Cloud e Datacenter Networking

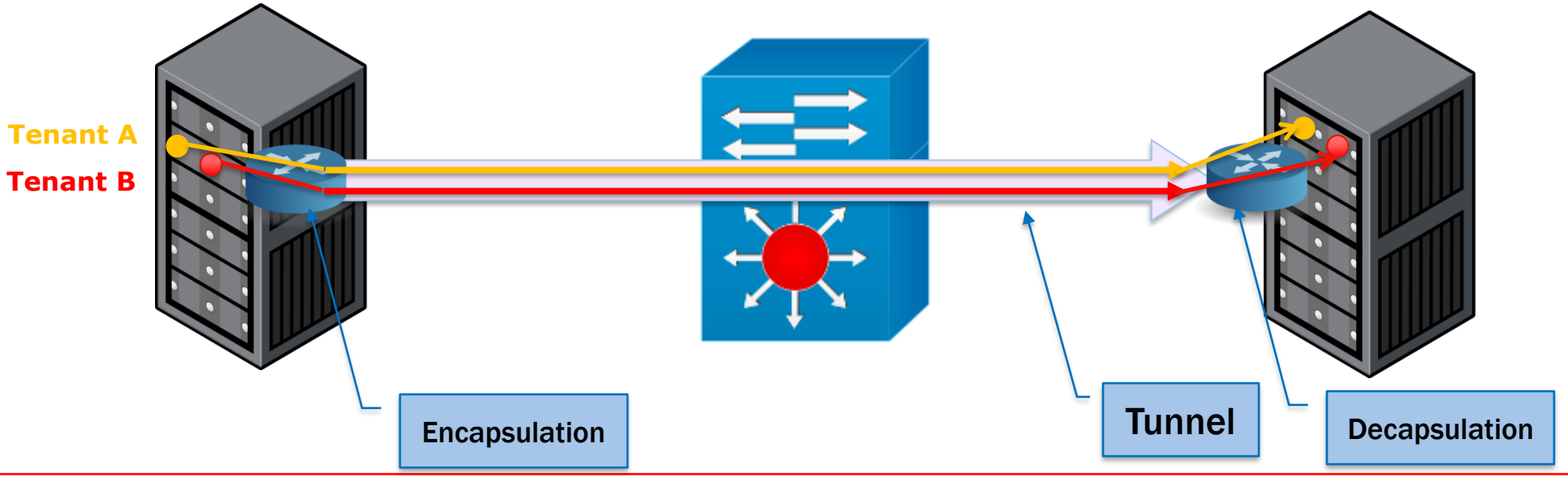
Università degli Studi di Napoli Federico II
Dipartimento di Ingegneria Elettrica e delle Tecnologie dell'Informazione DIETI
Laurea Magistrale in Ingegneria Informatica

Prof. Roberto Canonico

VXLAN demo with GNS3

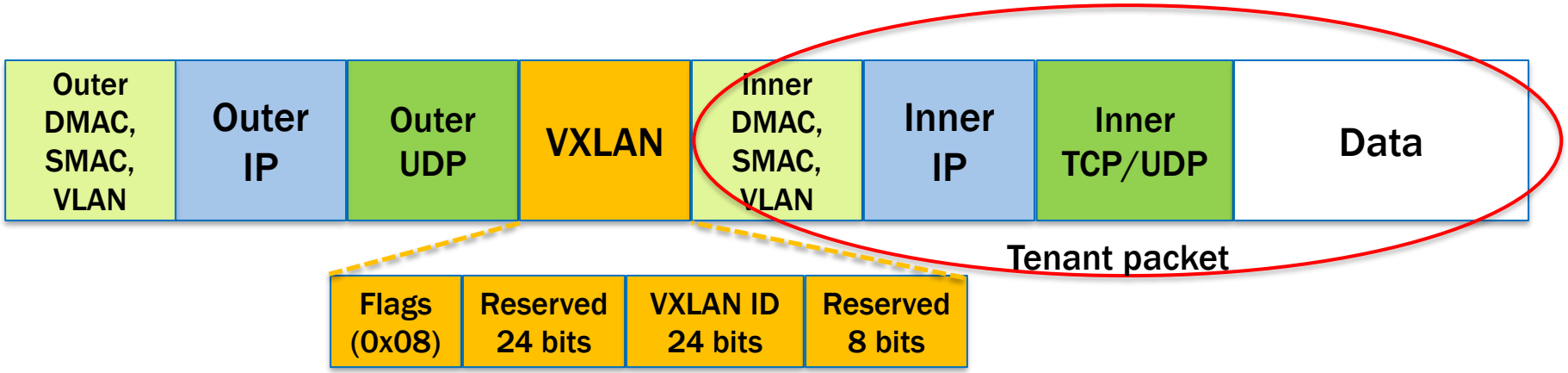
Network Virtualization using encapsulation

- VXLAN and NVGRE are two different network virtualization methods that use encapsulation and tunneling to create large numbers of virtual LANs for subnets that can extend across layer 2 and 3
- Encapsulation/decapsulation is performed by entities that could reside either in End Devices or in ToR edge switches (or in both)
- VXLAN is supported by Cisco and Vmware
- NVGRE was proposed by Microsoft, Intel, HP and Dell



VXLAN (RFC 7348)

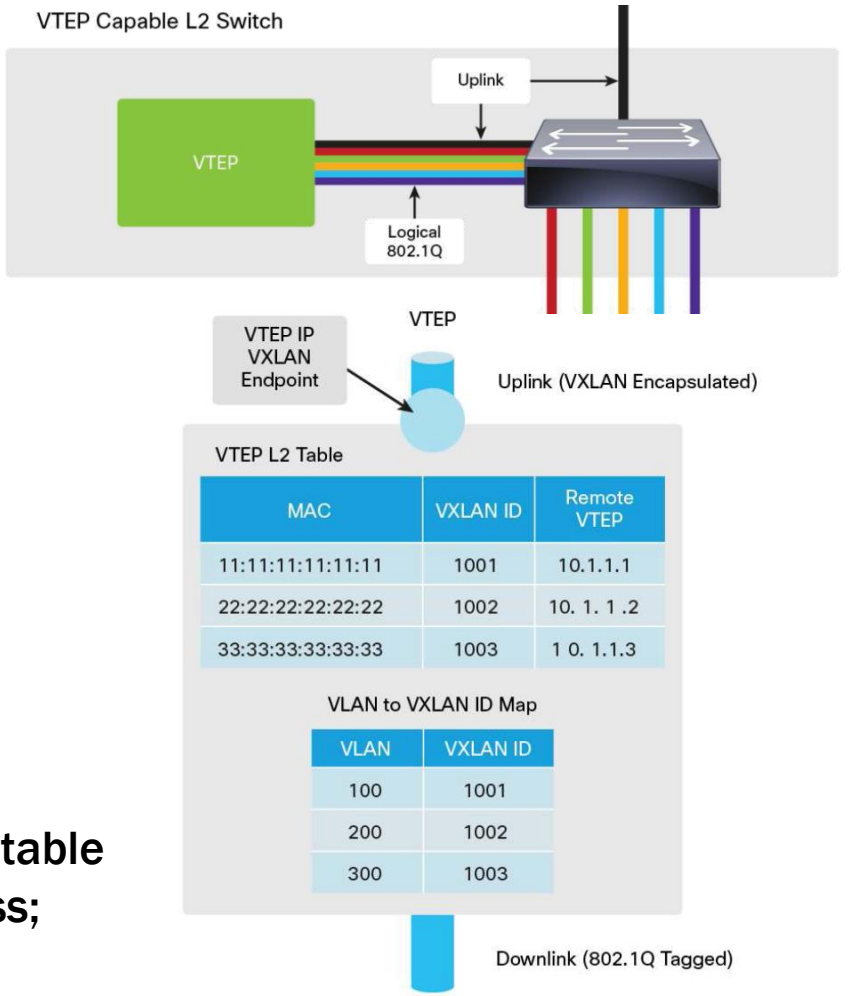
- Virtual eXtensible LAN (VXLAN) was originally proposed by Cisco and VMware to tunnel virtual layer 2 networks on a substrate layer 3 physical network



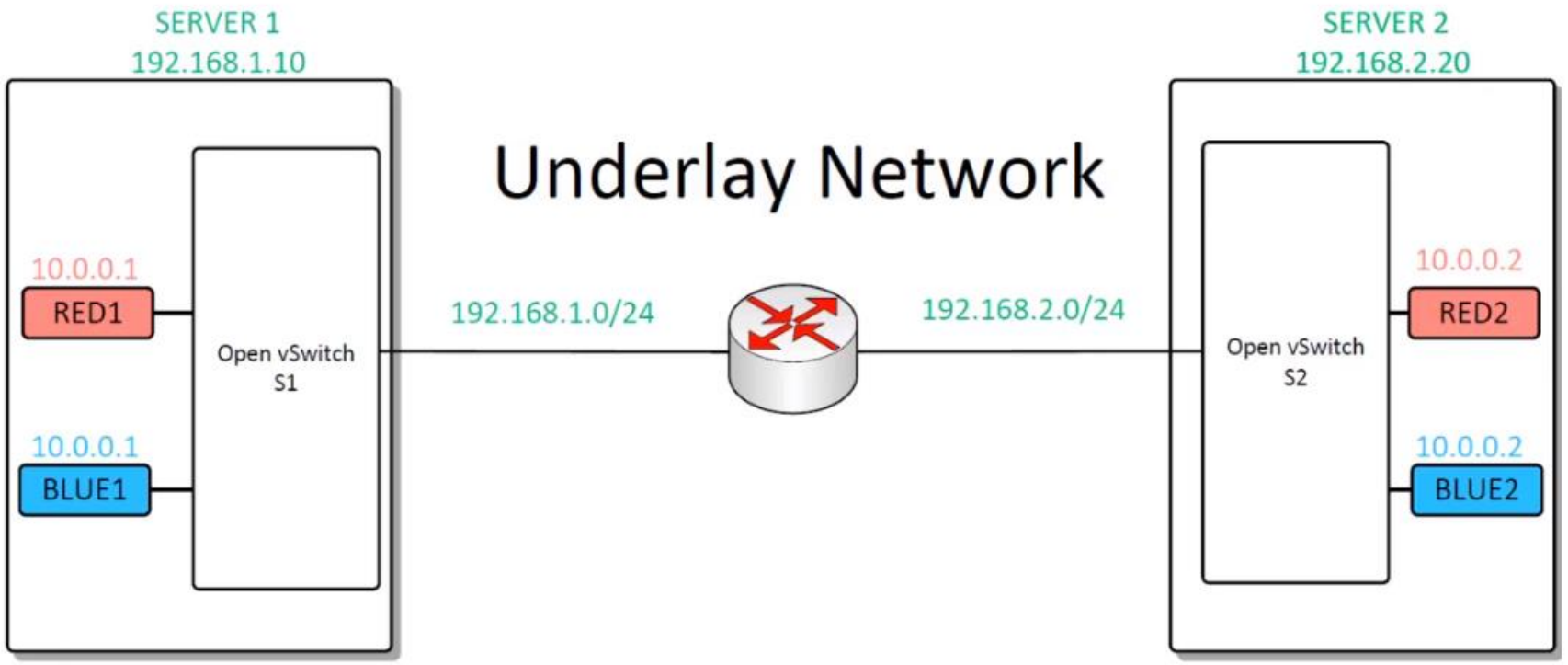
- VXLAN encapsulate packets in UDP tunnels with destination port number 4789
- In the shared L3 infrastructure, packets are identified by outer MAC addresses imposed by the infrastructure provider
- Tenants free to choose their own MAC addresses and VLAN IDs with no conflicts
- To avoid packet fragmentation in the shared infrastructure, it must support larger MTU values
- Encapsulation/decapsulation is performed at *VXLAN Tunnel End Points* (VTEPs)
- VXLAN ID allows to identify up to 2^{24} distinct virtual networks

VXLAN: VTEP encapsulation & decapsulation

- A VTEP has two logical interfaces: an uplink and a downlink
  - Uplink to encapsulate
  - Downlink to decapsulate
- The VTEP can be located either on a physical switch (e.g. a ToR) or within the hypervisor's virtual switch
- The *outer IP destination* address is that assigned to the destination VTEP
- The *outer IP source* address is that assigned to the VTEP sending the frame
- Packets received from a tenant's VM on the downlink are mapped to a VXLAN ID
  - A lookup is then performed in the VTEP Layer 2 table using the VXLAN ID and destination MAC address; this lookup provides the IP address of the destination VTEP
- Packets received from a VTEP on the uplink are mapped from the VXLAN ID to an IEEE 802.1Q VLAN ID and sent as Ethernet frames on the downlink to the VM



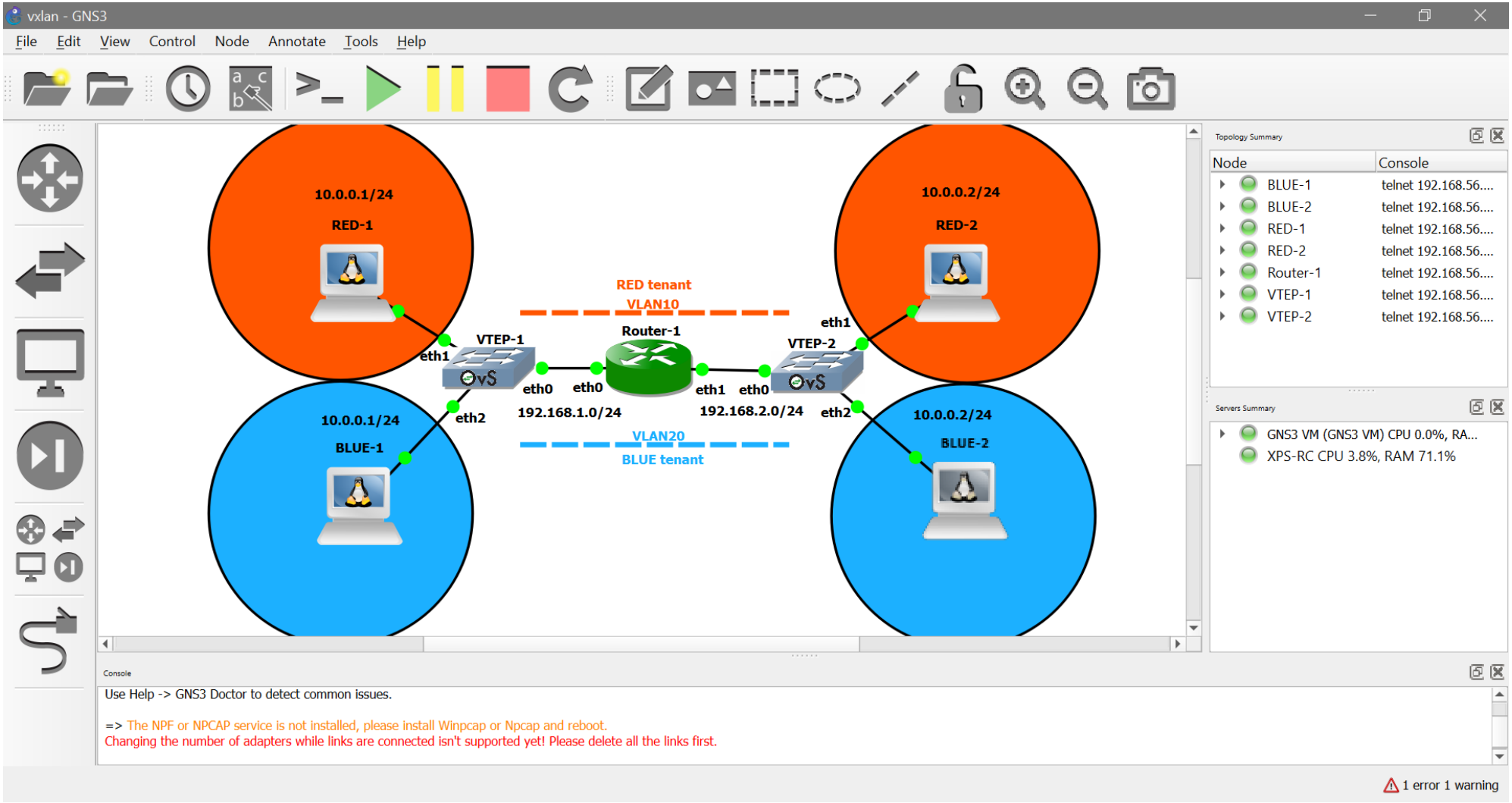
Demo: logical setup



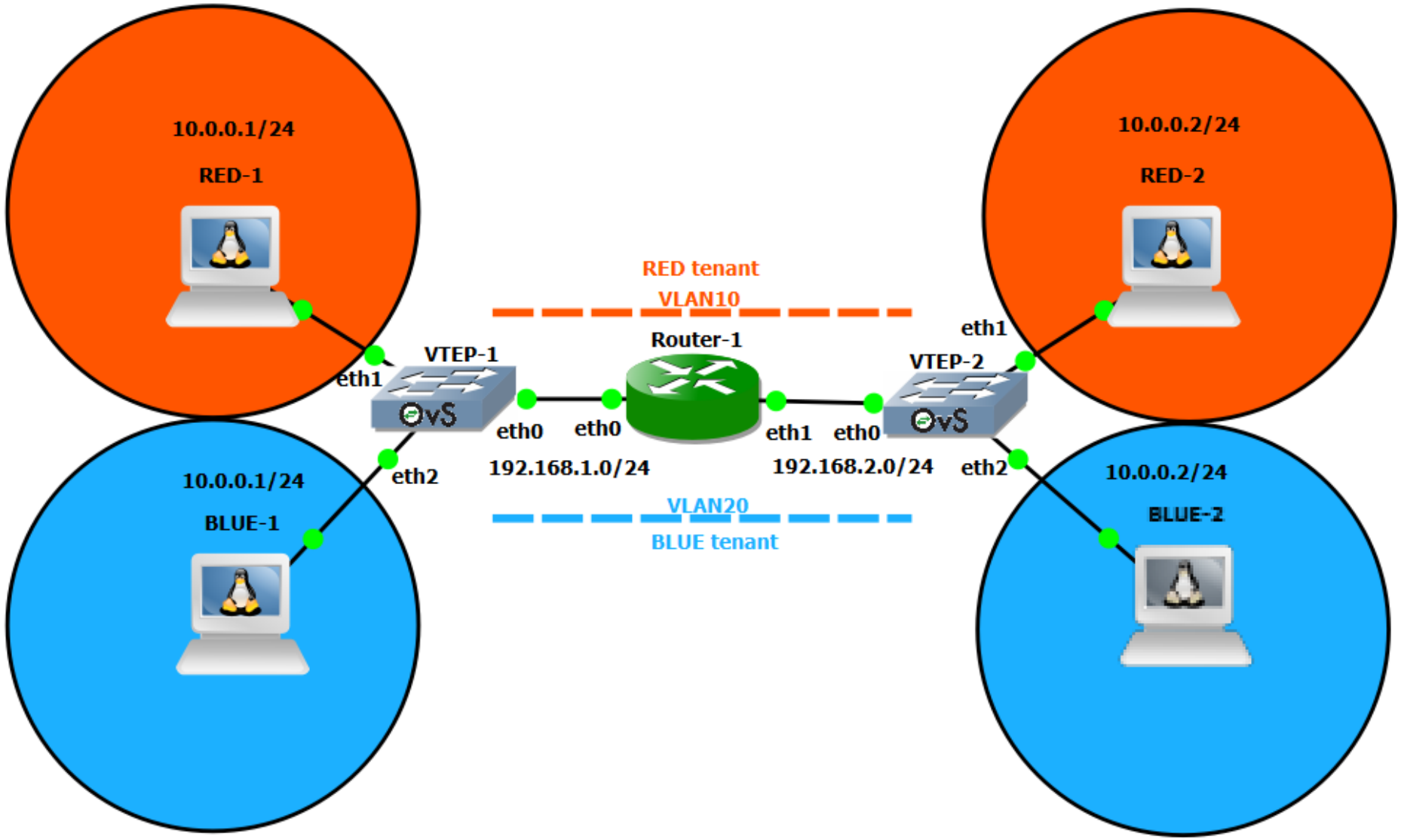
A similar demo based on mininet is presented here:

https://www.youtube.com/watch?v=QUmRUSqaAzc

Demo setup in GNS3 (1)



Demo setup in GNS3 (2)



Components of the demo setup in GNS3

- 4 end-systems implemented as Docker containers from the: **gns3/ubuntu:xenial** Docker image
- 1 Linux-based IP router implemented as a Docker container with an image derived from the: **kathara/base:debian10** Docker image
- 2 Open-vSwitch based VTEPs implemented as Docker containers from the: **gns3/openvswitch:latest** Docker image

Configuration of VTEPs

- Scripts to be executed to activate the two VTEPs

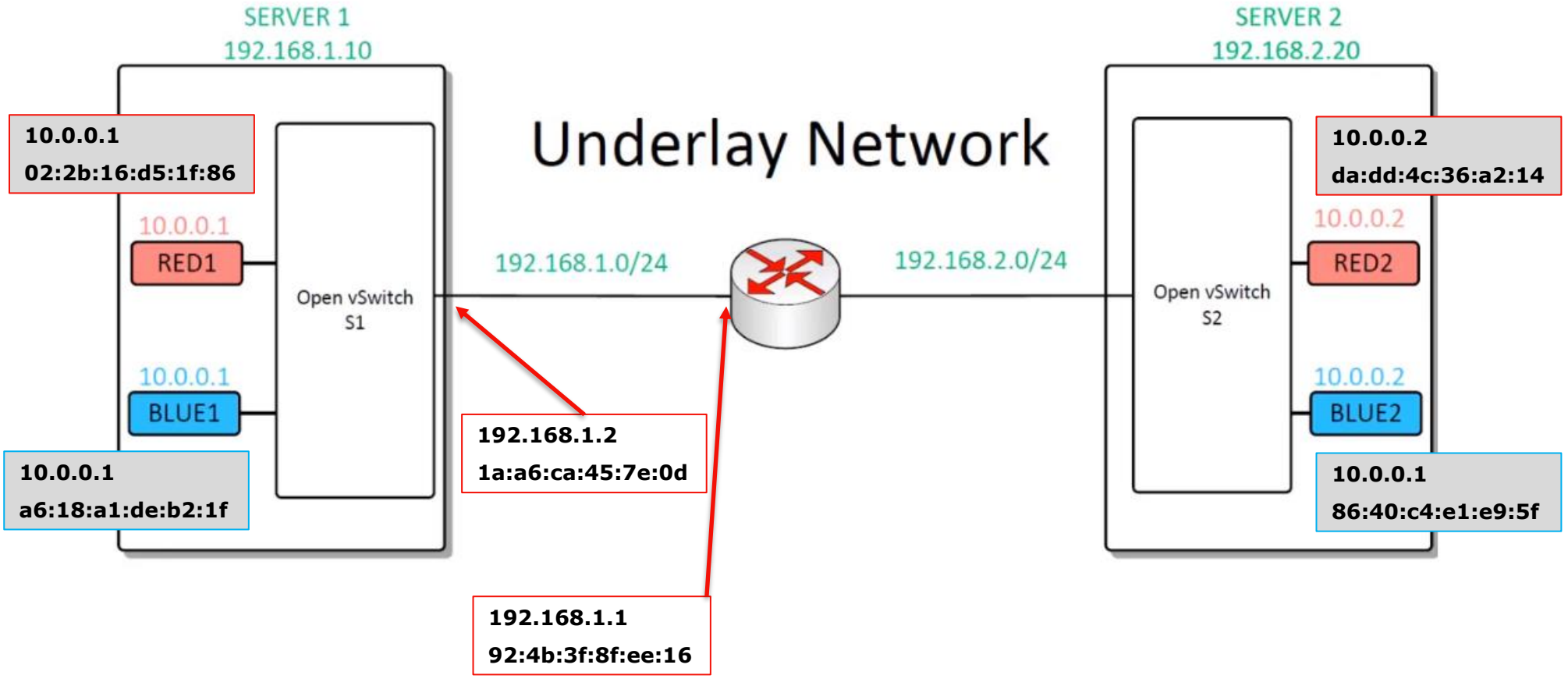
VTEP-1: ovs1.sh

```
#!/bin/sh
for br in `ovs-vsctl list-br`; do ovs-vsctl del-br ${br}; done
ovs-vsctl add-br ovs0
ovs-vsctl add-port ovs0 eth1 tag=10
ovs-vsctl add-port ovs0 vxlan1 tag=10 -- set interface vxlan1 type=vxlan \
          options:key=10 options:remote_ip=192.168.2.2
ovs-vsctl add-port ovs0 eth2 tag=20
ovs-vsctl add-port ovs0 vxlan2 tag=20 -- set interface vxlan2 type=vxlan \
          options:key=20 options:remote_ip=192.168.2.2
```

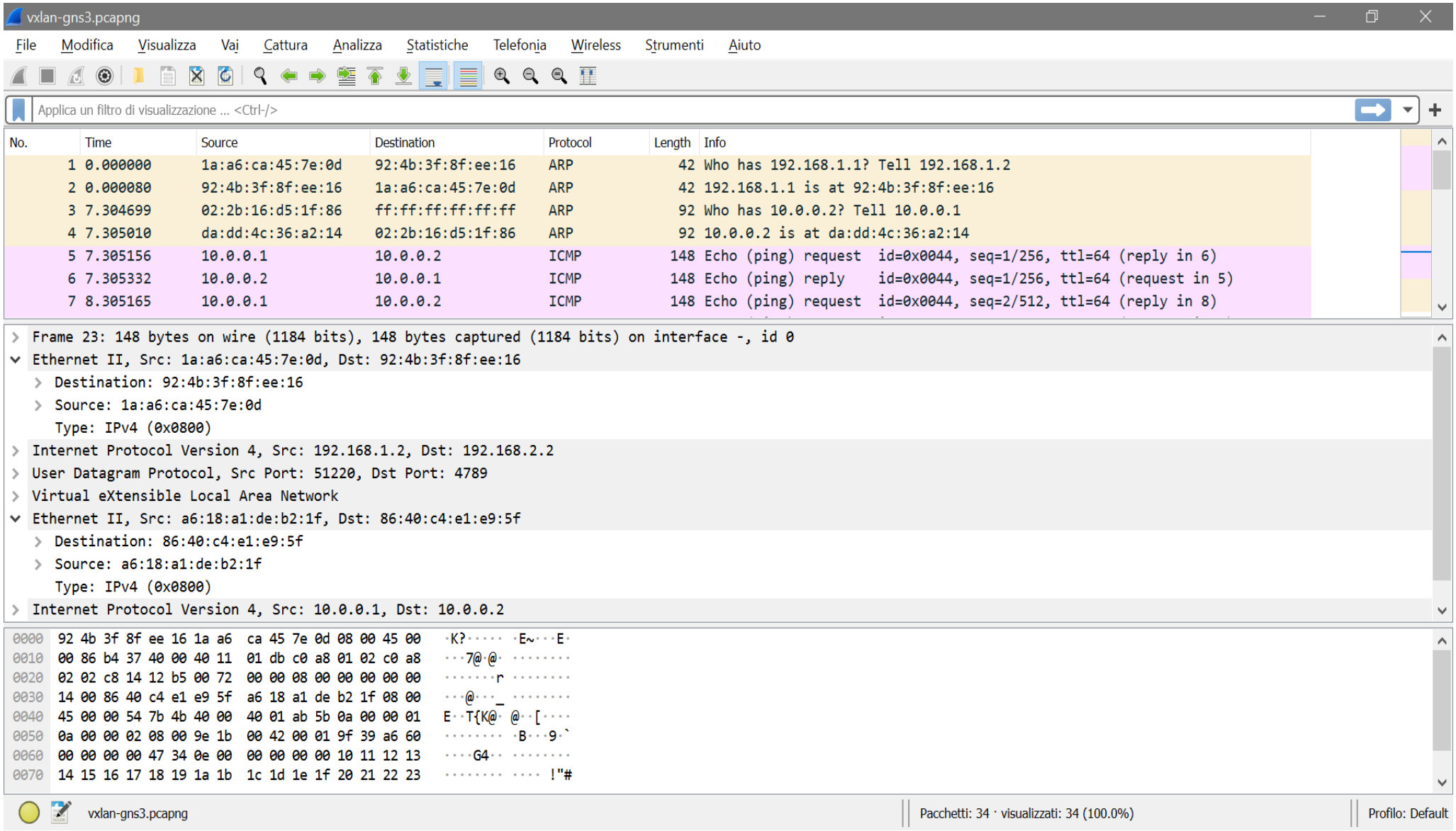
VTEP-2: ovs2.sh

```
#!/bin/sh
for br in `ovs-vsctl list-br`; do ovs-vsctl del-br ${br}; done
ovs-vsctl add-br ovs0
ovs-vsctl add-port ovs0 eth1 tag=10
ovs-vsctl add-port ovs0 vxlan1 tag=10 -- set interface vxlan1 type=vxlan \
          options:key=10 options:remote_ip=192.168.1.2
ovs-vsctl add-port ovs0 eth2 tag=20
ovs-vsctl add-port ovs0 vxlan2 tag=20 -- set interface vxlan2 type=vxlan \
          options:key=20 options:remote_ip=192.168.1.2
```

Demo: MAC and IP addresses in the GNS3 demo setup



Demo: analysis of packets with Wireshark



Some useful references

- Containers in GNS3 series: Advanced OpenVswitch switching
  https://gns3.com/community/blog/containers-in-gns3-series-advanc
- Connecting VMs Using Tunnels (Userspace)
  https://docs.openvswitch.org/en/latest/howto/userspace-tunneling/