLS: move_session: ▶
dest=TM_ACTIVE src=TM_INITIAL reinit_src=1
2023-10-09T11:39:21+00:00 2023-10-09 11:39:21 10.10.1.55:54272 TLS: tls_multi_process: ▶
initial untrusted session promoted to semi-trusted
2023-10-09T11:39:21+00:00 2023-10-09 11:39:21 10.10.1.55:54272 Delayed exit in 5 seconds
2023-10-09T11:39:21+00:00 2023-10-09 11:39:21 10.10.1.55:54272 SENT CONTROL [UNDEF]: ▶
'AUTH_FAILED' (status=1)
2023-10-09T11:39:21+00:00 2023-10-09 11:39:21 10.10.1.55:54272 SENT CONTROL [Epsilon]: ▶
'AUTH_FAILED' (status=1)
```

Client: The server is refused due to the Extended Key Usage (EKU) check:

```
2023-10-09T11:37:14+00:00 2023-10-09 11:37:14 VERIFY KU ERROR
2023-10-09T11:37:14+00:00 2023-10-09 11:37:14 OpenSSL: error:0A000086:SSL ▶
routines::certificate verify failed
2023-10-09T11:37:14+00:00 2023-10-09 11:37:14 TLS_ERROR: BIO read tls_read_plaintext error
2023-10-09T11:37:14+00:00 2023-10-09 11:37:14 TLS Error: TLS object -> incoming plaintext ▶
read error
2023-10-09T11:37:14+00:00 2023-10-09 11:37:14 TLS Error: TLS handshake failed
```

Link fragmentation is turned on, but the Peer has it turned off:

```
2023-10-02T08:05:54+00:00 2023-10-02 08:05:54 Beta/10.10.1.22:37547 IP packet with unknown ▶
IP version=0 seen
2023-10-02T08:06:05+00:00 2023-10-02 08:06:05 Beta/10.10.1.22:37547 FRAG_IN error ▶
flags=0x2a187bf3: FRAG_TEST not implemented
```

Server: Server has the compression enabled, but the client has it disabled:

```
2023-10-09T11:47:19+00:00 2023-10-09 11:47:19 Epsilon/10.10.1.55:57609 IP packet with ▶
unknown IP version=15 seen
```

Client: It has the compression disabled, but the server has it enabled and wants to enforce it:

```
2023-10-09T11:43:27+00:00 2023-10-09 11:43:27 Compression is not allowed since ▶
allow-compression is set to 'stub-only'
2023-10-09T11:43:27+00:00 2023-10-09 11:43:27 OPTIONS ERROR: server pushed compression ▶
settings that are not allowed and will result in a non-working connection. See also ▶
allow-compression in the manual.
2023-10-09T11:43:27+00:00 2023-10-09 11:43:27 ERROR: Failed to apply push options
2023-10-09T11:43:27+00:00 2023-10-09 11:43:27 Failed to open tun/tap interface
```

TLS Auth: The peer side uses a different key:

```
2023-08-08 09:33:47 Authenticate/Decrypt packet error: packet HMAC authentication failed
2023-08-08 09:33:47 TLS Error: incoming packet authentication failed from ▶
[AF_INET]10.10.1.44:47771
```

TLS Auth: The peer side does not have TLS-Auth enabled:

```
2023-08-08 09:37:38 TLS Error: cannot locate HMAC in incoming packet from ▶
[AF_INET]10.10.1.44:47878
```

TLS Crypt: The peer side uses a different key:

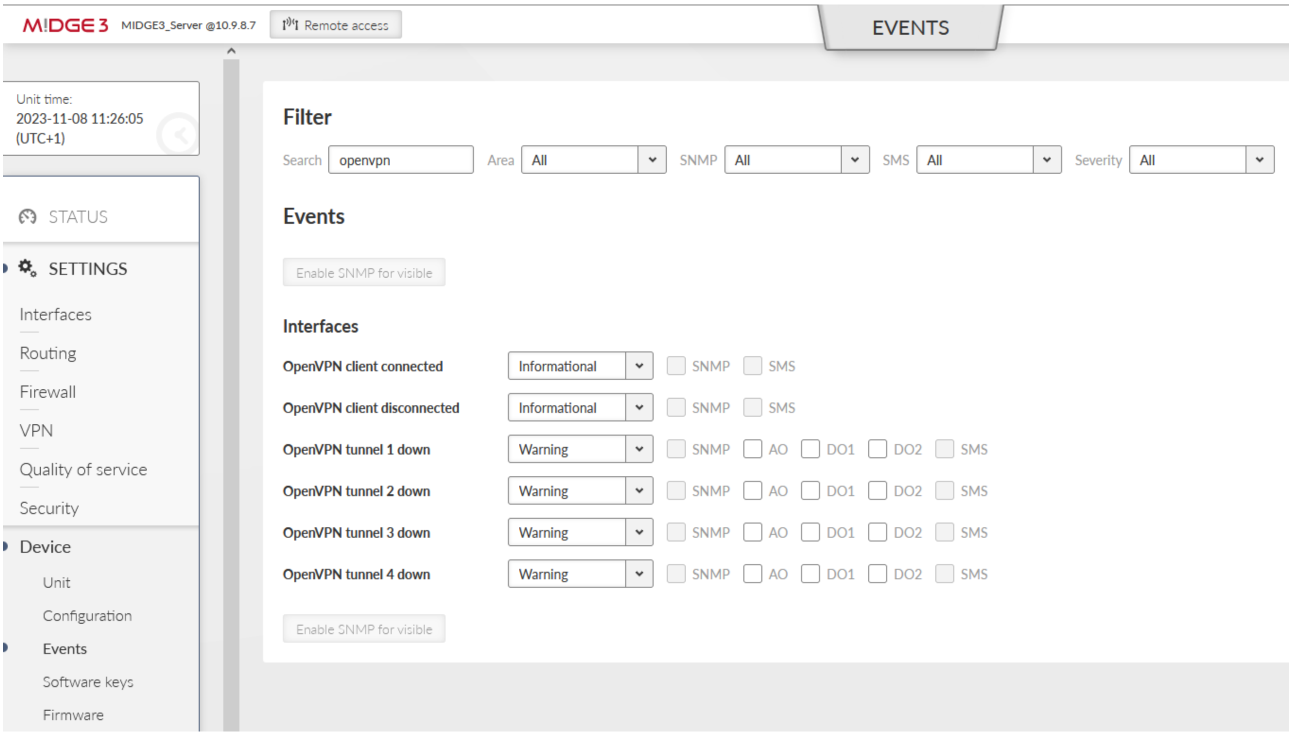
```
2023-08-08 11:17:33 tls-crypt unwrap error: packet authentication failed
2023-08-08 11:17:33 TLS Error: tls-crypt unwrapping failed from [AF_INET]10.10.1.44:56203
```

TLS Crypt: The peer side does not have TLS-Crypt enabled:

```
2023-08-08 11:18:34 tls-crypt unwrap error: packet too short
2023-08-08 11:18:34 TLS Error: tls-crypt unwrapping failed from [AF_INET]10.10.1.44:59311
```

2.2. Events

M!DGE3/RipEX2 devices support so called Events.



Unit time:
2023-11-08 11:26:05
(UTC+1)

Filter

Search: openvpn · Area: All · SNMP: All · SMS: All · Severity: All

Events

Interfaces

OpenVPN client connected	Informational	□ SNMP	□ SMS			
OpenVPN client disconnected	Informational	□ SNMP	□ SMS			
OpenVPN tunnel 1 down	Warning	□ SNMP	□ AO	□ DO1	□ DO2	□ SMS
OpenVPN tunnel 2 down	Warning	□ SNMP	□ AO	□ DO1	□ DO2	□ SMS
OpenVPN tunnel 3 down	Warning	□ SNMP	□ AO	□ DO1	□ DO2	□ SMS
OpenVPN tunnel 4 down	Warning	□ SNMP	□ AO	□ DO1	□ DO2	□ SMS

Fig. 47: Events

You can be notified about the clients being (dis)connected and about the tunnels being down via SNMP notifications, Alarm and Digital outputs or SMS.

You can also change the severities for particular events so it can e.g., display this important Event within the STATUS menu, or change the physical SYS LED diode color to red.

All Events can also be displayed and filtered within the DIAGNOSTICS > Events menu. Read more information about the Events in the manual.

3. M!DGE2 compatibility

OpenVPN is a well-known protocol and is compatible with any other OpenVPN enabled device, including older M!DGE2 cellular router.

You can do both, adding M!DGE2 into the existing M!DGE3/RipEX2 network or adding M!DGE3/RipEX2 to the existing M!DGE2 network.

Just configure the correct OpenVPN parameters and upload the keys and certificates.

Keep in mind that older M!DGE2 "Expert files" are no longer supported in M!DGE3/RipEX2 units. Configure the parameters manually (based on 2.1.1.0 firmware features).

Note

Keep the M!DGE2 software up-to-date. Check the Download section on *RACOM website*[1].

3.1. Adding M!DGE2 to the existing M!DGE3/RipEX2 network

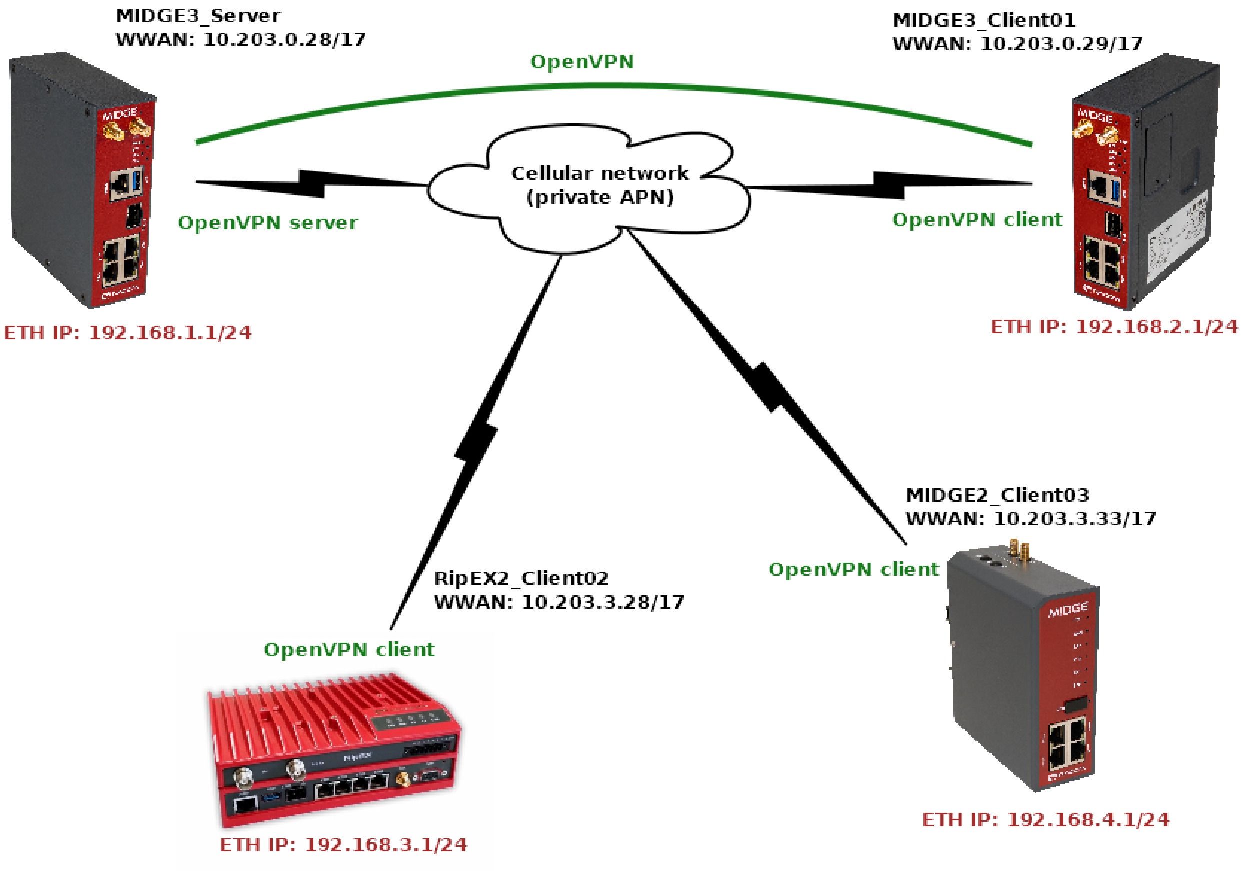


Fig. 48: Routed (TUN) OpenVPN topology, M!DGE2 added

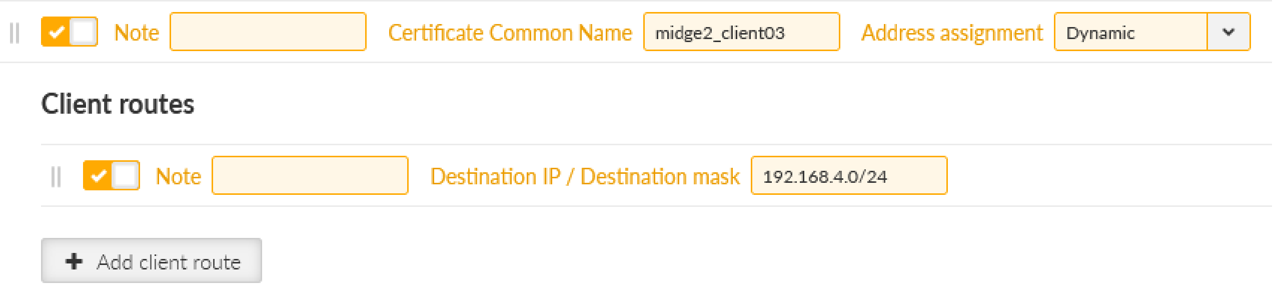
Note

There can be multiple ways of doing the same. Consider the following text to be just one of the options. You can choose a different approach.

[1] https://www.racom.eu/eng/products/cellular-router-midge.html#dnl_fwr

3.1.1. MIDGE3_Server

We start in the server by going to the SETTINGS > VPN > OpenVPN menu. Open the “Edit clients” menu and add a new client with a Certificate Common name equal to “midge2_client03”. Add its route to be 192.168.4.0/24.



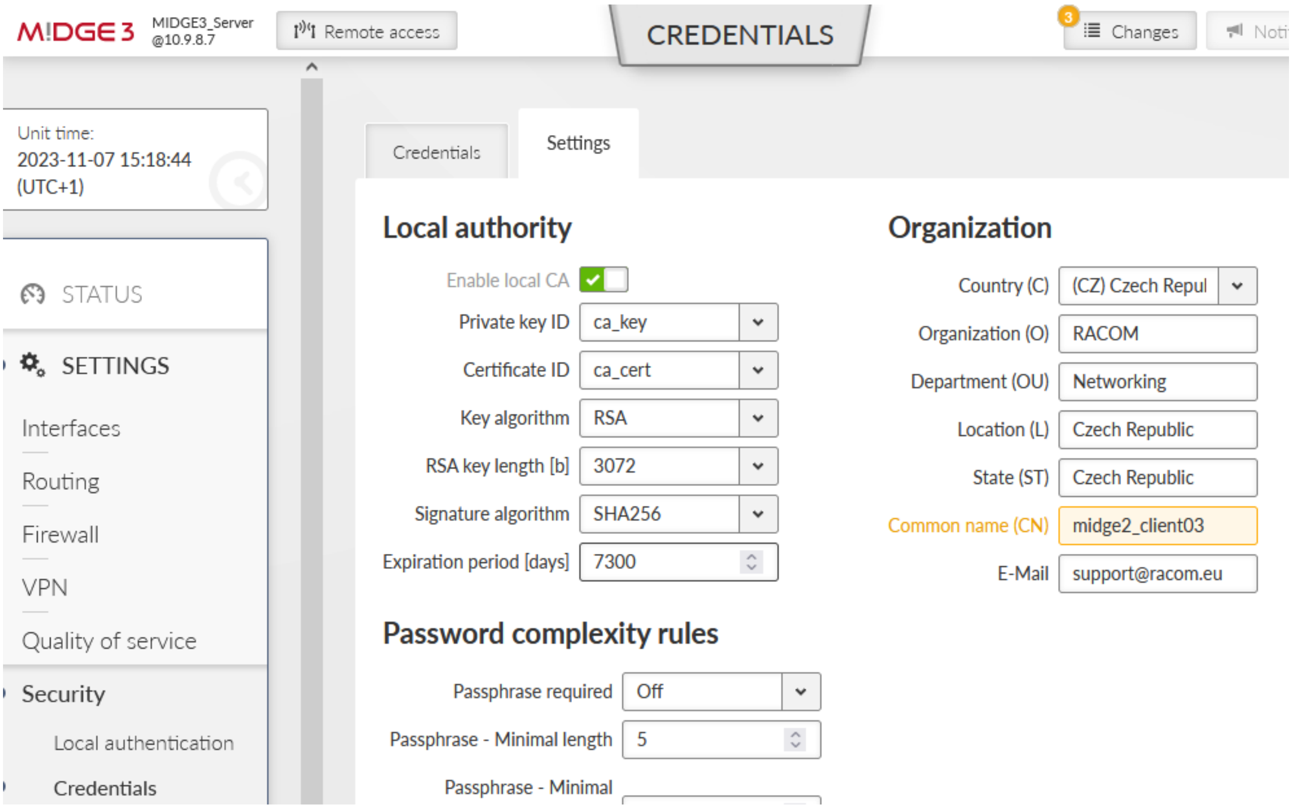
☑ Note: (empty)
Certificate Common Name: midge2_client03
Address assignment: Dynamic

Client routes

☑ Note: (empty)
Destination IP / Destination mask: 192.168.4.0/24

Fig. 49: MIDGE3_Server – adding MIDGE2_Client03 client

Go to the SETTINGS > Security > Credentials > Settings menu. Change the Common name (CN) parameter to “midge2_client03” so that newly created certificate has a correct CN.



Unit time:
2023-11-07 15:18:44
(UTC+1)

Local authority

Enable local CA: ☑
Private key ID: ca_key
Certificate ID: ca_cert
Key algorithm: RSA
RSA key length [b]: 3072
Signature algorithm: SHA256
Expiration period [days]: 7300

Password complexity rules

Passphrase required: Off
Passphrase - Minimal length: 5
Passphrase - Minimal

Organization

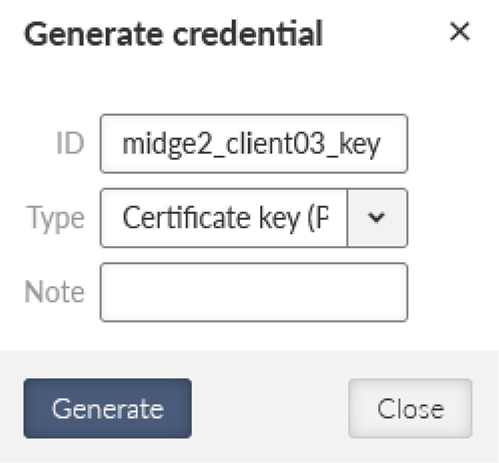
Country (C): (CZ) Czech Repul
Organization (O): RACOM
Department (OU): Networking
Location (L): Czech Republic
State (ST): Czech Republic
Common name (CN): midge2_client03
E-Mail: support@racom.eu

Fig. 50: MIDGE3_Server Common name change

Commit the changes to the OpenVPN.

Go to the SETTINGS > Security > Credentials menu. Generate a private key and a certificate for this M!DGE2 unit.

midge2_client03_key Certificate key (PRI)



Generate credential

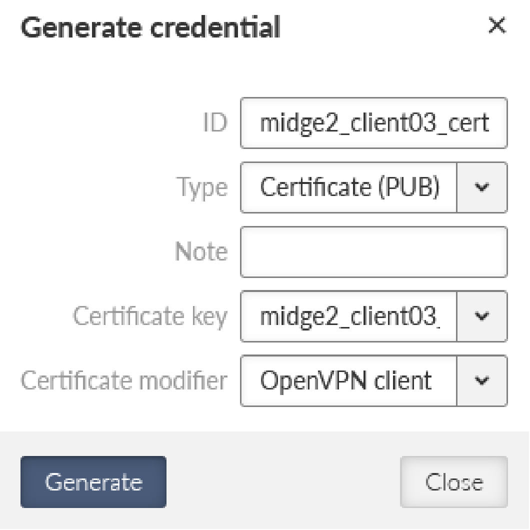
ID: midge2_client03_key

Type: Certificate key (P

Note:

Fig. 51: MIDGE3_Server – generating MIDGE2_Client03 private key

midge2_client03_cert Certificate (PUB)



Generate credential

ID: midge2_client03_cert

Type: Certificate (PUB)

Note:

Certificate key: midge2_client03

Certificate modifier: OpenVPN client

Fig. 52: MIDGE3_Server – generating MIDGE2_Client03 OpenVPN certificate

Download them into your PC together with the OpenVPN TLS Protection key (PRI) “ta_key” and CA chain (PUB) “ca_cert”.

Change the MIDGE3_Server’s Common name back to “midge3_server” and commit changes.

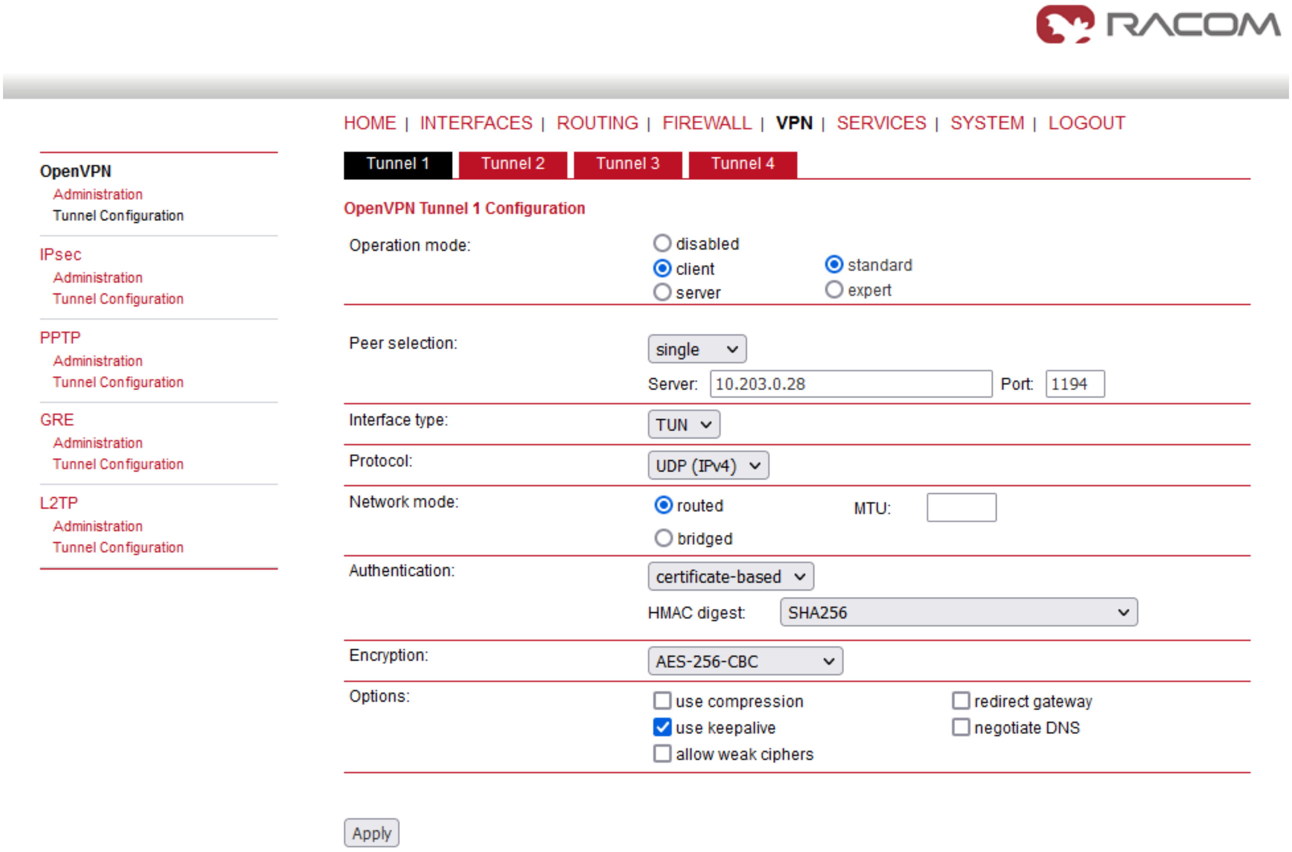
3.1.2. MIDGE2_Client03

The application note is not intended for M!DGE2 complete tutorial, but if you configure it from the factory settings, do not forget to:

- set the Ethernet IP to 192.168.4.1/24 and change its DHCP address range

- configure the Mobile interface correctly to suit your APN
  - our WWAN IP address will be 10.203.3.33/17
- set the hostname to “MIDGE2_Client03”
- set the correct NTP server and Time zone
- set the correct Common Name (CN) within the Keys&Certificates to be “midge2_client03”

Go to the VPN > OpenVPN > Tunnel Configuration menu.



OpenVPN Tunnel 1 Configuration

Operation mode:	○ disabled ● client ○ server	● standard ○ expert
Peer selection:	single Server: 10.203.0.28	Port: 1194
Interface type:	TUN	
Protocol:	UDP (IPv4)	
Network mode:	● routed ○ bridged	MTU:
Authentication:	certificate-based HMAC digest: SHA256	
Encryption:	AES-256-CBC	
Options:	□ use compression ☑ use keepalive □ allow weak ciphers	□ redirect gateway □ negotiate DNS

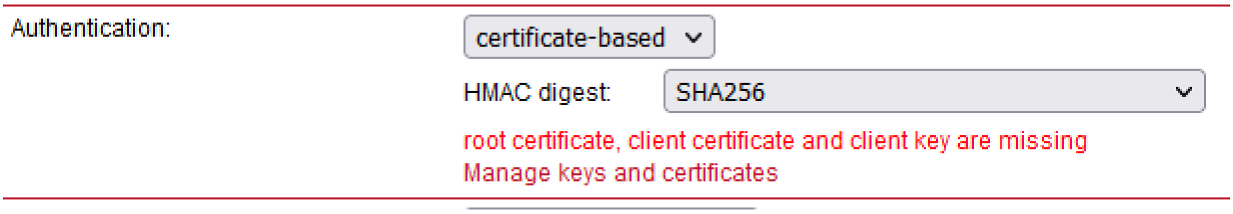
Fig. 53: MIDGE2_Client03 OpenVPN settings

Set the Operation mode to “client”.

- Set the Server IP to 10.203.0.28
- Uncheck the “use compression” and “allow weak ciphers” options
- Check the “use keepalive” option

Other parameters should stay in default and shall follow the server’s setup.

Apply the changes. Click on the Error message within the Authentication part of the menu.



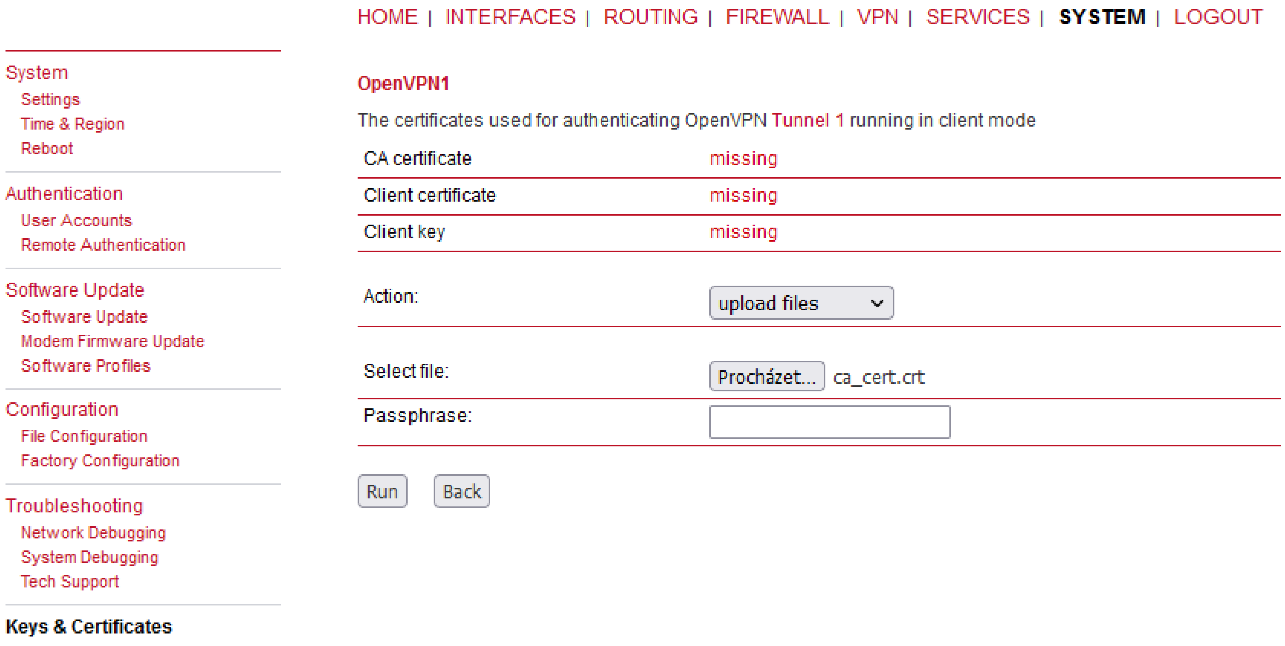
Authentication: certificate-based

HMAC digest: SHA256

root certificate, client certificate and client key are missing

Manage keys and certificates

Fig. 54: Upload the required certificates and keys

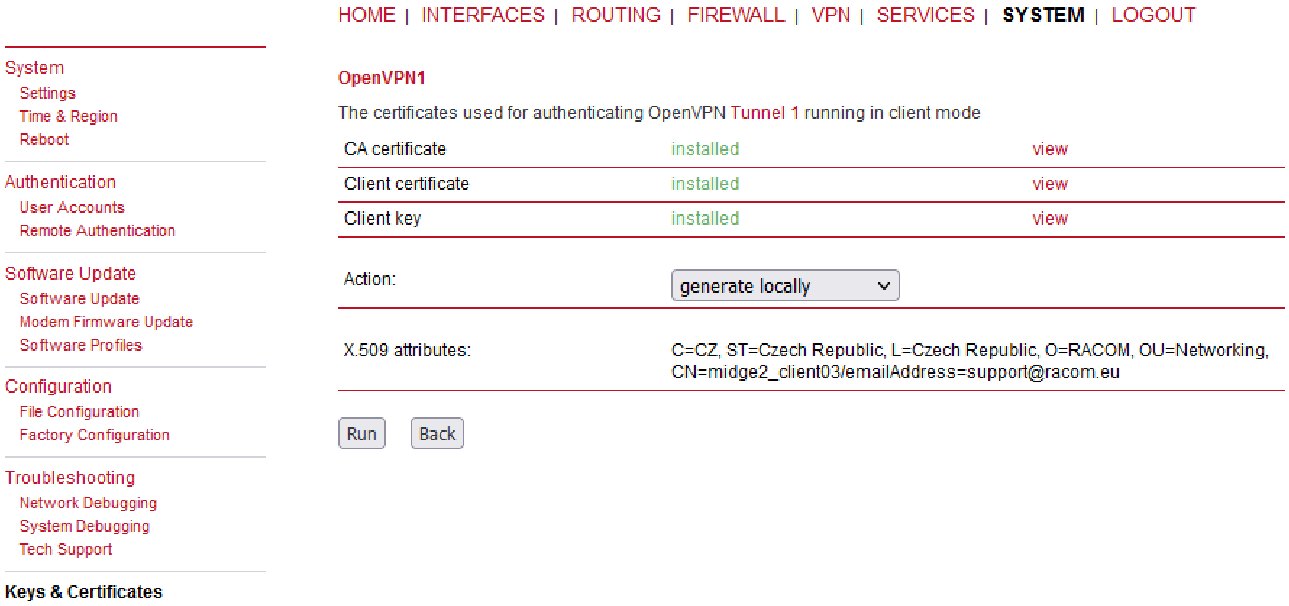


OpenVPN1

The certificates used for authenticating OpenVPN Tunnel 1 running in client mode

CA certificate	missing
Client certificate	missing
Client key	missing
Action:	upload files
Select file:	Procházet... ca_cert.crt
Passphrase:	

Fig. 55: MIDGE2_Client03 Keys & Certificates upload



OpenVPN1

The certificates used for authenticating OpenVPN Tunnel 1 running in client mode

CA certificate	installed	view
Client certificate	installed	view
Client key	installed	view
Action:	generate locally	
X.509 attributes:	C=CZ, ST=Czech Republic, L=Czech Republic, O=RACOM, OU=Networking, CN=midge2_client03/emailAddress=support@racom.eu	

Fig. 56: MIDGE2_Client03 Keys and Certificates uploaded successfully

But what about the OpenVPN TLS Protection key (PRI)? (TLS-Auth, ta.key)

This is not supported within the M!DGE2 web interface, but can be done different way.

Go to the VPN > OpenVPN > Tunnel configuration and click on the **Download** button. Save the ZIP file to your PC and unzip it. Open the openvpn-expert.conf file using a text editor and add one new line:

tls-auth ta.key 1

A complete file should be the same, or similar to:

```
client
remote 10.203.0.28 1194
proto udp
verb 3
auth-retry nointeract
nobind
auth-nocache
sndbuf 0
rcvbuf 0
ipchange "/etc/openvpn/tunnel0-ipchange"
passtos
up-restart
auth-retry nointeract
remap-usr1 SIGHUP
resolv-retry infinite
persist-key
persist-tun
auth SHA256
cipher AES-256-CBC
keepalive 10 60
ca ca.crt
cert openvpn-expert.crt
key openvpn-expert.key
dev tun
tls-auth ta.key 1
```

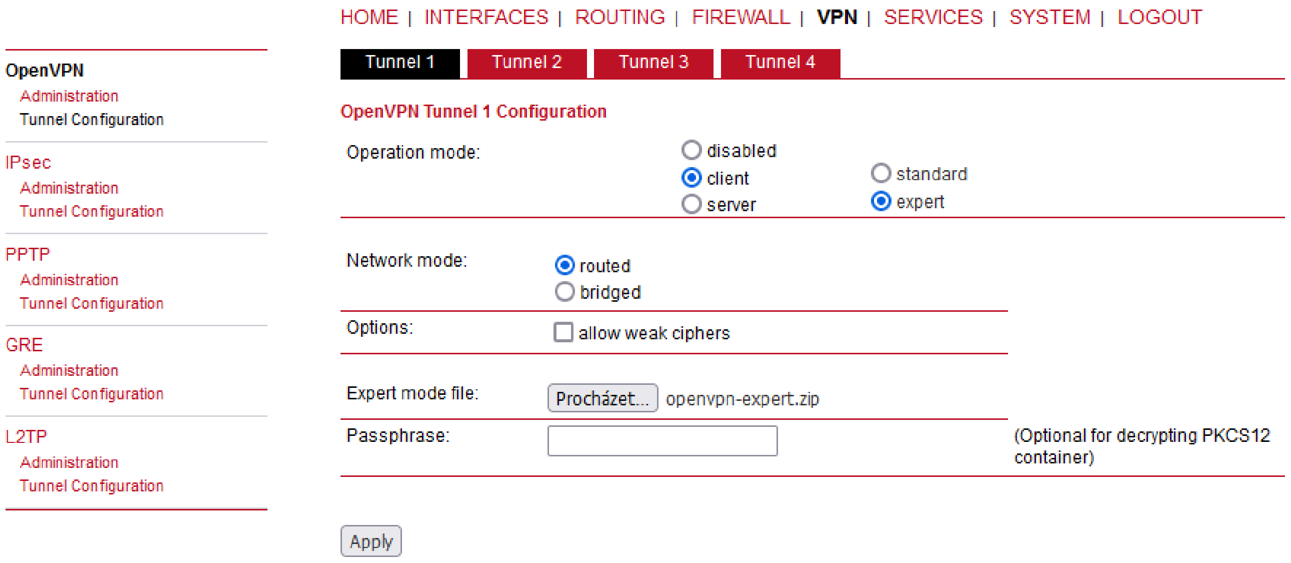
Now, copy & paste the OpenVPN TLS Protection key (PRI) file to the same folder/directory with the rest of the files within the Expert file. Name it “ta.key”. Zip all the files to the openvpn-expert.zip file again.

So, the differences are:

- the configuration file includes the tls-auth ta.key 1 directive
- the ZIP includes the ta.key file

Go back to the MIDGE2_Client03’s web interface and VPN > OpenVPN > Tunnel Configuration menu. **Erase** the current OpenVPN settings.

Set the Operation mode to “client” again, but set the “expert mode” now.



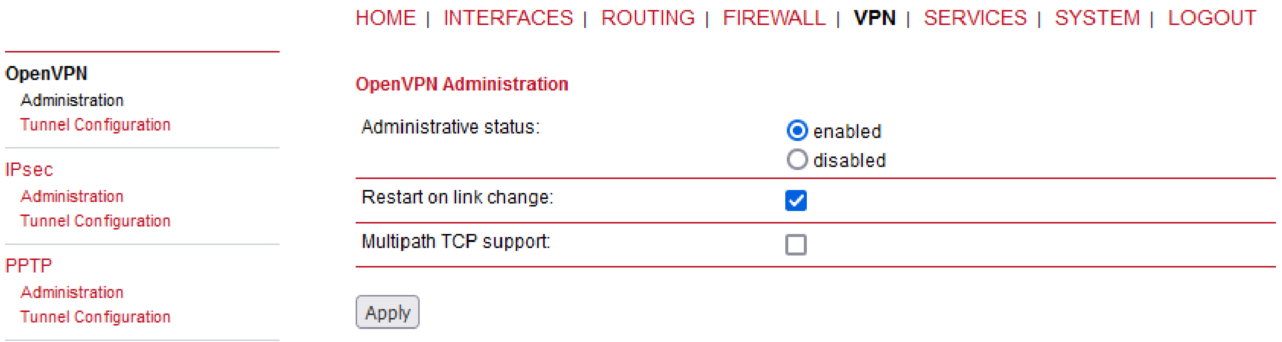
OpenVPN Tunnel 1 Configuration

Operation mode:	○ disabled ● client ○ server	○ standard ● expert
Network mode:	● routed ○ bridged	
Options:	□ allow weak ciphers	
Expert mode file:	Procházet... openvpn-expert.zip	
Passphrase:		(Optional for decrypting PKCS12 container)

Fig. 57: MIDGE2_Client03 OpenVPN expert file

Keep the “Routed” option set and uncheck the “allow weak ciphers” option. Select the updated ZIP file and **apply** the changes.

Go to the OpenVPN’s Administration menu and enable the tunnel.



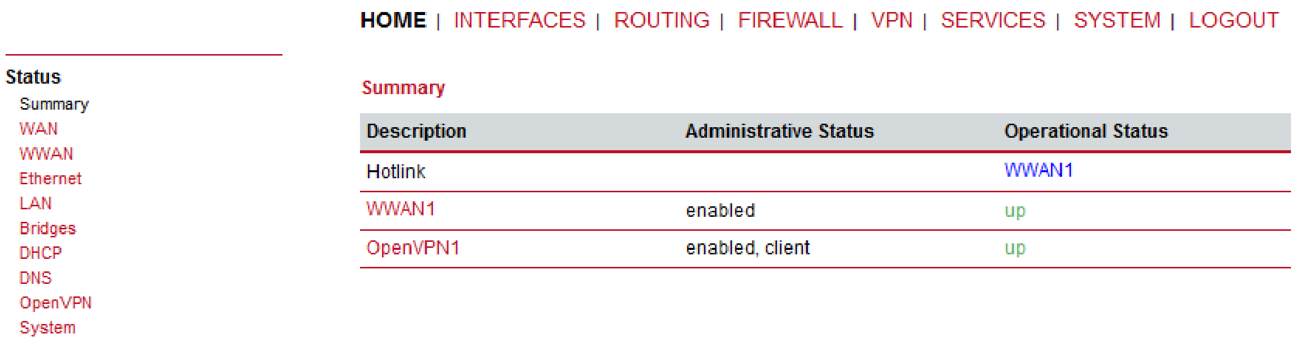
OpenVPN Administration

Administrative status:	● enabled ○ disabled
Restart on link change:	☑
Multipath TCP support:	□

Fig. 58: MIDGE2_Client03 OpenVPN Administration

3.1.3. Diagnostics

Go to the HOME menu and check if the client gets connected, or not.

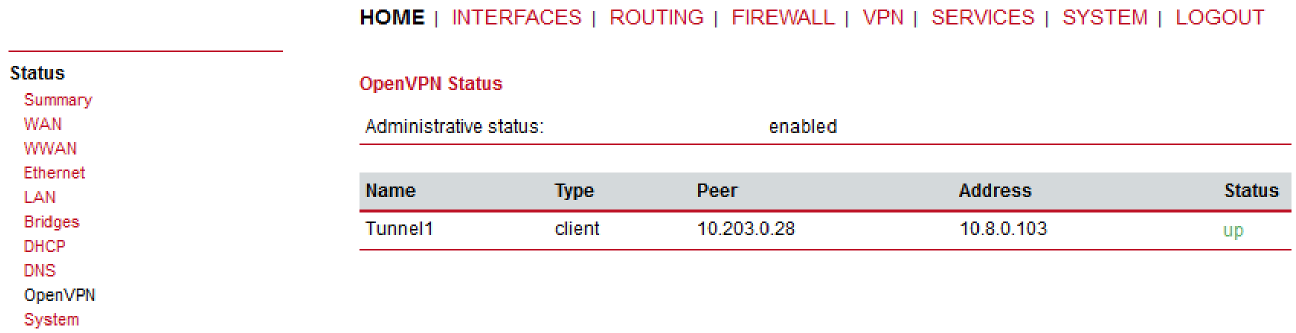

HOME | INTERFACES | ROUTING | FIREWALL | VPN | SERVICES | SYSTEM | LOGOUT
Status
Summary
WAN
WWAN
Ethernet
LAN
Bridges
DHCP
DNS
OpenVPN
System

Summary

Description	Administrative Status	Operational Status
Hotlink		WWAN1
WWAN1	enabled	up
OpenVPN1	enabled, client	up

Fig. 59: MIDGE2_Client03 successfully connected

If you go to the OpenVPN details, you should e.g. see the assigned OpenVPN IP address.


HOME | INTERFACES | ROUTING | FIREWALL | VPN | SERVICES | SYSTEM | LOGOUT
Status
Summary
WAN
WWAN
Ethernet
LAN
Bridges
DHCP
DNS
OpenVPN
System

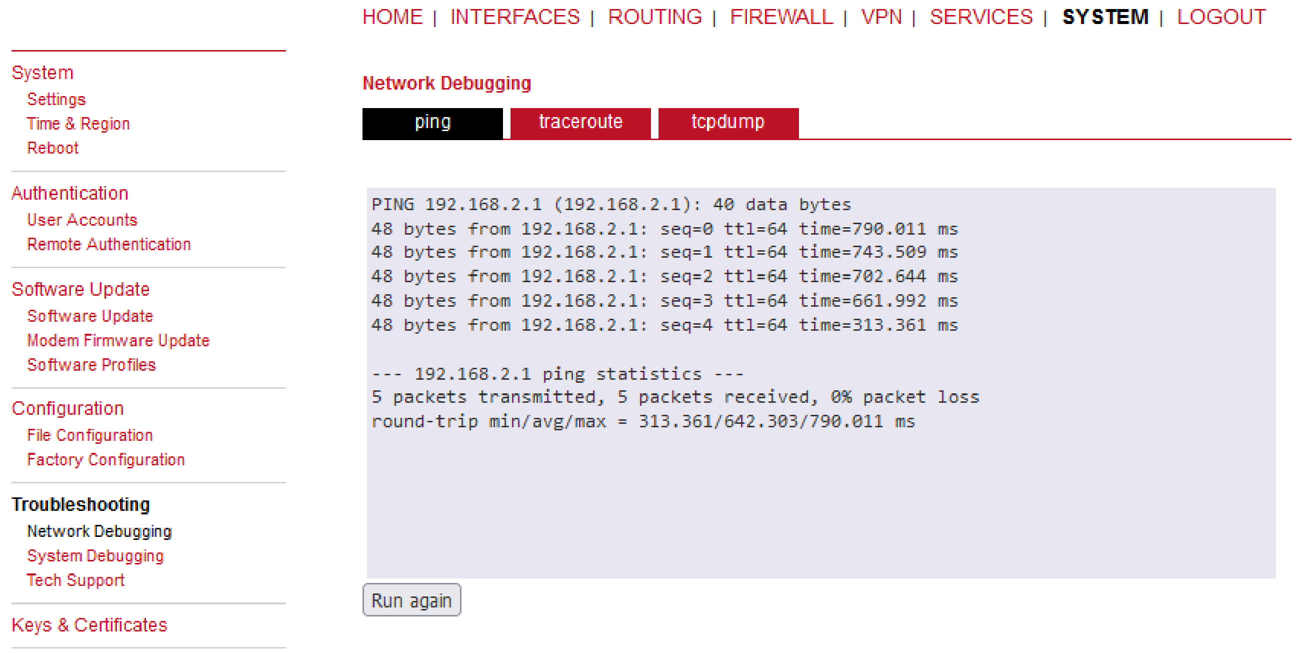
OpenVPN Status

Administrative status: enabled

Name	Type	Peer	Address	Status
Tunnel1	client	10.203.0.28	10.8.0.103	up

Fig. 60: MIDGE2_Client03 Detailed status

You can go to the SYSTEM > Network debugging > ping menu and try to ping 192.168.1.1 IP address of the server or 192.168.2.1 MIDGE3_Client02 IP address.

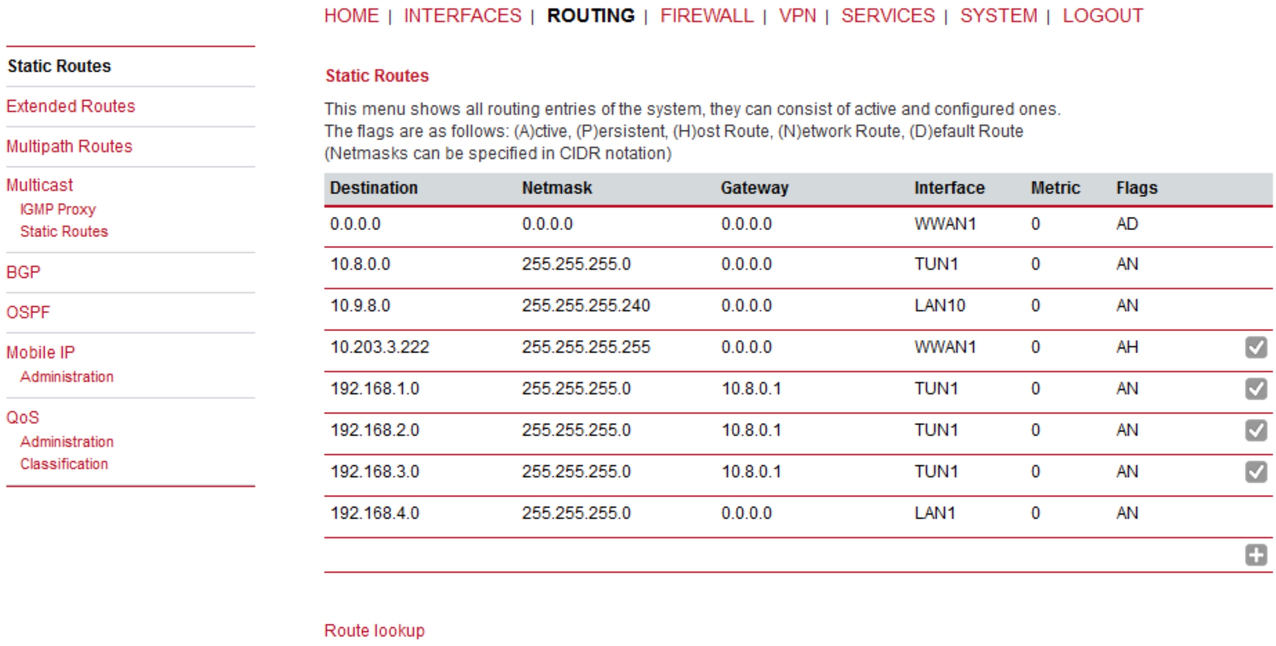


```
PING 192.168.2.1 (192.168.2.1): 40 data bytes
48 bytes from 192.168.2.1: seq=0 ttl=64 time=790.011 ms
48 bytes from 192.168.2.1: seq=1 ttl=64 time=743.509 ms
48 bytes from 192.168.2.1: seq=2 ttl=64 time=702.644 ms
48 bytes from 192.168.2.1: seq=3 ttl=64 time=661.992 ms
48 bytes from 192.168.2.1: seq=4 ttl=64 time=313.361 ms

--- 192.168.2.1 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 313.361/642.303/790.011 ms
```

Fig. 61: MIDGE2_Client03 pinging another client

You can check the routes within the ROUTING > Static routes menu. You should see all the remote networks accessible via TUN interface.



Static Routes

This menu shows all routing entries of the system, they can consist of active and configured ones.
The flags are as follows: (A)ctive, (P)ersistent, (H)ost Route, (N)etwork Route, (D)efault Route
(Netmasks can be specified in CIDR notation)

Destination	Netmask	Gateway	Interface	Metric	Flags	
0.0.0.0	0.0.0.0	0.0.0.0	WWAN1	0	AD	
10.8.0.0	255.255.255.0	0.0.0.0	TUN1	0	AN	
10.9.8.0	255.255.255.240	0.0.0.0	LAN10	0	AN	
10.203.3.222	255.255.255.255	0.0.0.0	WWAN1	0	AH	☑
192.168.1.0	255.255.255.0	10.8.0.1	TUN1	0	AN	☑
192.168.2.0	255.255.255.0	10.8.0.1	TUN1	0	AN	☑
192.168.3.0	255.255.255.0	10.8.0.1	TUN1	0	AN	☑
192.168.4.0	255.255.255.0	0.0.0.0	LAN1	0	AN	

Fig. 62: MIDGE2_Client03 Routing

Last, but not least, you can also download a Techsupport package and check the logs for any issues. Do it from the SYSTEM > Tech Support menu.

Logs can also be viewed online if you connect to the M!DGE2 SSH. You can also do advanced debugging directly from the Linux command line in M!DGE2.

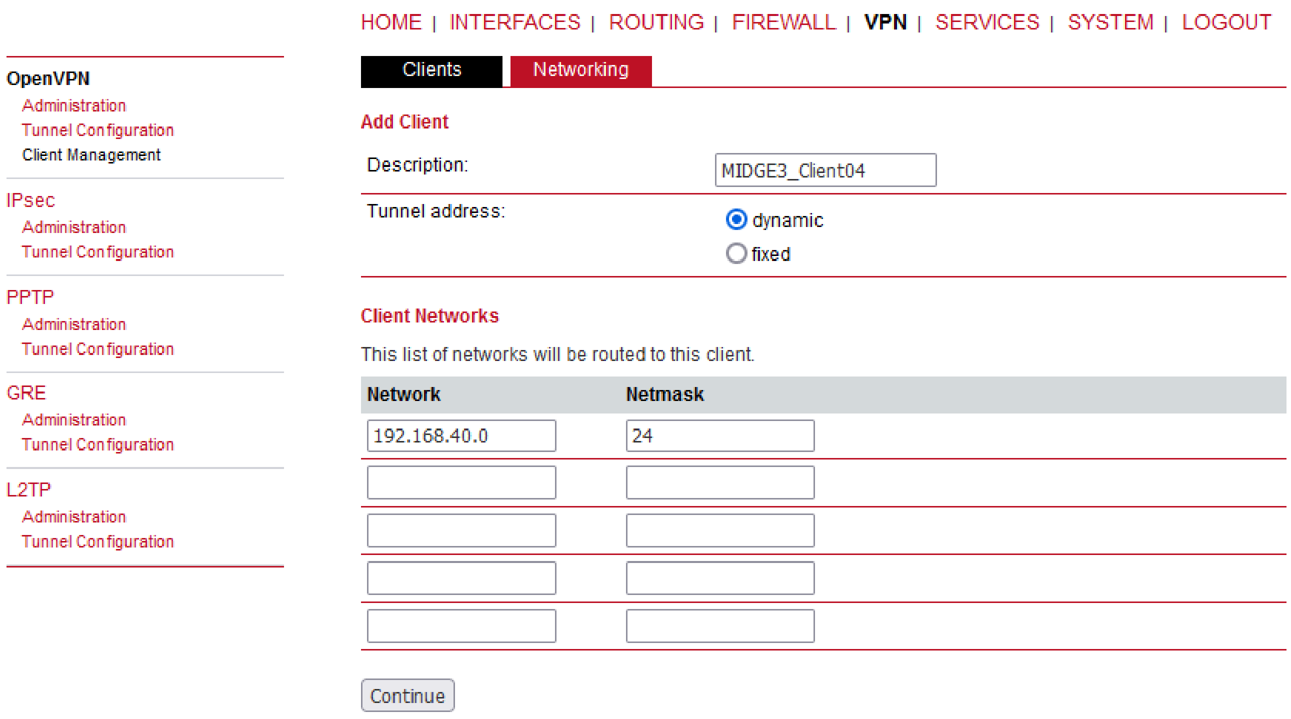
3.2. Adding M!DGE3/RipEX2 to the existing M!DGE2 network

Requirement can also be the other way. You have a working M!DGE2 network utilizing the OpenVPN connections. Now you bought M!DGE3 as well and just need to add a new client.

The application note will not show the step-by-step configuration with all the screenshots, but will help you with required steps. Most of the following steps are covered within the application note so just go through previous examples if you encounter any obstacle.

- Configure your M!DGE3/RipEX2 to connect to your APN and configure all other required parameters such as NTP, Network Interfaces, Common name, …
- Add a new client in your M!DGE2 server and set its network(s)
- Generate valid credentials (Keys & certificates) in your M!DGE2 server (or e.g. in some external CA)
- Upload/Add the client’s key, client’s certificate, CA chain and if required, the TLS-Auth/TLS-Crypt key to your M!DGE3/RipEX2
- Configure your M!DGE3/RipEX2’s OpenVPN with valid credentials and correct parameters to suit your server settings

Several screenshots to help you.



Add Client

Description: MIDGE3_Client04

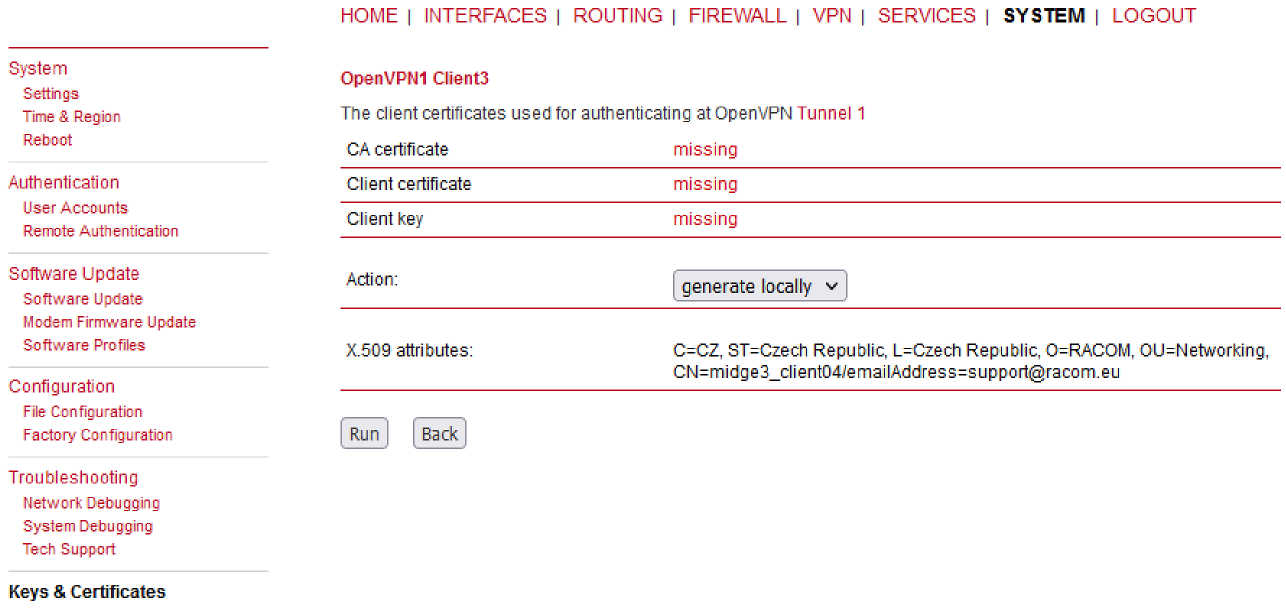
Tunnel address: ● dynamic ○ fixed

Client Networks

This list of networks will be routed to this client.

Network	Netmask
192.168.40.0	24

Fig. 63: Adding a new OpenVPN client in M!DGE2



OpenVPN1 Client3

The client certificates used for authenticating at OpenVPN Tunnel 1

CA certificate	missing
Client certificate	missing
Client key	missing

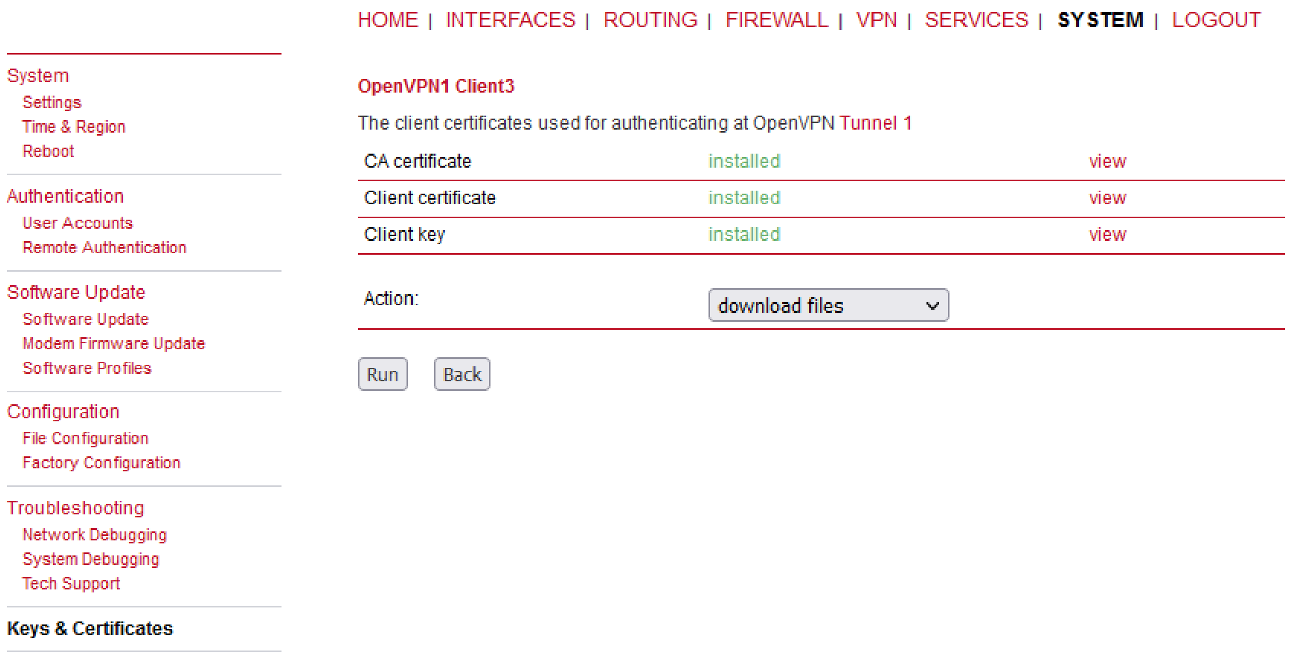
Action: generate locally

X.509 attributes: C=CZ, ST=Czech Republic, L=Czech Republic, O=RACOM, OU=Networking, CN=midge3_client04/emailAddress=support@racom.eu

Fig. 64: Generating new certificates in M!DGE2

Note

You may also change the Common name of the M!DGE2 server before generating the client's certificate even though there is no "Certificate modifier" option in M!DGE2 so the "Verify peer certificate usage" parameter in M!DGE3/RipEX2 should probably be disabled.



OpenVPN1 Client3

The client certificates used for authenticating at OpenVPN Tunnel 1

CA certificate	installed	view
Client certificate	installed	view
Client key	installed	view

Action: download files

Fig. 65: M!DGE2 downloading Keys & certificates

M!DGE2 OpenVPN server only supports a “Network topology” option **“Net /30**”. This is not configurable in clients, but is driven by the server. Just keep this in mind while interconnecting various devices within one OpenVPN network. The network is either “Net /30” or more up-to-date option “Subnet”. It cannot be combined. M!DGE2 and M!DGE3/RipEX2 support both options, but M!DGE2 cannot set it for the Server, it can only adapt to it as a client.

If you have any issues, contact our technical support at *support@racom.eu*[2].

[2] mailto:support@racom.eu

3.3. M!DGE2 OpenVPN server migration to M!DGE3

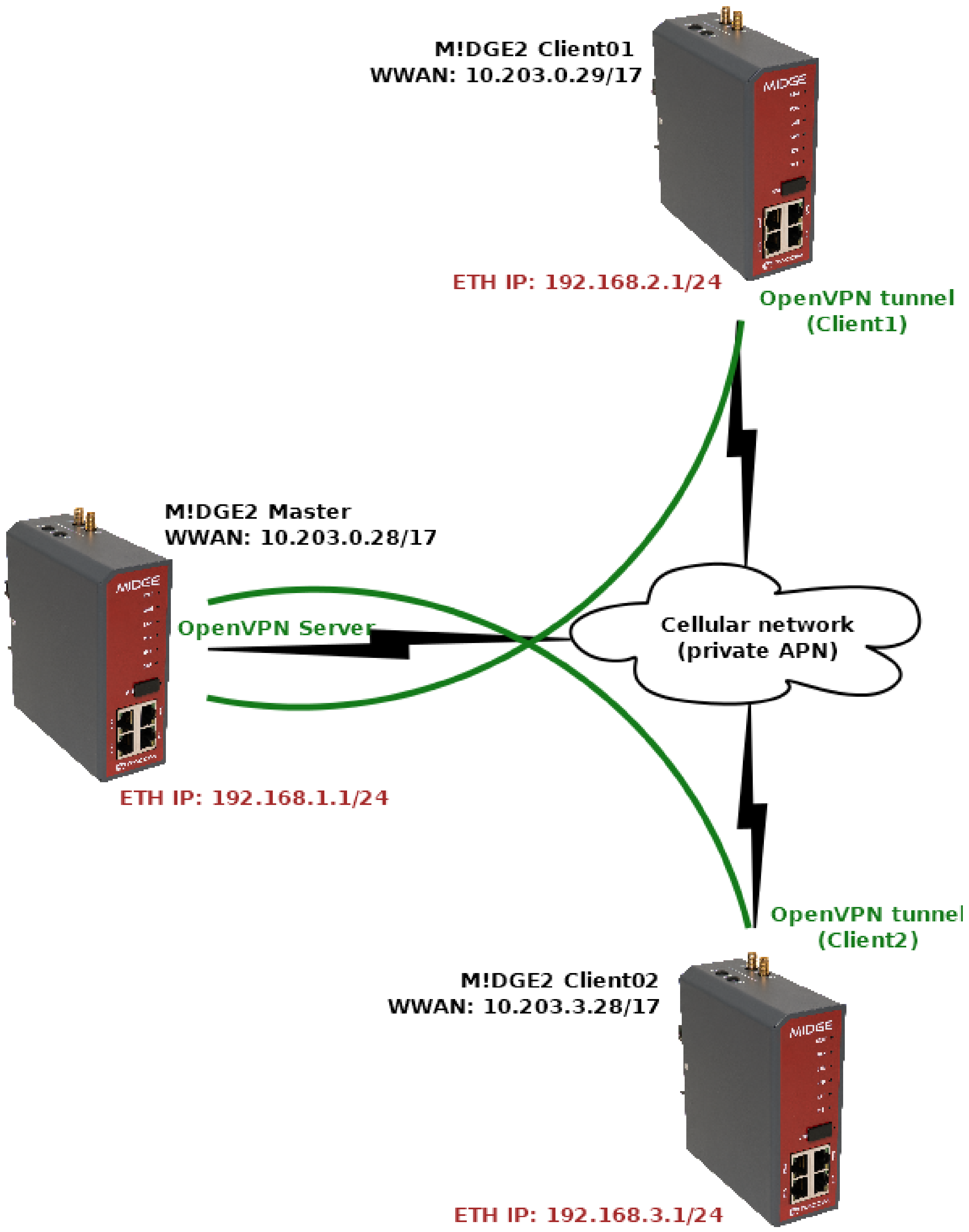


Fig. 66: Simple OpenVPN TUN/Routed topology – M!DGE2 only

Within this example, we will replace the M!DGE2 Master (OpenVPN server) unit by a new M!DGE3 router. We will not touch any configuration or credentials in remote OpenVPN clients. We just swap M!DGE2 by M!DGE3 in the central location.

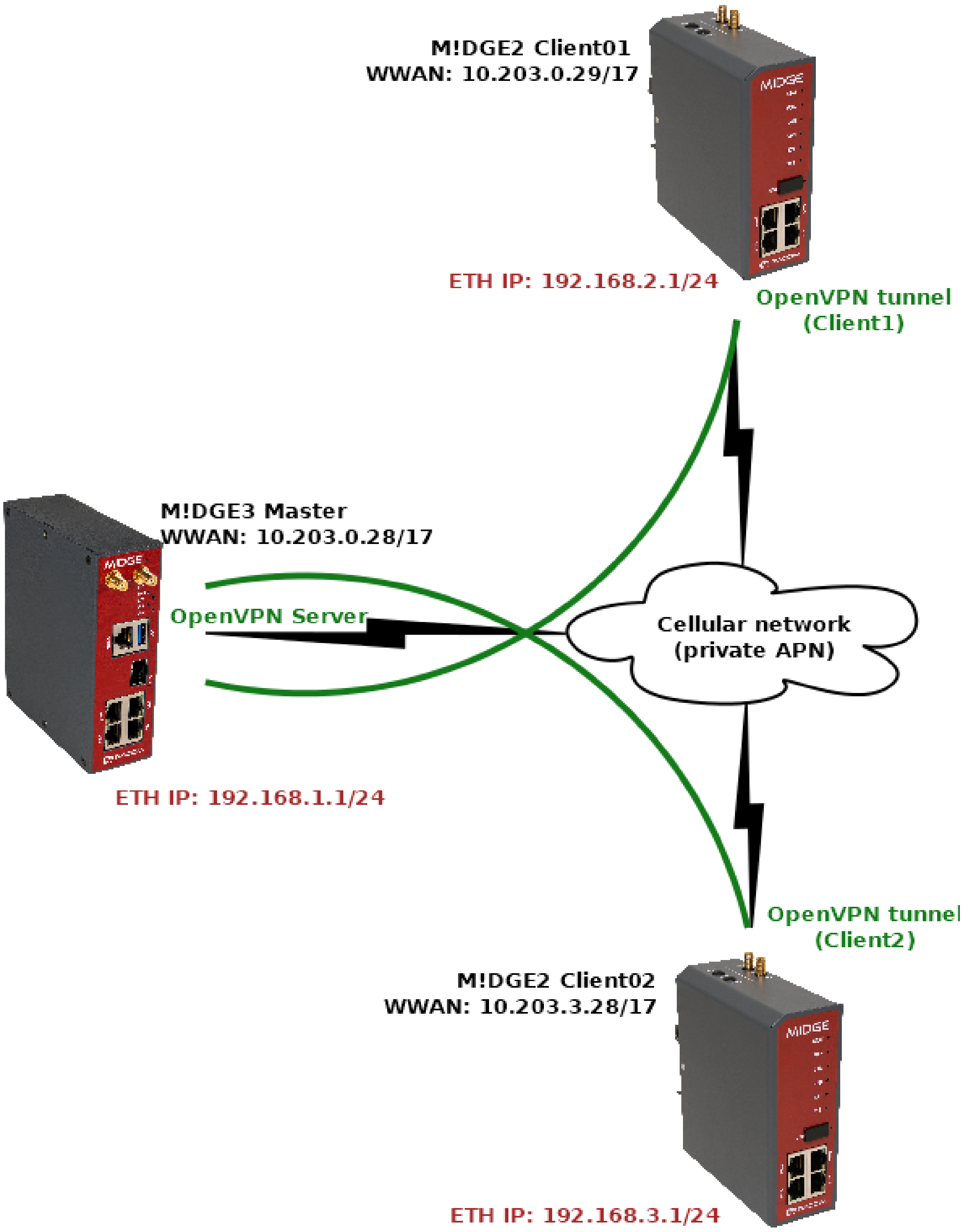
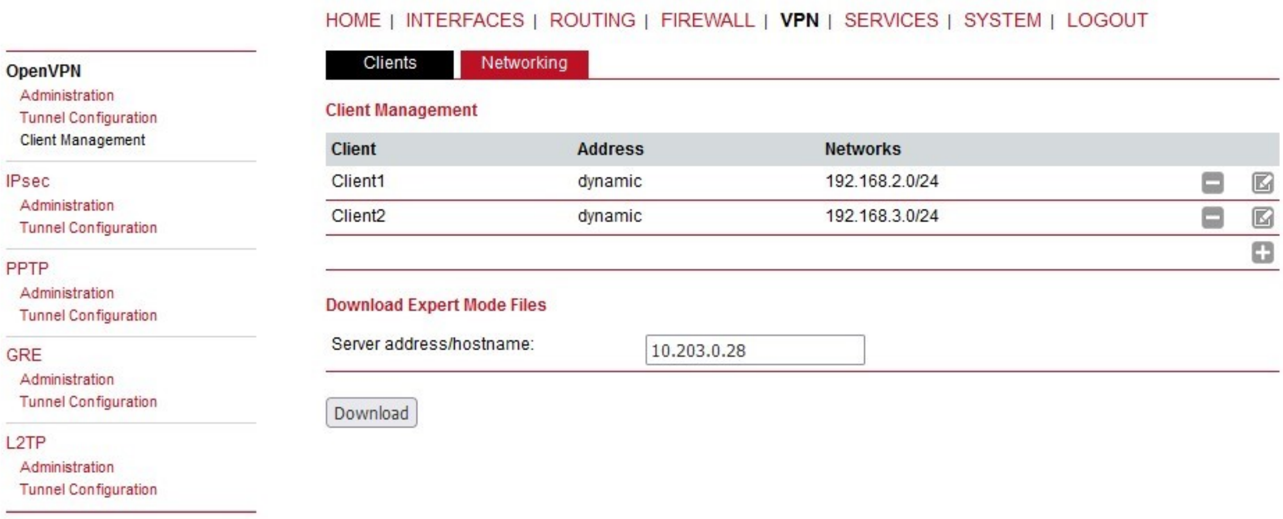


Fig. 67: The same topology, but with M!DGE3 router instead of the central M!DGE2

First of all, we suggest to download complete configuration files from all the units in your network, or at least the M!DGE2 Master so you do have backup files for future needs, or in case of any issues.

3.3.1. M!DGE2 Master (legacy OpenVPN server)

Except the configuration file (within SYSTEM > File configuration menu), we also suggest to backup/download an expert file for each OpenVPN client in case of swapping any client M!DGE2 by M!DGE3 as well (VPN > OpenVPN > Client management).



Client Management

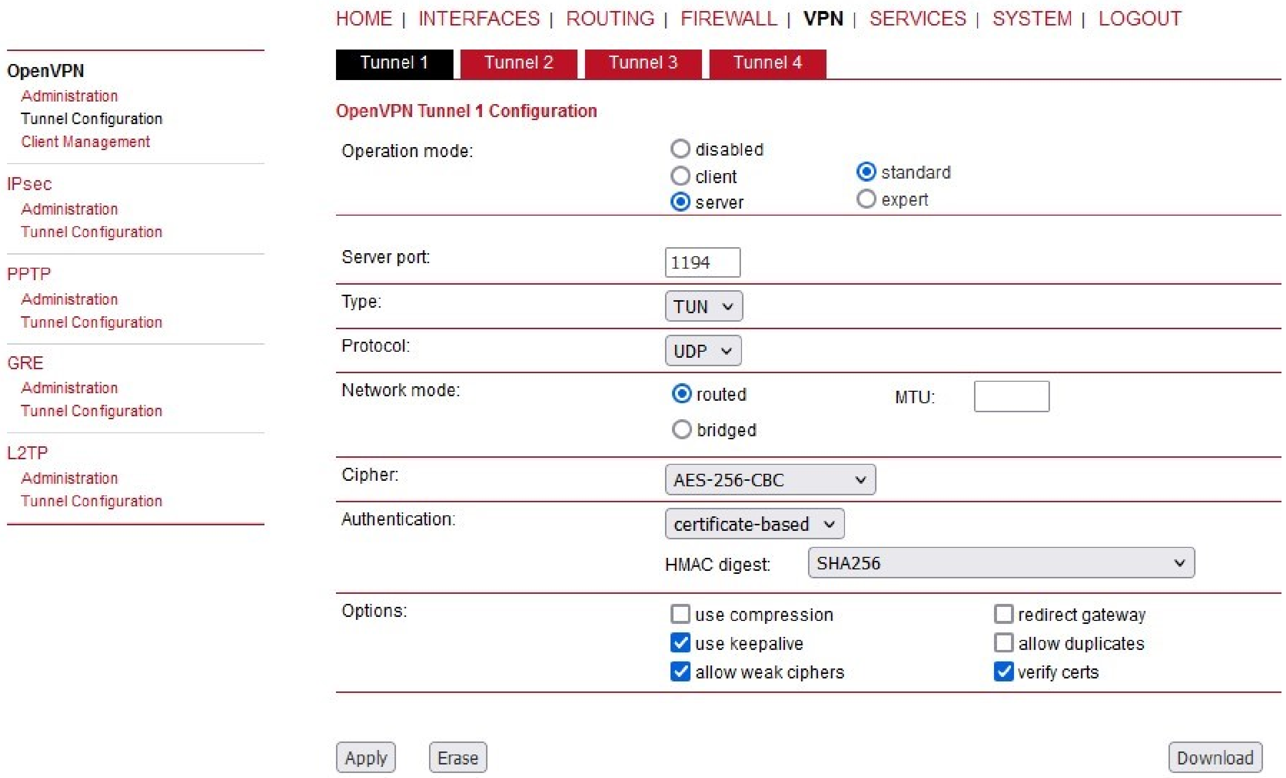
Client	Address	Networks
Client1	dynamic	192.168.2.0/24
Client2	dynamic	192.168.3.0/24

Download Expert Mode Files

Server address/hostname: 10.203.0.28

Fig. 68: M!DGE2 Master (server) OpenVPN client management

Keep an eye on our server’s OpenVPN settings, because you will need to manually set matching parameters in your new M!DGE3 server.



OpenVPN Tunnel 1 Configuration

Operation mode: ○ disabled ○ client ● server; ● standard ○ expert

Server port: 1194

Type: TUN

Protocol: UDP

Network mode: ● routed ○ bridged; MTU:

Cipher: AES-256-CBC

Authentication: certificate-based

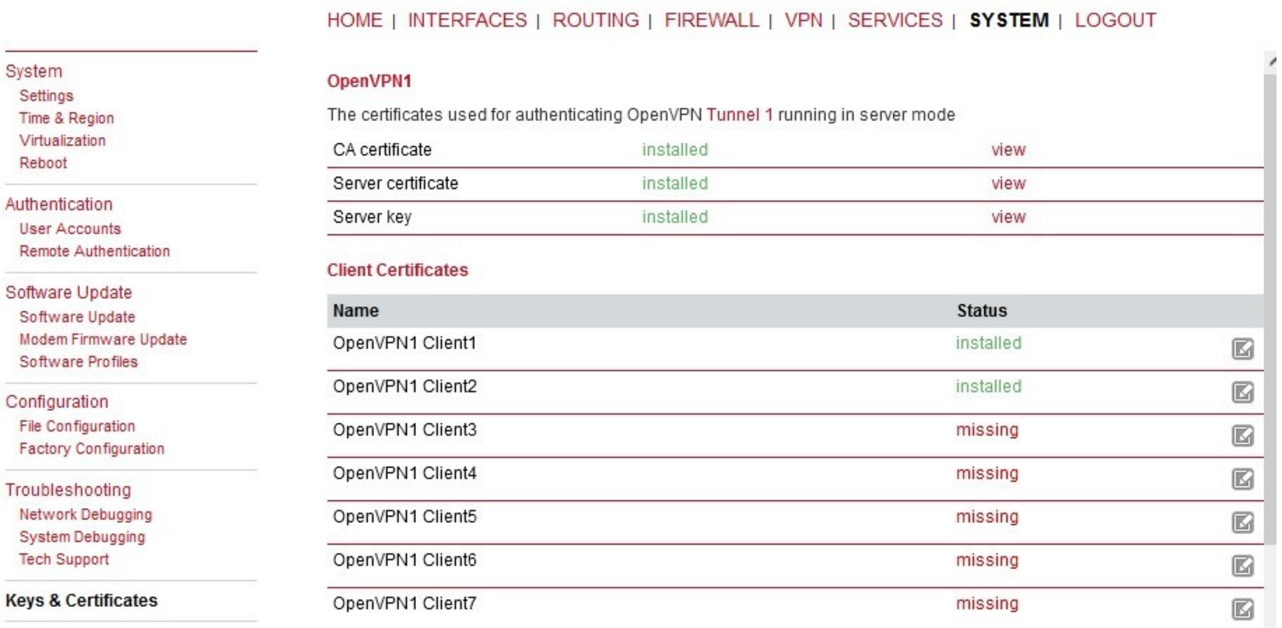
HMAC digest: SHA256

Options: □ use compression ☑ use keepalive ☑ allow weak ciphers □ redirect gateway □ allow duplicates ☑ verify certs

Fig. 69: M!DGE2 Master (server) OpenVPN settings

Last, but not least, check what are the Common Names (CNs) of your clients, because they probably do not match the set names within the Client management. E.g. in our example, the 1st client, called “Client1” within the GUI management, has the CN equal to “client0”. The CN is important due to control mechanism in M!DGE3. Mark all the CNs in some document.

Within M!DGE2 GUI, you can find them in the SYSTEM > Keys & Certificates menu. Open the OpenVPN details there.



OpenVPN1

The certificates used for authenticating OpenVPN Tunnel 1 running in server mode

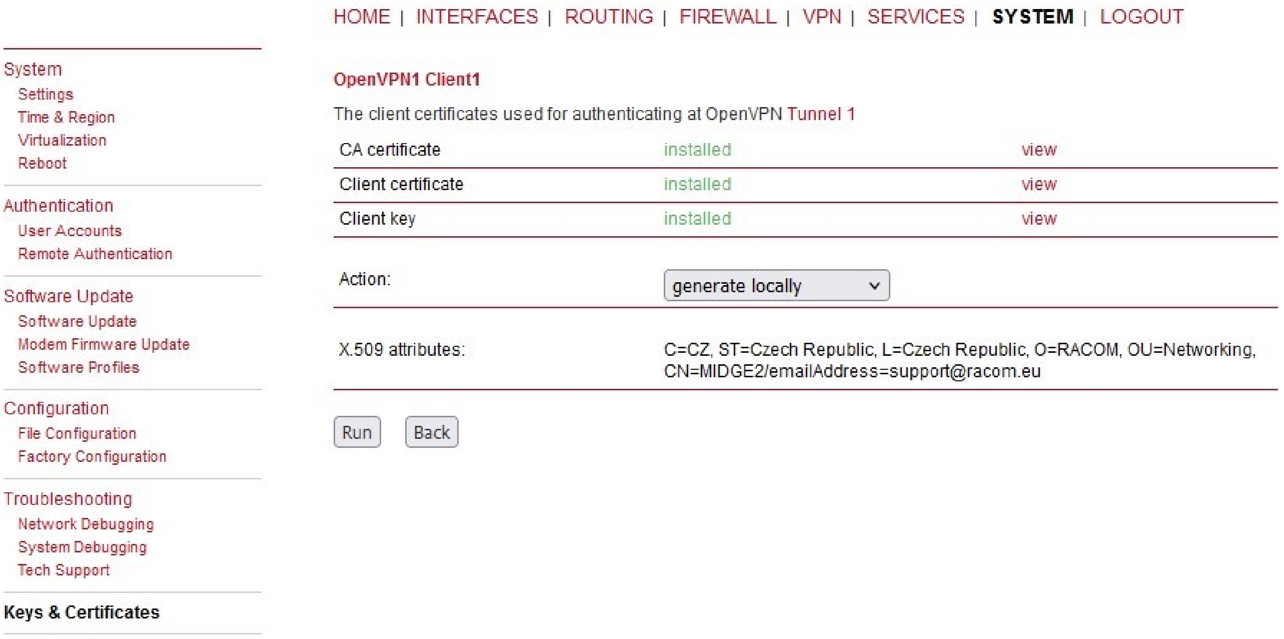
CA certificate	installed	view
Server certificate	installed	view
Server key	installed	view

Client Certificates

Name	Status
OpenVPN1 Client1	installed
OpenVPN1 Client2	installed
OpenVPN1 Client3	missing
OpenVPN1 Client4	missing
OpenVPN1 Client5	missing
OpenVPN1 Client6	missing
OpenVPN1 Client7	missing

Fig. 70: M!DGE2 Master (server) OpenVPN certificates

Open each client’s details.



OpenVPN1 Client1

The client certificates used for authenticating at OpenVPN Tunnel 1

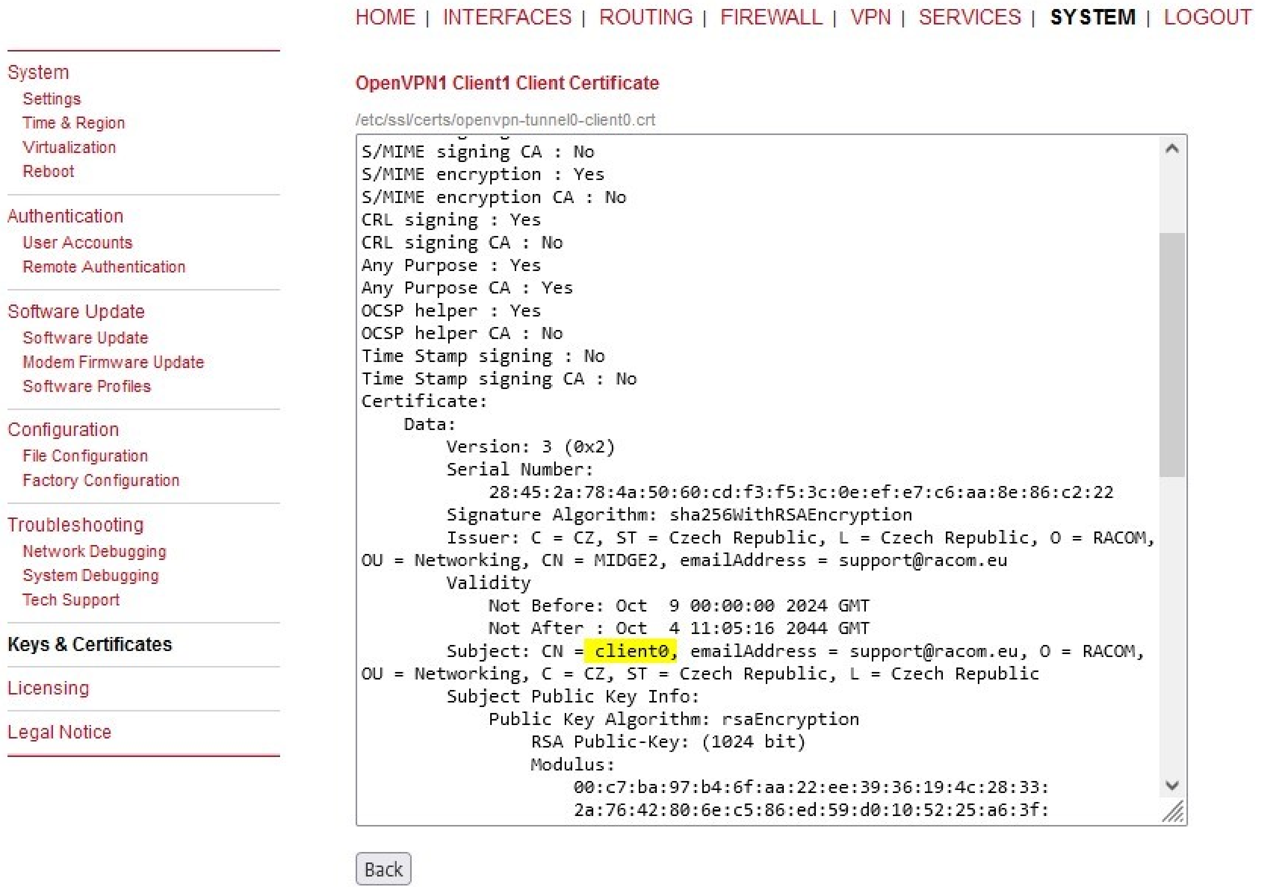
CA certificate	installed	view
Client certificate	installed	view
Client key	installed	view

Action: generate locally

X.509 attributes: C=CZ, ST=Czech Republic, L=Czech Republic, O=RACOM, OU=Networking, CN=MIDGE2/emailAddress=support@racom.eu

Fig. 71: Client1 OpenVPN certificates

Now, click on the “view” button next to the “Client certificate” line.



OpenVPN1 Client1 Client Certificate

/etc/ssl/certs/openvpn-tunnel0-client0.crt

```
S/MIME signing CA : No
S/MIME encryption : Yes
S/MIME encryption CA : No
CRL signing : Yes
CRL signing CA : No
Any Purpose : Yes
Any Purpose CA : Yes
OCSP helper : Yes
OCSP helper CA : No
Time Stamp signing : No
Time Stamp signing CA : No
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            28:45:2a:78:4a:50:60:cd:f3:f5:3c:0e:ef:e7:c6:aa:8e:86:c2:22
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: C = CZ, ST = Czech Republic, L = Czech Republic, O = RACOM,
OU = Networking, CN = MIDGE2, emailAddress = support@racom.eu
        Validity
            Not Before: Oct  9 00:00:00 2024 GMT
            Not After : Oct  4 11:05:16 2044 GMT
        Subject: CN = client0, emailAddress = support@racom.eu, O = RACOM,
OU = Networking, C = CZ, ST = Czech Republic, L = Czech Republic
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                RSA Public-Key: (1024 bit)
                Modulus:
                    00:c7:ba:97:b4:6f:aa:22:ee:39:36:19:4c:28:33:
                    2a:76:42:80:6e:c5:86:ed:59:d0:10:52:25:a6:3f:
```

Fig. 72: Client1’s CN

Mark down the correct CN – “client0” in our example. The 2nd client has “client1”.

Unzip all the downloaded files so we can import them into M!DGE3 Master (new OpenVPN server) now.

3.3.2. M!DGE3 Master (new OpenVPN server)

Start with upgrading the unit into the latest FW. Once completed, do the factory reset and start the configuration from the beginning.

Note

We suggest setting the M!DGE3 Master without the antenna connected or/and without the SIM card so that your currently used M!DGE2 server operation is not broken.

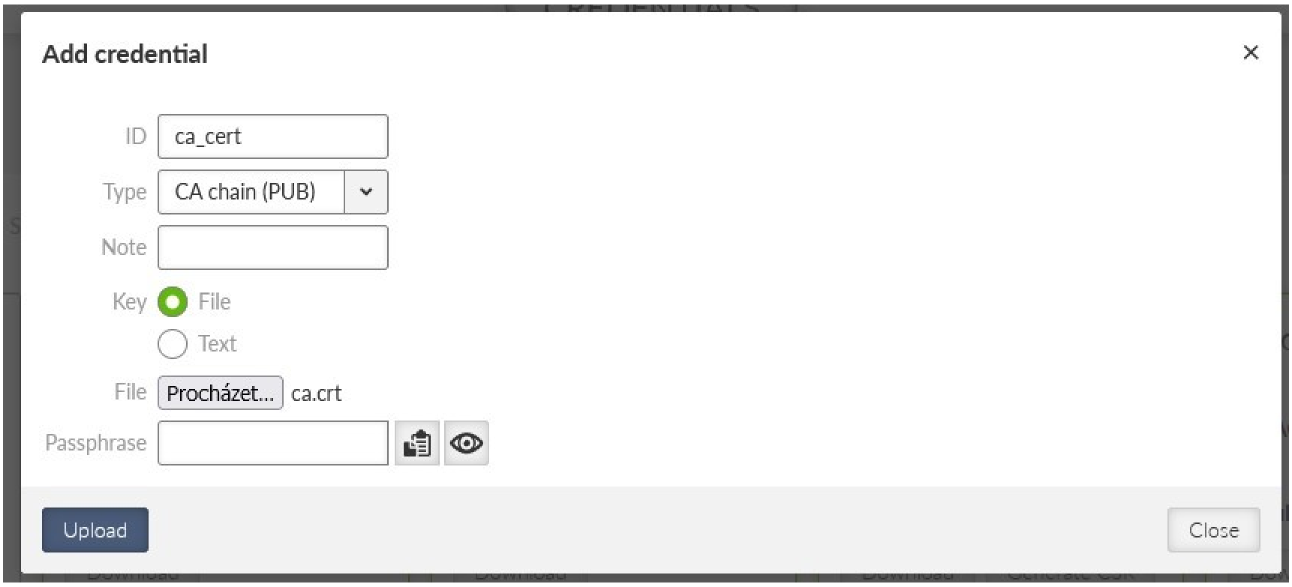
Focus on all configuration parameters to suit M!DGE2 settings. You can e.g. go through the M!DGE2 text configuration file and edit particular parameters in M!DGE3 GUI. Unfortunately, this configuration file cannot be uploaded into M!DGE3. Focus e.g. on

- Ethernet/Network interfaces
- Cellular (WWAN) connection incl. Supervision (Link testing)
- Routing (static, dynamic)
- Firewall, NAPT
- Unit name
- NTP time synchronization
- Security (users, management ports, …)
- Services (DHCP, DNS, SNMP, …)

Last, but not least, **OpenVPN**.

To be able to set up OpenVPN correctly, we need to upload all the credentials (Keys & certificates) required by the OpenVPN server. Do it from the SETTINGS > Security > Credentials menu.

Start with the CA certificate. Click on the "Add credential" button and select correct parameters and a correct file downloaded from the M!DGE2 Master (server) unit.



Add credential

ID: ca_cert

Type: CA chain (PUB)

Note:

Key: ● File ○ Text

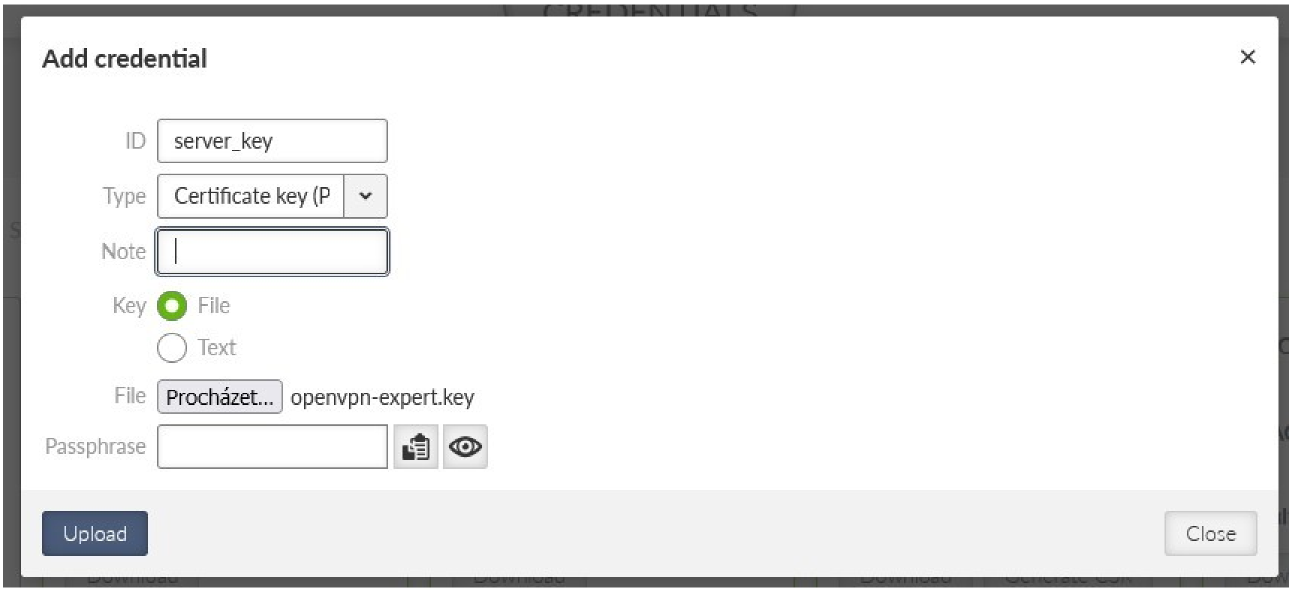
File: Procházet... ca.crt

Passphrase:

Fig. 73: Uploading the CA certificate into M!DGE3 Master (server) unit

The ID can by any, but unique within the M!DGE3 settings. Important is the "Type" which must be equal to "CA chain (PUB)". Select a correct file (usually saved as "ca.crt") and upload the file.

Upload the server’s key now.



Add credential

ID: server_key

Type: Certificate key (P

Note:

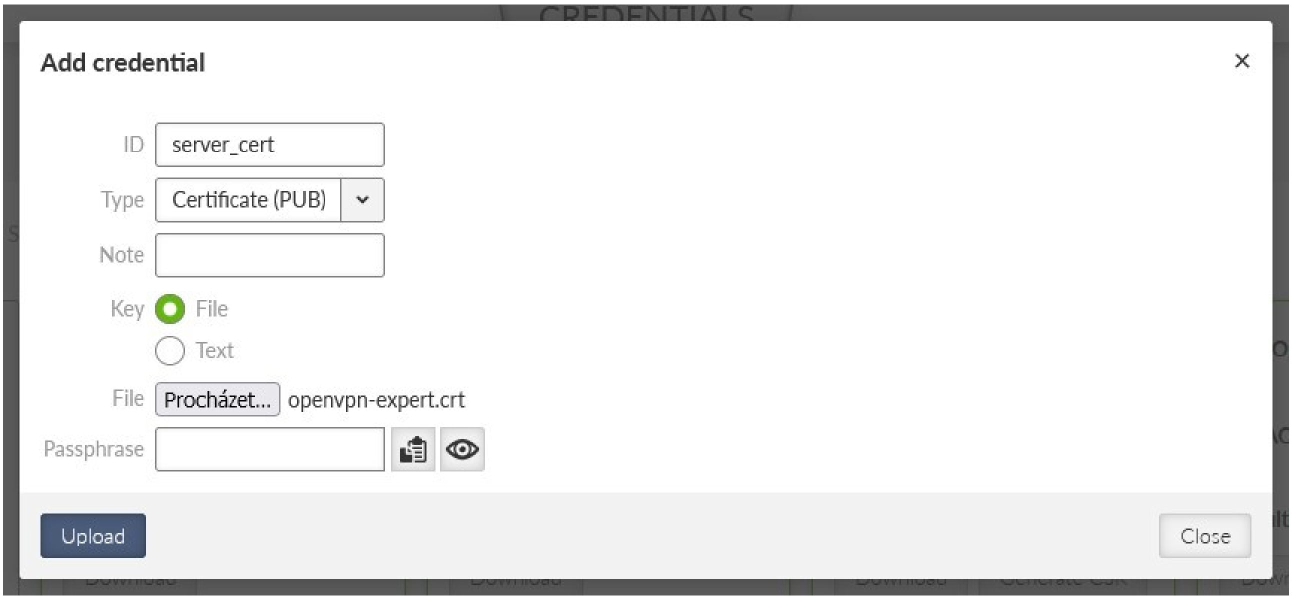
Key: ● File ○ Text

File: Procházet... openvpn-expert.key

Passphrase:

Fig. 74: Uploading the server key into M!DGE3 Master (server) unit

Upload the server’s certificate.



Add credential

ID: server_cert

Type: Certificate (PUB)

Note:

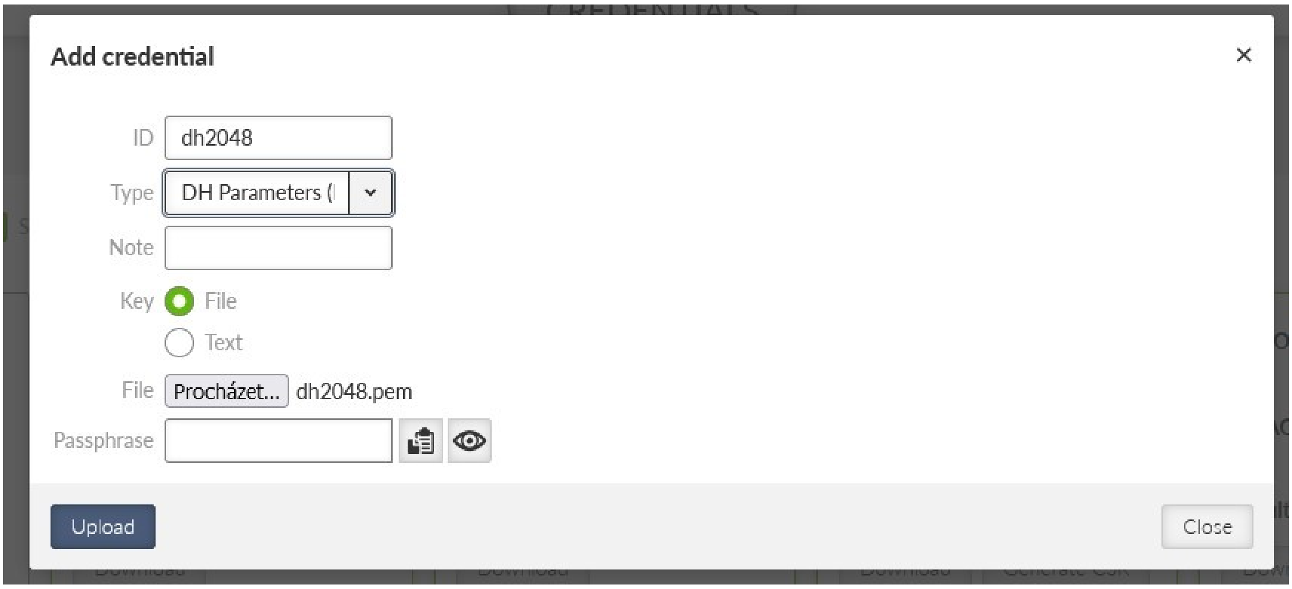
Key: ● File ○ Text

File: Procházet... openvpn-expert.crt

Passphrase:

Fig. 75: Uploading the server certificate into M!DGE3 Master (server) unit

And finish it with DH parameters.

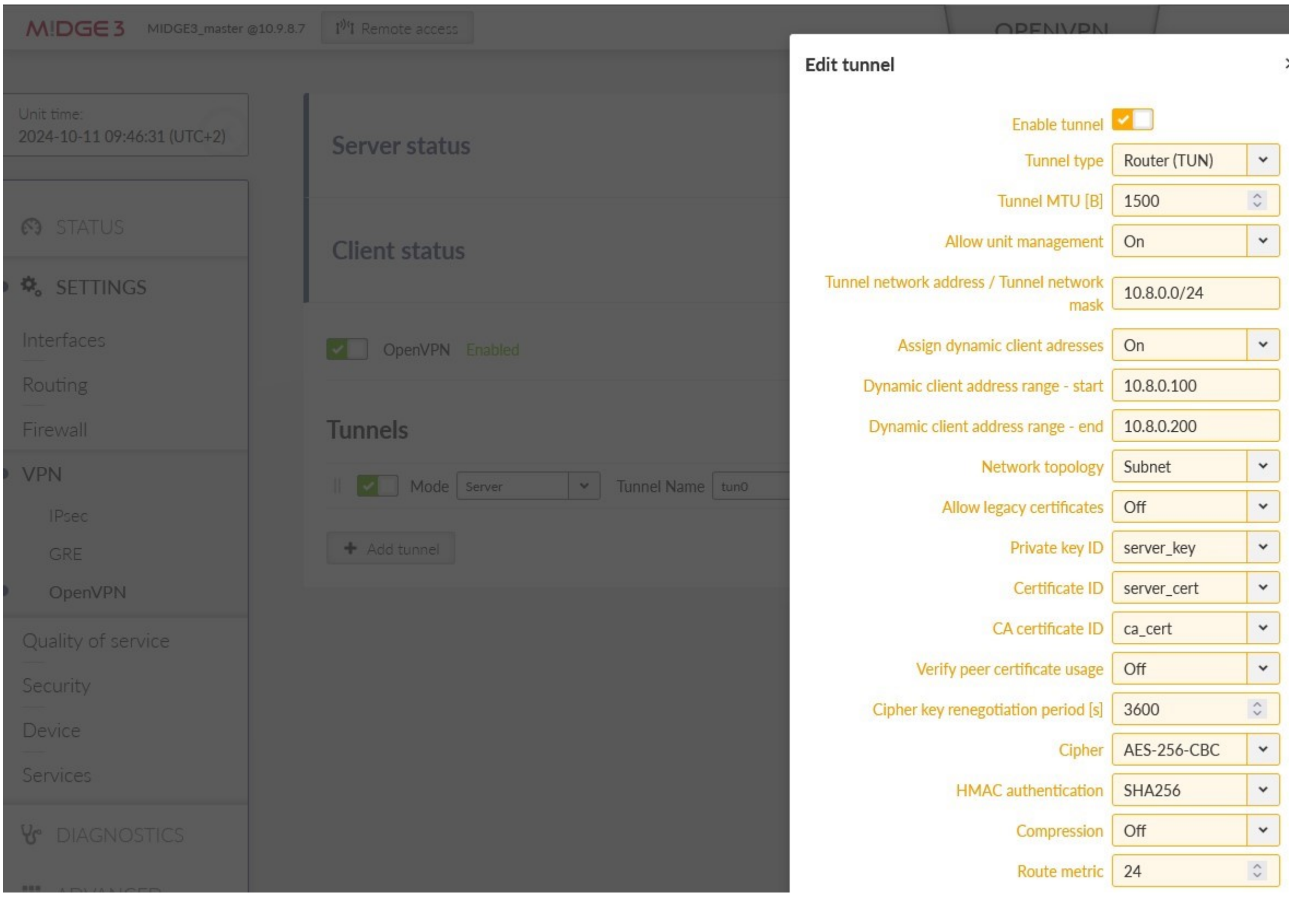


Add credential

ID	dh2048
Type	DH Parameters (|
Note	
Key	● File ○ Text
File	Procházet... dh2048.pem
Passphrase	

Fig. 76: Uploading the DH parameters into M!DGE3 Master (server) unit

Now, we can go to SETTINGS > VPN > OpenVPN menu. Add a new tunnel and set its Mode to “Server”.



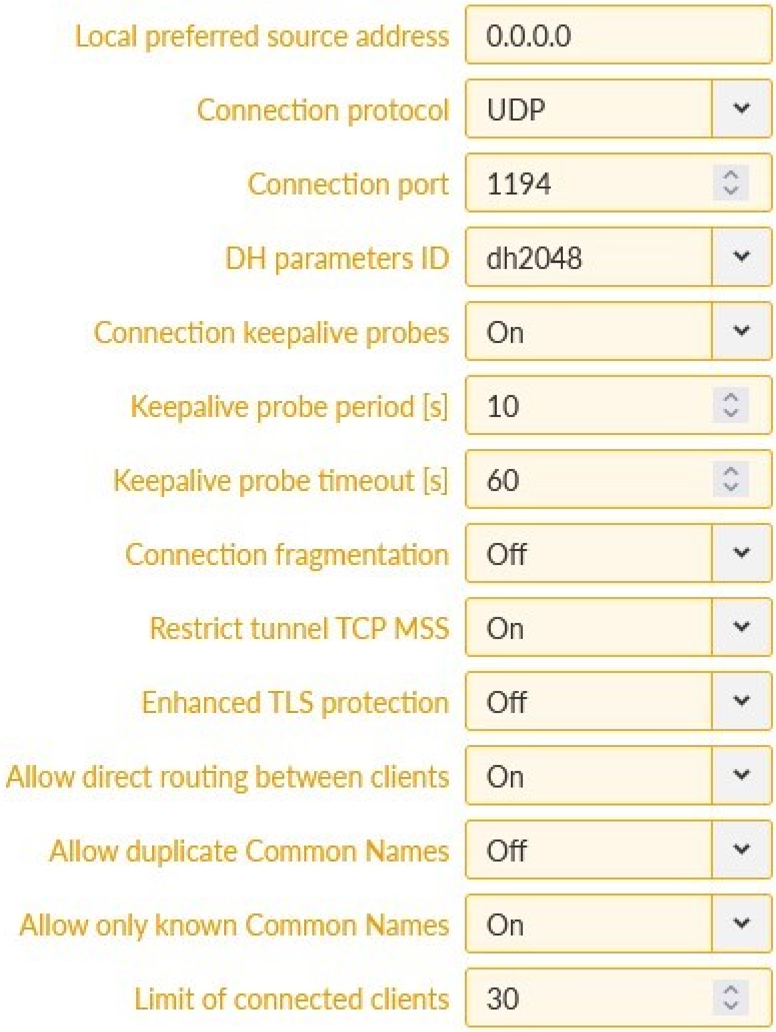
Edit tunnel

Enable tunnel	☑
Tunnel type	Router (TUN)
Tunnel MTU [B]	1500
Allow unit management	On
Tunnel network address / Tunnel network mask	10.8.0.0/24
Assign dynamic client adresses	On
Dynamic client address range - start	10.8.0.100
Dynamic client address range - end	10.8.0.200
Network topology	Subnet
Allow legacy certificates	Off
Private key ID	server_key
Certificate ID	server_cert
CA certificate ID	ca_cert
Verify peer certificate usage	Off
Cipher key renegotiation period [s]	3600
Cipher	AES-256-CBC
HMAC authentication	SHA256
Compression	Off
Route metric	24

Fig. 77: M!DGE3 Master (server) OpenVPN settings (1st part)

Focus on these parameters (and of course any other).

- **Tunnel type** – we use a routed/TUN option in our example
- **Tunnel network address / Tunnel network mask** – it can be the same as in the M!DGE2 settings, or it can be different – it is up to you now. It just cannot overlap with any other subnet in the network.
- **Assign dynamic client address** – this is important, because the OpenVPN server will automatically assign VPN IPs to each client from this range. You can keep it off – in such a case, each client must be configured with “Static” address assignment (later within clients’ settings). Static and dynamic ranges cannot overlap.
- **Network topology** – in M!DGE2, only “Net /30” option is implemented for the Server. In M!DGE3, up-to-date and modern “Subnet” option is recommended. Even the M!DGE2 clients can connect to this type as clients so we suggest using this “Subnet” topology option.
- Select proper **Private key ID, Certificate ID, CA certificate ID**.
- Turn off the “**Verify peer certificate usage**” if you did not use these modifiers in your network. Within M!DGE2, it was not possible to generate certificates with such modifiers, but could be imported manually from a different certificate management tools. **This can often cause issues** for the clients to connect with log messages incl. some “VERIFY KU ERROR” issues.
- Set a correct **Cipher** and **HMACauthentication**.
- Focus on “**Compression**” option – in M!DGE2 and older models, the compression was set to “On” by default so the data being transmitted are compressed and data usage is minimized. But using a compression can compromise data security (it can break encryption), so it is recommended to disable it (Off).



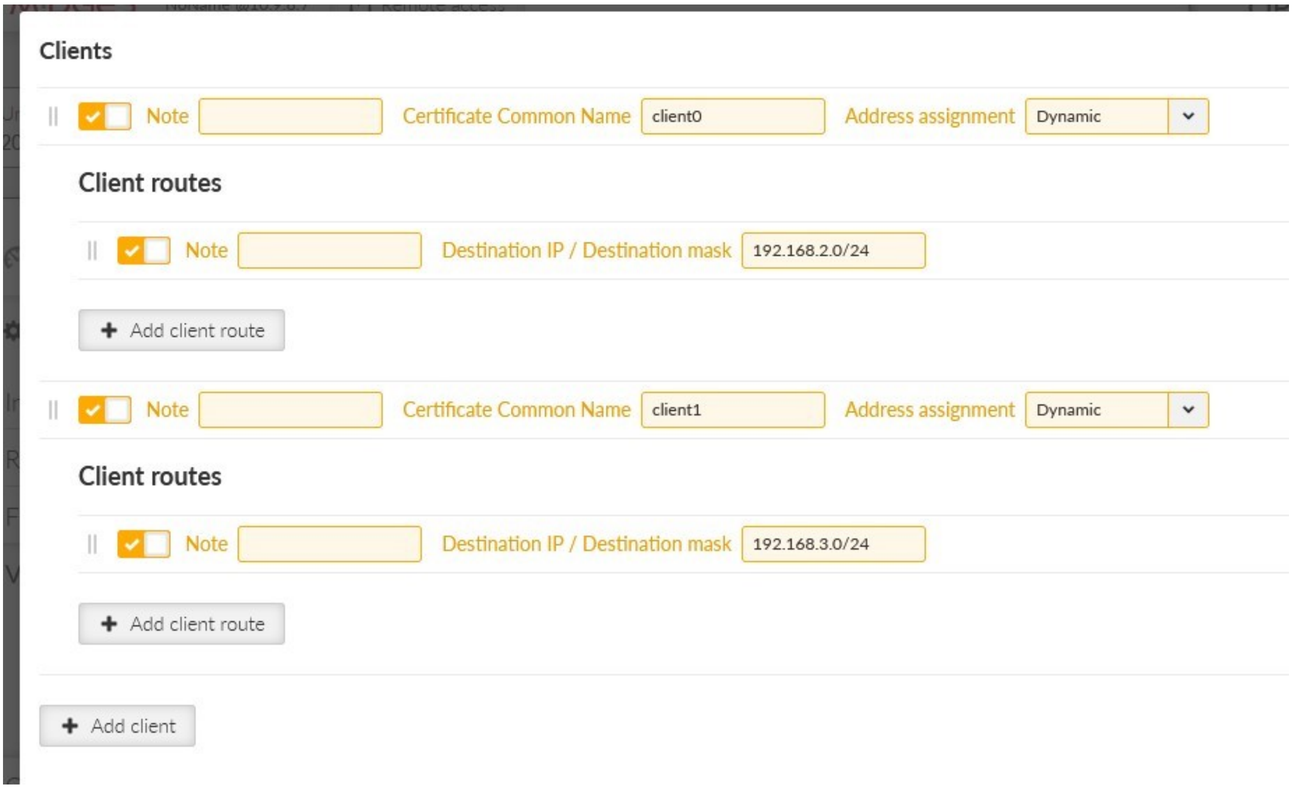
Local preferred source address	0.0.0.0
Connection protocol	UDP
Connection port	1194
DH parameters ID	dh2048
Connection keepalive probes	On
Keepalive probe period [s]	10
Keepalive probe timeout [s]	60
Connection fragmentation	Off
Restrict tunnel TCP MSS	On
Enhanced TLS protection	Off
Allow direct routing between clients	On
Allow duplicate Common Names	Off
Allow only known Common Names	On
Limit of connected clients	30

Fig. 78: M!DGE3 Master (server) OpenVPN settings (2nd part)

- Make sure to use either **UDP** or **TCP** correctly in all the units, together with its **port**.
- Select a correct **DH parameters ID**.
- Usually, enable the **keepalive** probes.
- If you utilized the **TLS-Auth** (ta.key), choose it within the “**Enhanced TLS protection**”. In such a case, you would need to add this “ta.key” file within the Credentials menu and choose it from the select box here. We also support TLS-Crypt option (higher security), but this is not supported by M!DGE2.
- If you need client-to-client communication, enable the “**Allow direct routing between clients**” option.
- **Common names** – in our example, we will not allow duplicate CNs and we will also allow known CNs only – they will be specified within the “Clients” configuration in the server’s settings.

There are many parameters which must match in all the routers. Double-check each parameter.

Once set, click on the “Edit clients” button and configure both M!DGE2 clients here.



Clients

☑ Note Certificate Common Name: client0 Address assignment: Dynamic

Client routes

☑ Note Destination IP / Destination mask: 192.168.2.0/24

☑ Note Certificate Common Name: client1 Address assignment: Dynamic

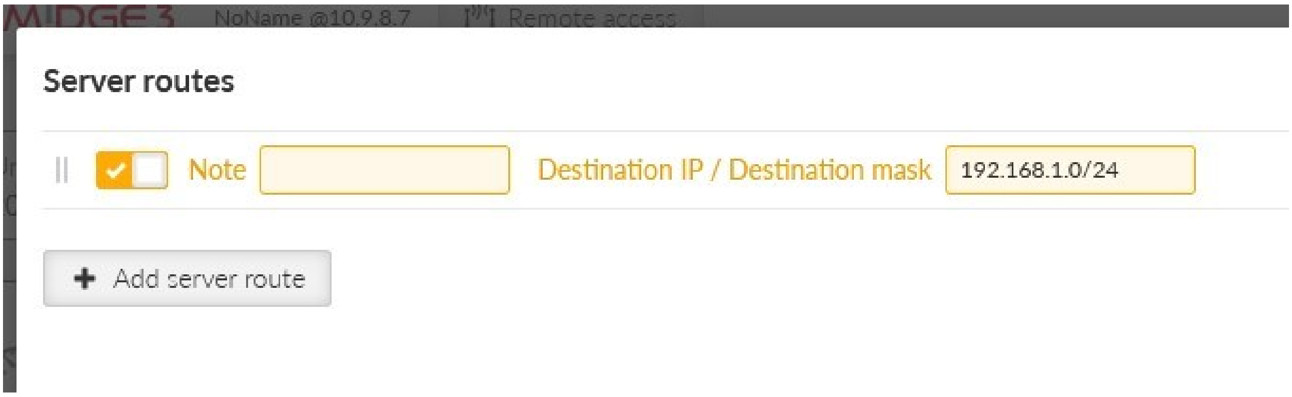
Client routes

☑ Note Destination IP / Destination mask: 192.168.3.0/24

Fig. 79: M!DGE3 Master (server) – configuration of clients

There are three important parameters. The **Certificate CN** must match the one installed in the client (client0 and client1 in our case). We chose the **Dynamic IP address assignment** and we also need to specify **subnets** which are accessible through the particular client M!DGE2 unit.

Close this window and edit the server routes. There is just one route in our example – 192.168.1.0/24. This route is pushed to each client.



Server routes

☑ Note Destination IP / Destination mask: 192.168.1.0/24

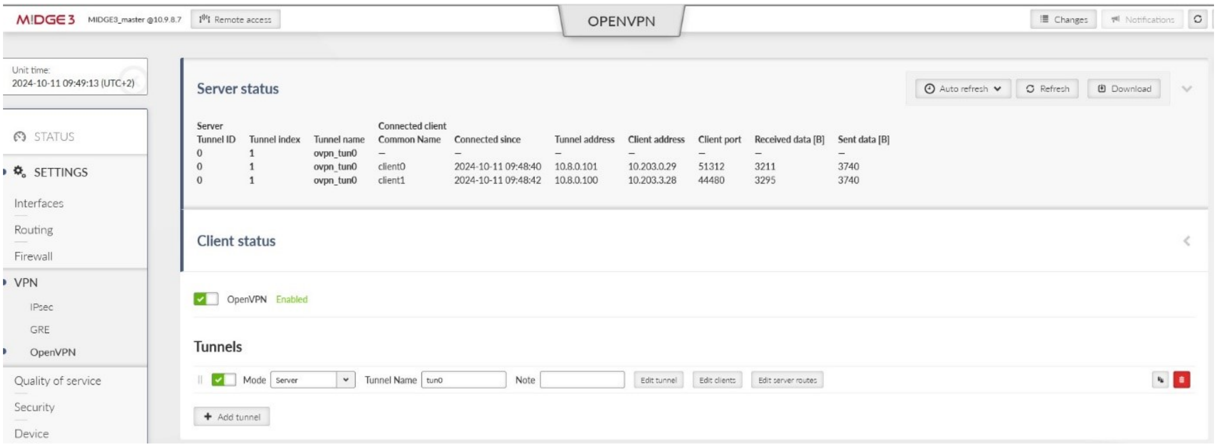
Fig. 80: M!DGE3 Master (server) – server routes

Follow your own settings, routing, diagram. This is just an example of course.

Apply all the changes.

Now, turn off the legacy M!DGE2 Master (server) router, remove its SIM card and insert it into our M!DGE3 router (while turned off as well). Swap the antenna connectors to this new M!DGE3 unit as well and turn it on (attach the power supply – focus on wiring, because different screw terminal is used and different voltage is allowed – with M!DGE3 we are on the safe side with 10-50 V, whereas M!DGE2 only had 12-24V).

If all comes up correctly, you should see both the clients connected.



Unit time:
2024-10-11 09:49:13 (UTC+2)

Server status

Server Tunnel ID	Tunnel index	Tunnel name	Connected client Common Name	Connected since	Tunnel address	Client address	Client port	Received data [B]	Sent data [B]
0	1	ovpn_tun0	–	–	–	–	–	–	–
0	1	ovpn_tun0	client0	2024-10-11 09:48:40	10.8.0.101	10.203.0.29	51312	3211	3740
0	1	ovpn_tun0	client1	2024-10-11 09:48:42	10.8.0.100	10.203.3.28	44480	3295	3740

Client status

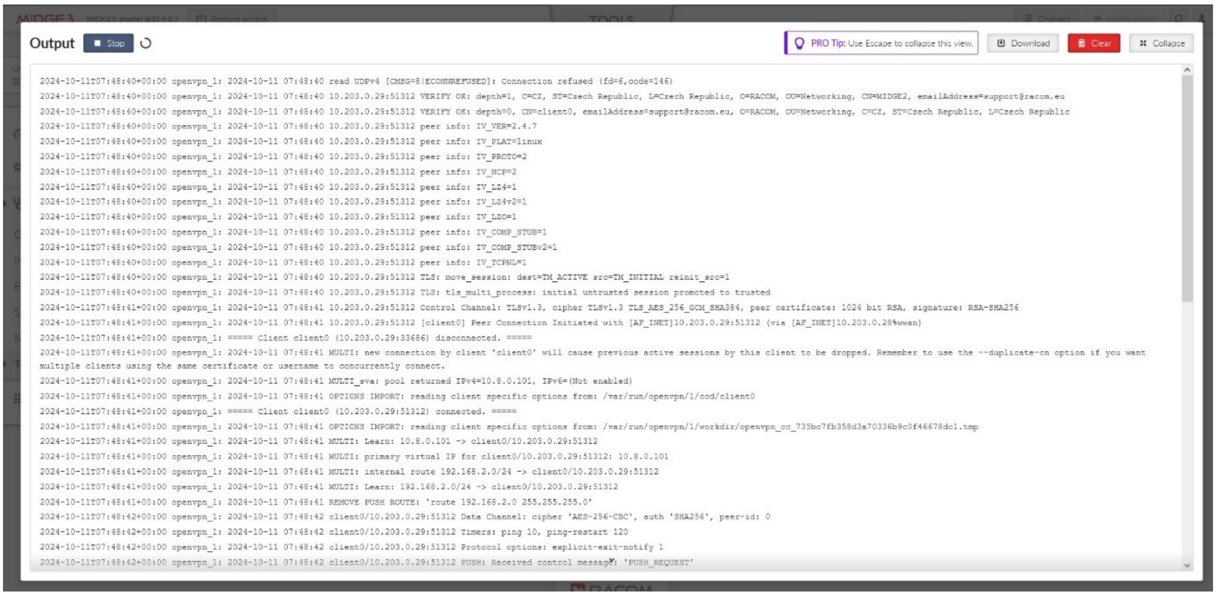
☑ OpenVPN Enabled

Tunnels

☑ Mode Server Tunnel Name tun0 Note

Fig. 81: M!DGE3 Master (server) – both clients connected

If you have any connectivity issues, follow the Troubleshooting Chapter 2 within this document. You can e.g. check the OpenVPN logs within the DIAGNOSTICS > Tools > Logs menu.



```
2024-10-11T07:48:40+00:00 openvpn_1: 2024-10-11 07:48:40 read UDPv4 [CMSG=8|ECONNREFUSED]: Connection refused (fd=6,code=146)
2024-10-11T07:48:40+00:00 openvpn_1: 2024-10-11 07:48:40 10.203.0.29:51312 VERIFY OK: depth=1, C=CZ, ST=Czech Republic, L=Czech Republic, O=RACOM, OU=Networking, CN=MIDGE2, emailAddress=support@racom.eu
2024-10-11T07:48:40+00:00 openvpn_1: 2024-10-11 07:48:40 10.203.0.29:51312 VERIFY OK: depth=0, CN=client0, emailAddress=support@racom.eu, O=RACOM, OU=Networking, C=CZ, ST=Czech Republic, L=Czech Republic
2024-10-11T07:48:40+00:00 openvpn_1: 2024-10-11 07:48:40 10.203.0.29:51312 peer info: IV_VER=2.4.7
2024-10-11T07:48:40+00:00 openvpn_1: 2024-10-11 07:48:40 10.203.0.29:51312 peer info: IV_PLAT=linux
2024-10-11T07:48:40+00:00 openvpn_1: 2024-10-11 07:48:40 10.203.0.29:51312 peer info: IV_PROTO=2
2024-10-11T07:48:40+00:00 openvpn_1: 2024-10-11 07:48:40 10.203.0.29:51312 peer info: IV_NCP=2
2024-10-11T07:48:40+00:00 openvpn_1: 2024-10-11 07:48:40 10.203.0.29:51312 peer info: IV_LZ4=1
2024-10-11T07:48:40+00:00 openvpn_1: 2024-10-11 07:48:40 10.203.0.29:51312 peer info: IV_LZ4v2=1
2024-10-11T07:48:40+00:00 openvpn_1: 2024-10-11 07:48:40 10.203.0.29:51312 peer info: IV_LZO=1
2024-10-11T07:48:40+00:00 openvpn_1: 2024-10-11 07:48:40 10.203.0.29:51312 peer info: IV_COMP_STUB=1
2024-10-11T07:48:40+00:00 openvpn_1: 2024-10-11 07:48:40 10.203.0.29:51312 peer info: IV_COMP_STUBv2=1
2024-10-11T07:48:40+00:00 openvpn_1: 2024-10-11 07:48:40 10.203.0.29:51312 peer info: IV_TCPNL=1
2024-10-11T07:48:40+00:00 openvpn_1: 2024-10-11 07:48:40 10.203.0.29:51312 TLS: move_session: dest=TM_ACTIVE src=TM_INITIAL reinit_src=1
2024-10-11T07:48:40+00:00 openvpn_1: 2024-10-11 07:48:40 10.203.0.29:51312 TLS: tls_multi_process: initial untrusted session promoted to trusted
2024-10-11T07:48:41+00:00 openvpn_1: 2024-10-11 07:48:41 10.203.0.29:51312 Control Channel: TLSv1.3, cipher TLSv1.3 TLS_AES_256_GCM_SHA384, peer certificate: 1024 bit RSA, signature: RSA-SHA256
2024-10-11T07:48:41+00:00 openvpn_1: 2024-10-11 07:48:41 10.203.0.29:51312 [client0] Peer Connection Initiated with [AF_INET]10.203.0.29:51312 (via [AF_INET]10.203.0.28%wwan)
2024-10-11T07:48:41+00:00 openvpn_1: ===== Client client0 (10.203.0.29:33686) disconnected. =====
2024-10-11T07:48:41+00:00 openvpn_1: 2024-10-11 07:48:41 MULTI: new connection by client 'client0' will cause previous active sessions by this client to be dropped. Remember to use the --duplicate-cn option if you want
multiple clients using the same certificate or username to concurrently connect.
2024-10-11T07:48:41+00:00 openvpn_1: 2024-10-11 07:48:41 MULTI_sva: pool returned IPv4=10.8.0.101, IPv6=(Not enabled)
2024-10-11T07:48:41+00:00 openvpn_1: 2024-10-11 07:48:41 OPTIONS IMPORT: reading client specific options from: /var/run/openvpn/1/ccd/client0
2024-10-11T07:48:41+00:00 openvpn_1: ===== Client client0 (10.203.0.29:51312) connected. =====
2024-10-11T07:48:41+00:00 openvpn_1: 2024-10-11 07:48:41 OPTIONS IMPORT: reading client specific options from: /var/run/openvpn/1/workdir/openvpn_cc_735bc7fb358d3a70336b9c0f46678dc1.tmp
2024-10-11T07:48:41+00:00 openvpn_1: 2024-10-11 07:48:41 MULTI: Learn: 10.8.0.101 -> client0/10.203.0.29:51312
2024-10-11T07:48:41+00:00 openvpn_1: 2024-10-11 07:48:41 MULTI: primary virtual IP for client0/10.203.0.29:51312: 10.8.0.101
2024-10-11T07:48:41+00:00 openvpn_1: 2024-10-11 07:48:41 MULTI: internal route 192.168.2.0/24 -> client0/10.203.0.29:51312
2024-10-11T07:48:41+00:00 openvpn_1: 2024-10-11 07:48:41 MULTI: Learn: 192.168.2.0/24 -> client0/10.203.0.29:51312
2024-10-11T07:48:41+00:00 openvpn_1: 2024-10-11 07:48:41 REMOVE PUSH ROUTE: 'route 192.168.2.0 255.255.255.0'
2024-10-11T07:48:42+00:00 openvpn_1: 2024-10-11 07:48:42 client0/10.203.0.29:51312 Data Channel: cipher 'AES-256-CBC', auth 'SHA256', peer-id: 0
2024-10-11T07:48:42+00:00 openvpn_1: 2024-10-11 07:48:42 client0/10.203.0.29:51312 Timers: ping 10, ping-restart 120
2024-10-11T07:48:42+00:00 openvpn_1: 2024-10-11 07:48:42 client0/10.203.0.29:51312 Protocol options: explicit-exit-notify 1
2024-10-11T07:48:42+00:00 openvpn_1: 2024-10-11 07:48:42 client0/10.203.0.29:51312 PUSH: Received control message: 'PUSH_REQUEST'
```

Fig. 82: M!DGE3 Master (server) – logs

You can also verify the functionality by pinging particular devices and/or seeing the Monitoring for encrypted data (UDP/1194).

4. OpenVPN Failover/backup option

In the most basic scenario, each client connects to one server defined by its parameters and its IP address. In more robust solution, we may have multiple servers due to high availability requirement, because in case of one server failure, the other one can work instead.

Another example could be that a particular M!DGE3/RipEX2 utilizes multiple WAN links and based on the active WAN (link priorities, dynamic routes, …) it can only reach one of more OpenVPN servers.

In such clients, we can configure multiple OpenVPN servers with different IPs and, if required, other parameters (Enhanced TLS protection, UDP/TCP protocol, port number, …). Priorities of the servers can be set. In normal operation and built-in OpenVPN “failover” option, once it connects to the lower priority server, it stays connected to it until there is some issue with this connectivity. In M!DGE3/RipEX2, we can define a fixed time after the currently active connection is closed and the higher priority servers can be used again.

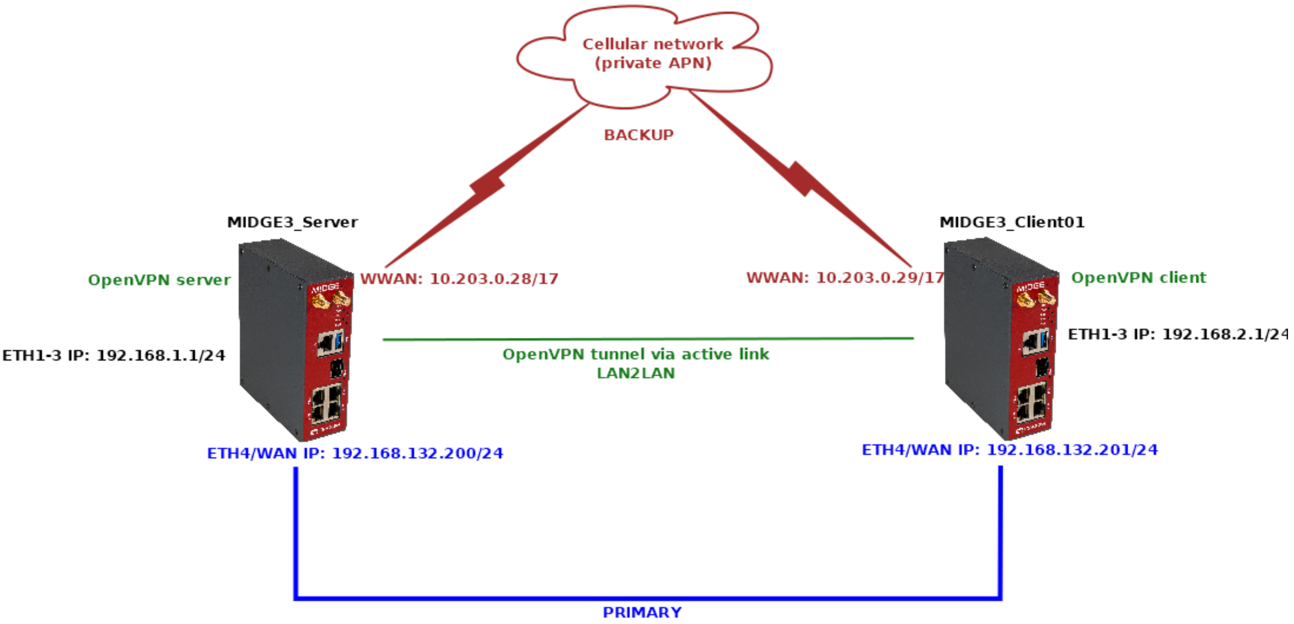
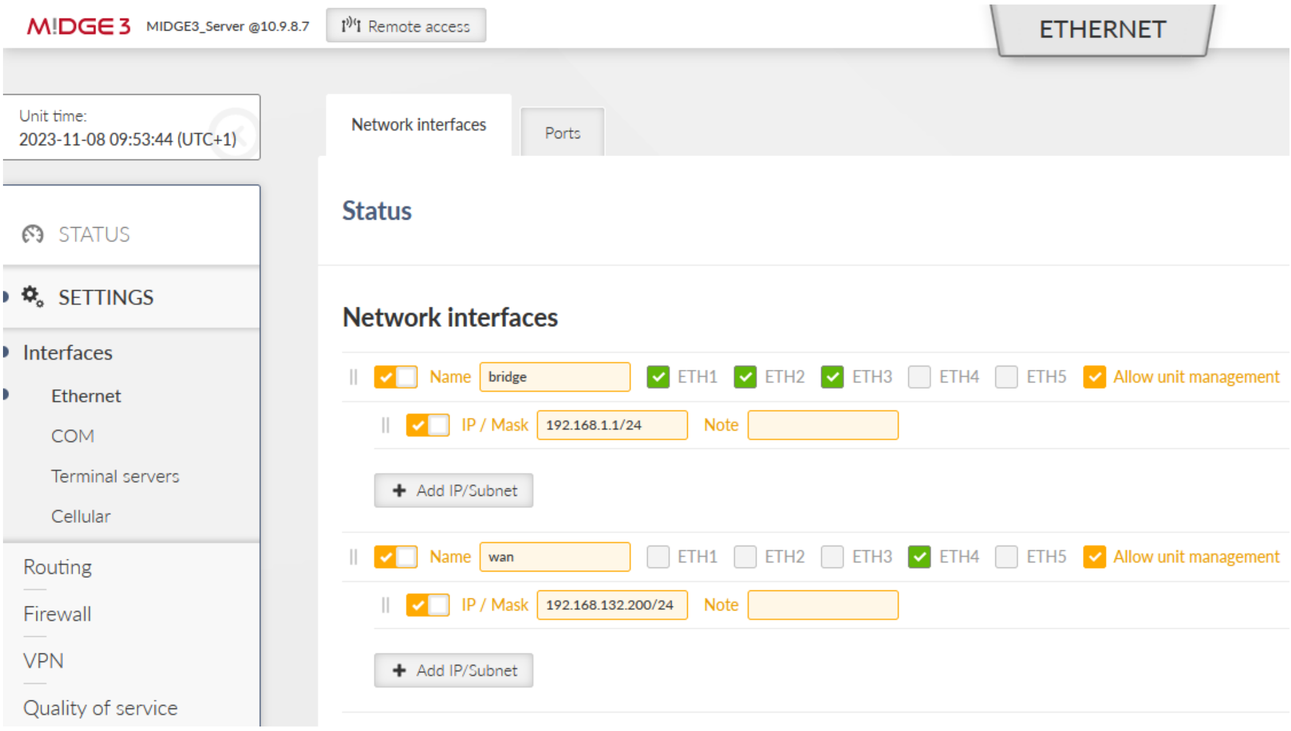


Fig. 83: Routed (TUN) OpenVPN topology, backup

If not already done, set up two M!DGE3 units by following the Chapter 1.1. The server can have RipEX2_Client02 configured as well, but we won’t focus on this client at all, only the MIDGE3_Client01 client.

4.1. MIDGE3_Server

Configure a new Network Interface within SETTINGS > Interfaces > Ethernet menu called “wan”, using ETH4 port. Set its IP address to 192.168.132.200/24.



Network interfaces

☑ Name: bridge ☑ ETH1 ☑ ETH2 ☑ ETH3 □ ETH4 □ ETH5 ☑ Allow unit management

☑ IP / Mask: 192.168.1.1/24 Note:

☑ Name: wan □ ETH1 □ ETH2 □ ETH3 ☑ ETH4 □ ETH5 ☑ Allow unit management

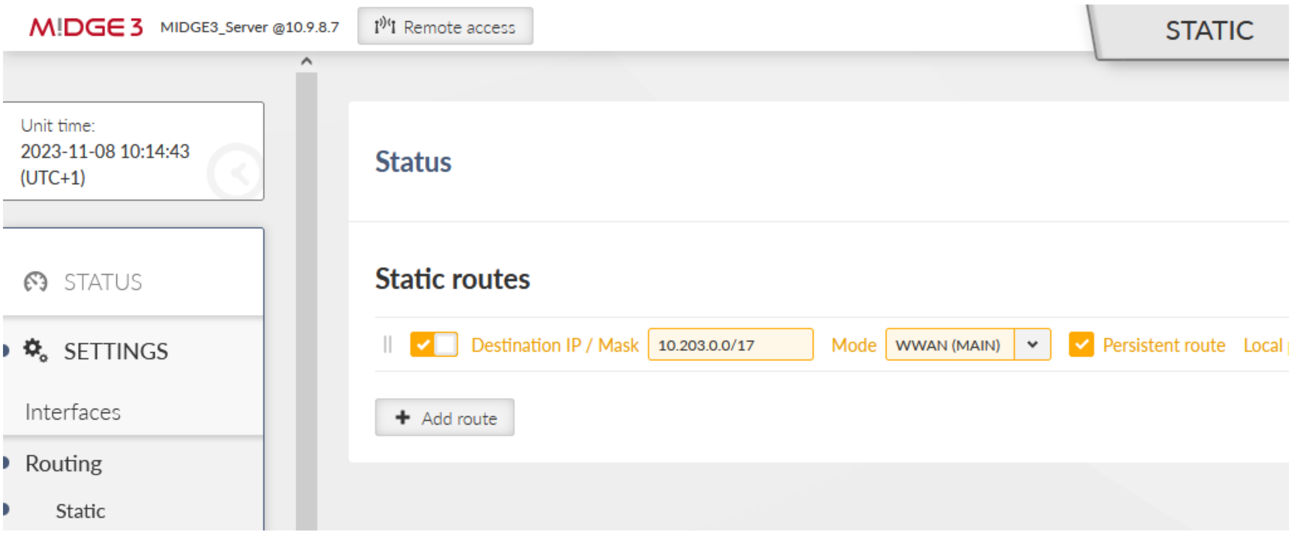
☑ IP / Mask: 192.168.132.200/24 Note:

Fig. 84: MIDGE3_Server WAN Network interface

Go to the SETTINGS > Routing > Static menu. Based on your APN settings, this may differ a lot, but we set a static route to cellular APN subnet 10.203.0.0/17 via WWAN (MAIN) persistently. We do not need to set any routes via the WAN for this scenario.

Note

More complex solutions can require multiple static routes, and/or Link management and dynamic routing.



Static routes

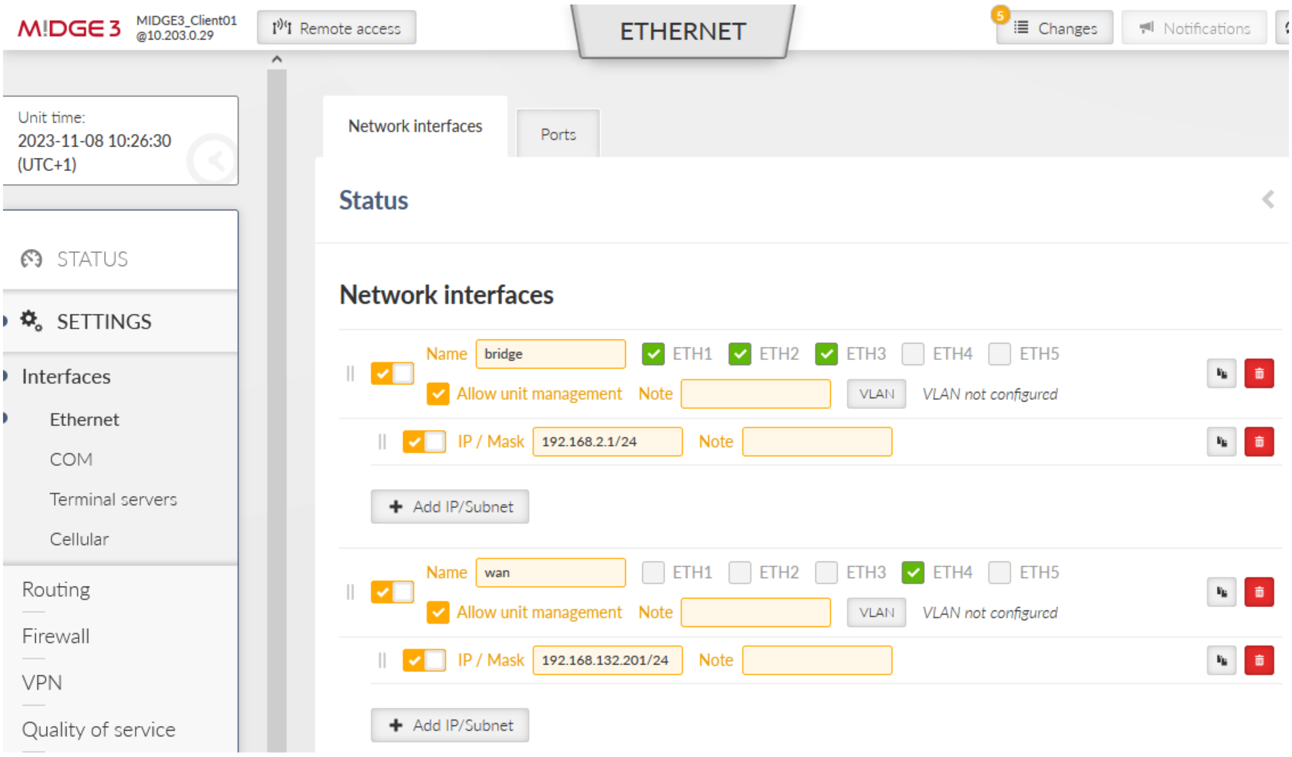
☑ Destination IP / Mask 10.203.0.0/17 Mode WWAN (MAIN) ☑ Persistent route Local

Fig. 85: MIDGE3_Server static routes

Commit the changes.

4.2. MIDGE3_Client01

Configure a new Network Interface within SETTINGS > Interfaces > Ethernet menu called “wan”, using ETH4 port. Set its IP address to 192.168.132.201/24.



Network interfaces

☑ Name bridge ☑ ETH1 ☑ ETH2 ☑ ETH3 □ ETH4 □ ETH5
☑ Allow unit management Note VLAN VLAN not configured
☑ IP / Mask 192.168.2.1/24 Note

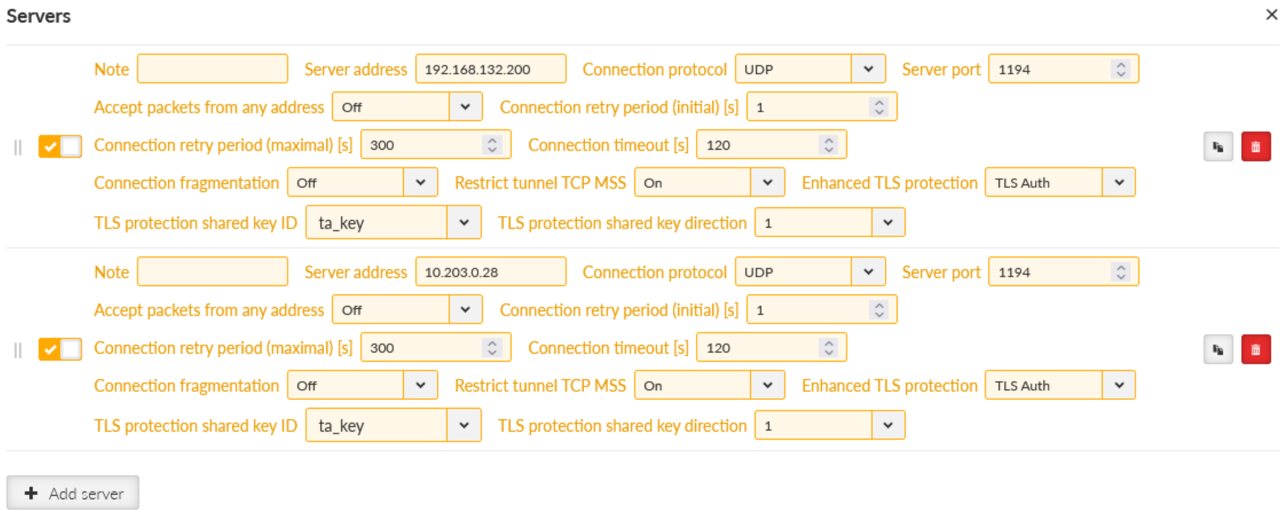
☑ Name wan □ ETH1 □ ETH2 □ ETH3 ☑ ETH4 □ ETH5
☑ Allow unit management Note VLAN VLAN not configured
☑ IP / Mask 192.168.132.201/24 Note

Fig. 86: MIDGE3_Client01 WAN Network interface

Go to the SETTINGS > Routing > Static menu. Based on your APN settings, this may differ a lot, but we set a static route to cellular APN subnet 10.203.0.0/17 via WWAN (MAIN) persistently. We do not need to set any routes via the WAN for this scenario.

Go to the SETTINGS > VPN > OpenVPN menu and click on the “Edit servers” button. Add one new server with server address 192.168.132.200. Set the TLS Protection shared key ID the same way as for the 10.203.0.28 server (TLS-Auth, direction ‘1’).

Switch the priorities so the 192.168.132.200 server is on the 1st line and 10.203.0.28 is on the 2nd line.



Servers

☑ Note: (empty) — Server address: 192.168.132.200 — Connection protocol: UDP — Server port: 1194
Accept packets from any address: Off — Connection retry period (initial) [s]: 1
Connection retry period (maximal) [s]: 300 — Connection timeout [s]: 120
Connection fragmentation: Off — Restrict tunnel TCP MSS: On — Enhanced TLS protection: TLS Auth
TLS protection shared key ID: ta_key — TLS protection shared key direction: 1

☑ Note: (empty) — Server address: 10.203.0.28 — Connection protocol: UDP — Server port: 1194
Accept packets from any address: Off — Connection retry period (initial) [s]: 1
Connection retry period (maximal) [s]: 300 — Connection timeout [s]: 120
Connection fragmentation: Off — Restrict tunnel TCP MSS: On — Enhanced TLS protection: TLS Auth
TLS protection shared key ID: ta_key — TLS protection shared key direction: 1

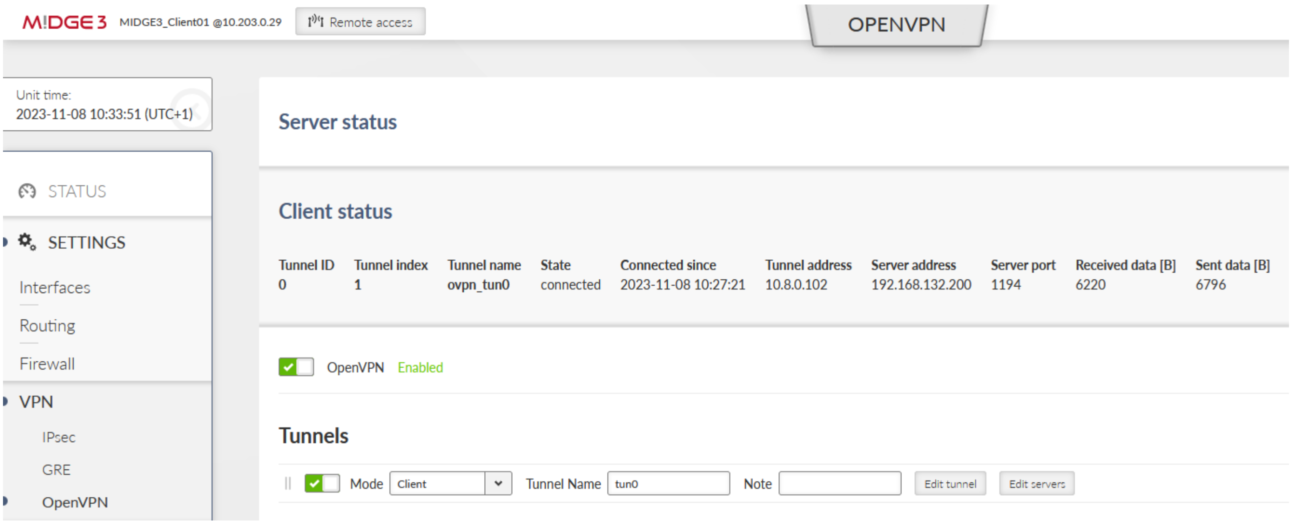
Fig. 87: MIDGE3_Client01 List of servers

Commit the changes.

Connect M!DGE3_Server and M!DGE3_Client01 via the Ethernet cable using the ETH4 ports!

4.3. Diagnostics

Currently, the OpenVPN tunnel should be established via the primary option. Go to the client’s OpenVPN settings and check the Status.



Unit time: 2023-11-08 10:33:51 (UTC+1)

Server status

Client status

Tunnel ID	Tunnel index	Tunnel name	State	Connected since	Tunnel address	Server address	Server port	Received data [B]	Sent data [B]
0	1	ovpn_tun0	connected	2023-11-08 10:27:21	10.8.0.102	192.168.132.200	1194	6220	6796

☑ OpenVPN Enabled

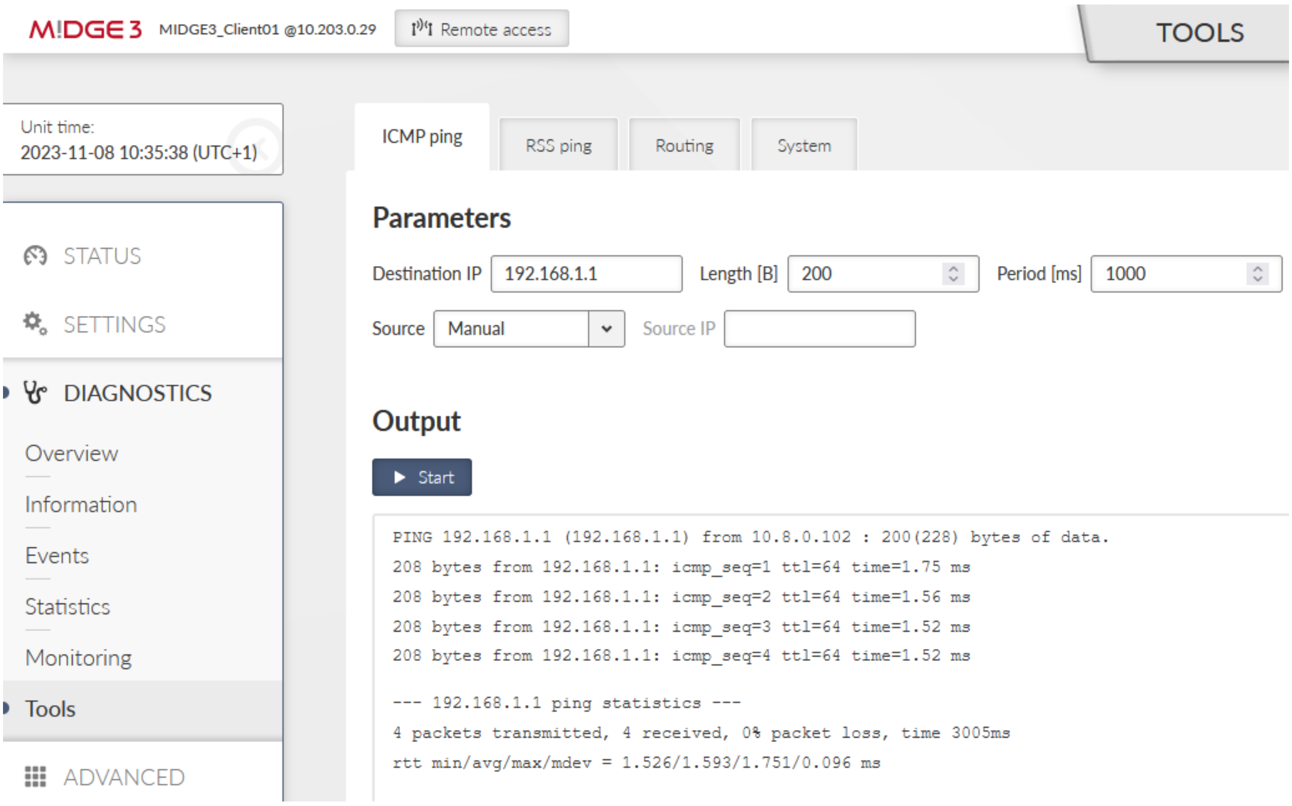
Tunnels

☑ Mode: Client — Tunnel Name: tun0 — Note: (empty)

Fig. 88: MIDGE3_Client01 OpenVPN Status

Focus on “Server address” – it should be 192.168.132.200. If not, you can reboot the unit and wait until the tunnel gets connected again.

Go to the DIAGNOSTICS > Tools > ICMP ping and try to ping 192.168.1.1 from the client. It should work with a very low RTT – it is approximately 1.5 ms in our example – because it is just via the Ethernet cable.



Parameters

Destination IP: 192.168.1.1
Length [B]: 200
Period [ms]: 1000
Source: Manual
Source IP:

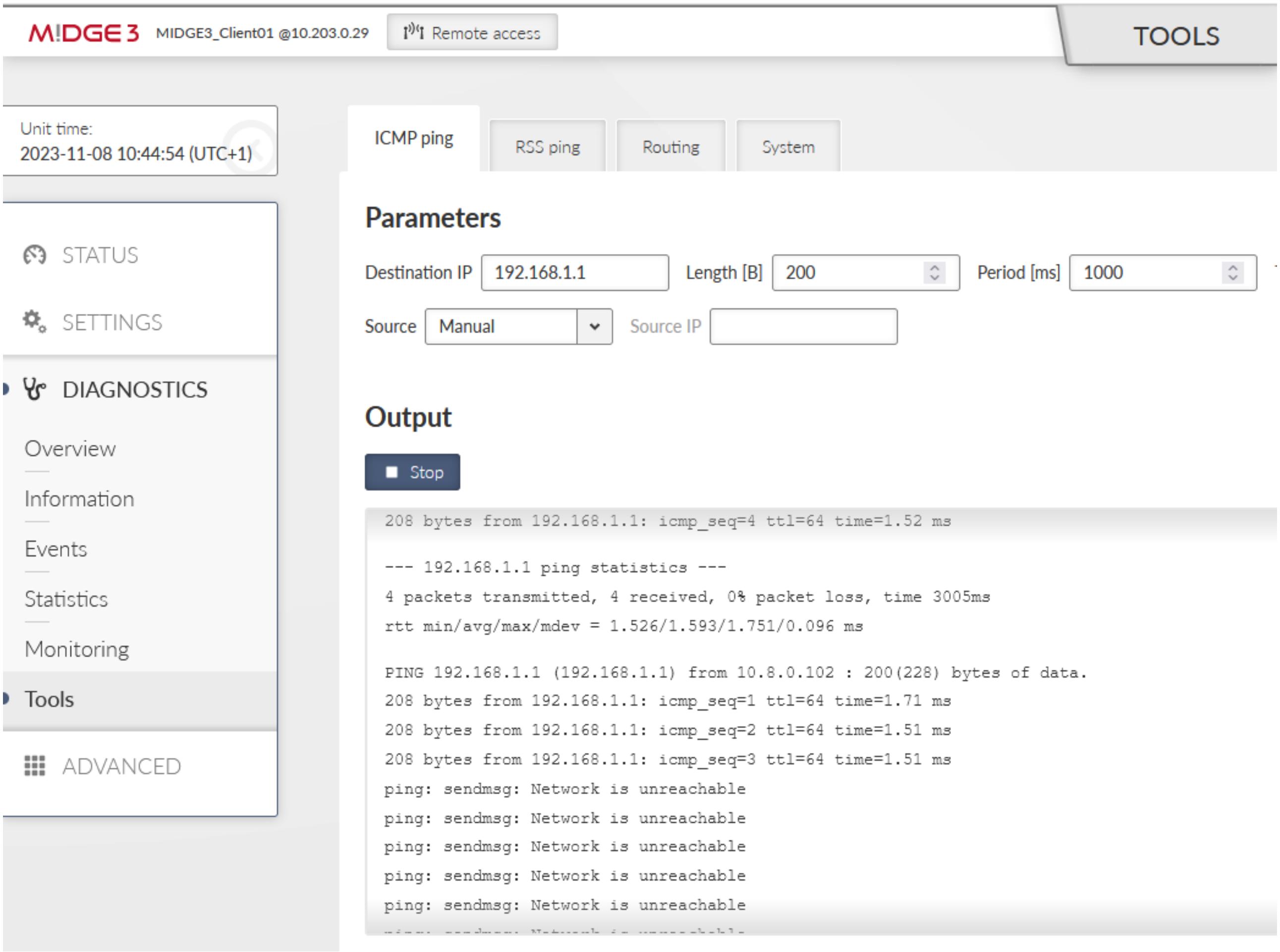
Output

```
PING 192.168.1.1 (192.168.1.1) from 10.8.0.102 : 200(228) bytes of data.
208 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=1.75 ms
208 bytes from 192.168.1.1: icmp_seq=2 ttl=64 time=1.56 ms
208 bytes from 192.168.1.1: icmp_seq=3 ttl=64 time=1.52 ms
208 bytes from 192.168.1.1: icmp_seq=4 ttl=64 time=1.52 ms

--- 192.168.1.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 1.526/1.593/1.751/0.096 ms
```

Fig. 89: MIDGE3_Client01 ICMP ping to the server’s bridge IP

Disconnect the ETH cable between the units. The ping should start failing until the OpenVPN mechanism reconnects via the cellular network. The RTTs should be much higher compared to ETH speed.



Destination IP: 192.168.1.1
Length [B]: 200
Period [ms]: 1000
Source: Manual
Source IP:

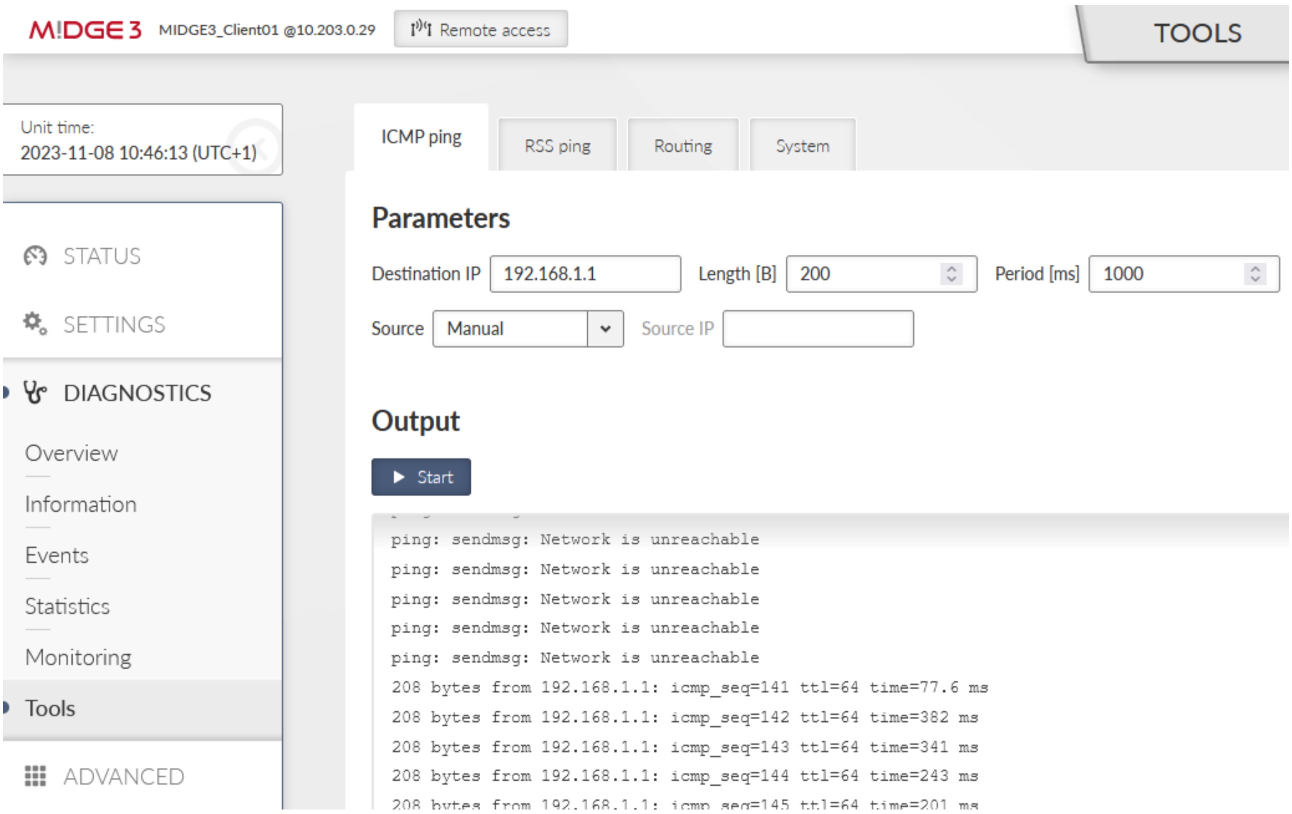
```
208 bytes from 192.168.1.1: icmp_seq=4 ttl=64 time=1.52 ms

--- 192.168.1.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 1.526/1.593/1.751/0.096 ms

PING 192.168.1.1 (192.168.1.1) from 10.8.0.102 : 200(228) bytes of data.
208 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=1.71 ms
208 bytes from 192.168.1.1: icmp_seq=2 ttl=64 time=1.51 ms
208 bytes from 192.168.1.1: icmp_seq=3 ttl=64 time=1.51 ms
ping: sendmsg: Network is unreachable
ping: sendmsg: Network is unreachable
ping: sendmsg: Network is unreachable
ping: sendmsg: Network is unreachable
ping: sendmsg: Network is unreachable
ping: sendmsg: Network is unreachable
```

Fig. 90: MIDGE3_Client01 ICMP ping failures

The switchover time can differ based on Keepalives set in the OpenVPN settings.



Parameters

Destination IP: 192.168.1.1 | Length [B]: 200 | Period [ms]: 1000

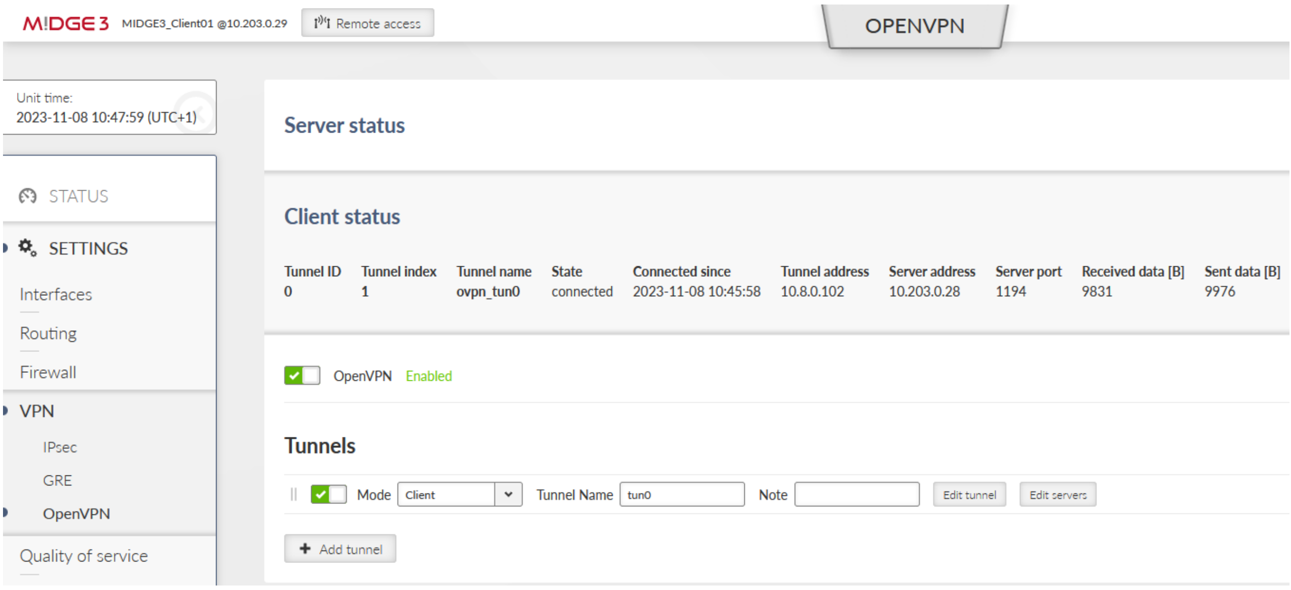
Source: Manual | Source IP:

Output

```
ping: sendmsg: Network is unreachable
ping: sendmsg: Network is unreachable
ping: sendmsg: Network is unreachable
ping: sendmsg: Network is unreachable
ping: sendmsg: Network is unreachable
208 bytes from 192.168.1.1: icmp_seq=141 ttl=64 time=77.6 ms
208 bytes from 192.168.1.1: icmp_seq=142 ttl=64 time=382 ms
208 bytes from 192.168.1.1: icmp_seq=143 ttl=64 time=341 ms
208 bytes from 192.168.1.1: icmp_seq=144 ttl=64 time=243 ms
208 bytes from 192.168.1.1: icmp_seq=145 ttl=64 time=201 ms
```

Fig. 91: MIDGE3_Client01 ICMP ping working correctly

You should also check the OpenVPN status – there should be 10.203.0.28 the Server address.



Server status

Client status

Tunnel ID	Tunnel index	Tunnel name	State	Connected since	Tunnel address	Server address	Server port	Received data [B]	Sent data [B]
0	1	ovpn_tun0	connected	2023-11-08 10:45:58	10.8.0.102	10.203.0.28	1194	9831	9976

☑ OpenVPN Enabled

Tunnels

☑ Mode: Client | Tunnel Name: tun0 | Note:

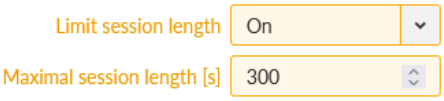
Fig. 92: MIDGE3_Client01 OpenVPN connection via the cellular network

Connect the Ethernet cable again.

The OpenVPN connection does not get re-established via the primary link now, because it only goes to the primary server if the current failover/backup fails. You can either disconnect the cellular antenna, or reboot the client. Wait until it gets connected via the primary ETH link again.

If you need to check the primary (higher priorities) link, set the “Limit session length” parameter in the client’s tunnel configuration. Set the “Maximal session length” to suit your requirements. The minimum time is 5 minutes and the maximum time is 1 day.

We will set it for 5 minutes only due to testing purposes.



Limit session length: On

Maximal session length [s]: 300

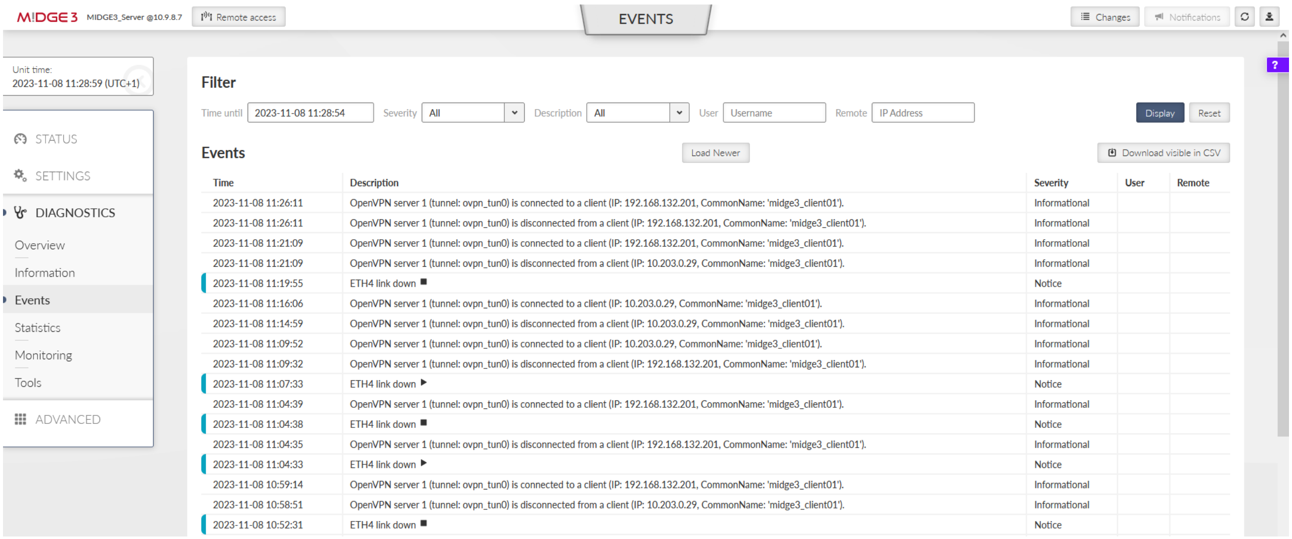
Fig. 93: MIDGE3_Client01 Limit session length

Commit changes.

You can do the test with the Ethernet cable again. Once the OpenVPN gets connected via the cellular network, attach the cable again. Within 5 minutes’ time, you should be re-connected via the primary Ethernet link again.

Keep in mind such settings reconnect the tunnel no matter which server is being used – so in our example, while operating via the primary Ethernet link – the VPN tunnel is still re-established every 5 minutes.

You may check the DIAGNOSTICS > Events menu.



Unit time:
2023-11-08 11:28:59 (UTC+1)

Filter

Time until: 2023-11-08 11:28:54 | Severity: All | Description: All | User: Username | Remote: IP Address

Events

Time	Description	Severity	User	Remote
2023-11-08 11:26:11	OpenVPN server 1 (tunnel: ovpn_tun0) is connected to a client (IP: 192.168.132.201, CommonName: 'midge3_client01').	Informational		
2023-11-08 11:26:11	OpenVPN server 1 (tunnel: ovpn_tun0) is disconnected from a client (IP: 192.168.132.201, CommonName: 'midge3_client01').	Informational		
2023-11-08 11:21:09	OpenVPN server 1 (tunnel: ovpn_tun0) is connected to a client (IP: 192.168.132.201, CommonName: 'midge3_client01').	Informational		
2023-11-08 11:21:09	OpenVPN server 1 (tunnel: ovpn_tun0) is disconnected from a client (IP: 10.203.0.29, CommonName: 'midge3_client01').	Informational		
2023-11-08 11:19:55	ETH4 link down ■	Notice		
2023-11-08 11:16:06	OpenVPN server 1 (tunnel: ovpn_tun0) is connected to a client (IP: 10.203.0.29, CommonName: 'midge3_client01').	Informational		
2023-11-08 11:14:59	OpenVPN server 1 (tunnel: ovpn_tun0) is disconnected from a client (IP: 10.203.0.29, CommonName: 'midge3_client01').	Informational		
2023-11-08 11:09:52	OpenVPN server 1 (tunnel: ovpn_tun0) is connected to a client (IP: 10.203.0.29, CommonName: 'midge3_client01').	Informational		
2023-11-08 11:09:32	OpenVPN server 1 (tunnel: ovpn_tun0) is disconnected from a client (IP: 192.168.132.201, CommonName: 'midge3_client01').	Informational		
2023-11-08 11:07:33	ETH4 link down ▶	Notice		
2023-11-08 11:04:39	OpenVPN server 1 (tunnel: ovpn_tun0) is connected to a client (IP: 192.168.132.201, CommonName: 'midge3_client01').	Informational		
2023-11-08 11:04:38	ETH4 link down ■	Notice		
2023-11-08 11:04:35	OpenVPN server 1 (tunnel: ovpn_tun0) is disconnected from a client (IP: 192.168.132.201, CommonName: 'midge3_client01').	Informational		
2023-11-08 11:04:33	ETH4 link down ▶	Notice		
2023-11-08 10:59:14	OpenVPN server 1 (tunnel: ovpn_tun0) is connected to a client (IP: 192.168.132.201, CommonName: 'midge3_client01').	Informational		
2023-11-08 10:58:51	OpenVPN server 1 (tunnel: ovpn_tun0) is disconnected from a client (IP: 10.203.0.29, CommonName: 'midge3_client01').	Informational		
2023-11-08 10:52:31	ETH4 link down ■	Notice		

Fig. 94: MIDGE3_Server Events history

5. Dynamic routing over OpenVPN L2

Dynamic routing protocols (Babel, BGP and OSPF) can only be operated over OpenVPN L2. It is not possible to utilize OpenVPN L3 due to internal routing in the server. Moreover, Babel uses IPv6 link addresses which are dynamically assigned and are not routed.

Be careful not to cause routing loops - i.e. routing OpenVPN back to the OpenVPN tunnel over and over again. The protective mechanism is not implemented in OpenVPN, it's only in IPsec.

The following text describes utilizing dynamic routing protocols Babel and BGP over **OpenVPN L2 tunnel (TAP, bridged option)**.

The example starts from the Routed (TUN) OpenVPN topology described in *Section 1.1, "Routed (TUN) OpenVPN topology"*. If not yet utilized, start your configuration there.

5.1. Babel dynamic routing

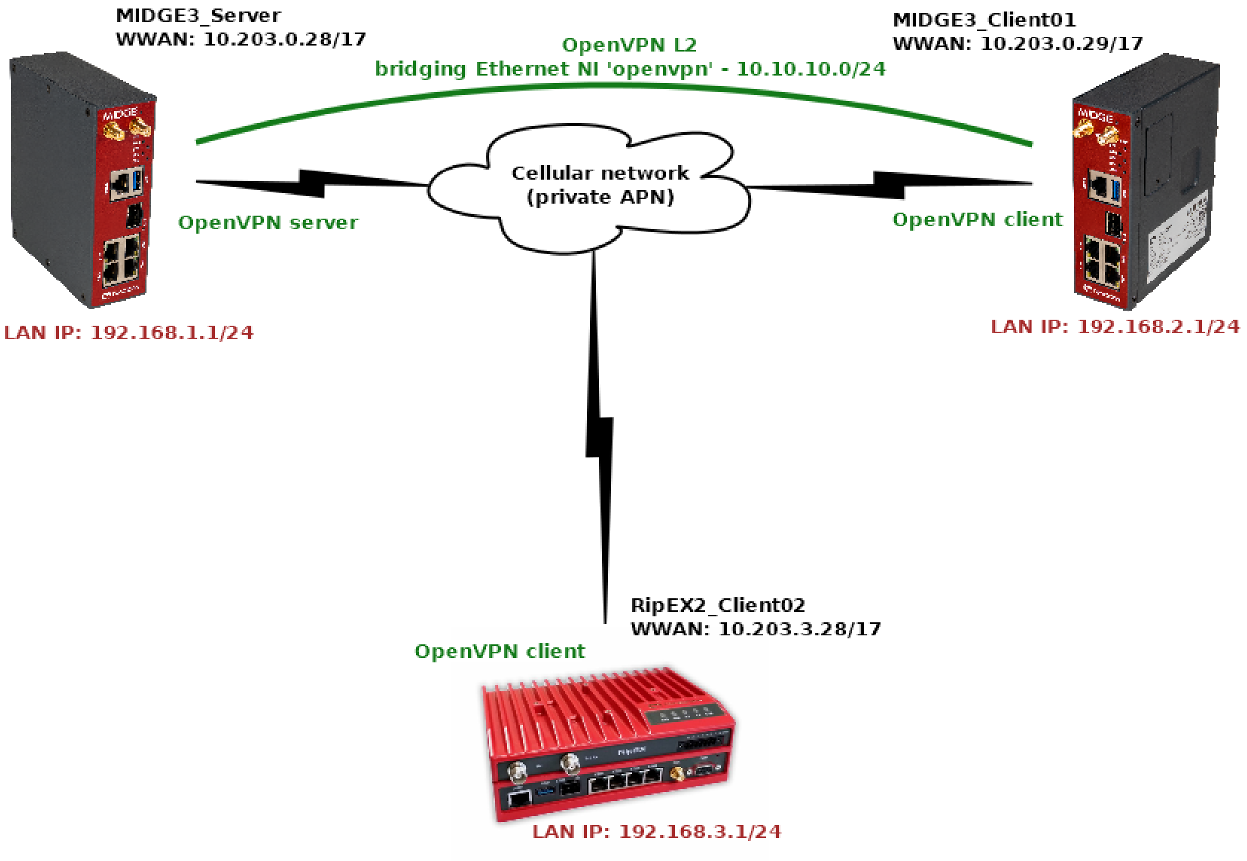
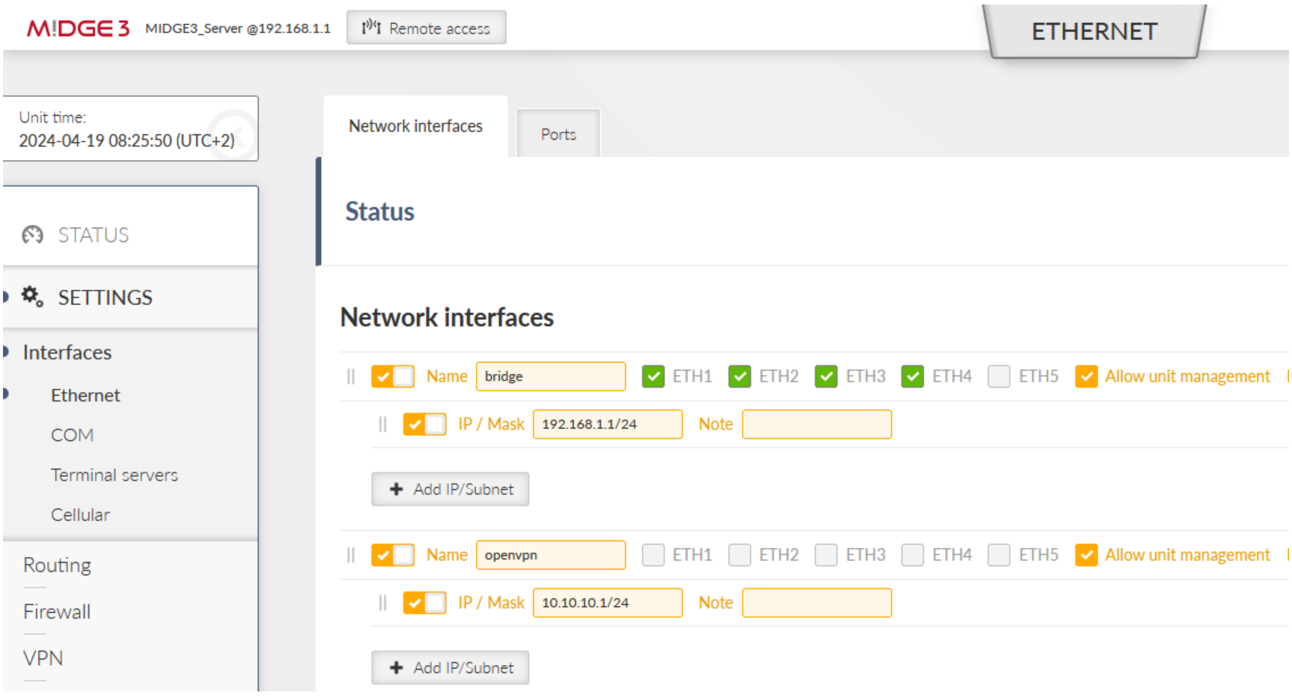


Fig. 95: Topology diagram

5.1.1. M!DGE3_Server

This M!DGE3 unit is set as the OpenVPN server. One of the most important configuration parameter is one additional Network interface within the SETTINGS > Interfaces > Ethernet menu. This NI is not attached to any of the physical ETH port, but is then set in OpenVPN L2 configuration only.



M!DGE3 MIDGE3_Server @192.168.1.1 Remote access ETHERNET

Unit time: 2024-04-19 08:25:50 (UTC+2)

STATUS
SETTINGS
Interfaces
Ethernet
COM
Terminal servers
Cellular
Routing
Firewall
VPN

Network interfaces | Ports

Status

Network interfaces

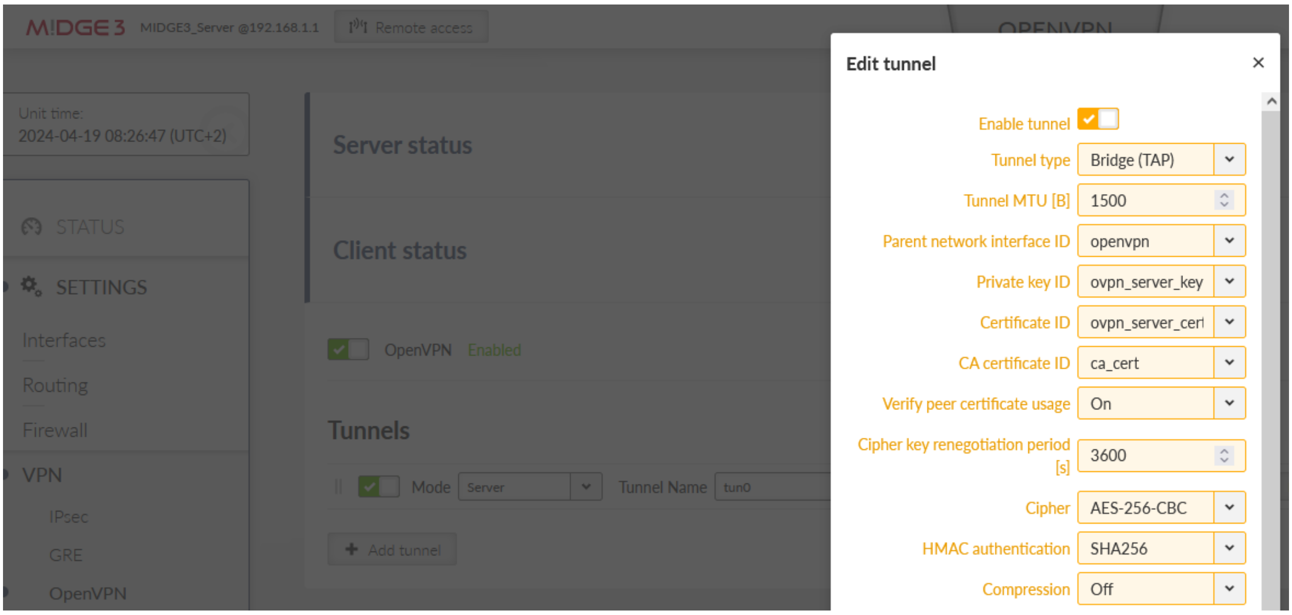
☑ Name bridge ☑ ETH1 ☑ ETH2 ☑ ETH3 ☑ ETH4 □ ETH5 ☑ Allow unit management
☑ IP / Mask 192.168.1.1/24 Note
Add IP/Subnet

☑ Name openvpn □ ETH1 □ ETH2 □ ETH3 □ ETH4 □ ETH5 ☑ Allow unit management
☑ IP / Mask 10.10.10.1/24 Note
Add IP/Subnet

Fig. 96: M!DGE3_Server Ethernet settings

We create a new Network interface (NI) called "openvpn" and its IP/mask is set to 10.10.10.1/24.

Go to the SETTINGS > VPN > OpenVPN menu and set/change the OpenVPN settings. Change the Tunnel type to Bridge (TAP) and set the Parent network interface ID to "openvpn".



M!DGE3 MIDGE3_Server @192.168.1.1 Remote access OPENVPN

Unit time: 2024-04-19 08:26:47 (UTC+2)

STATUS
SETTINGS
Interfaces
Routing
Firewall
VPN
IPsec
GRE
OpenVPN

Server status

Client status

☑ OpenVPN Enabled

Tunnels

☑ Mode Server Tunnel Name tun0

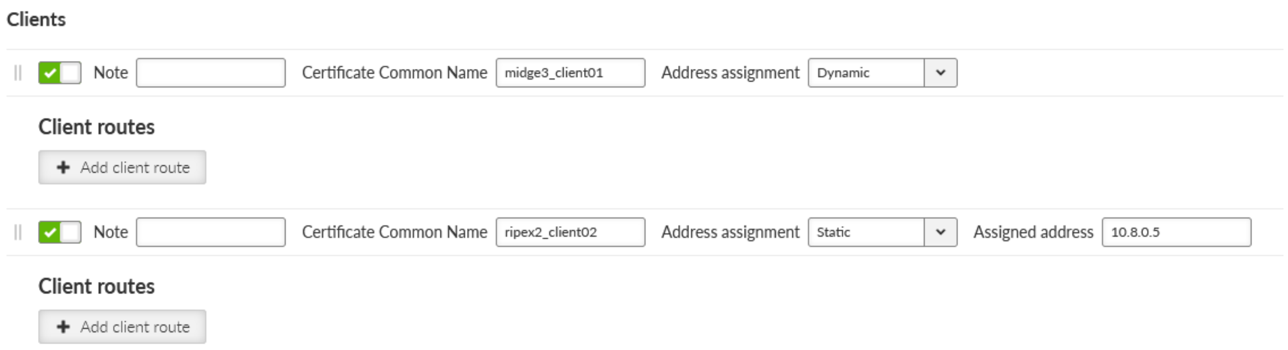
Add tunnel

Edit tunnel

Enable tunnel ☑
Tunnel type Bridge (TAP)
Tunnel MTU [B] 1500
Parent network interface ID openvpn
Private key ID ovpn_server_key
Certificate ID ovpn_server_cert
CA certificate ID ca_cert
Verify peer certificate usage On
Cipher key renegotiation period [s] 3600
Cipher AES-256-CBC
HMAC authentication SHA256
Compression Off

Fig. 97: M!DGE3_Server OpenVPN changes

Delete client routes for both the clients.



Clients

☑ Note Certificate Common Name midge3_client01 Address assignment Dynamic

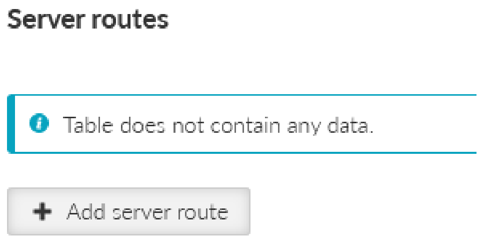
Client routes

☑ Note Certificate Common Name ripex2_client02 Address assignment Static Assigned address 10.8.0.5

Client routes

Fig. 98: M!DGE3_Server Deleting OpenVPN routes

Server routes' table is also empty. Routing is managed by Babel dynamic routing protocol.

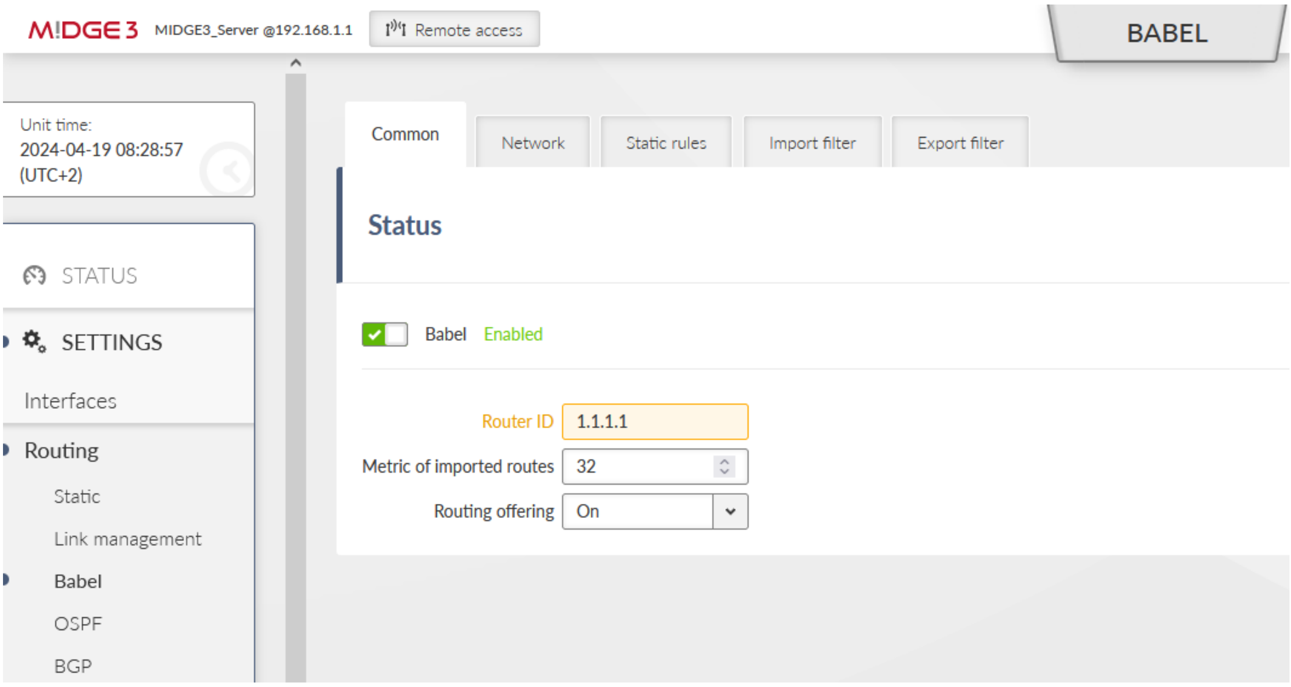


Server routes

Table does not contain any data.

Fig. 99: M!DGE3_Server OpenVPN empty server routes

Last, but not least important menu is SETTINGS > Routing > Babel. Enable it and set the ID to 1.1.1.1.



Status

☑ Babel Enabled

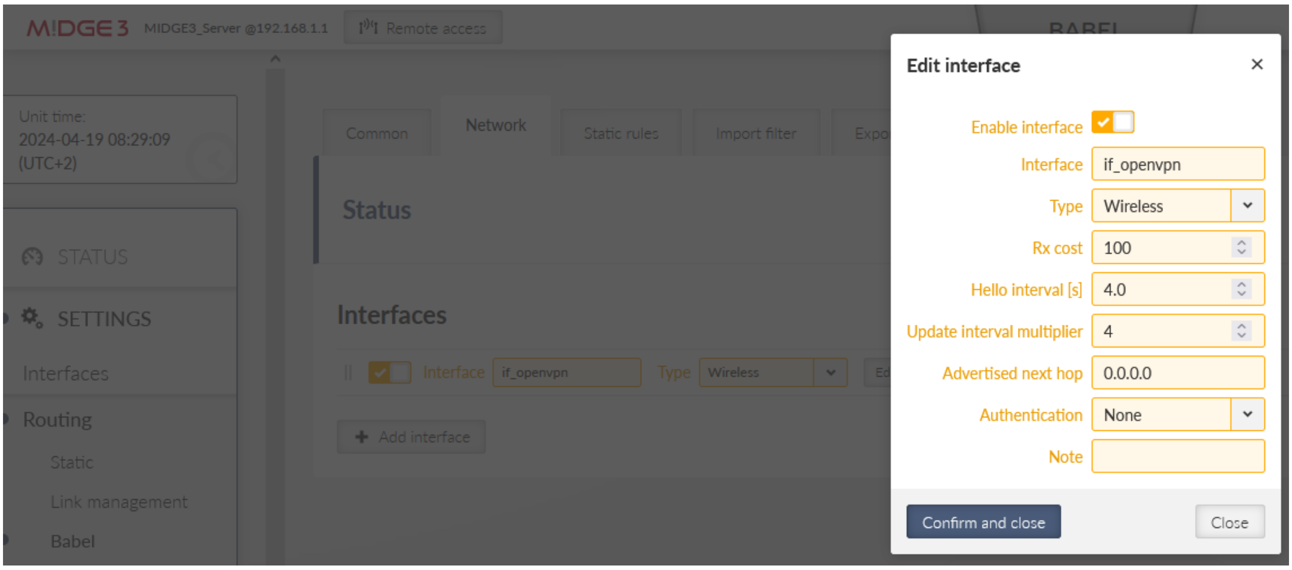
Router ID 1.1.1.1

Metric of imported routes 32

Routing offering On

Fig. 100: M!DGE3_Server Babel Common settings

Go to the Network tab and select the 'if_openvpn' interface. Other parameters may differ to suit your application.



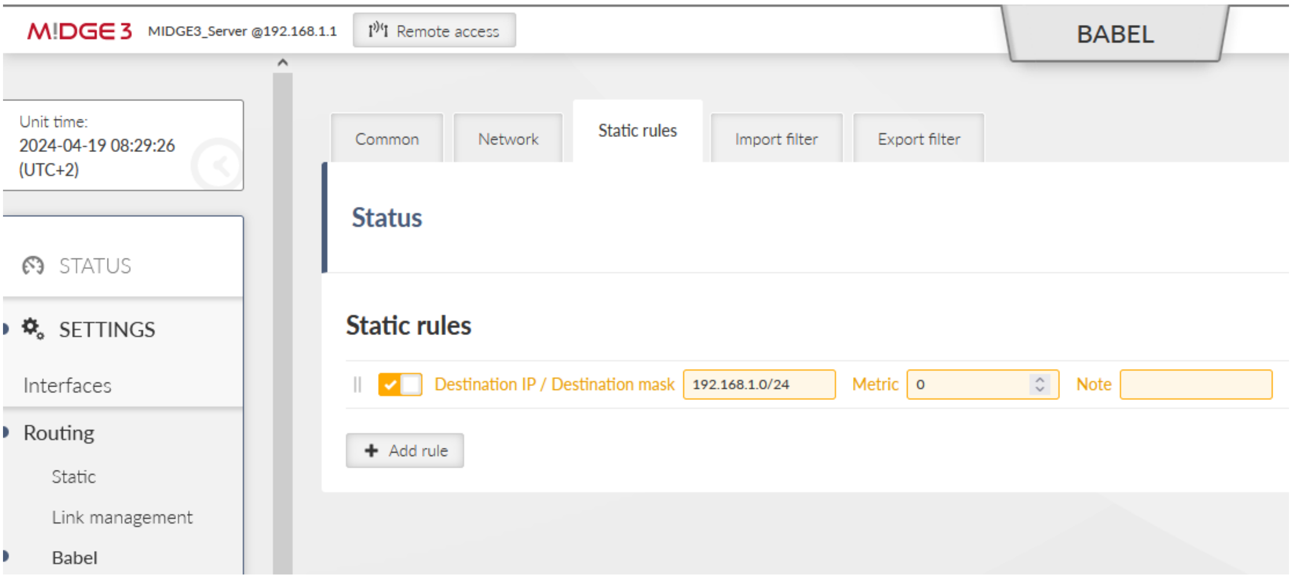
Edit interface

- Enable interface: ☑
- Interface: if_openvpn
- Type: Wireless
- Rx cost: 100
- Hello interval [s]: 4.0
- Update interval multiplier: 4
- Advertised next hop: 0.0.0.0
- Authentication: None
- Note:

Fig. 101: M!DGE3_Server Babel Network settings

Note the prefix "if_" which is always included in the real interface name of any NI within Ethernet menu.

Set the Static rule to be exported via the Babel protocol. This is the Master's LAN subnet.



Static rules

☑ Destination IP / Destination mask: 192.168.1.0/24 Metric: 0 Note:

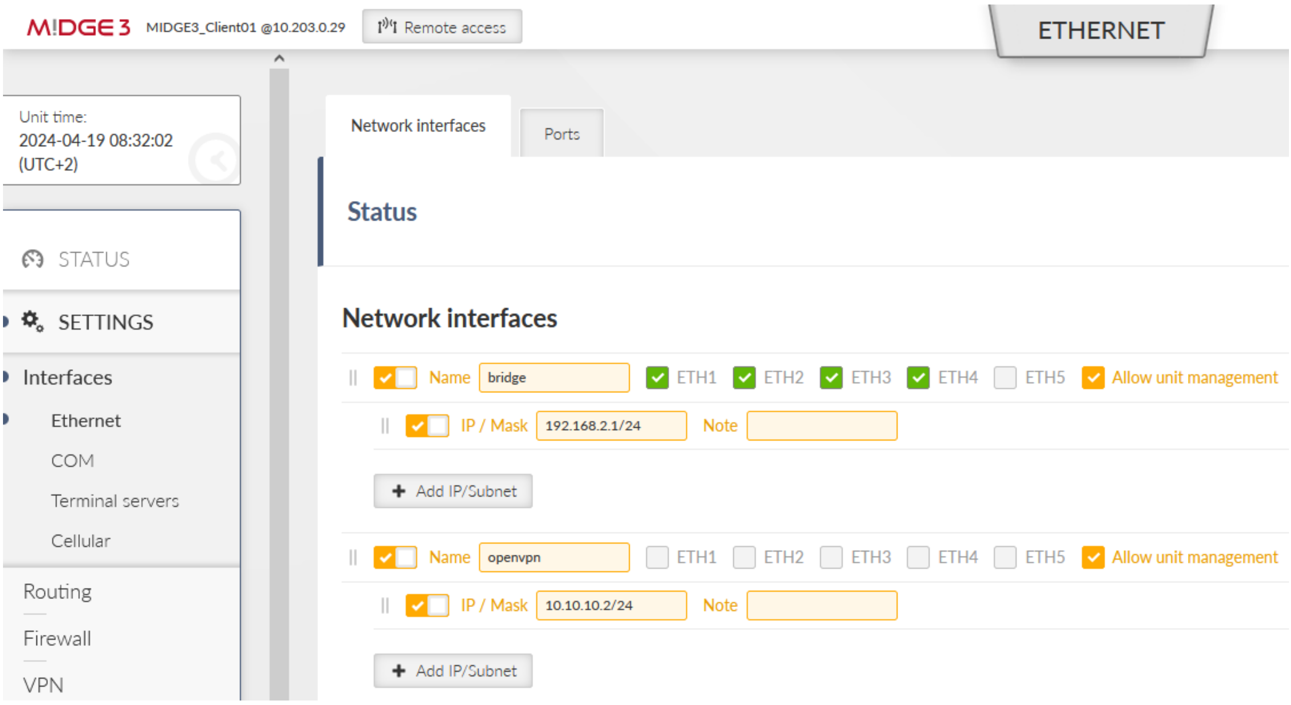
Fig. 102: M!DGE3_Server Babel Static rules

Save the changes.

5.1.2. MIDGE3_Client01

It is very the same configuration as in the Master unit.

OpenVPN Ethernet interface used for L2 bridge.



Network interfaces

☑ Name: bridge ☑ ETH1 ☑ ETH2 ☑ ETH3 ☑ ETH4 □ ETH5 ☑ Allow unit management

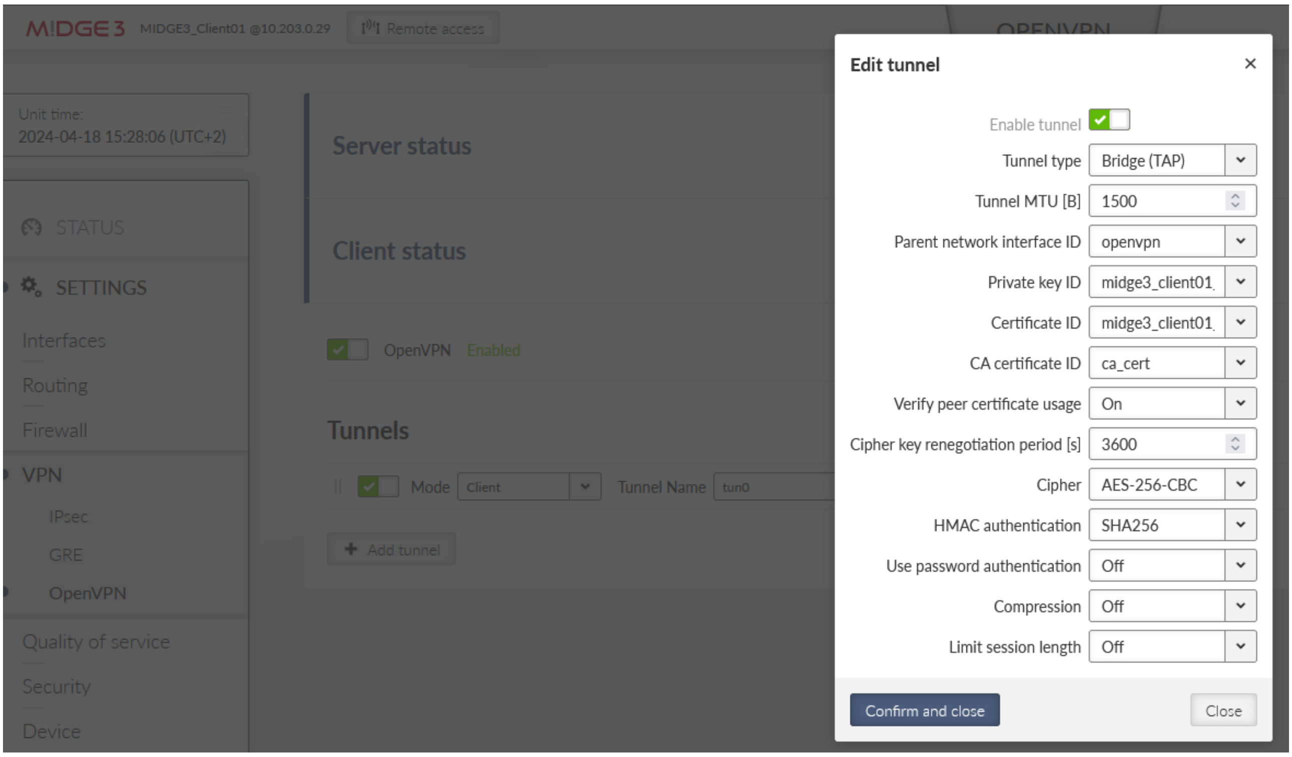
☑ IP / Mask: 192.168.2.1/24 Note:

☑ Name: openvpn □ ETH1 □ ETH2 □ ETH3 □ ETH4 □ ETH5 ☑ Allow unit management

☑ IP / Mask: 10.10.10.2/24 Note:

Fig. 103: M!DGE3_Client01 Ethernet settings

Configure OpenVPN tunnel appropriately. Select the Bridge (TAP) type and set the 'openvpn' Parent network interface ID.

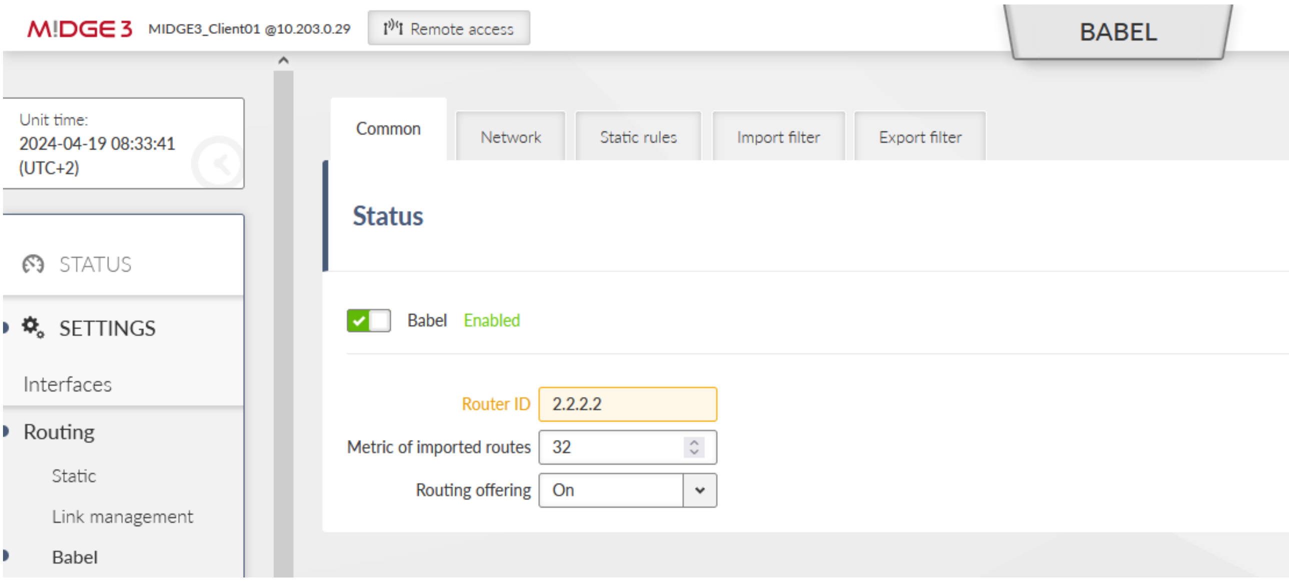


Edit tunnel

Enable tunnel	☑
Tunnel type	Bridge (TAP)
Tunnel MTU [B]	1500
Parent network interface ID	openvpn
Private key ID	midge3_client01
Certificate ID	midge3_client01
CA certificate ID	ca_cert
Verify peer certificate usage	On
Cipher key renegotiation period [s]	3600
Cipher	AES-256-CBC
HMAC authentication	SHA256
Use password authentication	Off
Compression	Off
Limit session length	Off

Fig. 104: M!DGE3_Client01 OpenVPN changes

Continue within the Babel settings. Set the ID to 2.2.2.2.



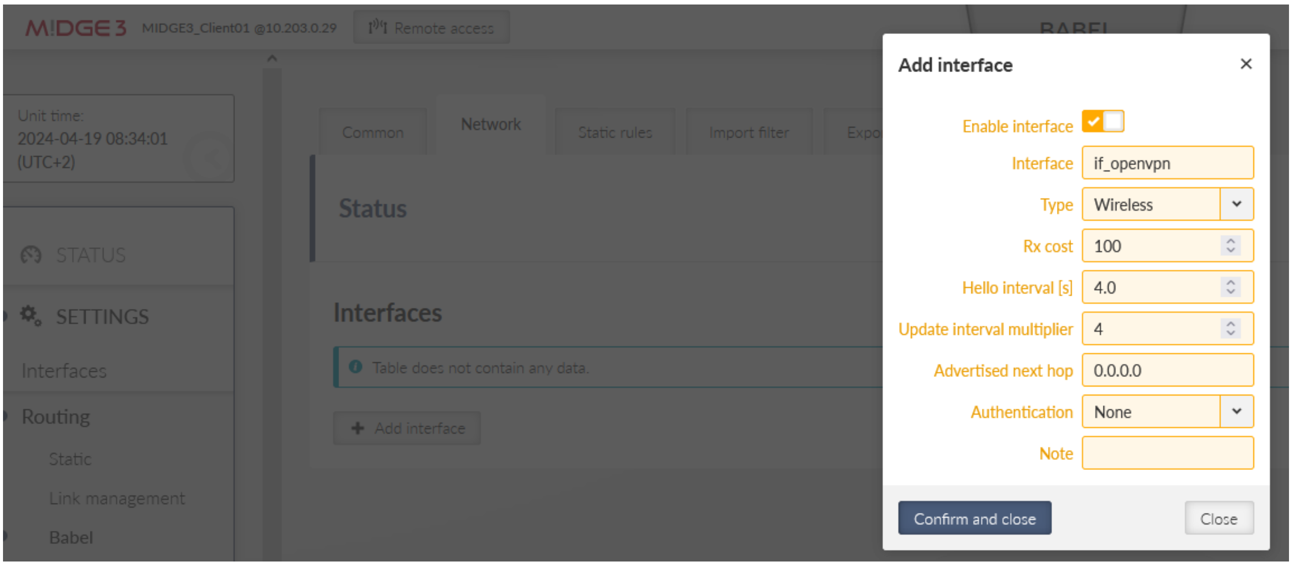
Status

Babel Enabled

Router ID	2.2.2.2
Metric of imported routes	32
Routing offering	On

Fig. 105: M!DGE3_Client01 Babel Common settings

Network tab is the same as in the Master unit.



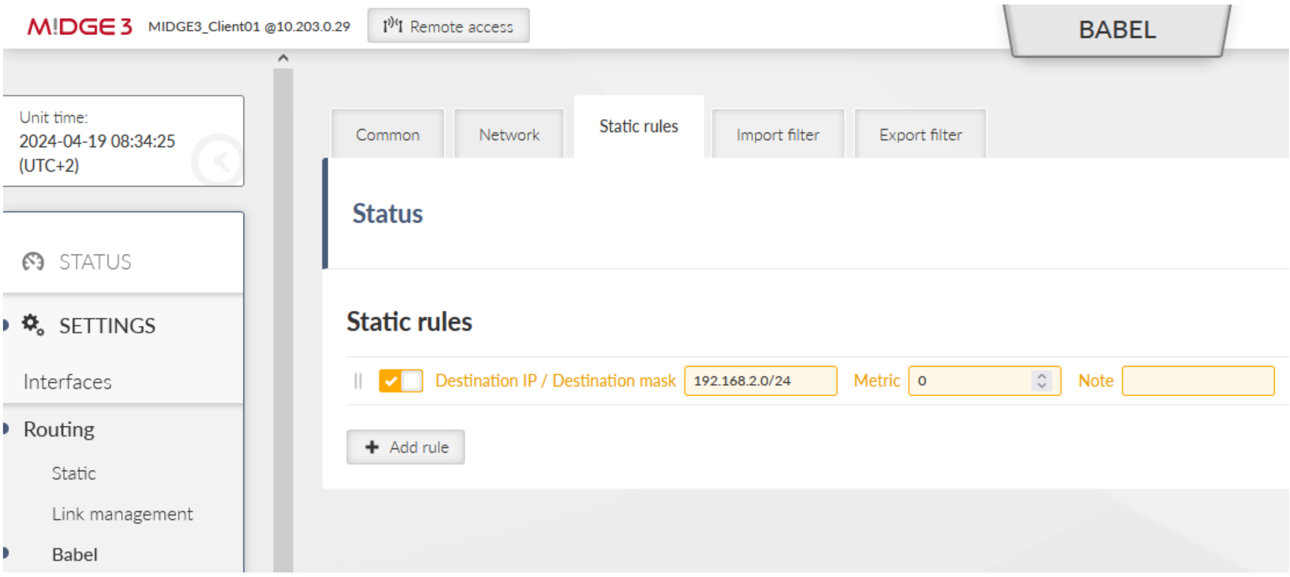
Add interface

☑ Enable interface

Interface	if_openvpn
Type	Wireless
Rx cost	100
Hello interval [s]	4.0
Update interval multiplier	4
Advertised next hop	0.0.0.0
Authentication	None
Note	

Fig. 106: M!DGE3_Client01 Babel Network settings

The Static rule exports local LAN 192.168.2.0/24.



Destination IP / Destination mask: 192.168.2.0/24
Metric: 0
Note:

Fig. 107: M!DGE3_Client01 Babel Static rules

Save the changes.

5.1.3. RipEX2_Client02

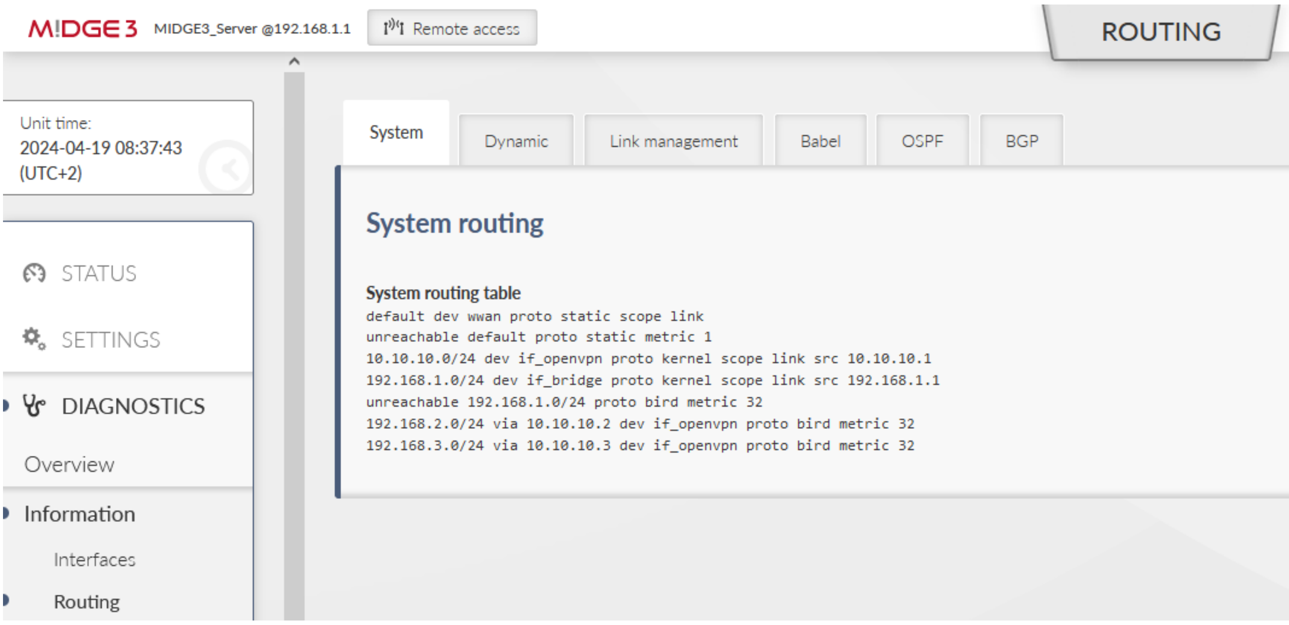
2nd client configuration is the same as the 1st client.

- Just set the correct 'openvpn' IP to 10.10.10.3/24 this time (SETTINGS > Interfaces > Ethernet menu).
- Change the OpenVPN settings.
- Local LAN is 192.168.3.0/24 - configure it within the Babel setup, set the Babel ID to 3.3.3.3

Save the changes.

5.1.4. Diagnostics

Check the ROUTING RULES in all the units. E.g. the server tables:



System routing

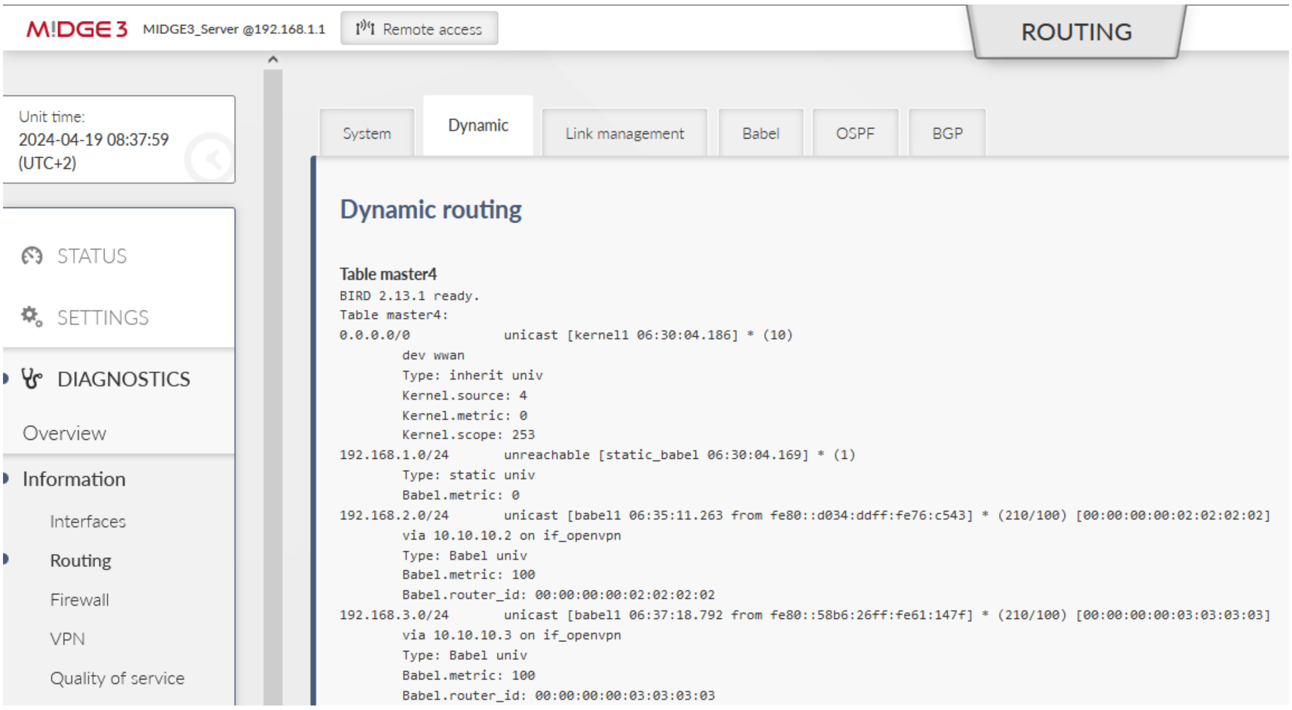
System routing table

```
default dev wwan proto static scope link
unreachable default proto static metric 1
10.10.10.0/24 dev if_openvpn proto kernel scope link src 10.10.10.1
192.168.1.0/24 dev if_bridge proto kernel scope link src 192.168.1.1
unreachable 192.168.1.0/24 proto bird metric 32
192.168.2.0/24 via 10.10.10.2 dev if_openvpn proto bird metric 32
192.168.3.0/24 via 10.10.10.3 dev if_openvpn proto bird metric 32
```

Fig. 108: M!DGE3_Server Diagnostics - System routing

You should see 192.168.2.0/24 and 192.168.3.0/24 to be reachable via the correct 'openvpn' IP address, dev 'if_openvpn', proto bird, metric 32.

More details within the Dynamic and/or Babel tabs:



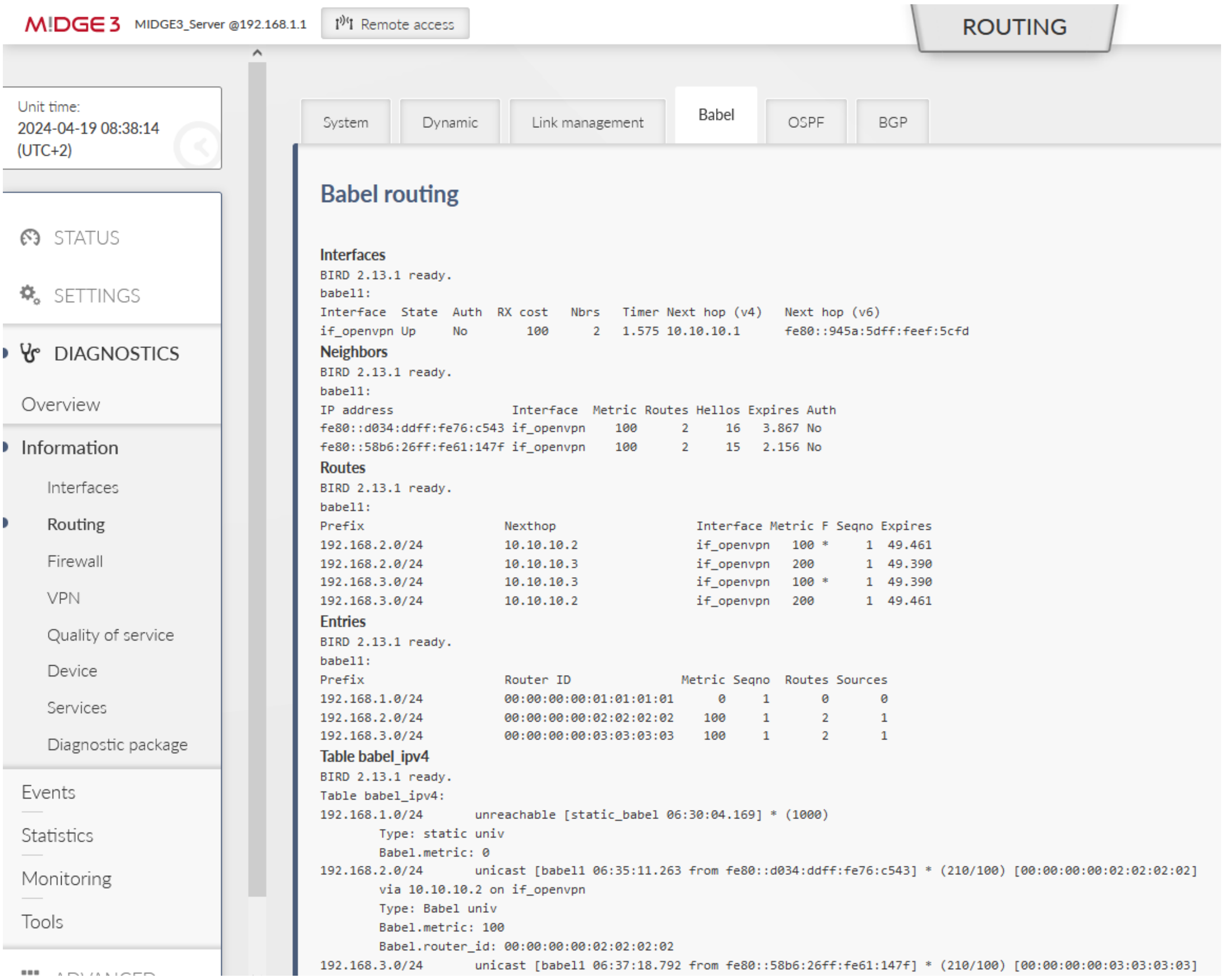
Dynamic routing

Table master4

```
BIRD 2.13.1 ready.
Table master4:
0.0.0.0/0            unicast [kernel1 06:30:04.186] * (10)
        dev wwan
        Type: inherit univ
        Kernel.source: 4
        Kernel.metric: 0
        Kernel.scope: 253
192.168.1.0/24       unreachable [static_babel 06:30:04.169] * (1)
        Type: static univ
        Babel.metric: 0
192.168.2.0/24       unicast [babel1 06:35:11.263 from fe80::d034:ddff:fe76:c543] * (210/100) [00:00:00:00:02:02:02:02]
        via 10.10.10.2 on if_openvpn
        Type: Babel univ
        Babel.metric: 100
        Babel.router_id: 00:00:00:00:02:02:02:02
192.168.3.0/24       unicast [babel1 06:37:18.792 from fe80::58b6:26ff:fe61:147f] * (210/100) [00:00:00:00:03:03:03:03]
        via 10.10.10.3 on if_openvpn
        Type: Babel univ
        Babel.metric: 100
        Babel.router_id: 00:00:00:00:03:03:03:03
```

Fig. 109: M!DGE3_Server Diagnostics - Dynamic routing

and



Babel routing

Interfaces

```
BIRD 2.13.1 ready.
babel1:
Interface  State  Auth  RX cost   Nbrs   Timer Next hop (v4)   Next hop (v6)
if_openvpn Up     No        100      2   1.575 10.10.10.1      fe80::945a:5dff:feef:5cfd
```

Neighbors

```
BIRD 2.13.1 ready.
babel1:
IP address                Interface  Metric Routes Hellos Expires Auth
fe80::d034:ddff:fe76:c543 if_openvpn    100      2     16   3.867 No
fe80::58b6:26ff:fe61:147f if_openvpn    100      2     15   2.156 No
```

Routes

```
BIRD 2.13.1 ready.
babel1:
Prefix                   Nexthop                   Interface Metric F Seqno Expires
192.168.2.0/24           10.10.10.2                if_openvpn   100 *     1  49.461
192.168.2.0/24           10.10.10.3                if_openvpn   200       1  49.390
192.168.3.0/24           10.10.10.3                if_openvpn   100 *     1  49.390
192.168.3.0/24           10.10.10.2                if_openvpn   200       1  49.461
```

Entries

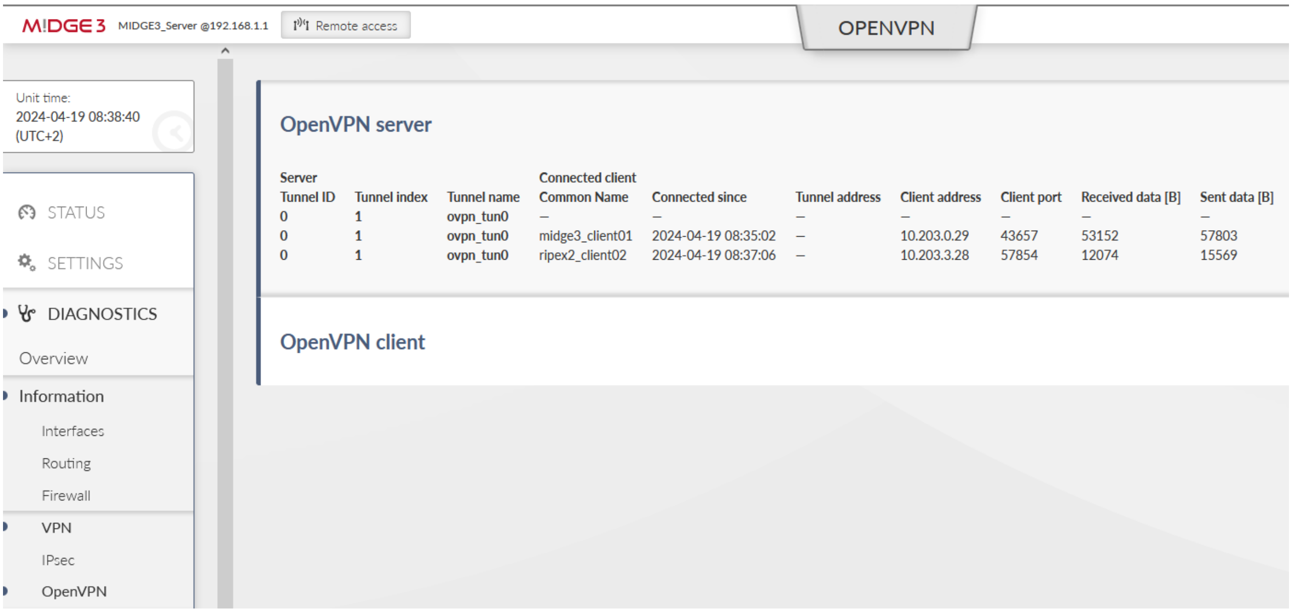
```
BIRD 2.13.1 ready.
babel1:
Prefix                   Router ID               Metric Seqno  Routes Sources
192.168.1.0/24           00:00:00:00:01:01:01:01      0     1       0       0
192.168.2.0/24           00:00:00:00:02:02:02:02    100     1       2       1
192.168.3.0/24           00:00:00:00:03:03:03:03    100     1       2       1
```

Table babel_ipv4

```
BIRD 2.13.1 ready.
Table babel_ipv4:
192.168.1.0/24       unreachable [static_babel 06:30:04.169] * (1000)
        Type: static univ
        Babel.metric: 0
192.168.2.0/24       unicast [babel1 06:35:11.263 from fe80::d034:ddff:fe76:c543] * (210/100) [00:00:00:00:02:02:02:02]
        via 10.10.10.2 on if_openvpn
        Type: Babel univ
        Babel.metric: 100
        Babel.router_id: 00:00:00:00:02:02:02:02
192.168.3.0/24       unicast [babel1 06:37:18.792 from fe80::58b6:26ff:fe61:147f] * (210/100) [00:00:00:00:03:03:03:03]
```

Fig. 110: M!DGE3_Server Diagnostics - Babel routing

Within the OpenVPN, you definitely should see both clients to be connected.



Unit time:
2024-04-19 08:38:40
(UTC+2)

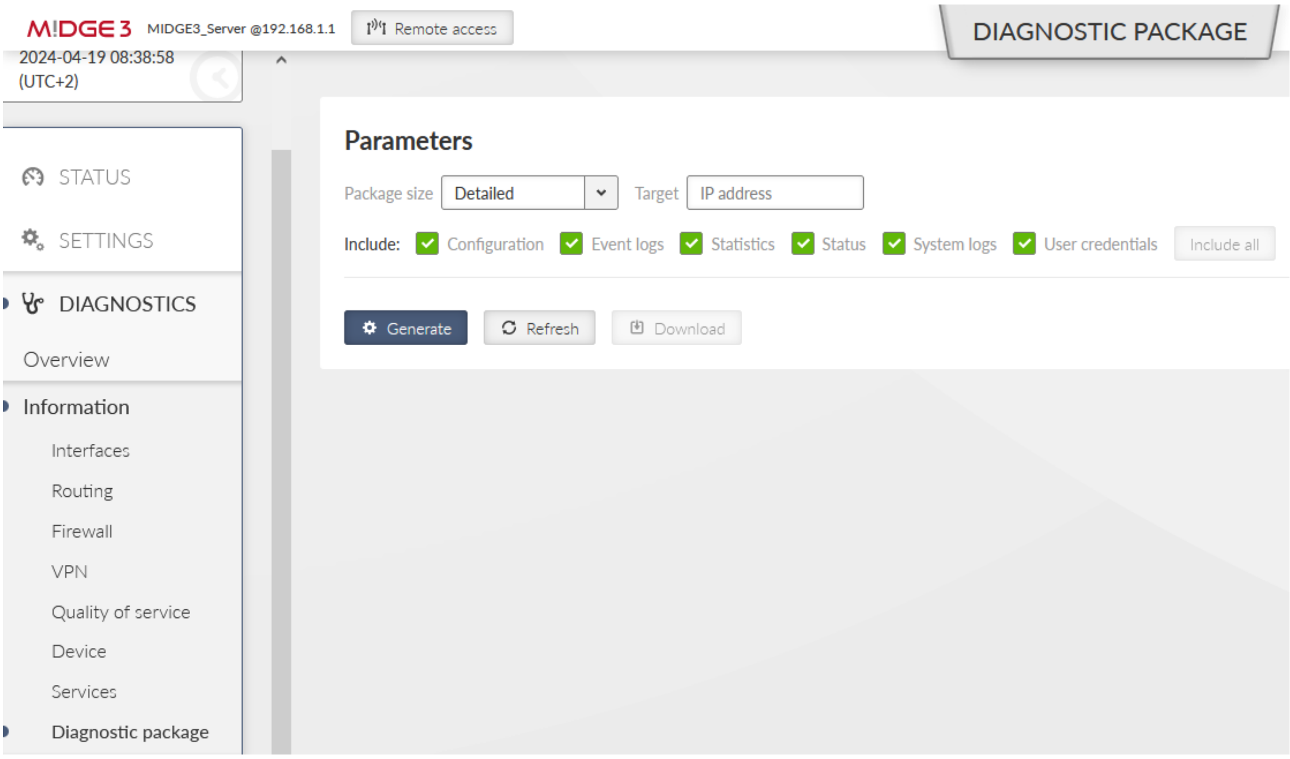
OpenVPN server

Server			Connected client						
Tunnel ID	Tunnel index	Tunnel name	Common Name	Connected since	Tunnel address	Client address	Client port	Received data [B]	Sent data [B]
0	1	ovpn_tun0	–	–	–	–	–	–	–
0	1	ovpn_tun0	midge3_client01	2024-04-19 08:35:02	–	10.203.0.29	43657	53152	57803
0	1	ovpn_tun0	ripex2_client02	2024-04-19 08:37:06	–	10.203.3.28	57854	12074	15569

OpenVPN client

Fig. 111: M!DGE3_Server Diagnostics - OpenVPN status

If you encounter any trouble, download the Diagnostic package, extract the status and logs and try to find out the reason for your issues. Doublecheck your configuration.



2024-04-19 08:38:58
(UTC+2)

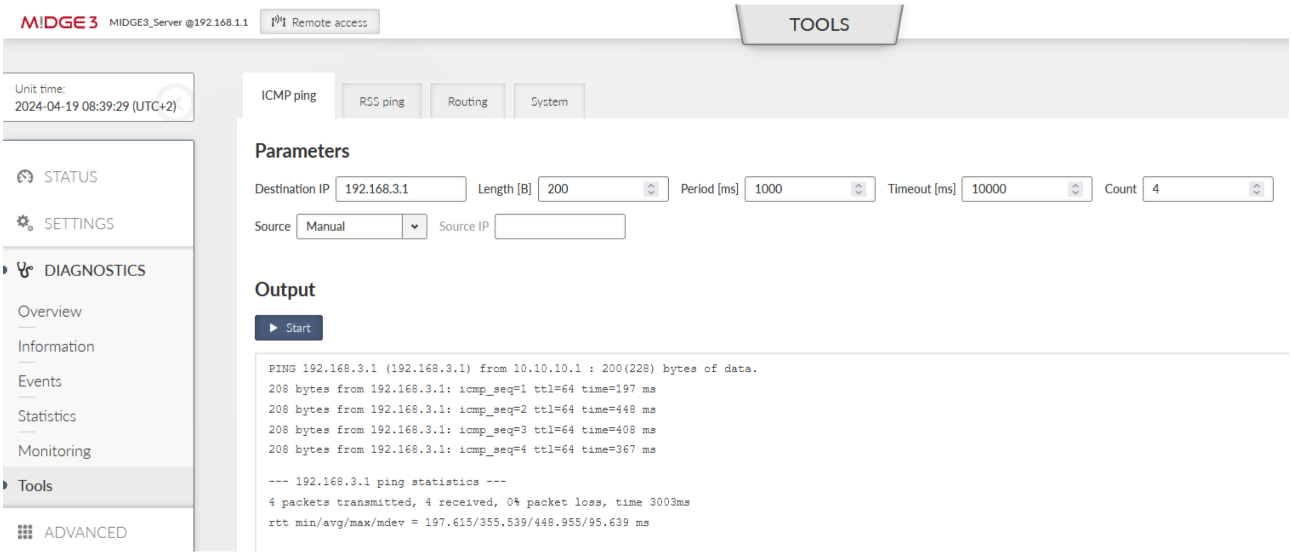
Parameters

Package size: Detailed Target: IP address

Include: ☑ Configuration ☑ Event logs ☑ Statistics ☑ Status ☑ System logs ☑ User credentials Include all

Fig. 112: M!DGE3_Server Diagnostics - Diagnostic package

We also suggest to verify the accessibility of M!DGE3/RipEX2 units and all its local devices via ICMP ping Tool. E.g.



Parameters

Destination IP 192.168.3.1 Length [B] 200 Period [ms] 1000 Timeout [ms] 10000 Count 4

Source Manual Source IP

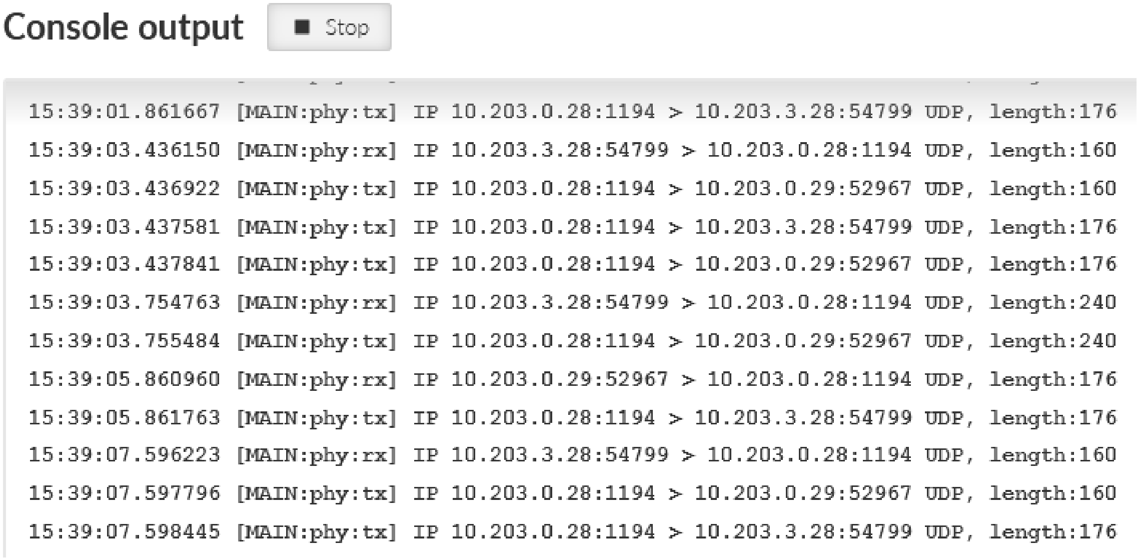
Output

```
PING 192.168.3.1 (192.168.3.1) from 10.10.10.1 : 200(228) bytes of data.
208 bytes from 192.168.3.1: icmp_seq=1 ttl=64 time=197 ms
208 bytes from 192.168.3.1: icmp_seq=2 ttl=64 time=448 ms
208 bytes from 192.168.3.1: icmp_seq=3 ttl=64 time=408 ms
208 bytes from 192.168.3.1: icmp_seq=4 ttl=64 time=367 ms

--- 192.168.3.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3003ms
rtt min/avg/max/mdev = 197.615/355.539/448.955/95.639 ms
```

Fig. 113: M!DGE3_Server Diagnostics - ICMP ping

You may also debug some issues via Monitoring - e.g. some not encrypted traffic instead of encrypted etc.



Console output

```
15:39:01.861667 [MAIN:phy:tx] IP 10.203.0.28:1194 > 10.203.3.28:54799 UDP, length:176
15:39:03.436150 [MAIN:phy:rx] IP 10.203.3.28:54799 > 10.203.0.28:1194 UDP, length:160
15:39:03.436922 [MAIN:phy:tx] IP 10.203.0.28:1194 > 10.203.0.29:52967 UDP, length:160
15:39:03.437581 [MAIN:phy:tx] IP 10.203.0.28:1194 > 10.203.3.28:54799 UDP, length:176
15:39:03.437841 [MAIN:phy:tx] IP 10.203.0.28:1194 > 10.203.0.29:52967 UDP, length:176
15:39:03.754763 [MAIN:phy:rx] IP 10.203.3.28:54799 > 10.203.0.28:1194 UDP, length:240
15:39:03.755484 [MAIN:phy:tx] IP 10.203.0.28:1194 > 10.203.0.29:52967 UDP, length:240
15:39:05.860960 [MAIN:phy:rx] IP 10.203.0.29:52967 > 10.203.0.28:1194 UDP, length:176
15:39:05.861763 [MAIN:phy:tx] IP 10.203.0.28:1194 > 10.203.3.28:54799 UDP, length:176
15:39:07.596223 [MAIN:phy:rx] IP 10.203.3.28:54799 > 10.203.0.28:1194 UDP, length:160
15:39:07.597796 [MAIN:phy:tx] IP 10.203.0.28:1194 > 10.203.0.29:52967 UDP, length:160
15:39:07.598445 [MAIN:phy:tx] IP 10.203.0.28:1194 > 10.203.3.28:54799 UDP, length:176
```

Fig. 114: M!DGE3_Server Diagnostics - Monitoring

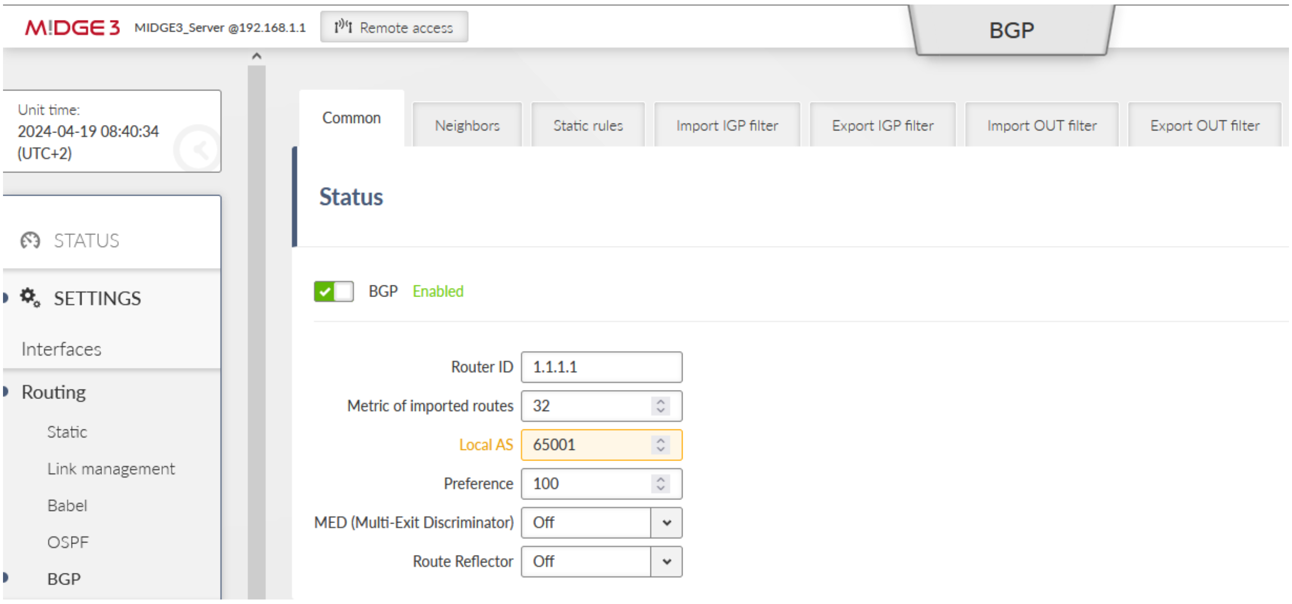
There should be UDP/1194 data for the OpenVPN traffic.

5.2. BGP dynamic routing

In case you prefer or you need to use BGP instead of Babel, follow the guide below.

5.2.1. M!DGE3_Server

We start in the Server M!DGE3. Disable the Babel protocol and enable BGP.



BGP Enabled

Router ID	1.1.1.1
Metric of imported routes	32
Local AS	65001
Preference	100
MED (Multi-Exit Discriminator)	Off
Route Reflector	Off

Fig. 115: M!DGE3_Server BGP Common settings

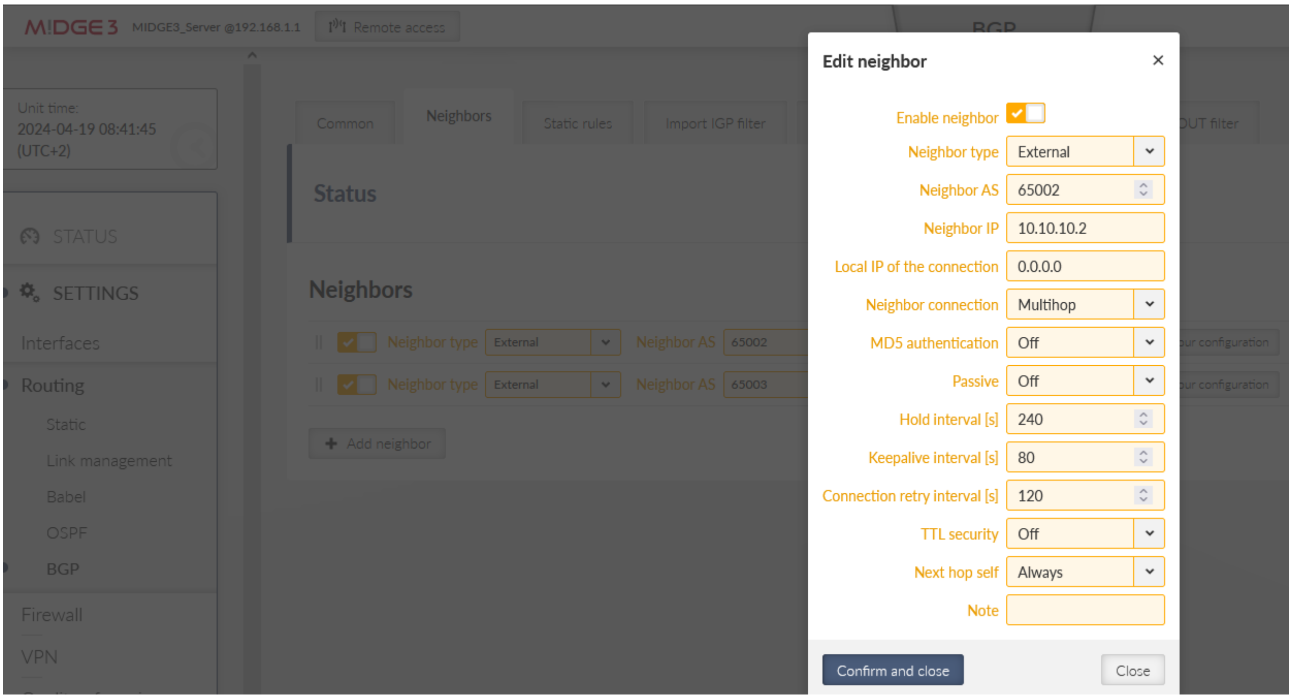
Keep the ID 1.1.1.1. Set the local AS to be 65001.

Go the the next tab Neighbors and add both neighbours. First one:

- Neighbor type: external
- Neighbor AS: 65002
- Neighbor IP: 10.10.10.2 (client01's openvpn IP)
- Neighbor connection: multihop
- Next hop self: always

Second one:

- Neighbor type: external
- Neighbor AS: 65003
- Neighbor IP: 10.10.10.3 (client02's openvpn IP)
- Neighbor connection: multihop
- Next hop self: always



Neighbors:

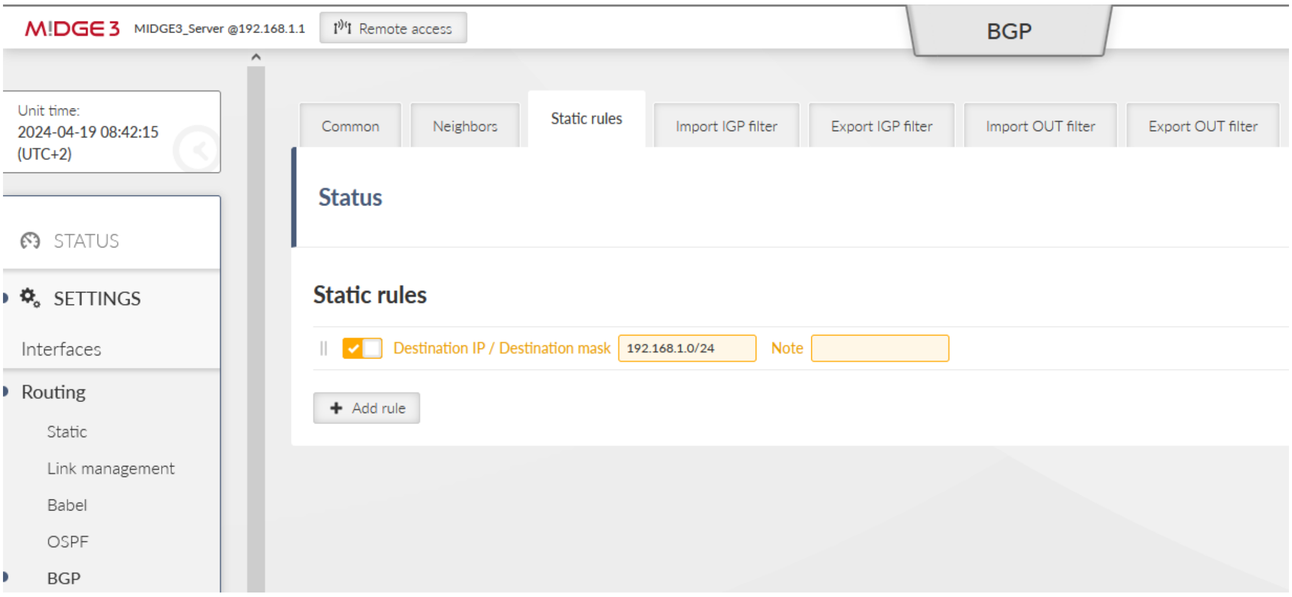
	Neighbor type	Neighbor AS
☑	External	65002
☑	External	65003

Edit neighbor

Enable neighbor	☑
Neighbor type	External
Neighbor AS	65002
Neighbor IP	10.10.10.2
Local IP of the connection	0.0.0.0
Neighbor connection	Multihop
MD5 authentication	Off
Passive	Off
Hold interval [s]	240
Keepalive interval [s]	80
Connection retry interval [s]	120
TTL security	Off
Next hop self	Always
Note	

Fig. 116: M!DGE3_Server BGP Neighbors settings

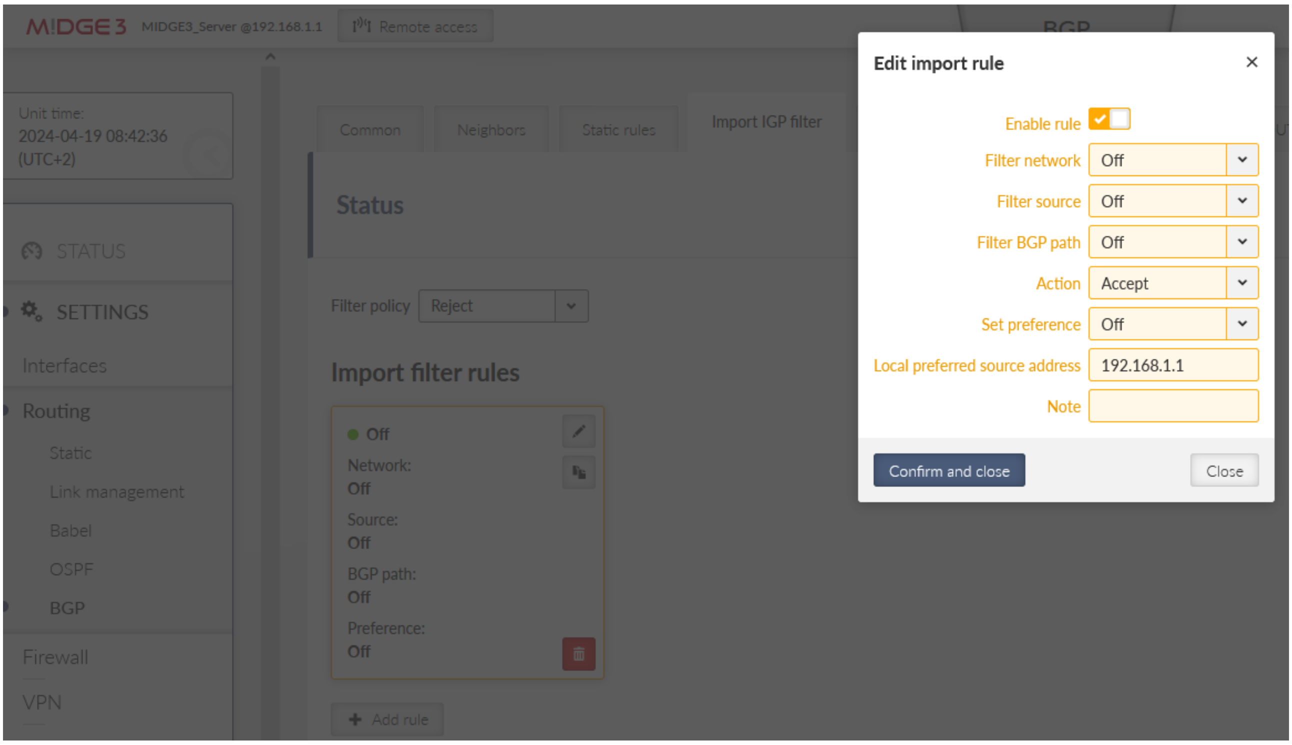
Within the Static route tab, fill in the local LAN 192.168.1.0/24.



	Destination IP / Destination mask	Note
☑	192.168.1.0/24	

Fig. 117: M!DGE3_Server BGP Static rules

And set the Import IGP filter for our local LAN IP.



Filter policy: Reject

Edit import rule

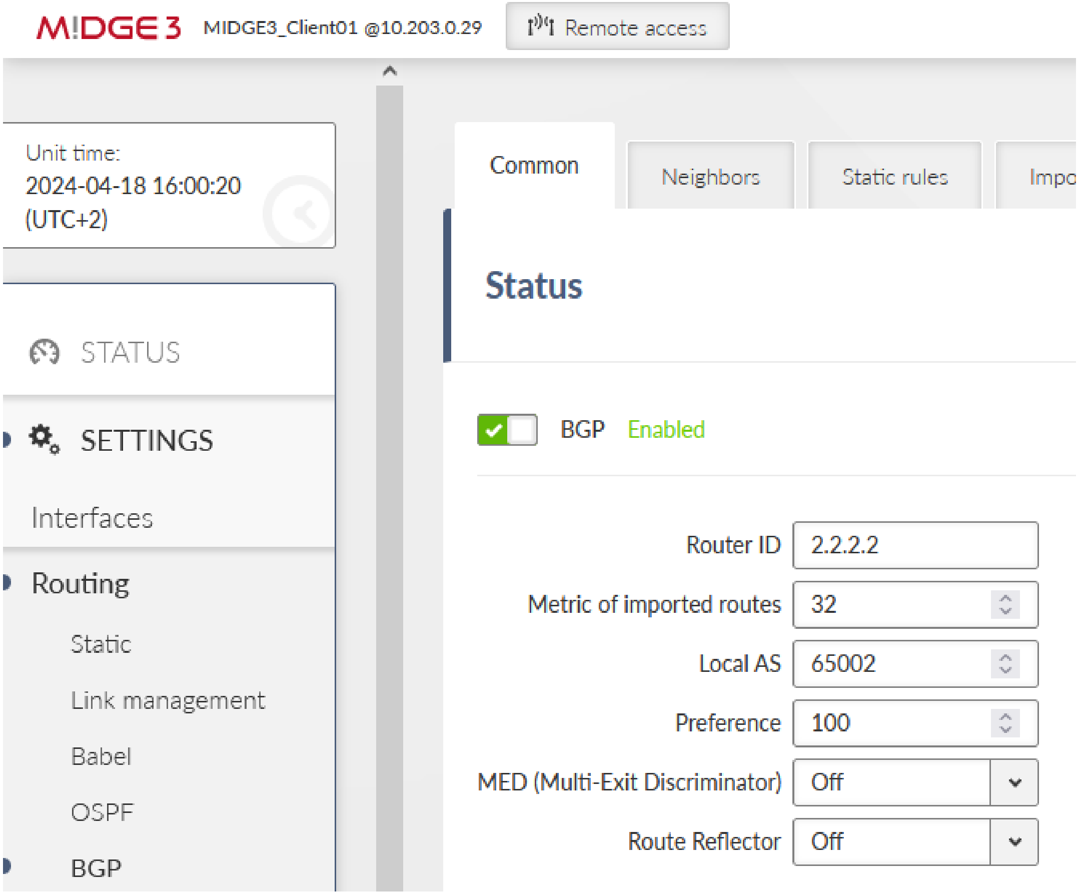
Enable rule	☑
Filter network	Off
Filter source	Off
Filter BGP path	Off
Action	Accept
Set preference	Off
Local preferred source address	192.168.1.1
Note	

Fig. 118: M!DGE3_Server BGP Import IGP filter settings

Save the changes.

5.2.2. M!DGE3_Client01

Disable Babel and enable BGP as well. Set the local AS to 65002.



BGP Enabled

Router ID 2.2.2.2

Metric of imported routes 32

Local AS 65002

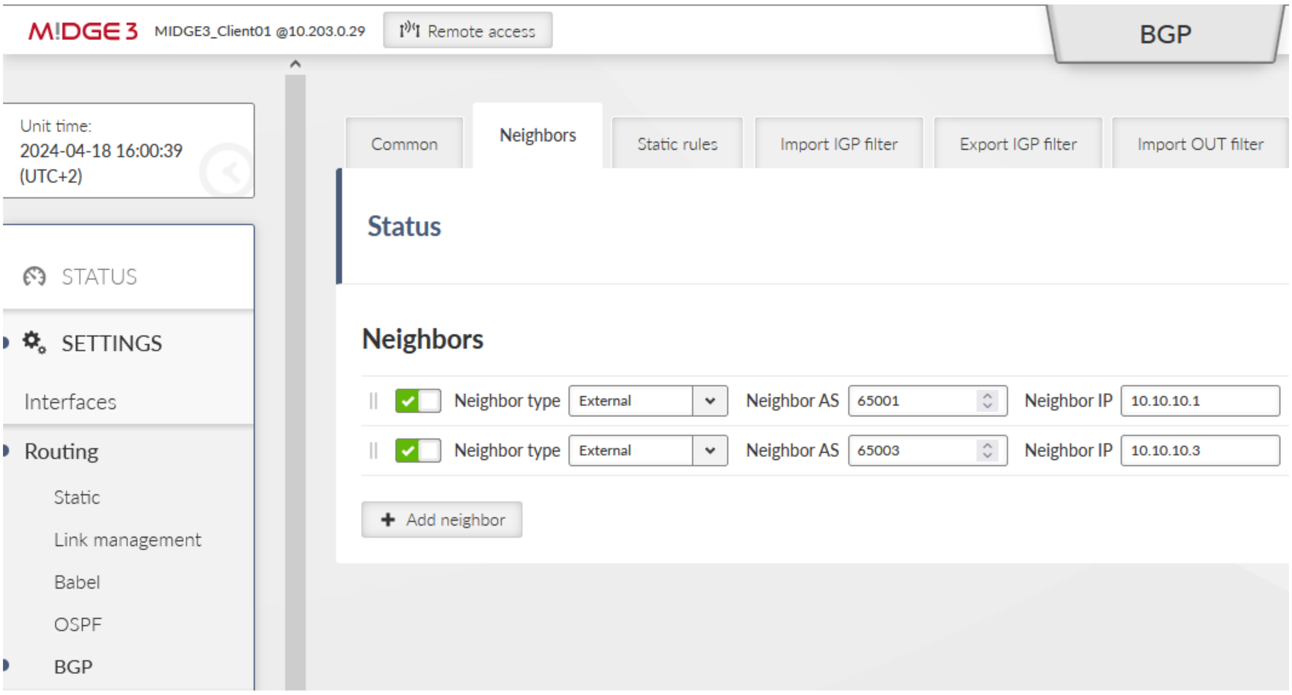
Preference 100

MED (Multi-Exit Discriminator) Off

Route Reflector Off

Fig. 119: M!DGE3_Client01 BGP Common settings

Neighbors:

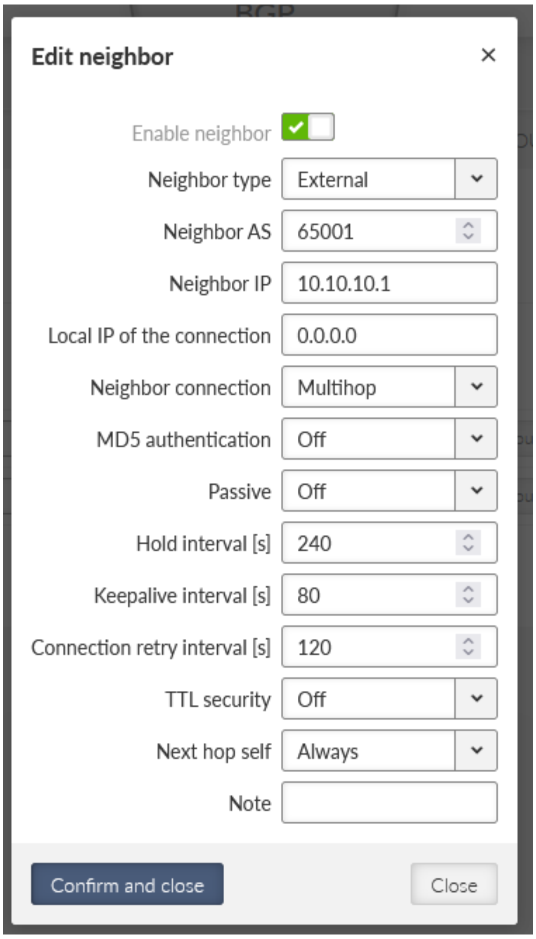


Neighbors

	Neighbor type	Neighbor AS	Neighbor IP
✓	External	65001	10.10.10.1
✓	External	65003	10.10.10.3

Fig. 120: M!DGE3_Client01 BGP Neighbors settings

Details for the 1st neighbor:



Edit neighbor

Enable neighbor ☑

Neighbor type: External

Neighbor AS: 65001

Neighbor IP: 10.10.10.1

Local IP of the connection: 0.0.0.0

Neighbor connection: Multihop

MD5 authentication: Off

Passive: Off

Hold interval [s]: 240

Keepalive interval [s]: 80

Connection retry interval [s]: 120

TTL security: Off

Next hop self: Always

Note:

Fig. 121: M!DGE3_Client01 BGP Neighbors details

Static rule to be exported is 192.168.2.0/24. Import IGP filter is our local LAN IP 192.168.2.1.

5.2.3. M!DGE3_Client02

Do the same for the 2nd client. Set its AS number to 65003.

5.2.4. Diagnostics

Do the similar diagnostics as for the Babel dynamic routing.

6. Compatibility with Linux or Windows servers/clients

The OpenVPN implementation uses generally know protocol and thus, it is compatible with any 3rd party device, including Linux and Windows servers/clients. So e.g., you can successfully connect from your laptop to the M!DGE3/RipEX2 via OpenVPN and get a secure connection to the complete network.

Within the 2.1.1.0 firmware, there is no option to export the .conf, .ovpn, .p12 and similar OpenVPN configuration files with or without the certificates. The same is valid for importing such configuration files. All must be done manually.

If you configure the Windows/Linux client, just configure the tunnel parameters to match the server settings (cipher, HMAC authentication, protocol, …) and upload valid credentials which can be downloaded from M!DGE3/RipEX2 Credentials menu.

Revision History

Revision 1.0	2024-03-13

First issue

Revision 1.1	2024-11-20

Added *Section 3.3, “M!DGE2 OpenVPN server migration to M!DGE3”*

Added *Section 5, “Dynamic routing over OpenVPN L2”*

Revision 1.2	2024-12-09

Minor text and stylistic improvements