EoIP

Introduction

Sub-menu: `/interface eoip`

Ethernet over IP (EoIP) Tunneling is a MikroTik RouterOS protocol based on **GRE RFC 1701** that creates an Ethernet tunnel between two routers on top of an IP connection. The EoIP tunnel may run over IPIP tunnel, PPTP tunnel, or any other connection capable of transporting IP.
When the bridging function of the router is enabled, all Ethernet traffic (all Ethernet protocols) will be bridged just as if there where a physical Ethernet interface and cable between the two routers (with bridging enabled). This protocol makes multiple network schemes possible.

Network setups with EoIP interfaces:

- Possibility to bridge LANs over the Internet
- Possibility to bridge LANs over encrypted tunnels
- Possibility to bridge LANs over 802.11b 'ad-hoc' wireless networks

The EoIP protocol encapsulates Ethernet frames in GRE (IP protocol number 47) packets (just like PPTP) and sends them to the remote side of the EoIP tunnel.

Property Description

Property	Description
allow-fast-path (*yes* | *no*; Default: **yes**)	Whether to allow FastPath processing. Must be disabled if IPsec tunneling is used.
arp (*disabled* | *enabled* | *proxy-arp* | *reply-only*; Default: **enabled)**	Address Resolution Protocol mode. - disabled - the interface will not use ARP - enabled - the interface will use ARP - proxy-arp - the interface will use the ARP proxy feature - reply-only - the interface will only reply to requests originated from matching IP address/MAC address combinations which are entered as static entries in the "/ip arp" table. No dynamic entries will be automatically stored in the "/ip arp" table. Therefore for communications to be successful, a valid static entry must already exist.
arp-timeout (*integer[/time]*; Default: **auto**)	Time interval in which ARP entries should time out.
clamp-tcp-mss (*yes* | *no*; Default: **yes**)	Controls whether to change MSS size for received TCP SYN packets. When enabled, a router will change the MSS size for received TCP SYN packets if the current MSS size exceeds the tunnel interface MTU (taking into account the TCP/IP overhead).The received encapsulated packet will still contain the original MSS, and only after decapsulation the MSS is changed.
comment (*string*; Default:)	Short description of the interface.
disabled (*yes* | *no* ; Default: **no**)	Whether an item is disabled.
dont-fragment (*inherit* | *no*; Default: **no**)	Whether to include DF bit in related packets: *no* - fragment if needed, *inherit* - use Dont Fragment flag of original packet. (Without Dont Fragment: inherit - packet may be fragmented).

dscp (*integer: 0-63*; Default: **inherited**)	DSCP value of packet. Inherited option means that dscp value will be inherited from packet which is going to be encapsulated.
ipsec-secret (*string*; Default:)	When secret is specified, router adds dynamic IPsec peer to remote-address with pre-shared key and policy (by default phase2 uses sha1/aes128cbc).
keepalive (*integer[/time],integer 0..4294967295*; Default: **10s,10**)	Tunnel keepalive parameter sets the time interval in which the tunnel running flag will remain even if the remote end of tunnel goes down. If configured time,retries fail, interface running flag is removed. Parameters are written in following format: `KeepaliveInterval,KeepaliveRetries` where `KeepaliveInterval` is time interval and `KeepaliveRetries` - number of retry attempts. By default keepalive is set to 10 seconds and 10 retries.
l2mtu (*integer; read-only*)	Layer2 Maximum transmission unit. Not configurable for EoIP. MTU in RouterOS
local-address (*IP*; Default:)	Source address of the tunnel packets, local on the router.
loop-protect	
loop-protect-disable-time	
loop-protect-send-interval	
mac-address (*MAC*; Default:)	Media Access Control number of an interface. The address numeration authority IANA allows the use of MAC addresses in the range from **00:00:5E:80:00:00 - 00:00:5E:FF:FF:FF** freely
mtu (*integer*; Default: **auto**)	Layer3 Maximum transmission unit
name (*string*; Default:)	Interface name
remote-address (*IP*; Default:)	IP address of remote end of EoIP tunnel
tunnel-id (*integer: 65536*; Default:)	Unique tunnel identifier, which must match other side of the tunnel

Configuration Examples

Parameter tunnel-id is a method of identifying a tunnel. It must be unique for each EoIP tunnel.

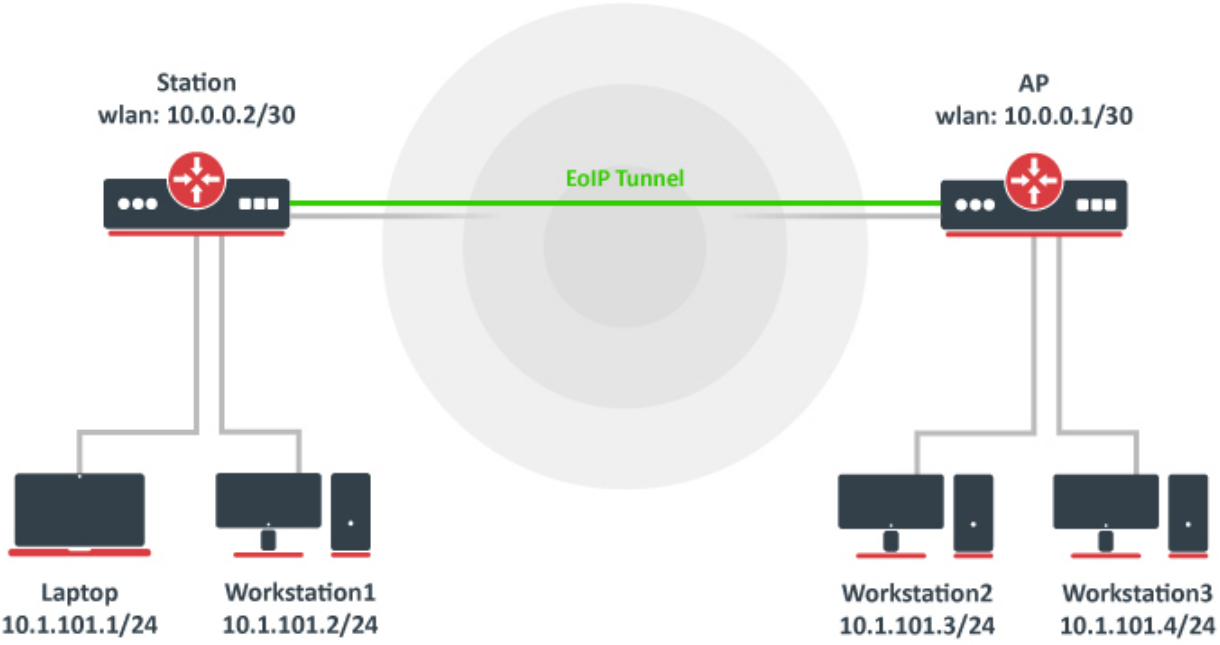
EoIP tunnel adds at least 42 byte overhead (8byte GRE + 14 byte Ethernet + 20 byte IP). MTU should be set to 1500 to eliminate packet fragmentation inside the tunnel (that allows transparent bridging of Ethernet-like networks so that it would be possible to transport full-sized Ethernet frame over the tunnel).

When bridging EoIP tunnels, it is highly recommended to set unique MAC addresses for each tunnel for the bridge algorithms to work correctly. For EoIP interfaces you can use MAC addresses that are in the range from **00:00:5E:80:00:00 - 00:00:5E:FF:FF:FF** , which IANA has reserved for such cases. Alternatively, you can set the second bit of the first byte to modify the auto-assigned address into a 'locally administered address', assigned by the network administrator, and thus use any MAC address, you just need to ensure they are unique between the hosts connected to one bridge.

Example

Let us assume we want to bridge two networks: 'Station' and 'AP'. By using EoIP setup can be made so that Station and AP LANs are in the same Layer2 broadcast domain.

Consider the following setup:



As you know wireless stations cannot be bridged, to overcome this limitation (not involving WDS) we will create an EoIP tunnel over the wireless link and bridge it with interfaces connected to local networks.

We will not cover wireless configuration in this example, let's assume that the wireless link is already established.

At first, we create an EoIP tunnel on our AP:

```
/interface eoip add name="eoip-remote" tunnel-id=0 remote-address=10.0.0.2 disabled=no
```

Verify the interface is created:

```
[admin@AP] > /interface eoip print
Flags: X - disabled; R - running
 0  R name="eoip-remote" mtu=auto actual-mtu=1458 l2mtu=65535 mac-address=FE:A5:6C:3F:26:C5 arp=enabled
      arp-timeout=auto loop-protect=default loop-protect-status=off loop-protect-send-interval=5s
      loop-protect-disable-time=5m local-address=0.0.0.0 remote-address=10.0.0.2 tunnel-id=0
      keepalive=10s,10 dscp=inherit clamp-tcp-mss=yes dont-fragment=no allow-fast-path=yes
```

Station router:

```
/interface eoip add name="eoip-main" tunnel-id=0 remote-address=10.0.0.1 disabled=no
```

Verify the interface is created:

```
[admin@Station] >  /interface eoip print
Flags: X - disabled; R - running
 0  R name="eoip-main" mtu=auto actual-mtu=1458 l2mtu=65535 mac-address=FE:4B:71:05:EA:8B arp=enabled
      arp-timeout=auto loop-protect=default loop-protect-status=off loop-protect-send-interval=5s
      loop-protect-disable-time=5m local-address=0.0.0.0 remote-address=10.0.0.1 tunnel-id=0
      keepalive=10s,10 dscp=inherit clamp-tcp-mss=yes dont-fragment=no allow-fast-path=yes
```

Next, we will bridge local interfaces with EoIP tunnel on our AP. If you already have a local bridge interface, simply add EoIP interface to it:

```
/interface bridge port add bridge=bridge1 interface=eoip-remote
```

The bridge port list should list all local LAN interfaces and the EoIP interface:

```
[admin@AP] > /interface bridge port print
Flags: I - INACTIVE; H - HW-OFFLOAD
Columns: INTERFACE, BRIDGE, HW, PVID, PRIORITY, PATH-COST, INTERNAL-PATH-COST, HORIZON
#    INTERFACE       BRIDGE   HW   PVID  PRIORITY  PATH-COST  INTERNAL-PATH-COST  HORIZON
0  H ether2          bridge1  yes     1  0x80             10                  10  none
1  H ether3          bridge1  yes     1  0x80             10                  10  none
2    eoip-remote     bridge1  yes     1  0x80             10                  10  none
```

On Station router, if you do not have a local bridge interface, create a new bridge and add both EoIP and local LAN interfaces to it:

```
/interface bridge add name=bridge1
/interface bridge port add bridge=bridge1 interface=ether2
/interface bridge port add bridge=bridge1 interface=eoip-main
```

Verify the bridge port section:

```
[admin@Station] > /interface bridge port print
Flags: I - INACTIVE; H - HW-OFFLOAD
Columns: INTERFACE, BRIDGE, HW, PVID, PRIORITY, PATH-COST, INTERNAL-PATH-COST, HORIZON
#    INTERFACE     BRIDGE   HW   PVID  PRIORITY  PATH-COST  INTERNAL-PATH-COST  HORIZON
0  H ether2        bridge1  yes     1  0x80             10                  10  none
2    eoip-main     bridge1  yes     1  0x80             10                  10  none
```

Now both sites are in the same Layer2 broadcast domain. You can set up IP addresses from the same network on both sites.