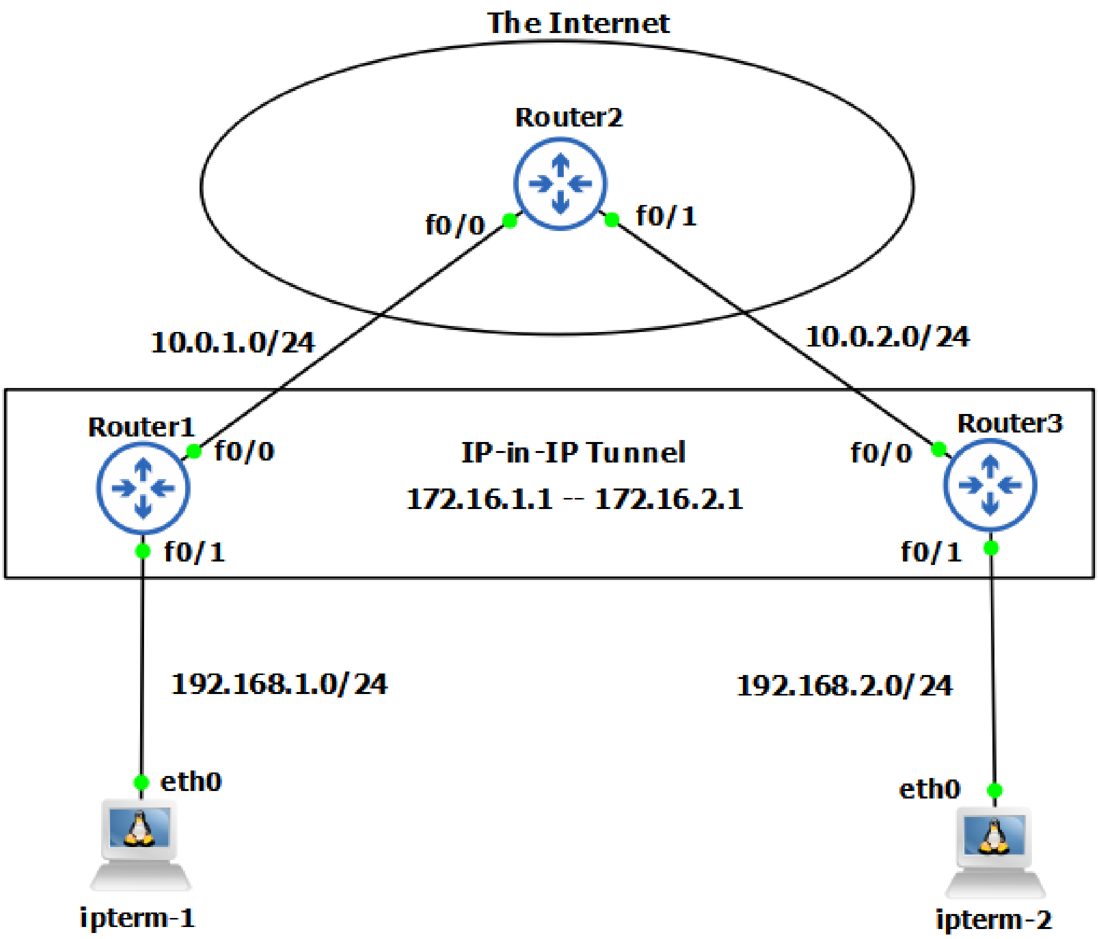
ELEC-H417: Lab 2 - Tunnels & Encapsulation

Obbjectives

- Configure a layer 3 IP-in-IP tunnel
- Understand GRE (Generic Routing Encapsulation) principles
- Understand how tunnels impact routing protocols
- Implement an IPv6-in-IPv4 tunnel

Topology



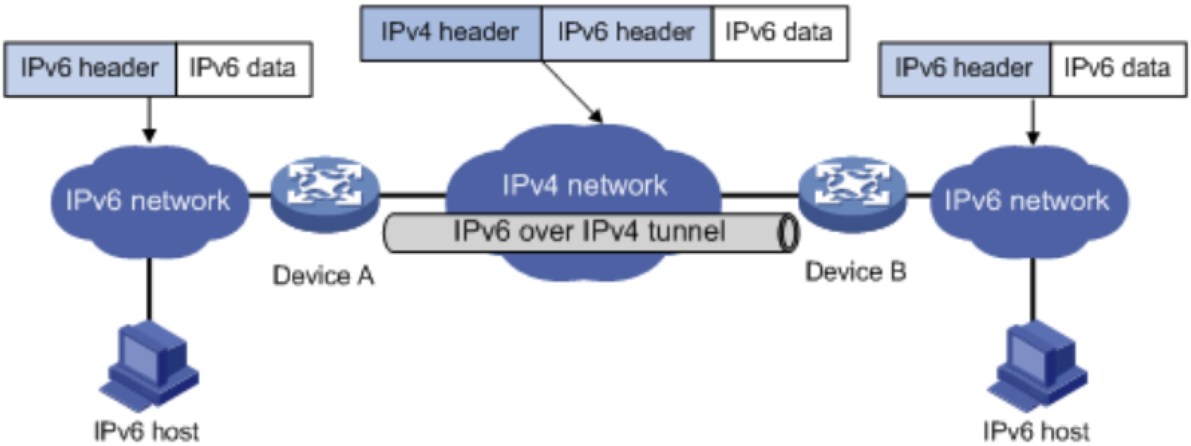
- **Guest hosts**
  - **ip-term1** (192.168.1.2/24 gateway 192.168.1.1)

    - **ip-term2** (192.168.2.2/24 gateway 192.168.2.1)
- **Routers**
    - **Router1**
        - f0/1 192.168.1.1/24
        - f0/0 10.0.1.1/24
    - **Router2** (The Internet)
        - f0/0 10.0.1.2/24
        - f0/1 10.0.2.2/24
    - **Router3**
        - f0/0 10.0.2.1/24
        - f0/1 192.168.2.1/24

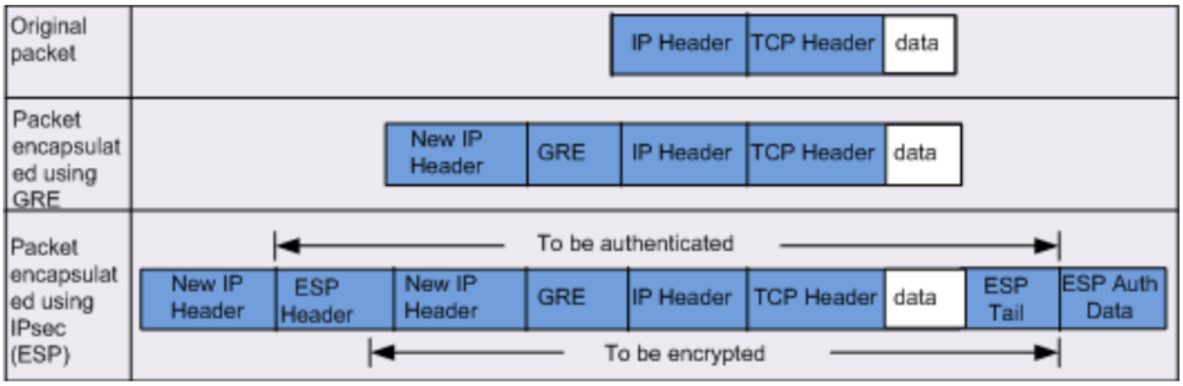
About Tunnels

A tunnel is a logical interface that acts as a logical connection between two endpoints. An IP-in-IP manual tunnel is a type of tunnel that has hard-coded source and destination addresses in the public IP domain, with a set IPv6 or IPv4 private addresses on the tunnel itself.

From a protocol point of view, the tunneled IP packets are re-encapsulated (IP in IP) to pass through the public IP domain. In practice, many in-IP protocols exist: IP-in-IP, IPv6-in-IPv4, L2TP (Ethernet in IP), and GRE (Generic Routing Encapsulation) which accommodates any kind of payload. GRE is often used in telecommunications systems to carry the traffic from and antenna to the core network of the operator.



Also, tunnels can be protected (integrity or ciphering), hence becoming VPN tunnels (see lab 4).



Commands

A tunnel is juste like a normal interface that has a source IP, a destination IP, and a transport mode. To create a tunnel interface, use the following commands:

```
Router1(config)# interface tunnel0
Router1(config-if)# ip address 172.16.1.1 255.255.255.0
Router1(config-if)# tunnel mode ipip
Router1(config-if)# tunnel soure 10.0.1.1
Router1(config-if)# tunnel destination 10.0.2.1
```

Any packet reaching the interface tunnel0 will be automatically encaspulated and sent
from the IP `164.15.18.1` to the destination IP `164.15.40.1` . Obviously, the IP `164.15.18.1` must exist on the current router and the destination IP must be routable, i.e., the router must know how to reach it. At the destination, the packet is de-encaspulated and routed by the destination router.

Please note that **both ends of the tunnel** must be configured with a tunnel interface to have a bi-directional tunnel.

Routing and tunnels

Impact on routing

How does a packet reach the interface tunnel0 ? Because it is routed there (by the existence of a route whose next hop is 172.16.1.2 . This can result from a static route (force through a tunnel) or a dynamic route computed automatically. Note that a

tunnel
interface is a real interface, which can be confirmed by

```
Router3# show ip interfaces brief
Interface       IP-Address  OK? Method Status                Protocol
FastEthernet0/0 10.0.2.1    YES NVRAM  up                    up
FastEthernet0/1 192.168.2.1 YES NVRAM  up                    up
FastEthernet1/0 unassigned  YES NVRAM  administratively down down
FastEthernet2/0 unassigned  YES NVRAM  administratively down down
Tunnel0         172.16.1.2  YES manual up                    up
```

In our example (and before the implementation of a tunnel), a packet sent from `ip-term1` will reach `ip-term2` in 3 hops (`ipterm1-R1-R2-R3-ipterm2`). Now let us implement a tunnel between `R1` and `R3`. Since `tunnel0` is a "real" interface (from an IP point-of-view), there is a direct link from `R1 [tunnel0]` to `R3 [tunnel0]`. Hence, the Layer3 route is now `ipterm1-R1-R3-ipterm2` (via tunnel0) which is 2 hops (better than the 3 hops route before).

Impact on routing protocols

Since `tunnel0` is a real interface (with an IP adrress and subnet), it is considered as a link
between `R1` and `R3` . This link can be propagated by means of a dynamic routing protocol, e.g., RIP. For instance, by configuring R1 and R3 like:

```
Router3(config)# router rip
Router3(config)# network 172.16.1.0
```

This will propagate the routing information about the subnet `172.16.1.0`. After a few minutes, the routing table is adjusted and the hop count between R1 and R3 will be 1 (via `tunnel0`).

Mission 1: No Tunnel

1. Double-click on the file with the “*.gns3project*” extension from the *Lab 2* to load the starting topology.
2. **(not in the report)** Open the topology, and check that all links are ok, i.e., that `ipterm-1` can ping `ipterm-2`. Check that dynamic routing (RIP) is active on R1,R2, and R3. Display the routing table at R1.

3. Traceroute between `ipterm-1` and `ipterm-2` . What are the hops in the route ? What is the path length ?

Mission 2: Site-to-site Tunnel

1. Implement a tunnel between R1 (172.16.1.1/24) and R3 (172.16.1.2/24).
2. Activate RIP routing on the tunnels interfaces at R1 and R3. Have a coffee break and display the routing tables at R1 and R3 (use `show ip route` and `sh ip rip database`). What changed ?
3. Traceroute between `ipterm-1` and `ipterm-2` . What are the hops in the route ? What is the path length ? Why ?
4. Using wireshark, Analyse the ping traffic (`ipterm-1` to `ipterm-2`) between R1 and R2 . What do you observe ? Show that there is encapsulation by the tunnel.

Mission 3: Deploy IPv6

IPv6 in CISCO

1. Enable the forwarding of IPv6 unicast datagrams (and ipv6 support in general):

```
Router1(config)# ipv6 unicast-routing
```

2. Set the IP:

```
Router1(config)# interface f0/1
Router1(config-if)# ipv6 address FC00:1::1/64
```

   Remember that here the subnet is of size 64 and is `FC00:1` . The host part is of size 64 also and the interface ID is here `1` .

3. Diplay the IPv6 interfaces in CISCO IOS:

```
Router1# show ipv6 interfaces brief
FastEthernet0/0 [up/up]
FastEthernet0/1 [up/up]
    FE80::C601:2BFF:FE63:1
    FC00:1::1
```

```
FastEthernet1/0 [administratively down/down]
FastEthernet2/0 [administratively down/down]
```

Do not forget to configure the *Router 3*

In Linux

On the ipterm, an IPv6 interface can be configured with:

```
ifconfig eth0 inet6 add FC00:1::2/64
```

Remember that an interface can have mulitple IPv6 addresses (hence the add keyword).
Validate using:

```
root@ipterm-1:~# ping6 -c3 FC00:1::1
PING FC00:1::1(fc00:1::1) 56 data bytes
64 bytes from fc00:1::1: icmp_seq=1 ttl=64 time=21.2 ms
64 bytes from fc00:1::1: icmp_seq=2 ttl=64 time=8.54 ms
64 bytes from fc00:1::1: icmp_seq=3 ttl=64 time=5.91 ms
```

Set the default route:

```
root@ipterm-1:~# ip -6 route add default via FC00:1::1
```

Do not forget the second host *ipterm-2*

Try to ping from `ipterm-1` to `ipterm-2`. What happend and why ?

Mission 4: IPv6-in-IPv4 tunnel

IPv6-in-IPv4 Tunnels (called *6in4 tunnels*) can be useful to connect subnets when an ISP is not IPv6-ready.

Cisco commands

Set (or change) the tunnel type with:

```
Router1(config)# interface tunnel0
Router1(config-if)# tunnel mode ipv6ip
```

Set the IP address of the tunnel interface:

```
Router1(config-if)# ipv6 address FC00:ABBA::1/64
```

RIPv6 allows to do dynamical routing in IPv6 subnets:

```
Router1(config)# ipv6 router rip LAB
```

Where LAB is the name of our RIP instance (RIPv6 can have multiple instances). Then configure each interface to be included in the RIP routing, e.g.:

```
Router1(config)# interface f0/1
Router1(config-if)# ipv6 rip LAB enable
Router1(config-if)# exit
Router1(config)# interface tunnel0
Router1(config-if)# ipv6 rip LAB enable
```

Display the IPv6 routing table with:

```
Router1# show ipv6 route
```

The mission itself

1. Set the IPv6 interfaces as follows:
    - **Router 1**
        - interface f0/1 with FC00:1::1/64
        - interface tunnel0 FC00:ABBA::1/64 (**do not forget to set tunnel type to ipv6ip !**)
    - **Router 3**
        - interface f0/1 with FC00:2::1/64
        - interface tunnel0 FC00:ABBA::2/64
    - **ipterm-1** interface eth0 FC00:1::2/64 and default gateway FC00:1::1/64
    - **ipterm-2** interface eth0 FC00:2::2/64 and default gateway FC00:2::1/64

2. Validate by pinging between `Router 1` (FC00:ABBA::1) and `Router 3` (FC00:ABBA::1). The tunnel should allow that, even if `Router 2` has no IPv6 configured.
3. Instanciate a RIPv6 on `Router 1` and `Router 3`. Remember to install it on the interface `tunnel 0` **and** `fastEthernet 0/1` !
4. Validate from R3:

   ```
   Router3# ping FC00:ABBA::1
   Router3# ping FC00:1::2
   ```

   and from the ipterm2:

   ```
   root@ipterm-2:~# ping6 -c2 FC00:1::2
   ```

5. With wireshark (between R1 and R2), capture this last set of ping packets between `ipterm-1` and `ipterm-2` and show that the tunnel has encapsulated the IPv6 into IPv4
6. What happend when you do a simple *ping* command (in IPv4) ?