# Application Note

# GRE Tunnel

Advantech Czech s.r.o., Sokolska 71, 562 04 Usti nad Orlici, Czech Republic
Document No. APP-0011-EN, revision from 15th November, 2024.

# Used symbols

⚠ Danger – Information regarding user safety or potential damage to the router.

❗ Attention – Problems that can arise in specific situations.

ℹ Information – Useful tips or information of special interest.

# Contents

# List of Figures

# List of Tables

# 1. GRE Protocol

*Generic Routing Encapsulation* (GRE) is a tunneling protocol developed by Cisco Systems that can encapsulate a wide variety of network layer protocols inside virtual point-to-point links over an Internet Protocol internetwork. GRE tunnel creates a connection of two LANs into one, looking from inside as homogeneous. GRE is used when IP packets need to be sent from one network to another, without being parsed or treated like IP packets by any intervening routers.

GRE encapsulates the original data – inner packet meant for deliver to the remote network – into the outer packet. This packet is sent through the GRE tunnel, intervening routers are routing it as the outer packet into the destination network where outer packet is removed and original packet is routed to the target. Unlike IP-to-IP tunnel, the GRE tunnel can be used for the transport of multicast and IPv6 packets between connected networks. Following picture graphically displays the principle of the GRE tunnel (on the left) and an example of encapsulation of IPv6 packets for transport through IPv4 network (on the right).

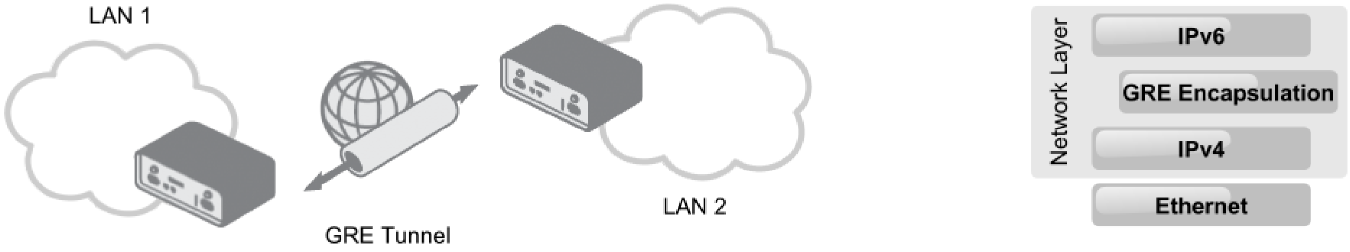


Figure 1: Principle of the GRE tunnel, encapsulation example

**RE protocol advantages:** GRE tunnels encase multiple protocols over a single-protocol backbone, GRE tunnels provide workarounds for networks with limited hops, GRE tunnels connect discontinuous sub-networks, GRE tunnels allow VPNs across wide area networks (WANs).

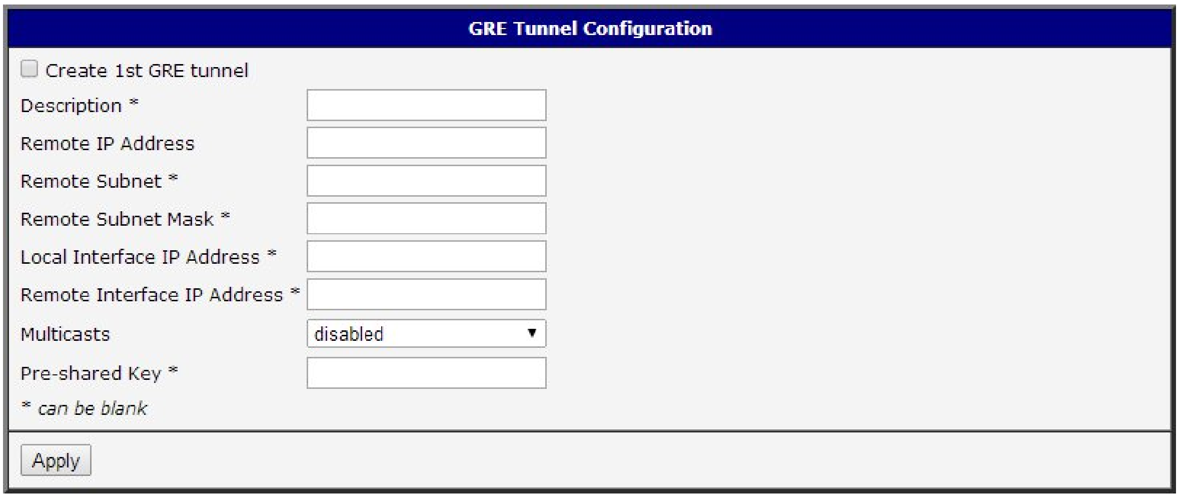
**Examples of the GRE protocol usage:** In conjunction with PPTP to create VPNs, in conjunction with IPsec VPNs to allow passing of routing information between connected networks, in Mobility protocols, Linux and BSD can establish ad-hoc IP over GRE tunnels which are interoperable with Cisco equipment.

GRE protocol provides stateless private connection, but is not encrypted (secured) protocol. It doesn't use any encryption like e.g. ESP (Encapsulating Security Payload) in IPsec protocol. GRE protocol is specified in RFC 2784 and RFC 2890. It is determined by number 47 in the Protocol field in the IP header.

# 2. GRE Tunnel Configuration

It is possible to configure up to four GRE tunnels. To open the GRE tunnel configuration page, click the *GRE* menu item in the *Configuration* section. The menu item will expand and you will see four separate configuration pages: *1st Tunnel*, *2nd Tunnel*, *3rd Tunnel* and *4th Tunnel*.

Only IPv4 tunnels are supported in Advantech routers.



**GRE Tunnel Configuration**

□ Create 1st GRE tunnel

| | |
|---|---|
| Description * | |
| Remote IP Address | |
| Remote Subnet * | |
| Remote Subnet Mask * | |
| Local Interface IP Address * | |
| Remote Interface IP Address * | |
| Multicasts | disabled ▾ |
| Pre-shared Key * | |

** can be blank*

Figure 2: GRE tunnel configuration

There are possible settings for every one of four GRE tunnels in the figure 2. The tunnel can be activated by checking the *Create 1st GRE tunnel* item. The items of settings are following:

| Item | Description |
|---|---|
| Description | Optional description of the tunnel. |
| Remote IP Address | IP address of the remote side of the tunnel |
| Local Interface IP Address | IP address of the local side of the tunnel |
| Remote Interface IP Address | IP address of the remote side of the tunnel |
| Remote Subnet | IP address of the network behind the remote side of the tunnel |
| Remote Subnet Mask | Mask of the network behind the remote side of the tunnel |
| Multicasts | Enables/disables multicast:<br>• **disabled** – multicast disabled<br>• **enabled** – multicast enabled |
| Pre-shared Key | An optional value that defines the 32 bit shared key in numeric format, through which the filtered data through the tunnel. This key must be defined on both routers as same, otherwise the router will drop received packets. Using this key, the data do not provide a tunnel through. |

Table 1: GRE tunnel configuration

All the changes in settings will apply after pressing the *Apply* button.

**Attention, GRE tunnel doesn't connect itself via NAT.** If you need to create tunnel through NAT, use IP-to-IP tunnel (IP packets encapsulated to IP packets) or GRE over IPsec (secured IPsec tunnel and then GRE encapsulation inside of the IPsec tunnel).

# 3. GRE Configuration Examples

## 3.1 GRE Tunnel Between Advantech Routers

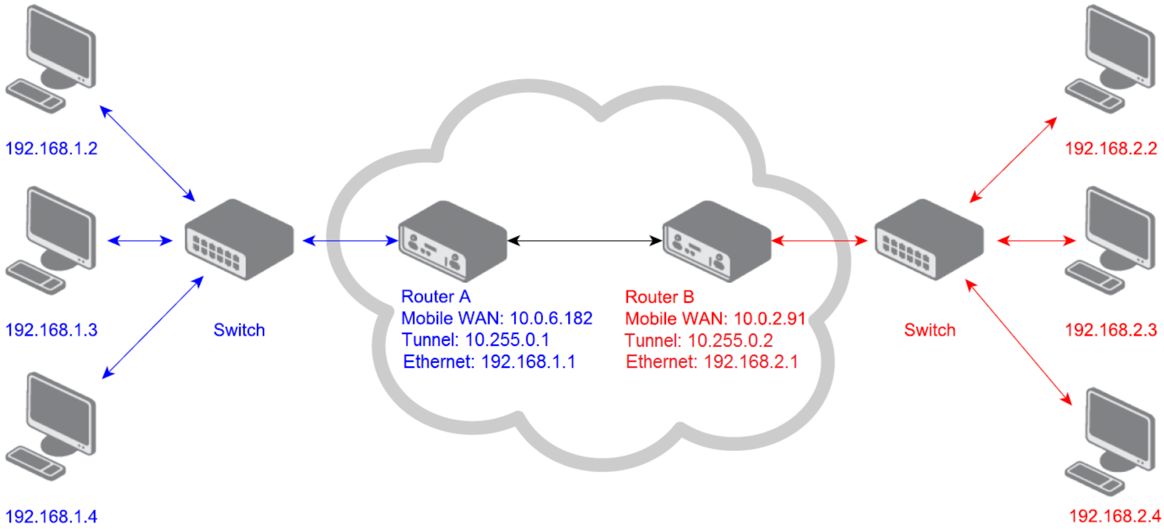
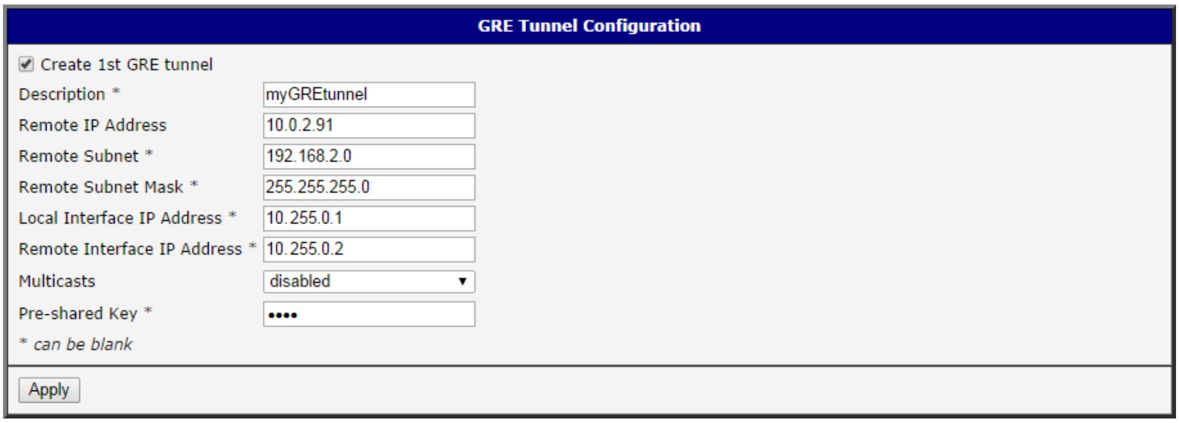


Figure 3: Topology of the Advantech to Advantech router configuration example

This is the example how to connect two LANs via GRE tunnel between two Advantech routers. The default gateway for stations in the blue network will be the Router A, for stations in the red network it will be the Router B. GRE tunnel parameters set on both routers are shown on the next figures:



**GRE Tunnel Configuration**

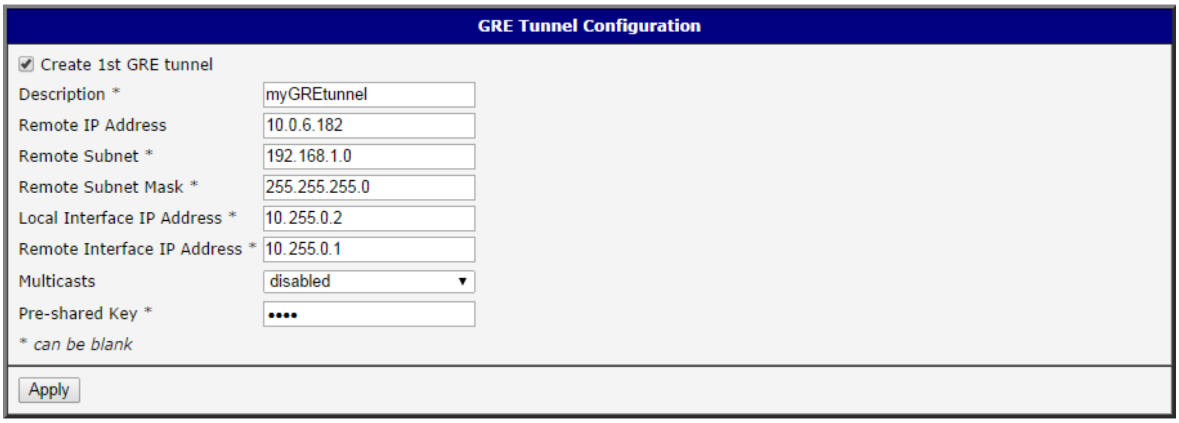
☑ Create 1st GRE tunnel

| | |
|---|---|
| Description * | myGREtunnel |
| Remote IP Address | 10.0.2.91 |
| Remote Subnet * | 192.168.2.0 |
| Remote Subnet Mask * | 255.255.255.0 |
| Local Interface IP Address * | 10.255.0.1 |
| Remote Interface IP Address * | 10.255.0.2 |
| Multicasts | disabled ▾ |
| Pre-shared Key * | •••• |

* *can be blank*

Figure 4: Router A (blue network) – GRE tunnel configuration

After the GRE tunnel activation, there will be created the new network interface "gre1" in every router. It can be viewed in the *Network* item in the *Status* section – see the figure:



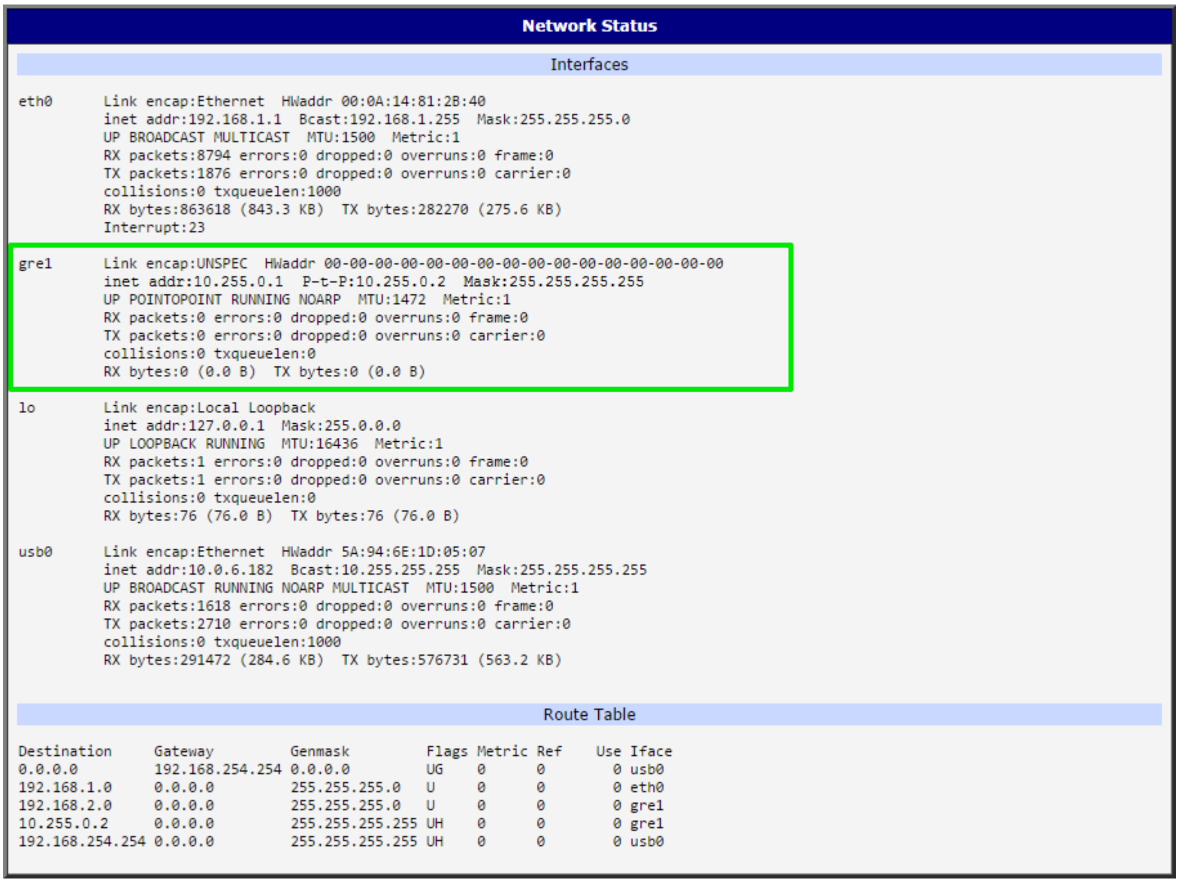
**GRE Tunnel Configuration**

☑ Create 1st GRE tunnel

| | |
|---|---|
| Description * | myGREtunnel |
| Remote IP Address | 10.0.6.182 |
| Remote Subnet * | 192.168.1.0 |
| Remote Subnet Mask * | 255.255.255.0 |
| Local Interface IP Address * | 10.255.0.2 |
| Remote Interface IP Address * | 10.255.0.1 |
| Multicasts | disabled |
| Pre-shared Key * | •••• |

** can be blank*

Figure 5: Router B (red network) – GRE tunnel configuration



**Network Status**

Interfaces

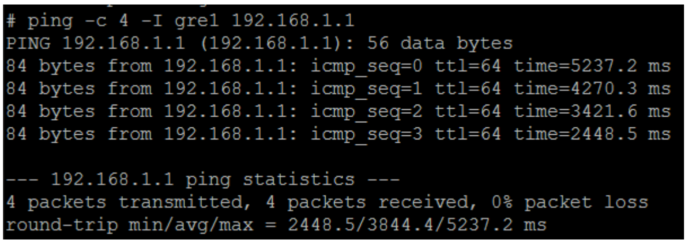
```
eth0      Link encap:Ethernet  HWaddr 00:0A:14:81:2B:40
          inet addr:192.168.1.1  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:8794 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1876 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:863618 (843.3 KB)  TX bytes:282270 (275.6 KB)
          Interrupt:23

gre1      Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          inet addr:10.255.0.1  P-t-P:10.255.0.2  Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP  MTU:1472  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:1 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:76 (76.0 B)  TX bytes:76 (76.0 B)

usb0      Link encap:Ethernet  HWaddr 5A:94:6E:1D:05:07
          inet addr:10.0.6.182  Bcast:10.255.255.255  Mask:255.255.255.255
          UP BROADCAST RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:1618 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2710 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:291472 (284.6 KB)  TX bytes:576731 (563.2 KB)
```

Route Table

```
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.254.254 0.0.0.0         UG    0      0        0 usb0
192.168.1.0     0.0.0.0         255.255.255.0   U     0      0        0 eth0
192.168.2.0     0.0.0.0         255.255.255.0   U     0      0        0 gre1
10.255.0.2      0.0.0.0         255.255.255.255 UH    0      0        0 gre1
192.168.254.254 0.0.0.0         255.255.255.255 UH    0      0        0 usb0
```

Figure 6: Network Status – network interface gre1

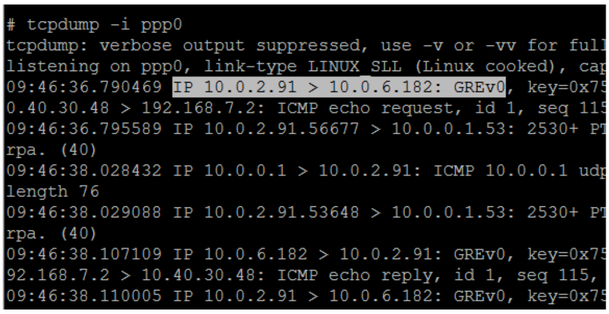
Now the connection between the networks via the GRE tunnel should work. It can be verified e.g. with the ping program after logging in one of the routers via telnet or SSH. In the fig. 13 there's console of the Router B (192.168.2.1) with the program ping and its result shown. The -c switch tells the number of requests, the -I switch tells the interface used (gre1).



```
# ping -c 4 -I gre1 192.168.1.1
PING 192.168.1.1 (192.168.1.1): 56 data bytes
84 bytes from 192.168.1.1: icmp_seq=0 ttl=64 time=5237.2 ms
84 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=4270.3 ms
84 bytes from 192.168.1.1: icmp_seq=2 ttl=64 time=3421.6 ms
84 bytes from 192.168.1.1: icmp_seq=3 ttl=64 time=2448.5 ms

--- 192.168.1.1 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 2448.5/3844.4/5237.2 ms
```

Figure 7: Program ping via gre1 network interface

To verify the usage of the GRE protocol, the tcpdump program for packet analysis can be run in one of the routers. See marked row in the next figure (GREv0). Here the tcpdump program was run with the -i switch telling which network interface listen on (ppp0 for watching the Mobile WAN communication running on this interface).



```
# tcpdump -i ppp0
tcpdump: verbose output suppressed, use -v or -vv for full
listening on ppp0, link-type LINUX_SLL (Linux cooked), cap
09:46:36.790469 IP 10.0.2.91 > 10.0.6.182: GREv0, key=0x75
0.40.30.48 > 192.168.7.2: ICMP echo request, id 1, seq 115
09:46:36.795589 IP 10.0.2.91.56677 > 10.0.0.1.53: 2530+ PT
rpa. (40)
09:46:38.028432 IP 10.0.0.1 > 10.0.2.91: ICMP 10.0.0.1 udp
length 76
09:46:38.029088 IP 10.0.2.91.53648 > 10.0.0.1.53: 2530+ PT
rpa. (40)
09:46:38.107109 IP 10.0.6.182 > 10.0.2.91: GREv0, key=0x75
92.168.7.2 > 10.40.30.48: ICMP echo reply, id 1, seq 115,
09:46:38.110005 IP 10.0.2.91 > 10.0.6.182: GREv0, key=0x75
```

Figure 8: Tcpdump program for the packet analysis – verifying the GRE communication

## 3.2 GRE Tunnel Between Advantech Router and OS Linux

The example of the GRE tunnel between Advantech Router and OS Linux is shown here. Linux is also running on the Advantech router, so it is a simple example to configure.

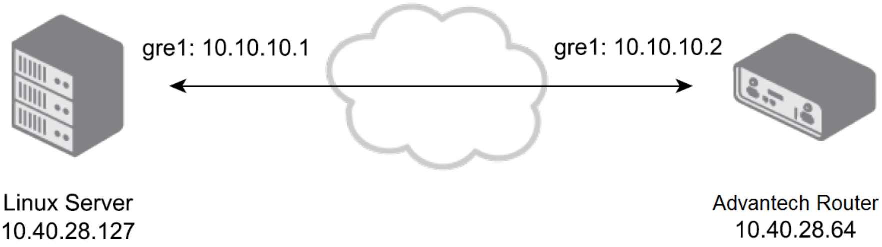
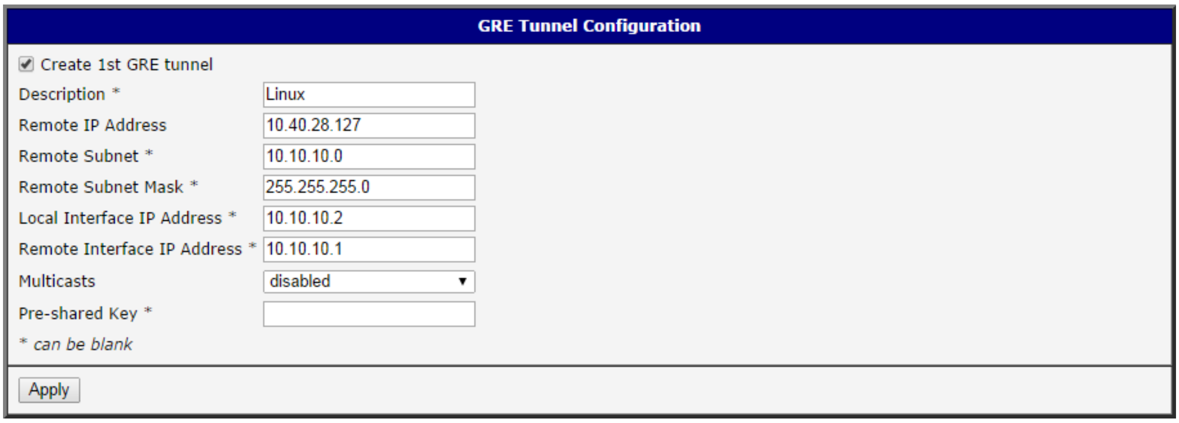


Figure 9: Example – GRE tunnel between Advantech router and OS Linux

For the topology and IP addresses in this example, the GRE tunnel in the Advantech router has to be set up the following way:



**GRE Tunnel Configuration**

☑ Create 1st GRE tunnel

| | |
|---|---|
| Description * | Linux |
| Remote IP Address | 10.40.28.127 |
| Remote Subnet * | 10.10.10.0 |
| Remote Subnet Mask * | 255.255.255.0 |
| Local Interface IP Address * | 10.10.10.2 |
| Remote Interface IP Address * | 10.10.10.1 |
| Multicasts | disabled ▾ |
| Pre-shared Key * | |

** can be blank*

Figure 10: GRE tunnel configuration in the Advantech router

In the OS Linux, run the terminal and create the other side of the GRE tunnel the following way. First, verify the Linux kernel module allowing the GRE tunnel is present. It can be done by these commands:

```
\$ sudo modprobe ip\_gre
\$ lsmod | grep gre
```

If the `gre` module is present in the kernel, the output will look like this:

```
ip\_gre                   22432  0
gre                      12989  1 ip\_gre
```

Now it is possible to create the GRE tunnel using the following commands:

```
\$ sudo ip tunnel add gre1 mode gre remote 10.40.28.64 local 10.40.28.127
ttl 255
\$ sudo ip link set gre1 up
\$ sudo ip addr add 10.10.10.1\/24 dev gre1
```

Verifying the tunnel's creation is possible by typing the `ip route show` command. Routing rules for the newly created network interface `gre1` are shown. Also, after running the `ifconfig` program offering information about network interfaces, the newly created interface is displayed. For shutting down or deleting the GRE interface, these commands can be used:

```
\$ sudo ip link set gre1 down
\$ sudo ip tunnel del gre1
```

The mentioned commands can also be used in the Advantech router (e.g., via SSH or telnet command line access) since the OS Linux is also running in the Advantech routers and the `ip` program is available in the router (see Command Line Interface application note).

## 3.3 GRE Tunnel Between Advantech Router and Cisco Router

This is the example of the GRE tunnel configuration between the Advantech and the Cisco router. The topology and addresses are on the figure below:

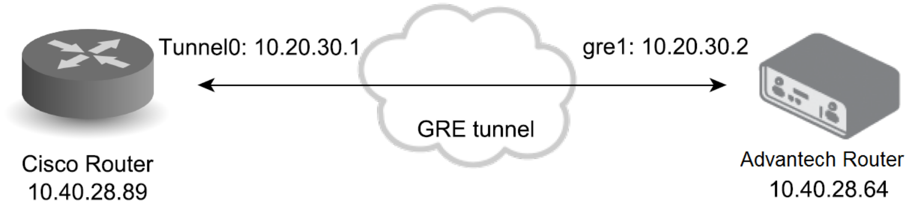
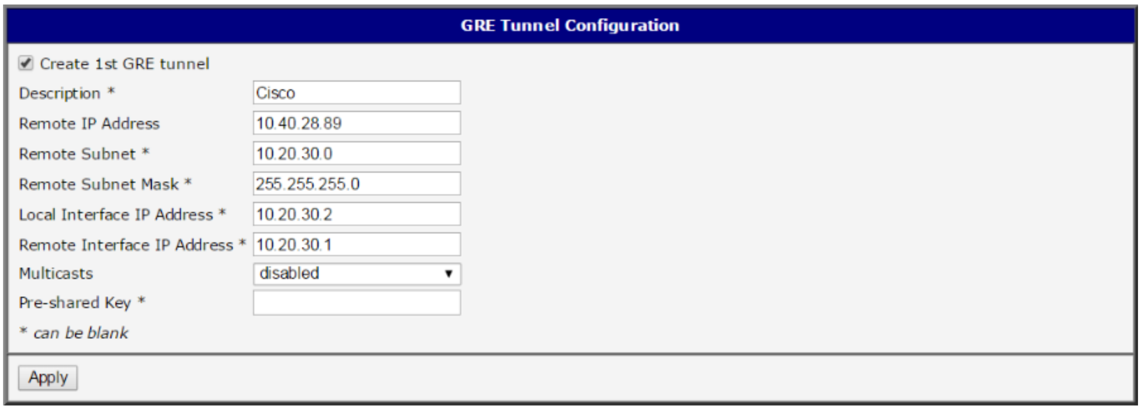


Figure 11: Example – GRE tunnel between Advantech router and Cisco router

Configure the Advantech router this way:



**GRE Tunnel Configuration**

|  |  |
|---|---|
| ☑ Create 1st GRE tunnel | |
| Description * | Cisco |
| Remote IP Address | 10.40.28.89 |
| Remote Subnet * | 10.20.30.0 |
| Remote Subnet Mask * | 255.255.255.0 |
| Local Interface IP Address * | 10.20.30.2 |
| Remote Interface IP Address * | 10.20.30.1 |
| Multicasts | disabled |
| Pre-shared Key * | |

* *can be blank*

Figure 12: Advantech router – GRE tunnel configuration

Log into the console of the Cisco router (e.g. via telnet or serial line) and enter into the configuration terminal typing the `config terminal` command. Now you can create the GRE tunnel using following commands:

```
Router(config)\# interface Tunnel0
Router(config-if)\# ip address 10.20.30.1 255.255.255.0
Router(config-if)\# tunnel source 10.40.28.89
Router(config-if)\# tunnel destination 10.40.28.64
Router(config-if)\# end
```

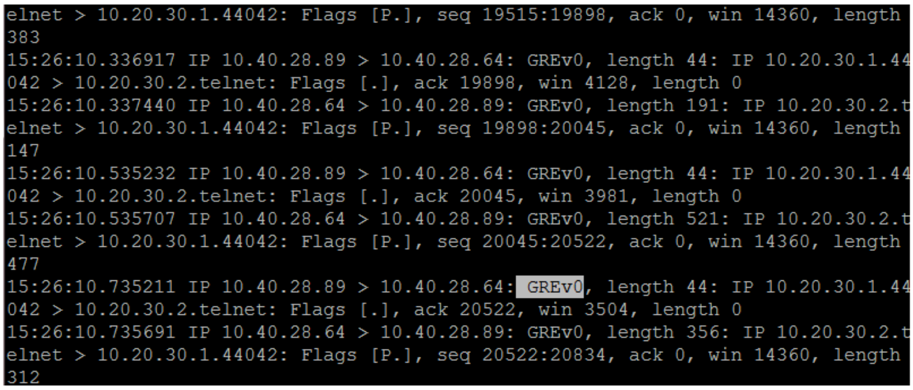
Optionally adjust the packet length for the added overhead to prevent unnecessary packet fragmentation. You can add the route for stations connected behind the router.

```
Router(config-if)\# ip mtu 1400
Router(config-if)\# ip tcp adjust-mss 1360
Router(config)\# ip route 192.168.1.0 255.255.255.0 10.20.30.1
```

You can view the running configuration typing the `show running-config` command (when out of the configuration terminal). There should be Tunnel0 network interface present and configured as done before.

For deeper knowledge of Cisco router settings, see the Cisco documentation.

Now the `ping` program should work with the successful result (from the Cisco router to Advantech router via GRE tunnel – to the 10.20.30.2 address and vice versa). To verify the GRE encapsulation you can e.g. from the Cisco router's console log in via telnet or SSH to the Advantech router (`telnet 10.20.30.2`) and run there the `tcpdump` program for packet analysis. All the captured packets will have a GRE protocol mark – see the next figure.



```
elnet > 10.20.30.1.44042: Flags [P.], seq 19515:19898, ack 0, win 14360, length
383
15:26:10.336917 IP 10.40.28.89 > 10.40.28.64: GREv0, length 44: IP 10.20.30.1.44
042 > 10.20.30.2.telnet: Flags [.], ack 19898, win 4128, length 0
15:26:10.337440 IP 10.40.28.64 > 10.40.28.89: GREv0, length 191: IP 10.20.30.2.t
elnet > 10.20.30.1.44042: Flags [P.], seq 19898:20045, ack 0, win 14360, length
147
15:26:10.535232 IP 10.40.28.89 > 10.40.28.64: GREv0, length 44: IP 10.20.30.1.44
042 > 10.20.30.2.telnet: Flags [.], ack 20045, win 3981, length 0
15:26:10.535707 IP 10.40.28.64 > 10.40.28.89: GREv0, length 521: IP 10.20.30.2.t
elnet > 10.20.30.1.44042: Flags [P.], seq 20045:20522, ack 0, win 14360, length
477
15:26:10.735211 IP 10.40.28.89 > 10.40.28.64: GREv0, length 44: IP 10.20.30.1.44
042 > 10.20.30.2.telnet: Flags [.], ack 20522, win 3504, length 0
15:26:10.735691 IP 10.40.28.64 > 10.40.28.89: GREv0, length 356: IP 10.20.30.2.t
elnet > 10.20.30.1.44042: Flags [P.], seq 20522:20834, ack 0, win 14360, length
312
```

Figure 13: Tcpdump program – GRE encapsulation check

## 3.4 GRE over IPsec tunnel

Example of creating the GRE tunnel inside of the IPsec tunnel between two Advantech routers is showed here. This secured (encrypted) connection can be used to transport the routing information (protocols) between the networks.

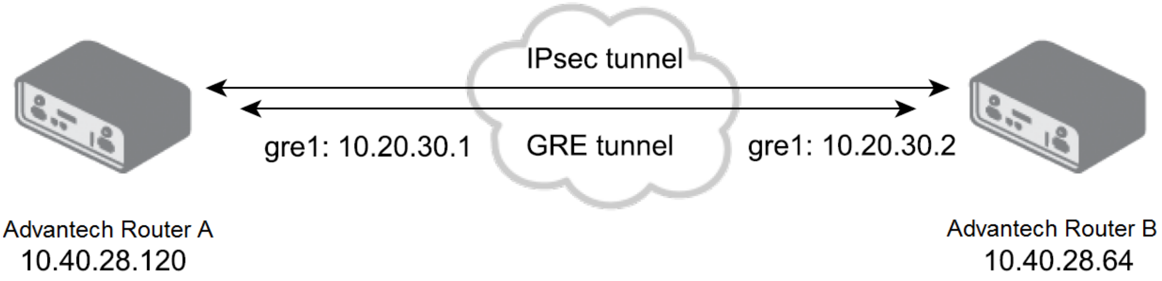
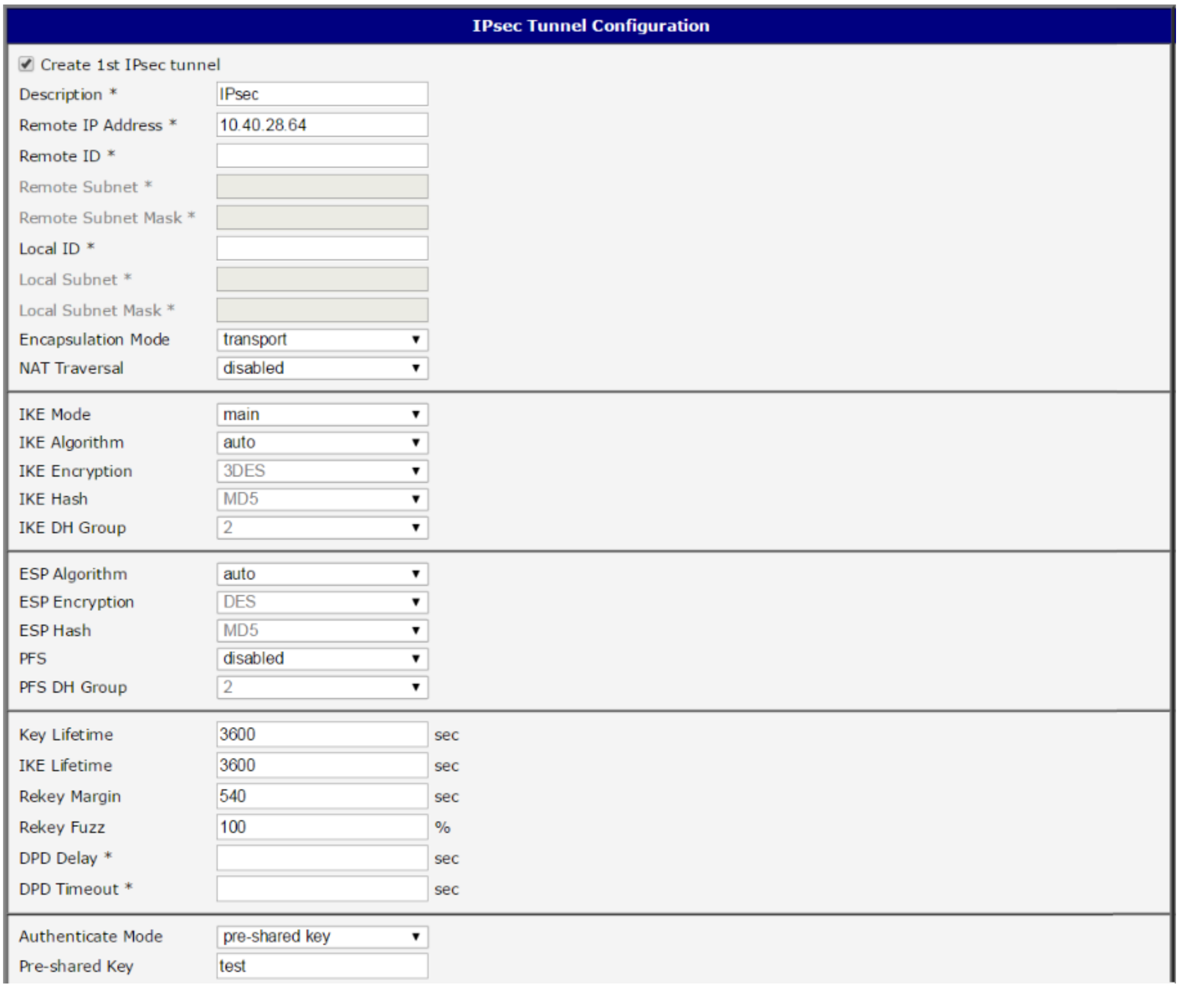


Figure 14: Topology of the GRE over IPsec example

For GRE over IPsec, the IPsec connection has to be established and also GRE tunnel has to be set up on both routers. There's IPsec and GRE setup of the Router A and Router B on the following pictures.

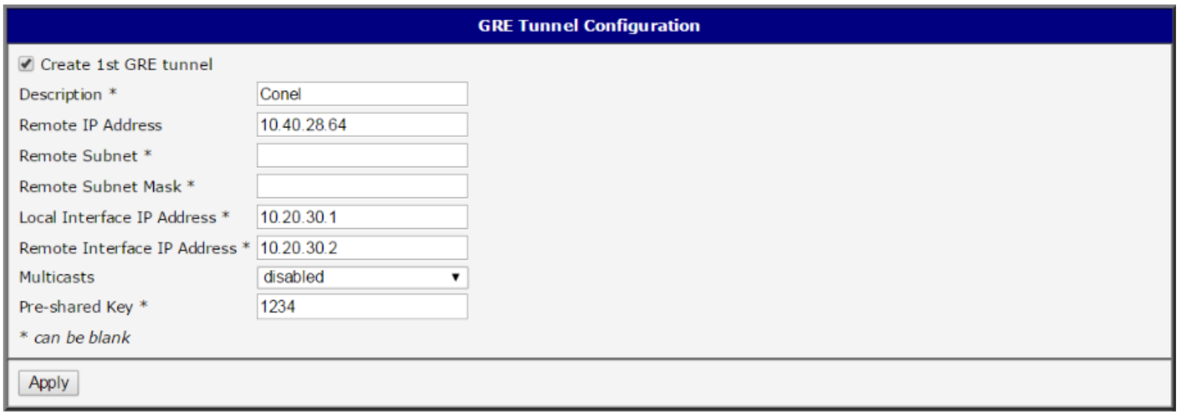


**IPsec Tunnel Configuration**

☑ Create 1st IPsec tunnel

| | | |
|---|---|---|
| Description * | IPsec | |
| Remote IP Address * | 10.40.28.64 | |
| Remote ID * | | |
| Remote Subnet * | | |
| Remote Subnet Mask * | | |
| Local ID * | | |
| Local Subnet * | | |
| Local Subnet Mask * | | |
| Encapsulation Mode | transport ▾ | |
| NAT Traversal | disabled ▾ | |
| IKE Mode | main ▾ | |
| IKE Algorithm | auto ▾ | |
| IKE Encryption | 3DES ▾ | |
| IKE Hash | MD5 ▾ | |
| IKE DH Group | 2 ▾ | |
| ESP Algorithm | auto ▾ | |
| ESP Encryption | DES ▾ | |
| ESP Hash | MD5 ▾ | |
| PFS | disabled ▾ | |
| PFS DH Group | 2 ▾ | |
| Key Lifetime | 3600 | sec |
| IKE Lifetime | 3600 | sec |
| Rekey Margin | 540 | sec |
| Rekey Fuzz | 100 | % |
| DPD Delay * | | sec |
| DPD Timeout * | | sec |
| Authenticate Mode | pre-shared key ▾ | |
| Pre-shared Key | test | |

Figure 15: Router A – IPsec configuration (*IPsec* item in the *Customization* section)



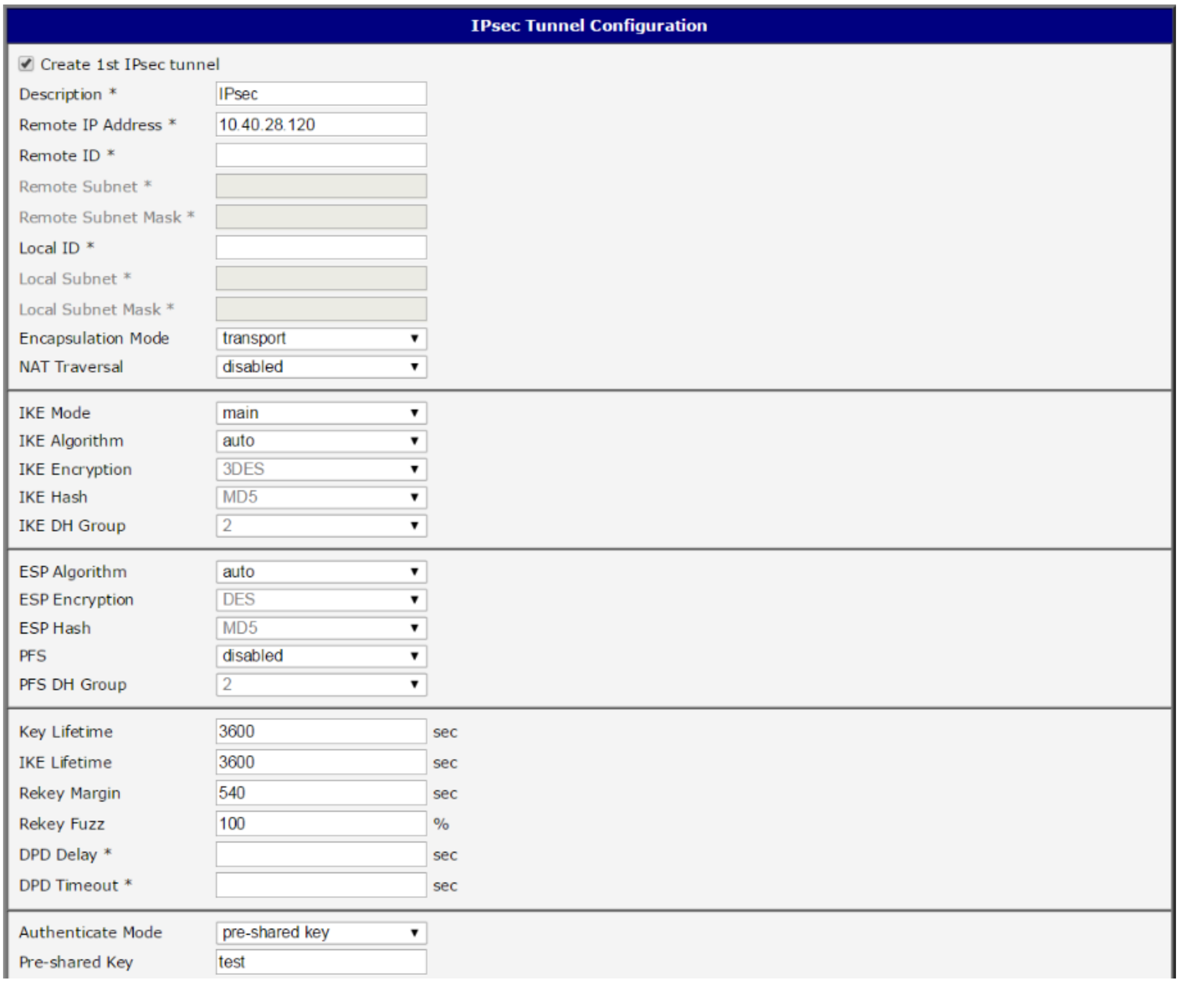
GRE Tunnel Configuration

☑ Create 1st GRE tunnel

| | |
|---|---|
| Description * | Conel |
| Remote IP Address | 10.40.28.64 |
| Remote Subnet * | |
| Remote Subnet Mask * | |
| Local Interface IP Address * | 10.20.30.1 |
| Remote Interface IP Address * | 10.20.30.2 |
| Multicasts | disabled |
| Pre-shared Key * | 1234 |

* *can be blank*

Figure 16: Router A – GRE configuration

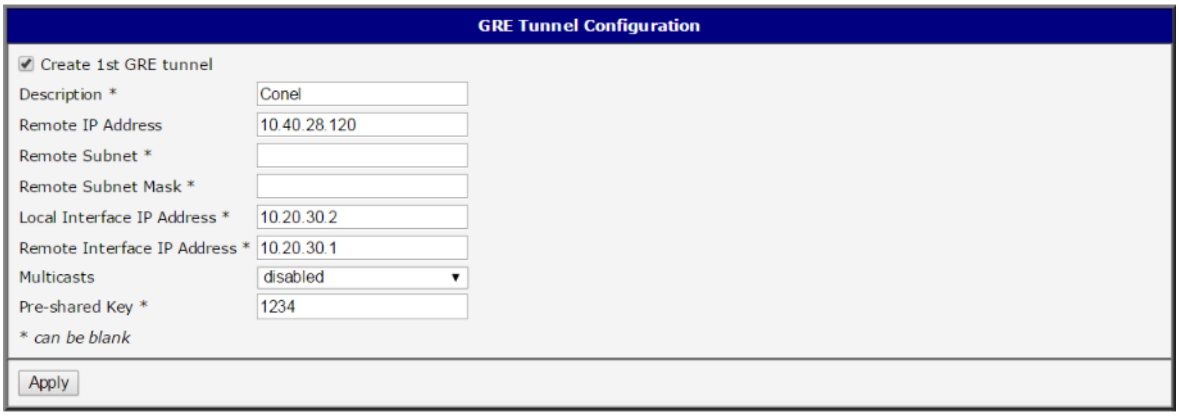


IPsec Tunnel Configuration

☑ Create 1st IPsec tunnel

| | | |
|---|---|---|
| Description * | IPsec | |
| Remote IP Address * | 10.40.28.120 | |
| Remote ID * | | |
| Remote Subnet * | | |
| Remote Subnet Mask * | | |
| Local ID * | | |
| Local Subnet * | | |
| Local Subnet Mask * | | |
| Encapsulation Mode | transport | |
| NAT Traversal | disabled | |
| IKE Mode | main | |
| IKE Algorithm | auto | |
| IKE Encryption | 3DES | |
| IKE Hash | MD5 | |
| IKE DH Group | 2 | |
| ESP Algorithm | auto | |
| ESP Encryption | DES | |
| ESP Hash | MD5 | |
| PFS | disabled | |
| PFS DH Group | 2 | |
| Key Lifetime | 3600 | sec |
| IKE Lifetime | 3600 | sec |
| Rekey Margin | 540 | sec |
| Rekey Fuzz | 100 | % |
| DPD Delay * | | sec |
| DPD Timeout * | | sec |
| Authenticate Mode | pre-shared key | |
| Pre-shared Key | test | |

Figure 17: Router B – IPsec configuration (*IPsec* item in the *Customization* section)



**GRE Tunnel Configuration**

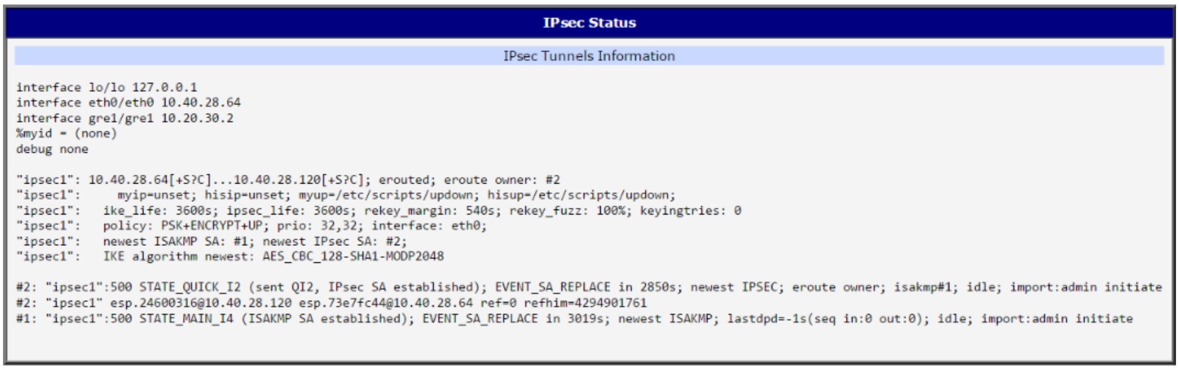
☑ Create 1st GRE tunnel

| | |
|---|---|
| Description * | Conel |
| Remote IP Address | 10.40.28.120 |
| Remote Subnet * | |
| Remote Subnet Mask * | |
| Local Interface IP Address * | 10.20.30.2 |
| Remote Interface IP Address * | 10.20.30.1 |
| Multicasts | disabled |
| Pre-shared Key * | 1234 |

** can be blank*

Figure 18: Router B – GRE configuration

When right configured, both routers will have the *established* information in the IPsec status – *IPsec* item in the *Status* section (see also the *System Log*).



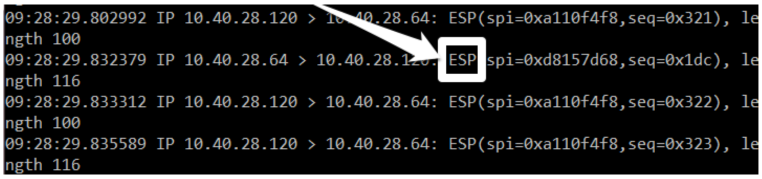
**IPsec Status**

IPsec Tunnels Information

```
interface lo/lo 127.0.0.1
interface eth0/eth0 10.40.28.64
interface gre1/gre1 10.20.30.2
%myid = (none)
debug none

"ipsec1": 10.40.28.64[+S?C]...10.40.28.120[+S?C]; erouted; eroute owner: #2
"ipsec1":     myip=unset; hisip=unset; myup=/etc/scripts/updown; hisup=/etc/scripts/updown;
"ipsec1":   ike_life: 3600s; ipsec_life: 3600s; rekey_margin: 540s; rekey_fuzz: 100%; keyingtries: 0
"ipsec1":   policy: PSK+ENCRYPT+UP; prio: 32,32; interface: eth0;
"ipsec1":   newest ISAKMP SA: #1; newest IPsec SA: #2;
"ipsec1":   IKE algorithm newest: AES_CBC_128-SHA1-MODP2048

#2: "ipsec1":500 STATE_QUICK_I2 (sent QI2, IPsec SA established); EVENT_SA_REPLACE in 2850s; newest IPSEC; eroute owner; isakmp#1; idle; import:admin initiate
#2: "ipsec1" esp.24600316@10.40.28.120 esp.73e7fc44@10.40.28.64 ref=0 refhim=4294901761
#1: "ipsec1":500 STATE_MAIN_I4 (ISAKMP SA established); EVENT_SA_REPLACE in 3019s; newest ISAKMP; lastdpd=-1s(seq in:0 out:0); idle; import:admin initiate
```

Figure 19: Router B – IPsec Status, tunnel established

Encryption of the GRE tunnel by IPsec can be verified after logging in the both routers via telnet or SSH. E.g. on the router B run the tcpdump program with parameters for filtering the ESP protocol (IPsec): `tcpdump -s0 protochain 50`. From the router's A console then log in the router B via telnet or SSH and via GRE tunnel – the 10.20.30.2 address – so the captured communication is via GRE tunnel. When writing in the router's A console, the tcpdump program on the router B will capture the encrypted ESP packets. The communication is running via GRE tunnel and is IPsec encrypted.



```
09:28:29.802992 IP 10.40.28.120 > 10.40.28.64: ESP(spi=0xa110f4f8,seq=0x321), le
ngth 100
09:28:29.832379 IP 10.40.28.64 > 10.40.28.120: ESP(spi=0xd8157d68,seq=0x1dc), le
ngth 116
09:28:29.833312 IP 10.40.28.120 > 10.40.28.64: ESP(spi=0xa110f4f8,seq=0x322), le
ngth 100
09:28:29.835589 IP 10.40.28.120 > 10.40.28.64: ESP(spi=0xa110f4f8,seq=0x323), le
ngth 116
```

Figure 20: Router B – ESP packets captured by tcpdump program

# 4. Related Documents

You can obtain product-related documents on the **Engineering Portal** at *icr.advantech.com*.

To access your router's documents or firmware, go to the *Router Models* page, locate the required model, and select the appropriate tab below.

Documents that are common to all models and describe specific functionality areas are available on the *Application Notes* page.

The **Router Apps** installation packages and manuals are available on the *Router Apps* page.

If you are interested in further options for extending router functionality, either through scripts or custom Router Apps, please see the information available on the *Development* page.