PRINCIPLES, MODELS AND APPLICATIONS FOR DISTRIBUTED SYSTEMS M – Module 2

Network Overlay
Lab Session

Prof. Davide Borsatti – *davide.borsatti@unibo.it*

Department of Electrical, Electronic and Information Engineering
"Guglielmo Marconi"

Assignment 2 – Basic Topology

- Write a bash script that build this simple topology with two network namespaces. You can choose the IP network that you want. Check their connectivity with `ping`

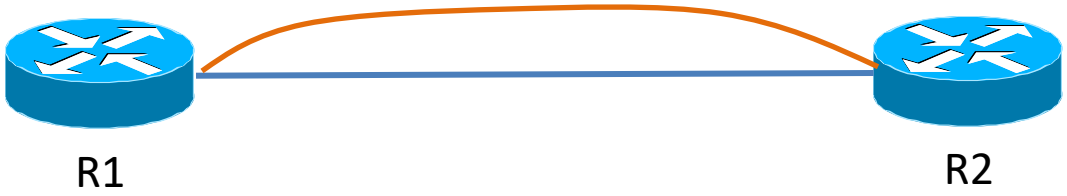
- To get started:

```
ip netns add R1
ip link add ____ type veth peer name ____
ip link set VETH_NAME netns R1
ip netns exec R1 ip addr add ...
```

Assignment 2 – GRE Tunnel

- Write a bash script that setup a GRE tunnel between the two network namespaces.

➡ You need to activate the kernel module to enable GRE tunnels: sudo modprobe ip_gre



GRE Tunnel

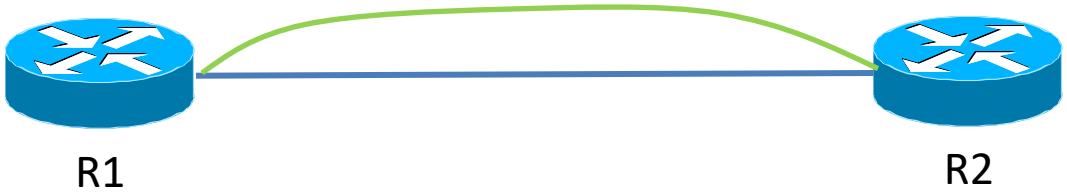
- To get started:

```
ip tunnel add GRE_DEV_NAME mode gre remote REMOTE_TUNNEL_IP local
LOCAL_TUNNEL_IP ttl 63

ip link set dev GRE_DEV_NAME up
ip addr add <gre_ip_subnet> dev GRE_DEV_NAME
```

Assignment 2 – VXLAN Tunnel

- Write a bash script that setup a VXLAN tunnel between the two network namespaces.



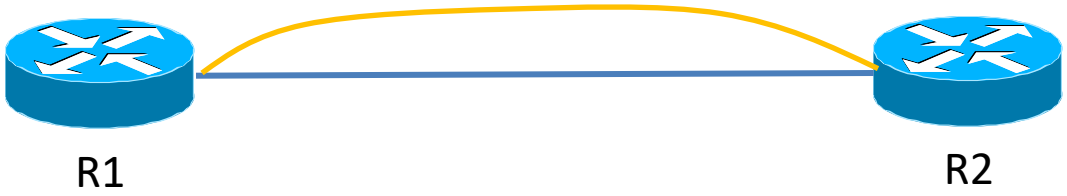
- To get started:

```
ip link add VXLAN_INTERFACE_NAME type vxlan id VNI local TUNNEL_IP_LOCAL remote
TUNNEL_IP_REMOTE nolearning dstport 4789

ip link set VXLAN_INTERFACE_NAME up
```

Assignment 2 – VPN with WireGuard

- Write a bash script that build this simple topology with two network namespaces. You can choose the IP network that you want.



- Single Router config (check official guide):

```
ip link add dev wg0 type wireguard
wg genkey | sudo tee <private_key_filename>.key
wg set wg0 private-key <private_key_filename>.key
ip link set wg0 up
wg set wg0 peer <peer-public-key> endpoint <peer-ip>:<peer-port>
allowed-ips <ip-ranges-allowed>
```

`wg` to see the status of active WireGuard interfaces

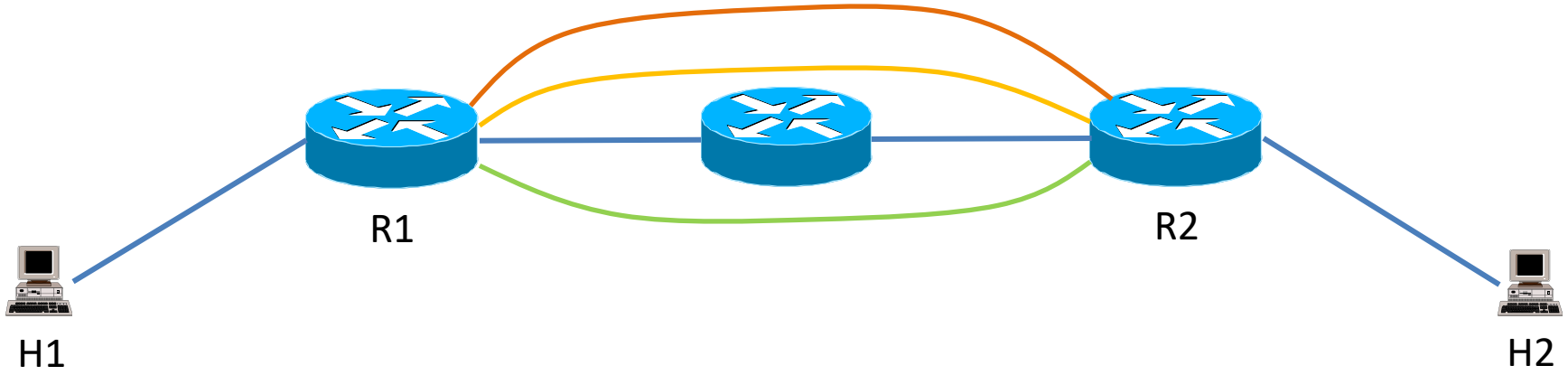
Assignment 2 – Tunnel Comparisons

- With a tool like WireShark or Tcpdump, capture the traffic flowing through the tunnels and analyze their differences.

- Observe the tunnels' performance (e.g., latency and bandwidth) differences (if any) with tools like ping and iperf3.

Assignment 2 - Extra

- Enrich the scripts to support the more complex network topology shown below.
- **N.B.** To connect the H1 and H2 network together with one of the tunnel, you should create a virtual bridge (some guidelines) in both routers containing the interface of the tunnel and the interface toward the host.



Linux Network Namespaces

`ip nets`

- Network namespace management
  - `ip netns add NAMESPACE`
    - Create a namespace
  - `ip netns del NAMESPACE`
    - Delete a namespace
  - `ip netns list`
    - List the existing namespaces
  - `ip netns exec NAMESPACE COMMAND`
    - Execute a specific COMMAND in namespace NAMESPACE
  - `ip netns monitor`
    - Reports events related to namespaces

Virtual Ethernet links

`ip link`

- `ip link add NAME type veth`
  - NAME is the symbolic name of the device
  - TYPE is the type of network device

Veth = Virtual Ethernet

- The logical equivalent of an Ethernet cable with two plugs at the ends

Example

- `ip link add NAME1 type veth peer name NAME2`
  - Create a ethernet link
  - The logical names of the two plugs are NAME1 and NAME2
- `ip link set NAME NAMESPACE`
  - Connects the NAME plug of the veth to NAMESPACE

Configuring a network interface

```
ip addr add <address>/<prefix-length> dev <interface>
```

Add a new ip address to the interface

```
ip addr del <address>/<prefix-length> dev <interface>
```

Remove an ip address from the interface

```
ip link set <interface> up
ip link set <interface> down
```

Activate or deactivate an interface

```
ip route add default via <default gateway IP address>
```

Add a new default gateway

```
ip route add <network prefix>/<prefix length> via <next hop IP address>
```

Add a new fixed route

GRE tunnel set up

```
Modprobe ip_gre
```

```
ip tunnel add <name> mode gre remote <remote IP> local <local IP> ttl
<number>
```

- Local IP: must be an IP number which is already configured on an interface of the local gateway (the one the configuration is run)
- Remote IP: Must be an IP number on the same network of <local IP> which is already configured on an interface of the remote gateway at the other edge of the tunnel
- TTL: says the ttl value to be used when packets get off the tunnel

VxLAN tunnel setup

```
ip link add <name> type vxlan id <VNI> local <local IP> remote <remote IP> nolearning dstport <port number>
```

- Local IP: must be an IP number which is already configured on an interface of the local gateway (the one the configuration is run)
- Remote IP: Must be an IP number reachable by the machine (i.e., it should be already configured on an interface of the machine at the other end of the tunnel)
- Port number: VxLAN is carried on top of UDP, therefore a port number must be specified
- VNI: Identifier of the VXLAN tunnel, it should be the same on both sides