University of Rome Tor Vergata
ICT and Internet Engineering

Network and System Defense

Alessandro Pellegrini, Angelo Tulumello

A.A. 2023/2024

Virtual LANs

Angelo Tulumello

Slides by Prof. Marco Bonola

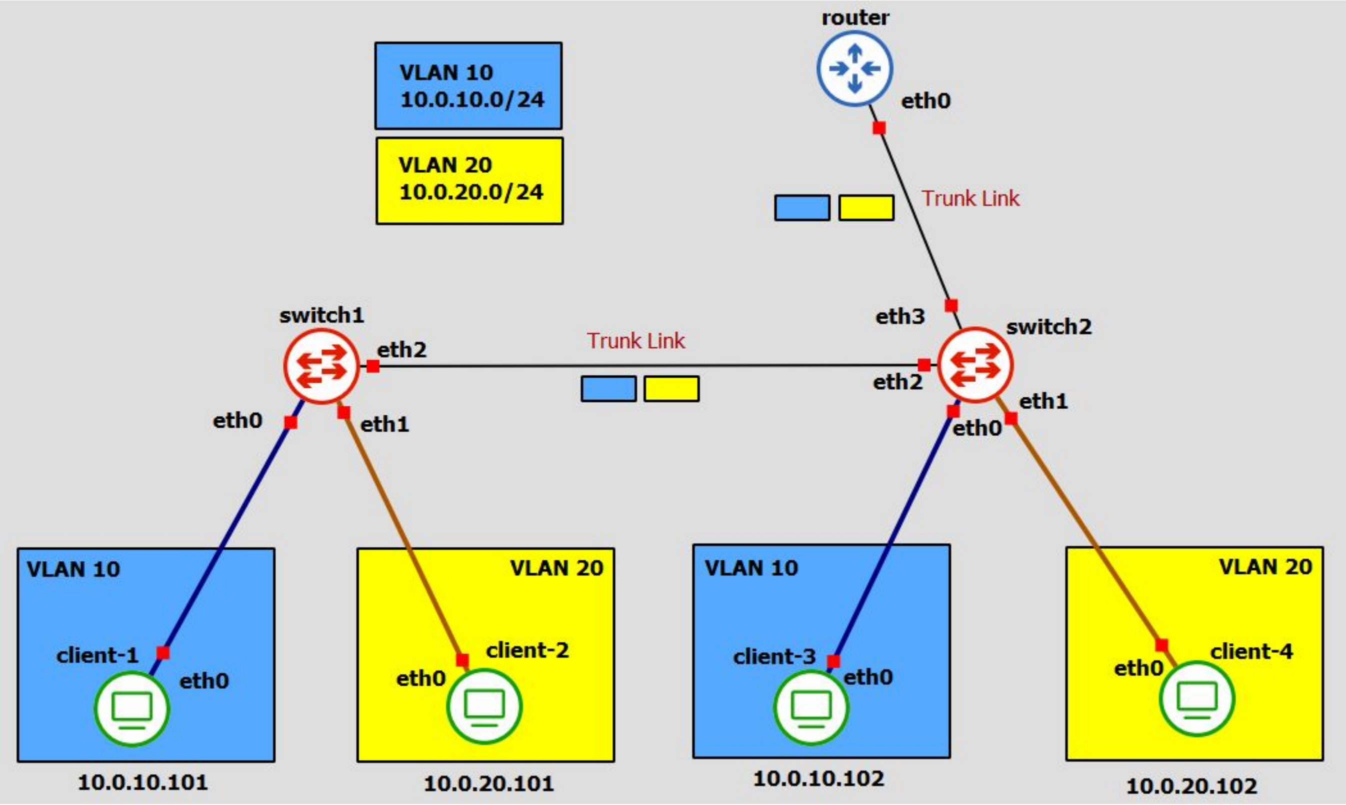
Other Slides Set

from Prof Salsano’s ITP Course

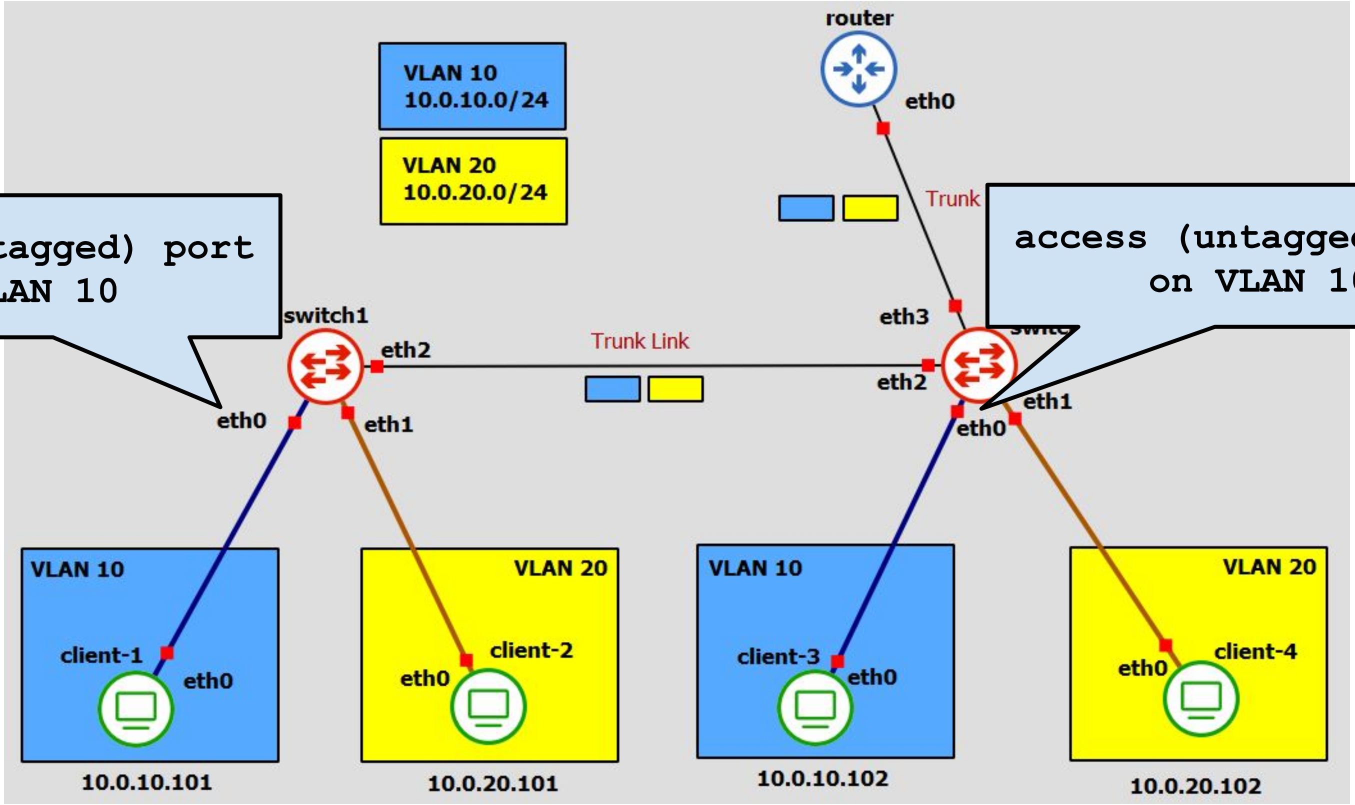
Lab3: 2 VLANs, 2 switch, 1 Router

Topology

client 1 da VLAN 10, client 2 da VLAN 20 collegate a switch. mediante Access LInk, uguale per client 3 e 4. Tra i due switch c'è TRUNK LINK, per il passaggio tra le VLAN, dove avviene la comunicazione. I due switch hanno stessa immagine, con 4 interfacce. Lo switch 2 è collegato al router, usato per fare il routing tra VLAN10 e VLAN20, NON POSSIAMO ALTRIMENTI ANDARE tra le due VLAN. (infatti il TRUNK Link non permette "scambio di VLAN", per ogni cavo una sola VLAN.



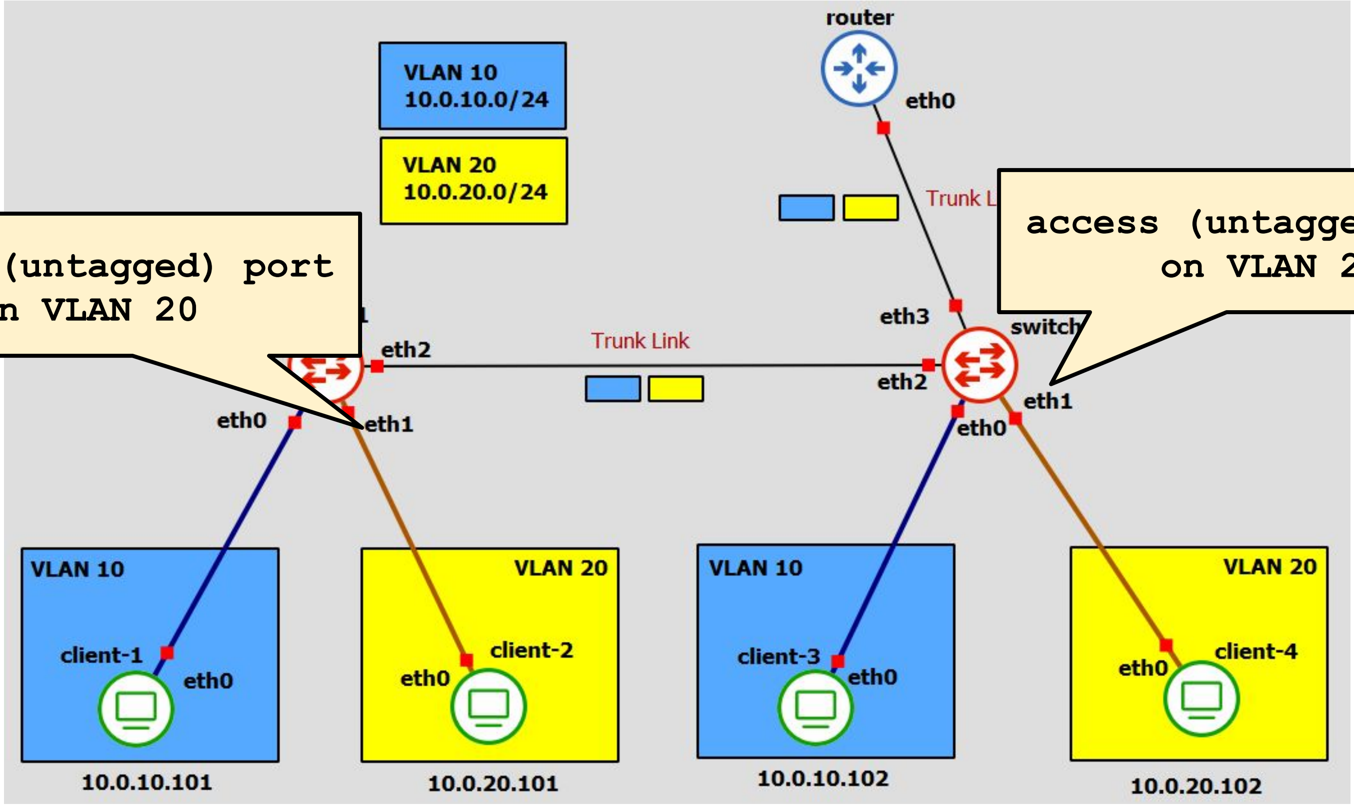
Topology



access (untagged) port on VLAN 10

access (untagged) port on VLAN 10

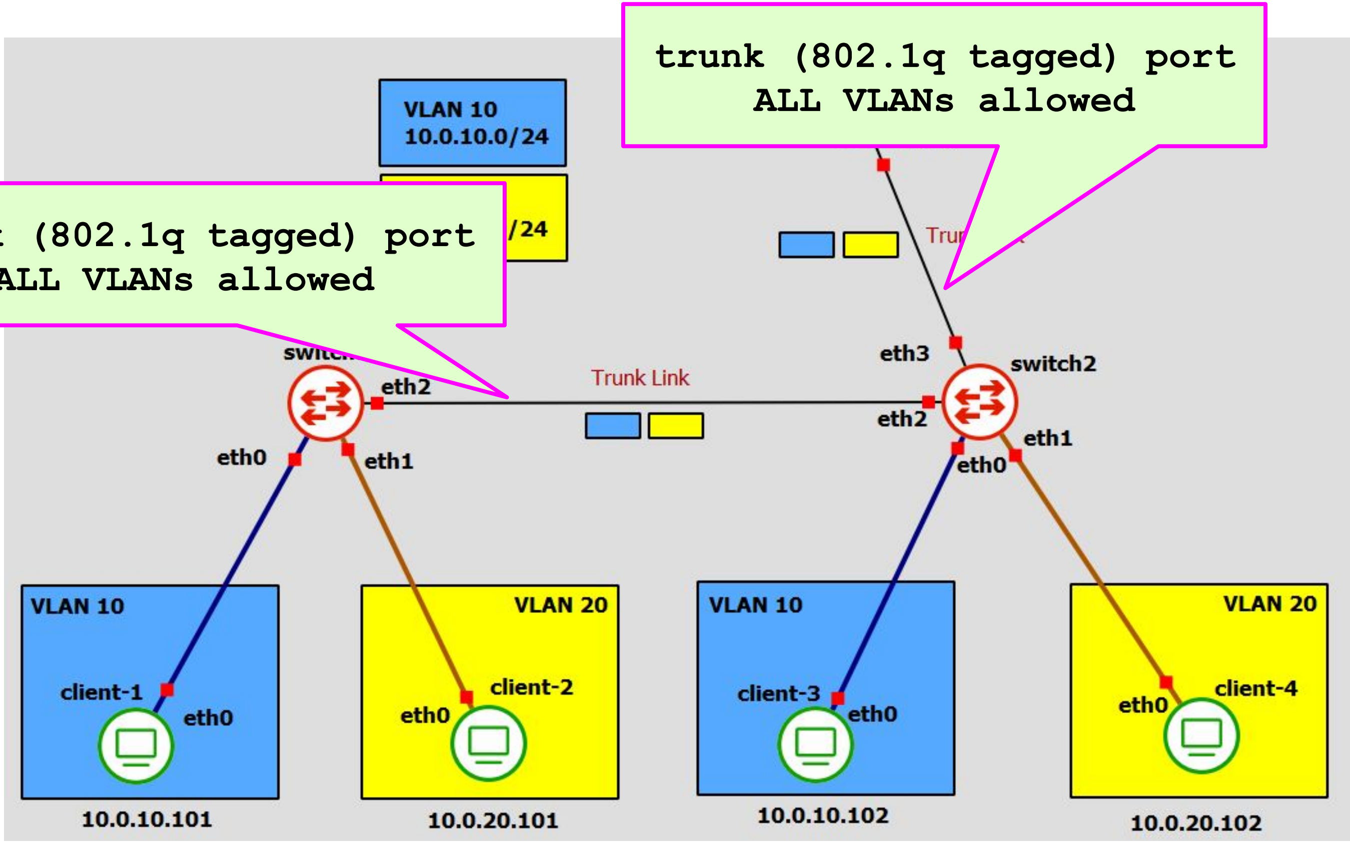
Topology



access (untagged) port on VLAN 20

access (untagged) port on VLAN 20

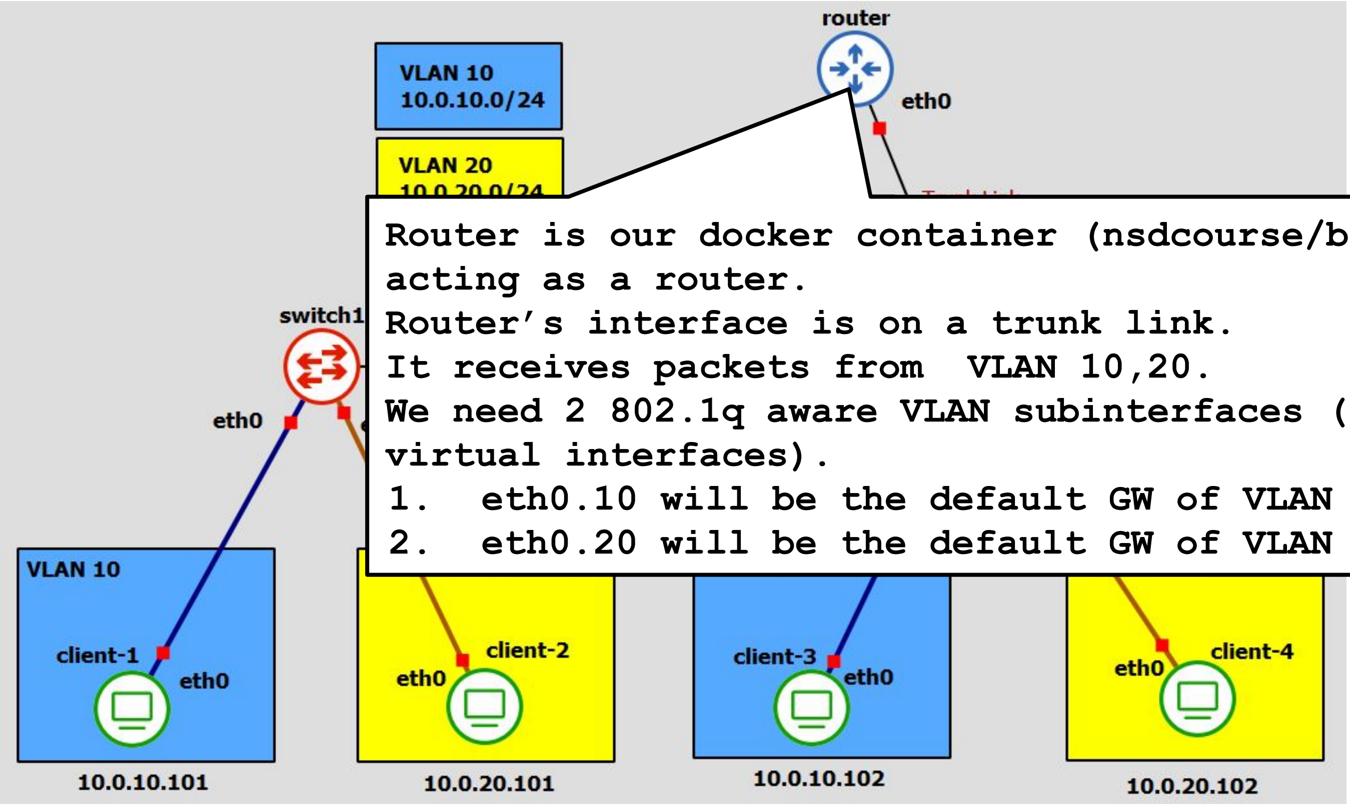
Topology



trunk (802.1q tagged) port
ALL VLANs allowed

trunk (802.1q tagged) port
ALL VLANs allowed

Topology



Router is our docker container (nsdcourse/basenet) acting as a router.

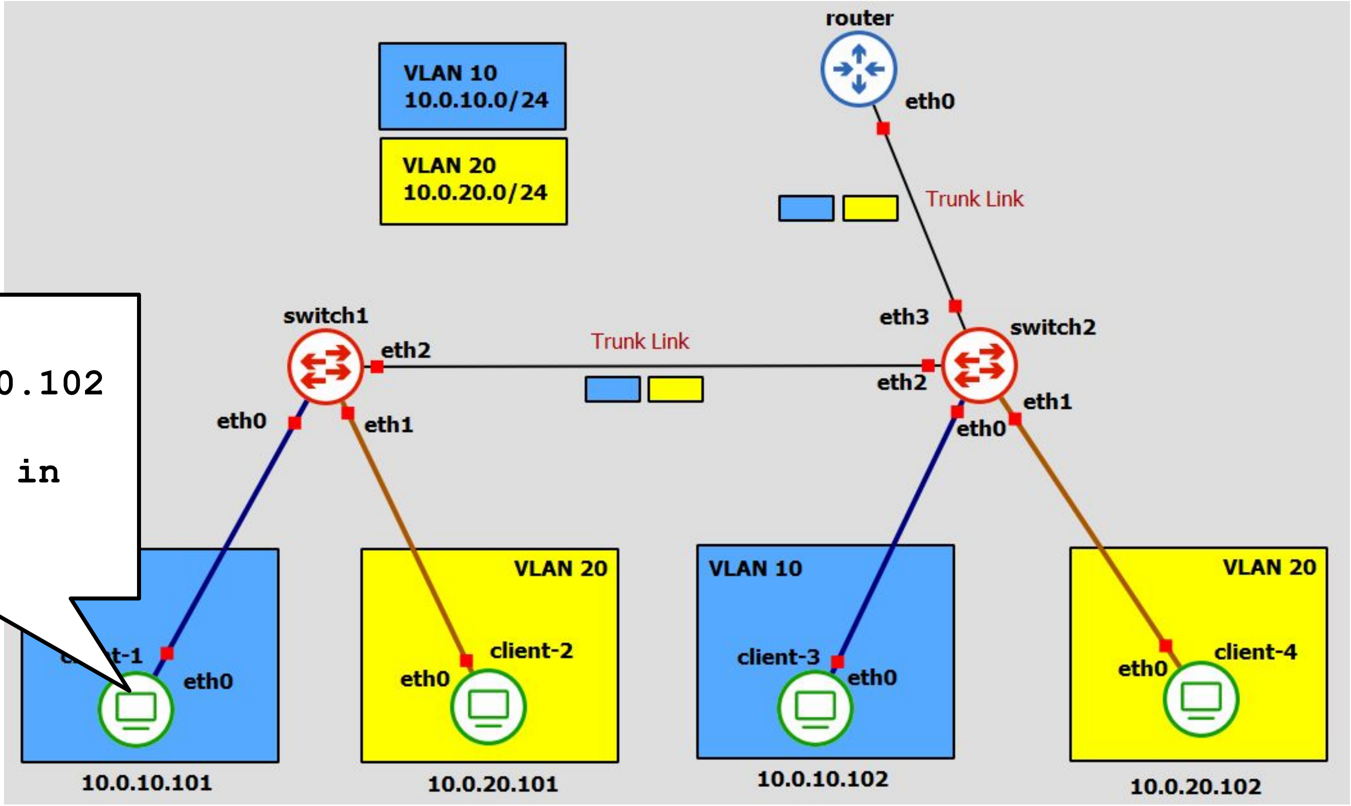
Router's interface is on a trunk link.

It receives packets from VLAN 10,20.

We need 2 802.1q aware VLAN subinterfaces (i.e. virtual interfaces).

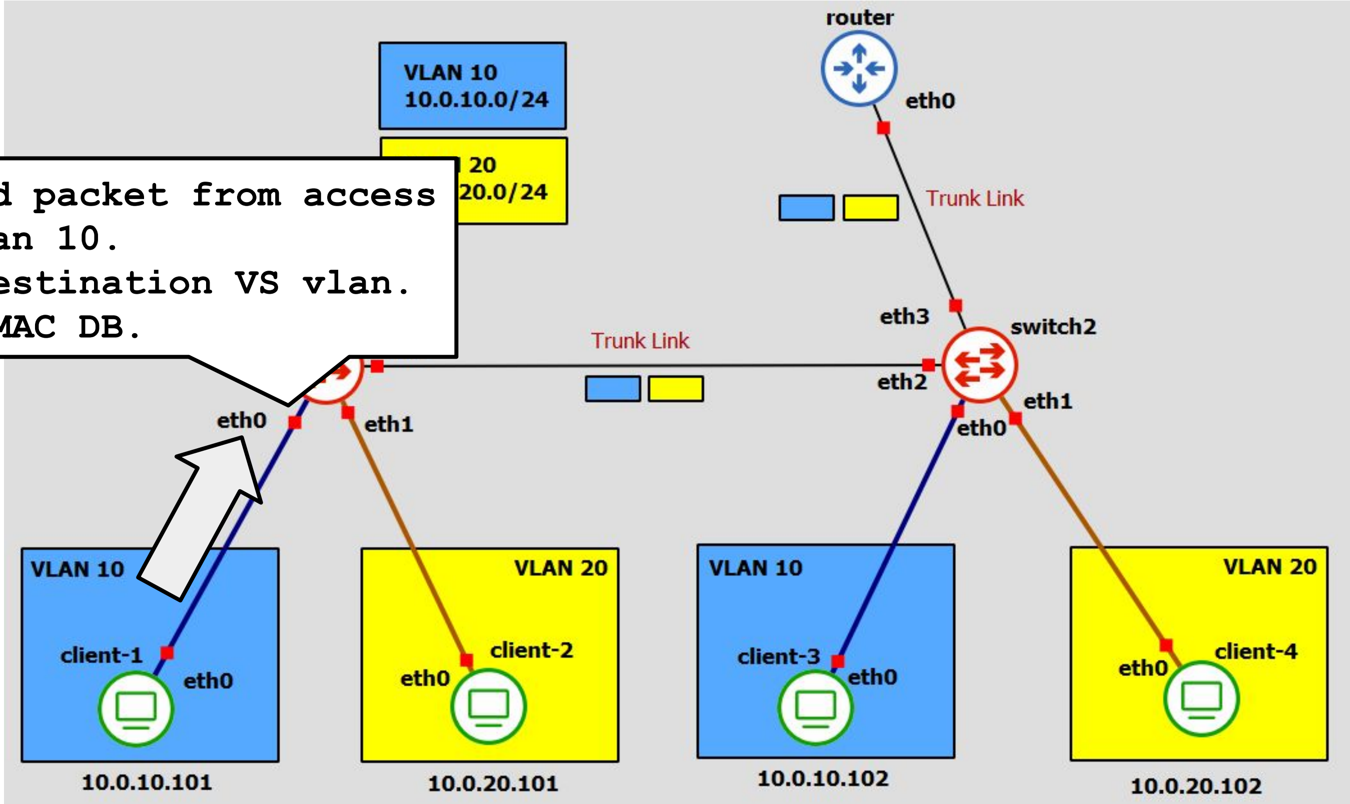
1. eth0.10 will be the default GW of VLAN 10
2. eth0.20 will be the default GW of VLAN 20

Forwarding Operations



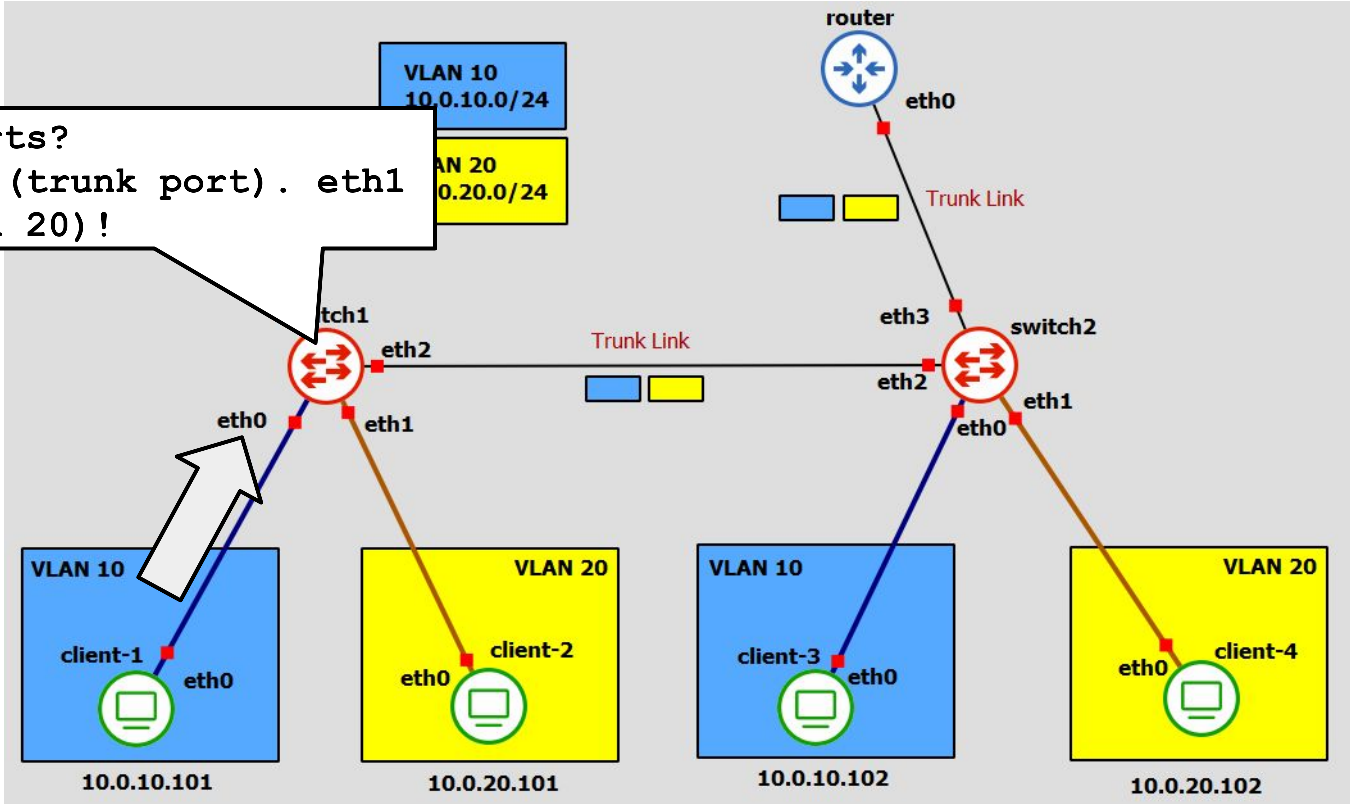
ping 10.0.10.102 (assuming destination in ARP cache)

Forwarding Operations



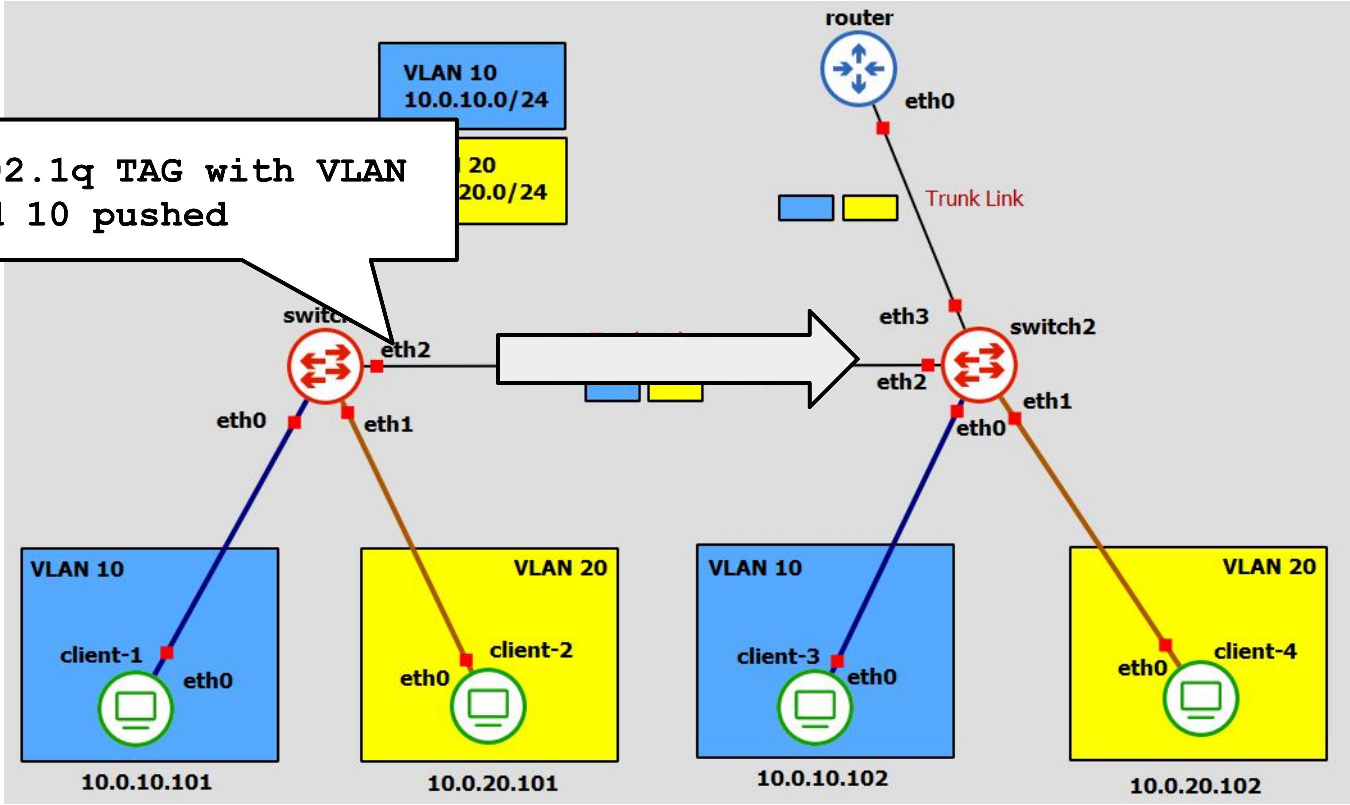
untagged packet from access port vlan 10.
check destination VS vlan.
update MAC DB.

Forwarding Operations



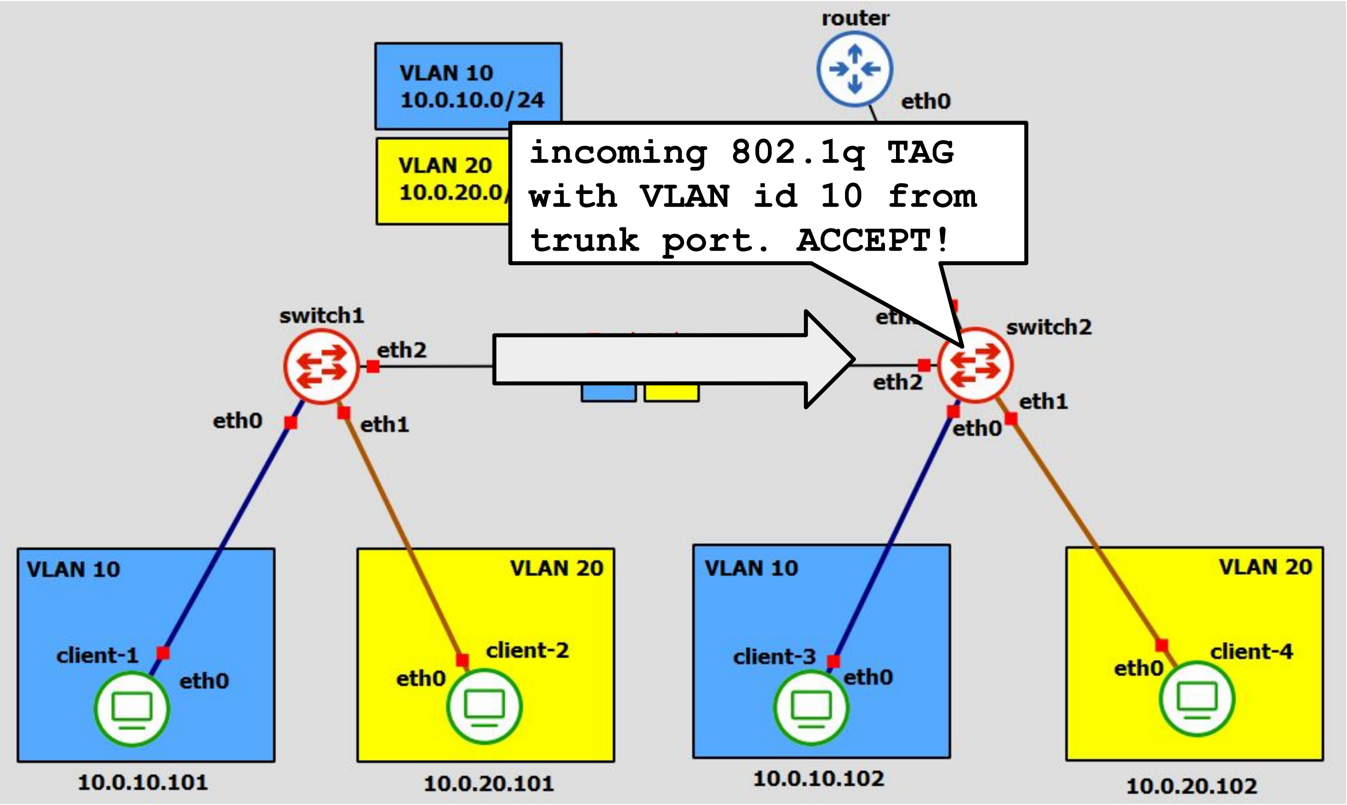
```
output ports?
only eth2 (trunk port). eth1
is in vlan 20)!
```

Forwarding Operations



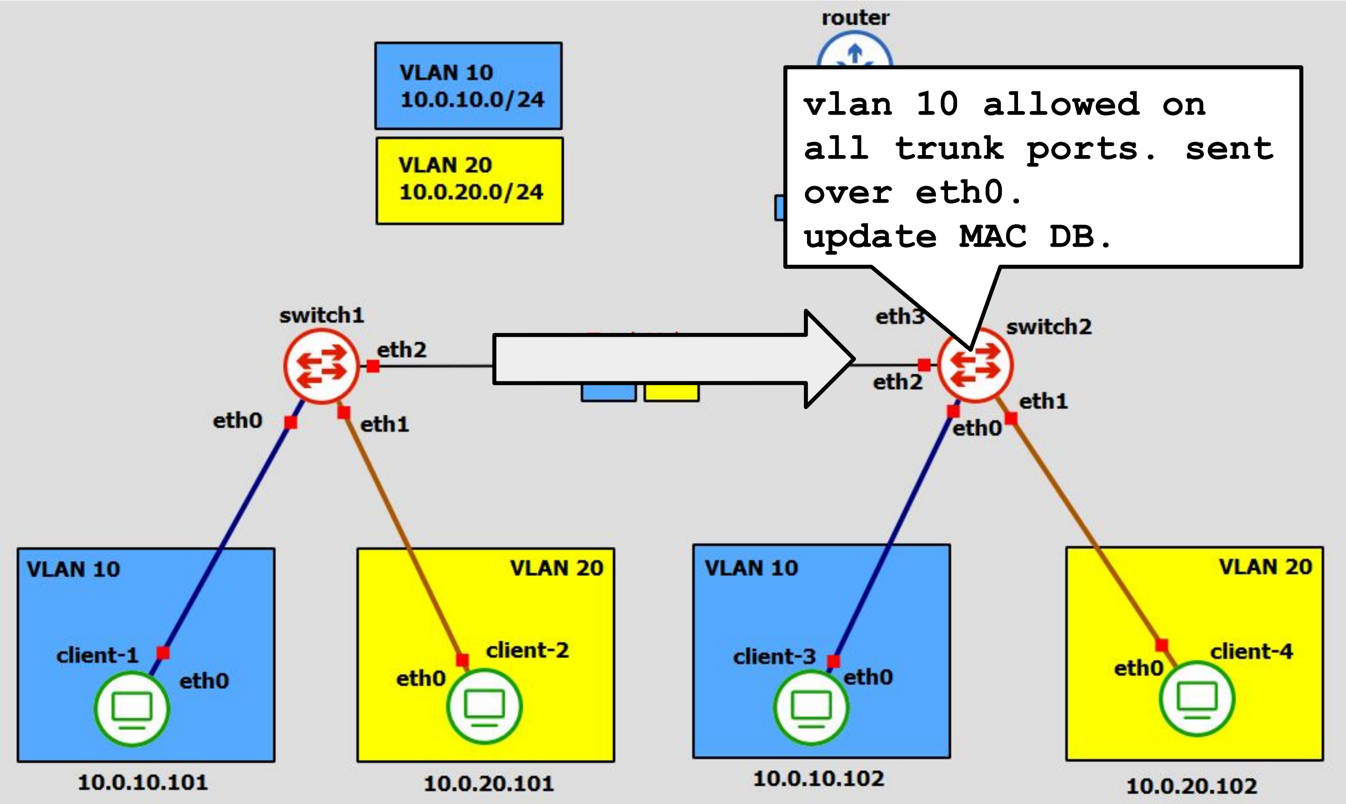
802.1q TAG with VLAN id 10 pushed

Forwarding Operations



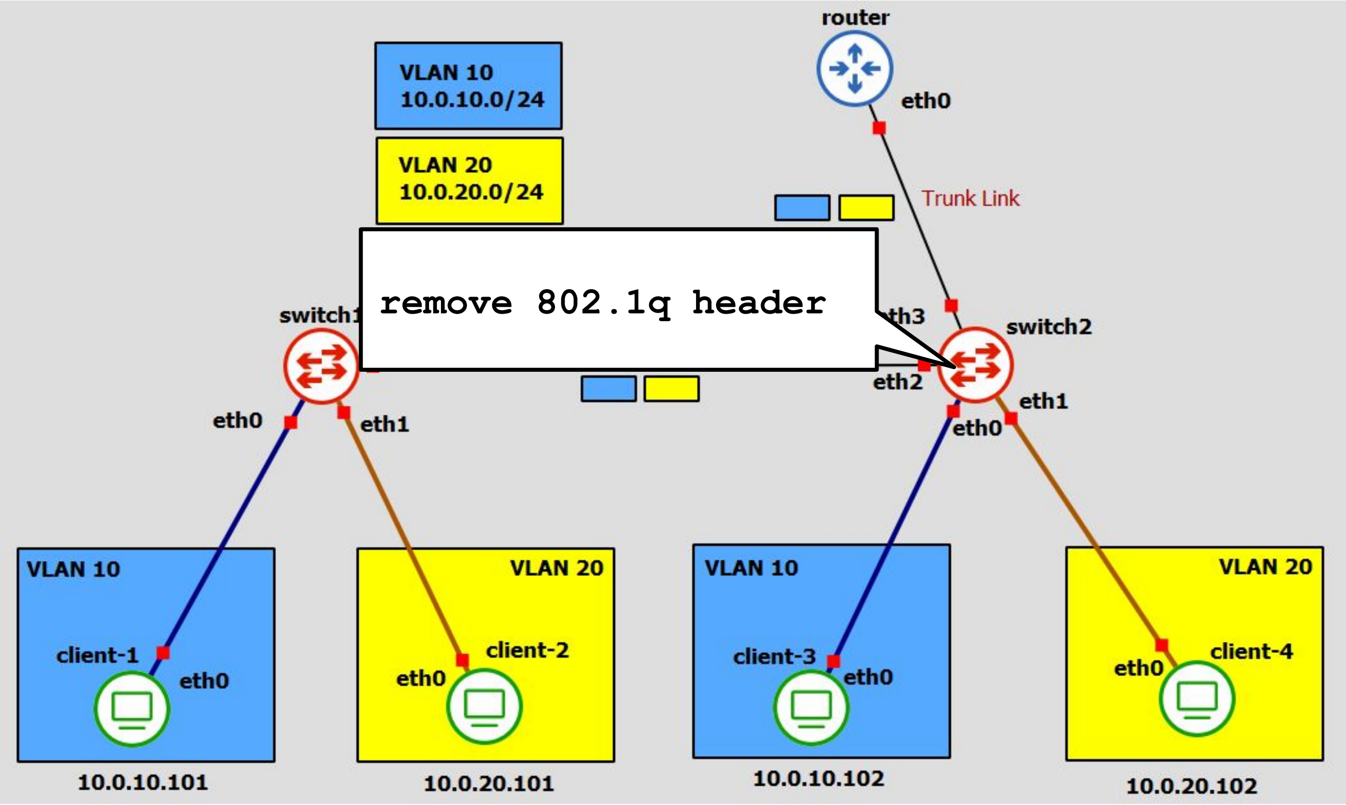
```
incoming 802.1q TAG
with VLAN id 10 from
trunk port. ACCEPT!
```

Forwarding Operations



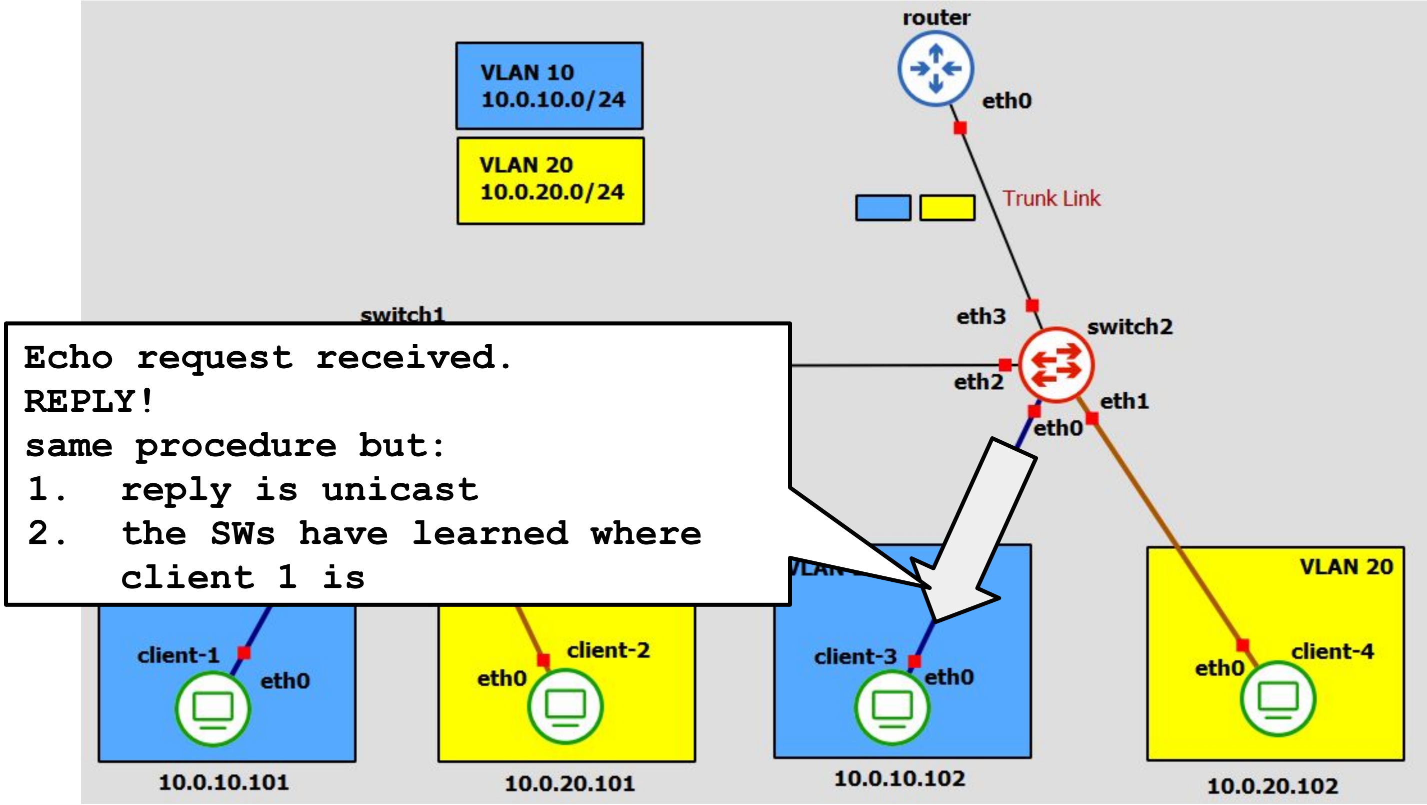
```
vlan 10 allowed on
all trunk ports. sent
over eth0.
update MAC DB.
```

Forwarding Operations



remove 802.1q header

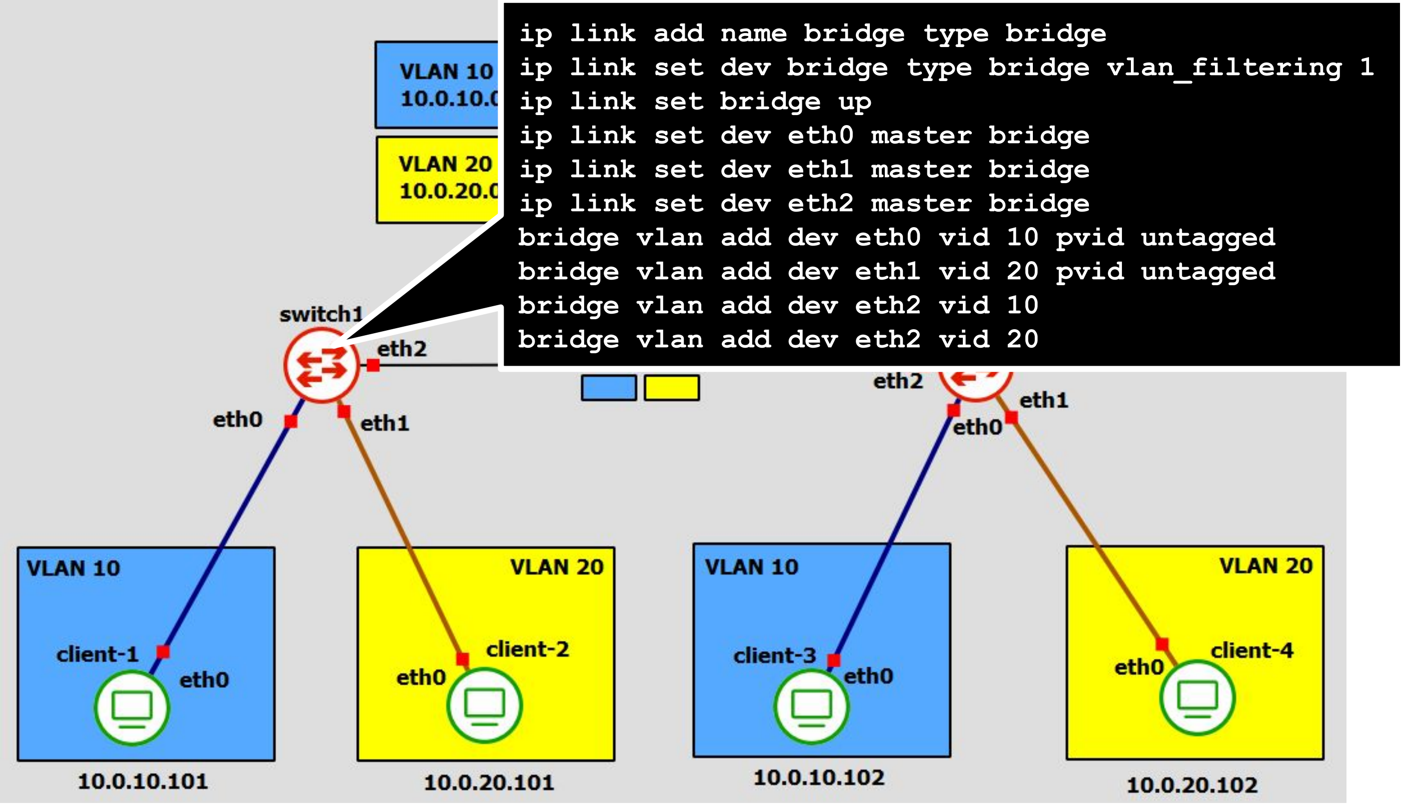
Forwarding Operations



```
Echo request received.
REPLY!
same procedure but:
1.   reply is unicast
2.   the SWs have learned where
     client 1 is
```

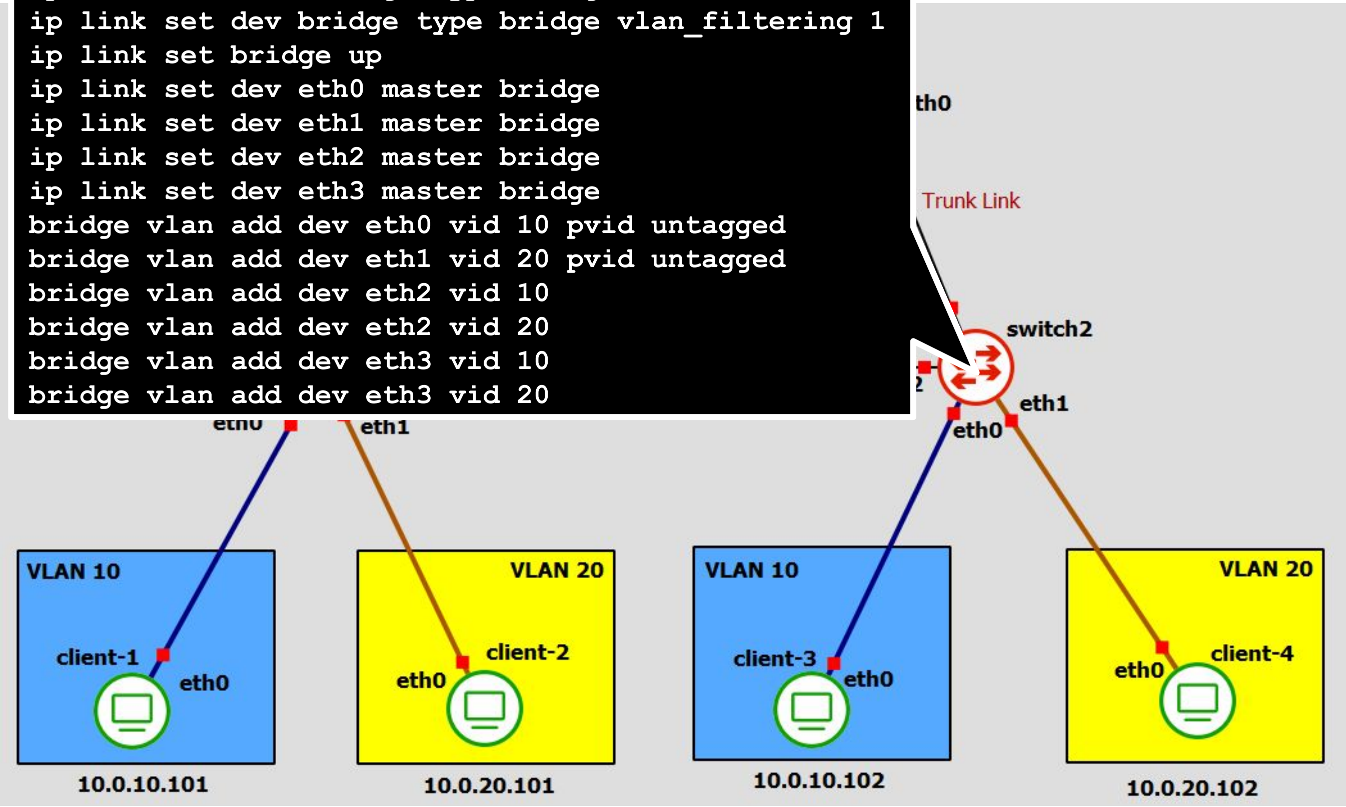
Configuration

- crea interfaccia bridge
- attivazione bridge
- aggiungiamo le interfacce facenti parti del bridge eth0,..eth2
- con i due comandi "untagged" gestiamo la config del bridge, selezioniamo l'opzione vlan, e per il device eth0 aggiungiamo l'id 10, di tipo pvid untagged, cioè è la primaria. poichè l'unica, possiamo mandare pkt untagged nel link. Si vede con "bridge vlan show".
- infine mettiamo trunk link, in cui ci passano entrambe le vlan.



```
ip link add name bridge type bridge
ip link set dev bridge type bridge vlan_filtering 1
ip link set bridge up
ip link set dev eth0 master bridge
ip link set dev eth1 master bridge
ip link set dev eth2 master bridge
bridge vlan add dev eth0 vid 10 pvid untagged
bridge vlan add dev eth1 vid 20 pvid untagged
bridge vlan add dev eth2 vid 10
bridge vlan add dev eth2 vid 20
```

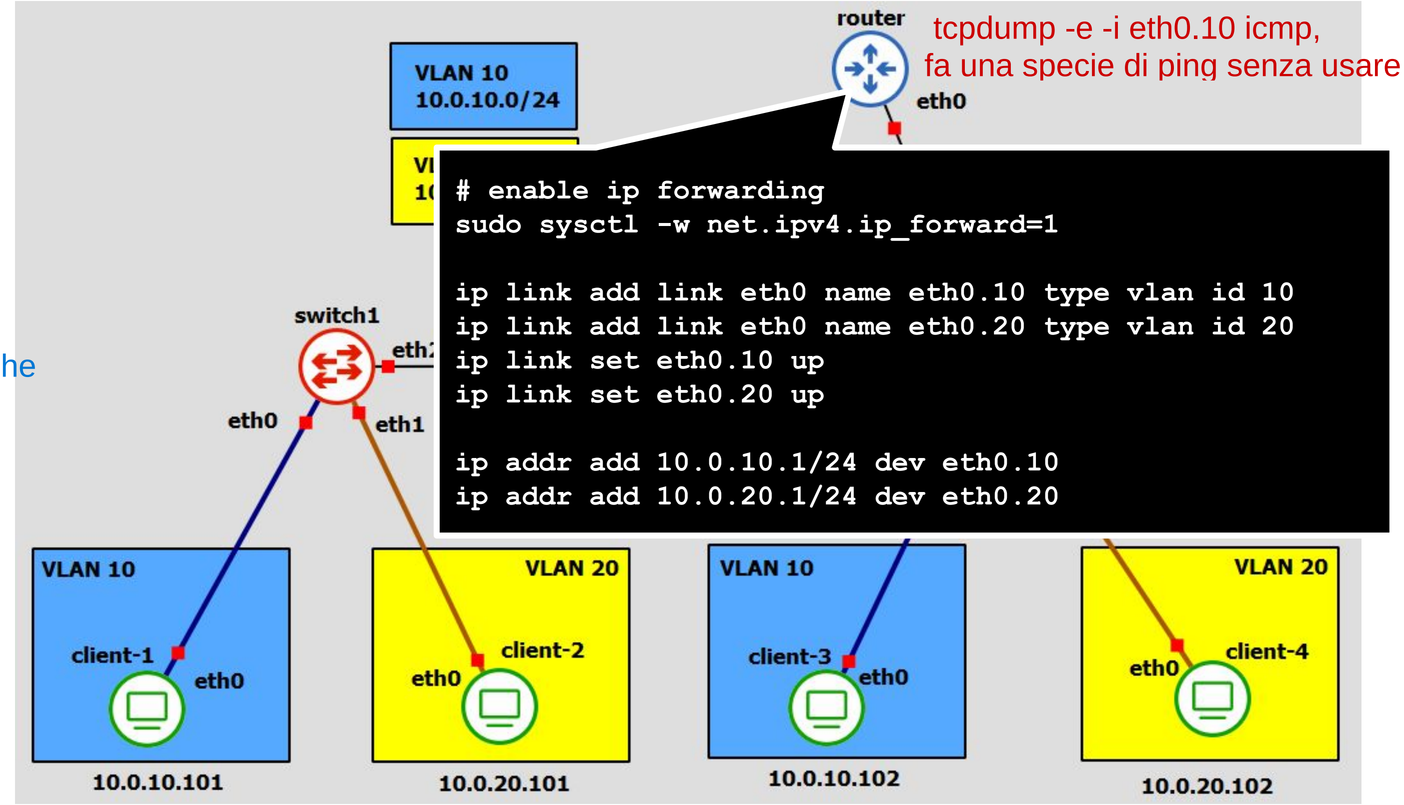
Configuration



```
ip link add name bridge type bridge
ip link set dev bridge type bridge vlan_filtering 1
ip link set bridge up
ip link set dev eth0 master bridge
ip link set dev eth1 master bridge
ip link set dev eth2 master bridge
ip link set dev eth3 master bridge
bridge vlan add dev eth0 vid 10 pvid untagged
bridge vlan add dev eth1 vid 20 pvid untagged
bridge vlan add dev eth2 vid 10
bridge vlan add dev eth2 vid 20
bridge vlan add dev eth3 vid 10
bridge vlan add dev eth3 vid 20
```

Configuration

- dico al S.O che deve attivare routing, cioè ip forwarding
- creiamo virtual interfaces, eth0.10 perchè collego eth' con vlan 10
  abbiamo creato eth0 che può dividere il traffico a seconda del tag, e fare il redirect.
- poi faccio enable

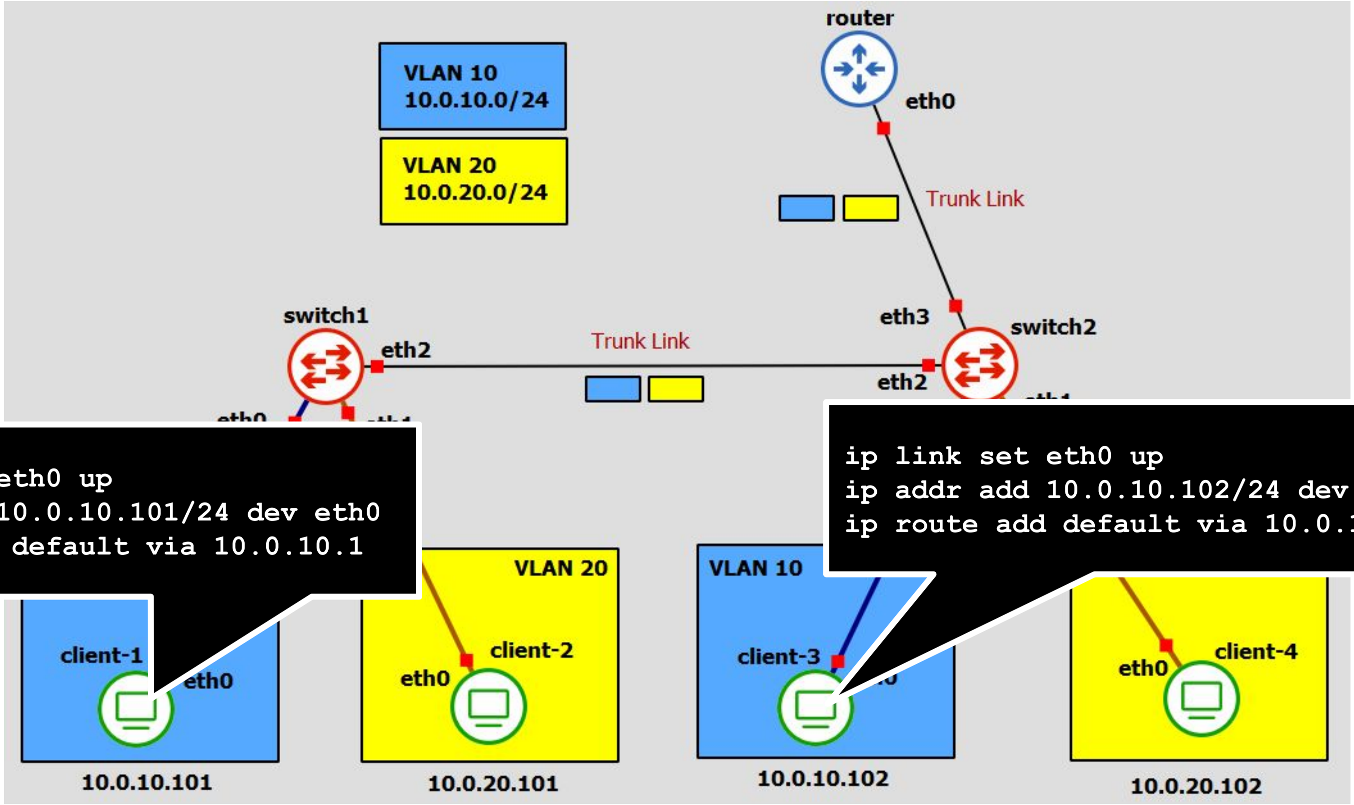


tcpdump -e -i eth0.10 icmp,
fa una specie di ping senza usare wireshark

```
# enable ip forwarding
sudo sysctl -w net.ipv4.ip_forward=1

ip link add link eth0 name eth0.10 type vlan id 10
ip link add link eth0 name eth0.20 type vlan id 20
ip link set eth0.10 up
ip link set eth0.20 up

ip addr add 10.0.10.1/24 dev eth0.10
ip addr add 10.0.20.1/24 dev eth0.20
```

attivo indirizzi ip che configuriamo dentro il router, associandola alla vlan 10 okt da client 1 passa per switch1, switch2, router, switch2 e client3. Abbiamo pkt doppi per questo.

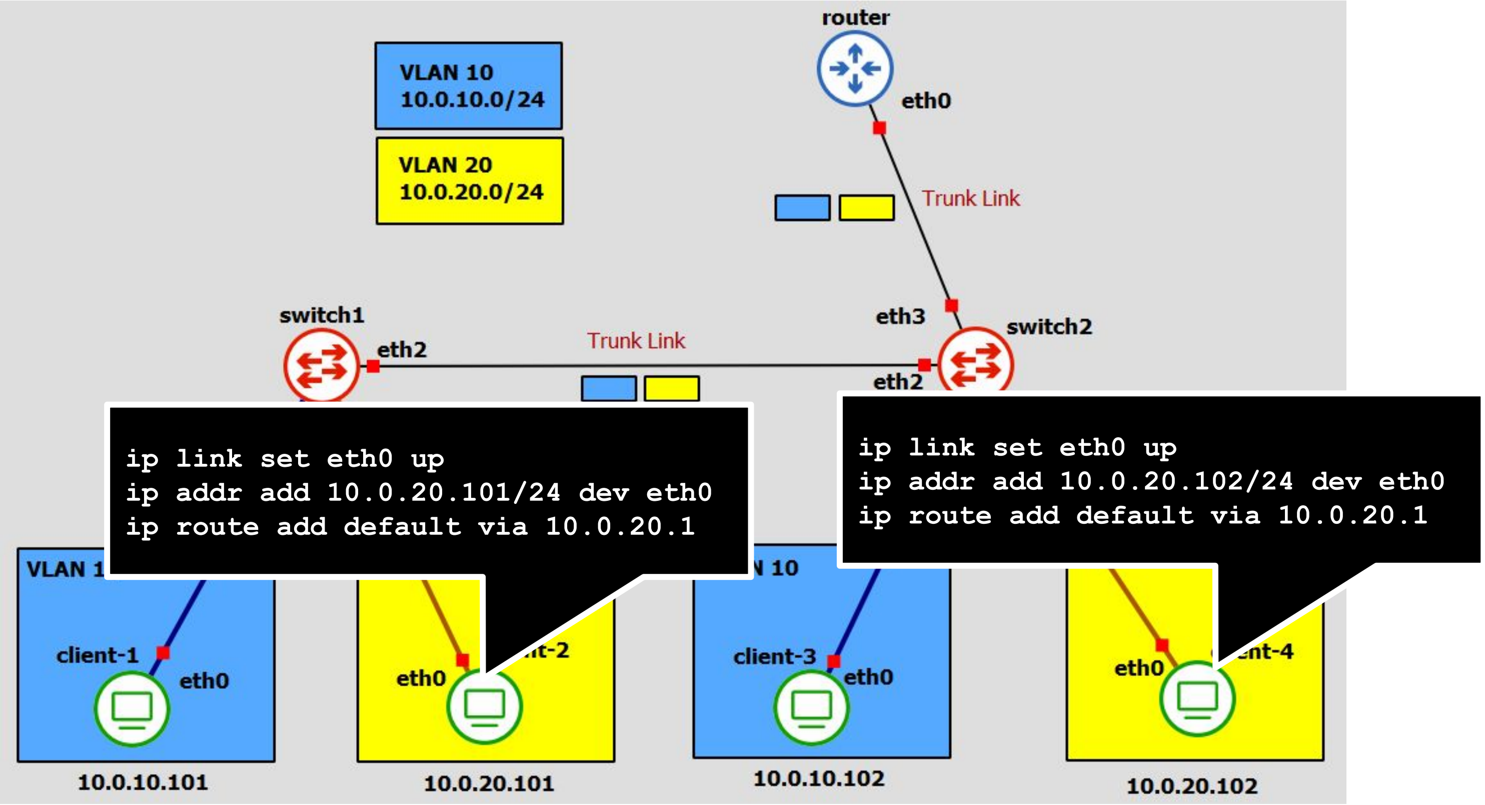
Configuration



```
ip link set eth0 up
ip addr add 10.0.10.101/24 dev eth0
ip route add default via 10.0.10.1
```

```
ip link set eth0 up
ip addr add 10.0.10.102/24 dev eth0
ip route add default via 10.0.10.1
```

Configuration



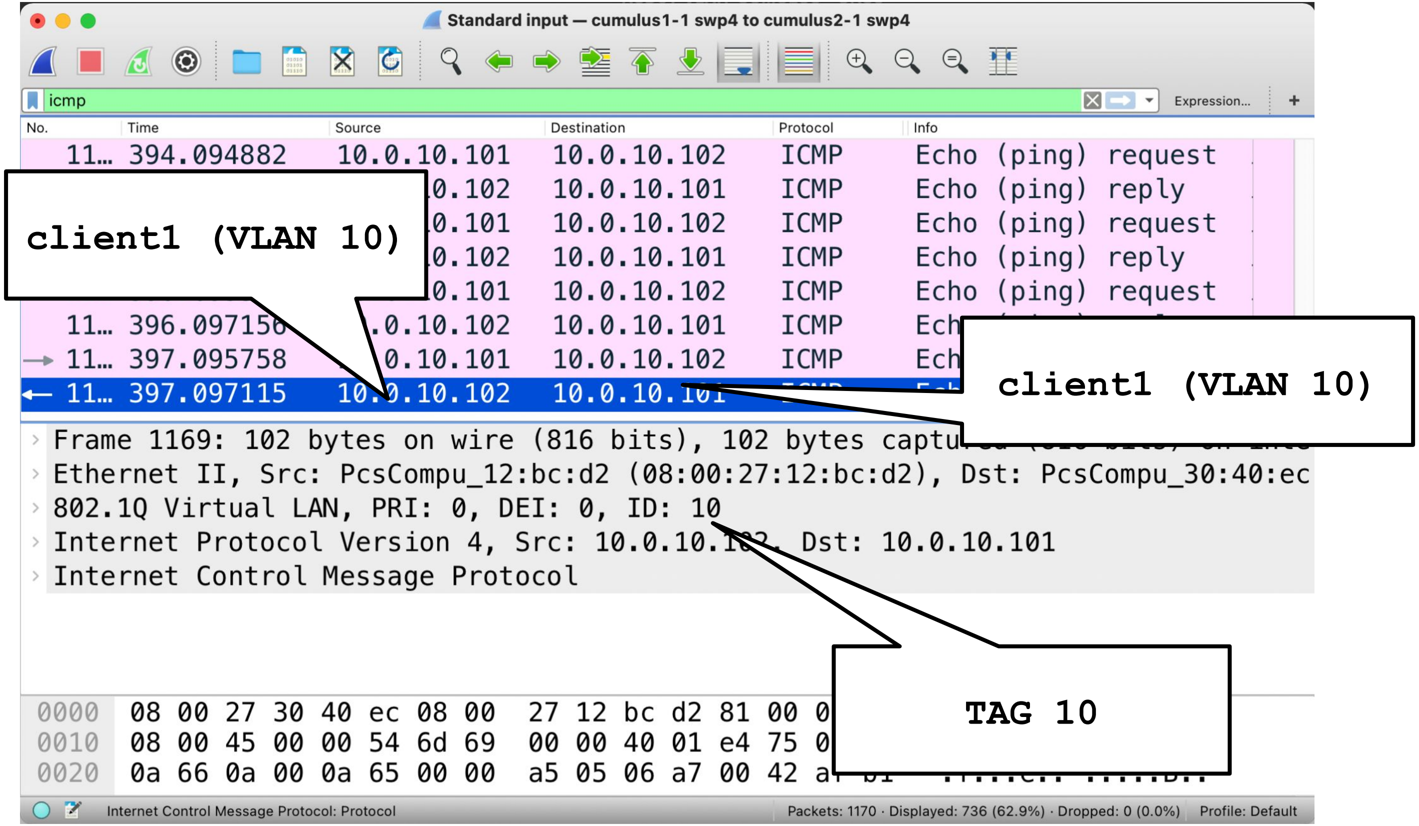
```
ip link set eth0 up
ip addr add 10.0.20.101/24 dev eth0
ip route add default via 10.0.20.1
```

```
ip link set eth0 up
ip addr add 10.0.20.102/24 dev eth0
ip route add default via 10.0.20.1
```

Check the actual VLAN separation

1. broadcast packets from client1 only visible by client3
2. trunk link correctly tag the packet from client1 to client2
3. for inter-VLAN communication we need IP forwarding!

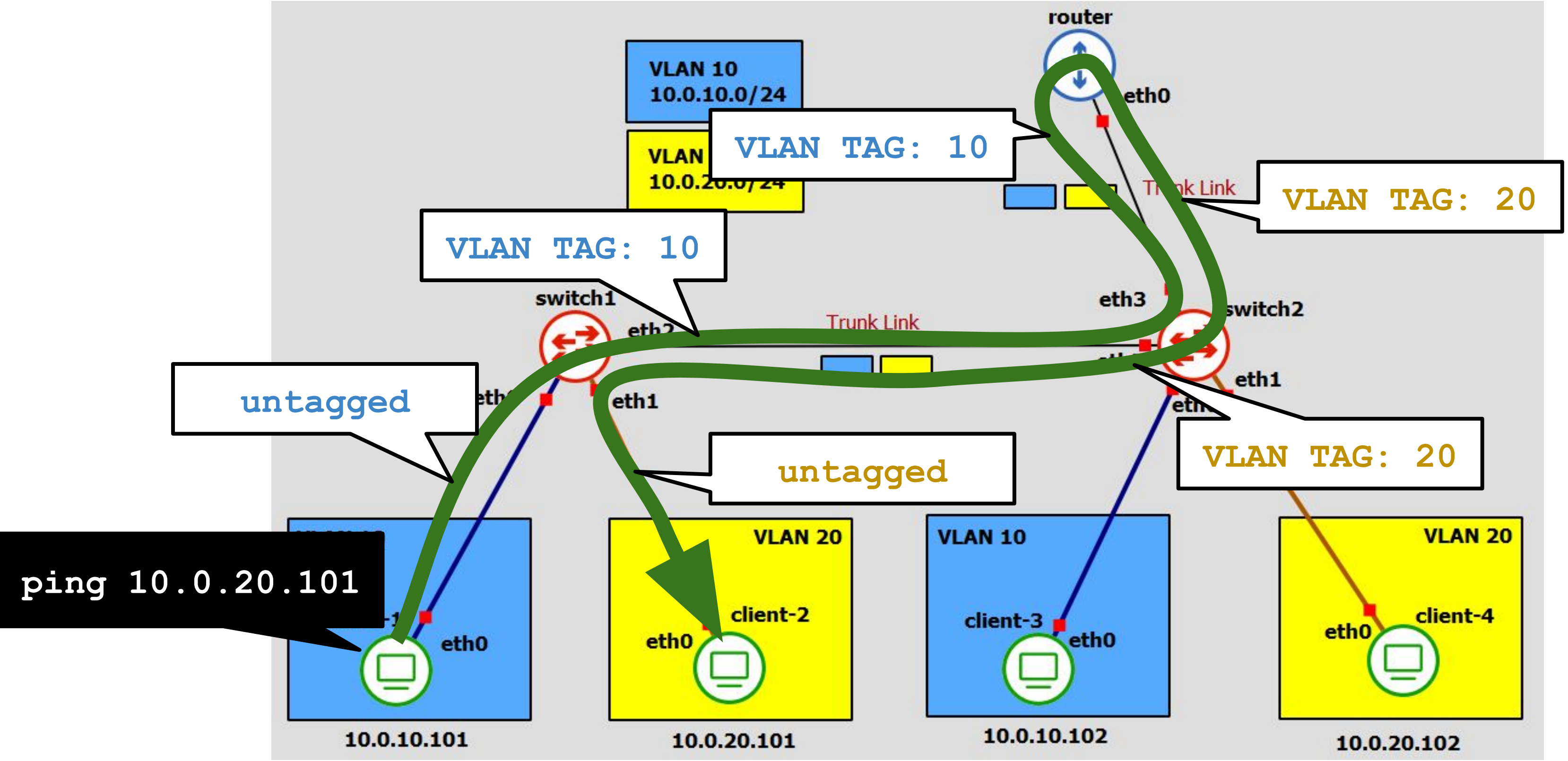
further check: statically bind an IP in VLAN 10 to client2 MAC address. ping this IP address. You will see packets in the link between switch1 and client2



```
No.     Time           Source          Destination     Protocol   Info
  11…   394.094882     10.0.10.101     10.0.10.102     ICMP       Echo (ping) request
                          0.102        10.0.10.101     ICMP       Echo (ping) reply
                          0.101        10.0.10.102     ICMP       Echo (ping) request
                          0.102        10.0.10.101     ICMP       Echo (ping) reply
                          0.101        10.0.10.102     ICMP       Echo (ping) request
  11…   396.097156     .0.10.102       10.0.10.101     ICMP       Ech
→ 11…   397.095758      0.10.101       10.0.10.102     ICMP       Ech
← 11…   397.097115     10.0.10.102     10.0.10.101

> Frame 1169: 102 bytes on wire (816 bits), 102 bytes captu
> Ethernet II, Src: PcsCompu_12:bc:d2 (08:00:27:12:bc:d2), Dst: PcsCompu_30:40:ec
> 802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 10
> Internet Protocol Version 4, Src: 10.0.10.102, Dst: 10.0.10.101
> Internet Control Message Protocol

0000  08 00 27 30 40 ec 08 00  27 12 bc d2 81 00 0
0010  08 00 45 00 00 54 6d 69  00 00 40 01 e4 75 0
0020  0a 66 0a 00 0a 65 00 00  a5 05 06 a7 00 42 a

Internet Control Message Protocol: Protocol     Packets: 1170 · Displayed: 736 (62.9%) · Dropped: 0 (0.0%)     Profile: Default
```

tagged packet on the trunk link between switch1 and switch2

Communicating between VLANs? Only via R1!!!



VLAN TAG: 10

VLAN TAG: 20

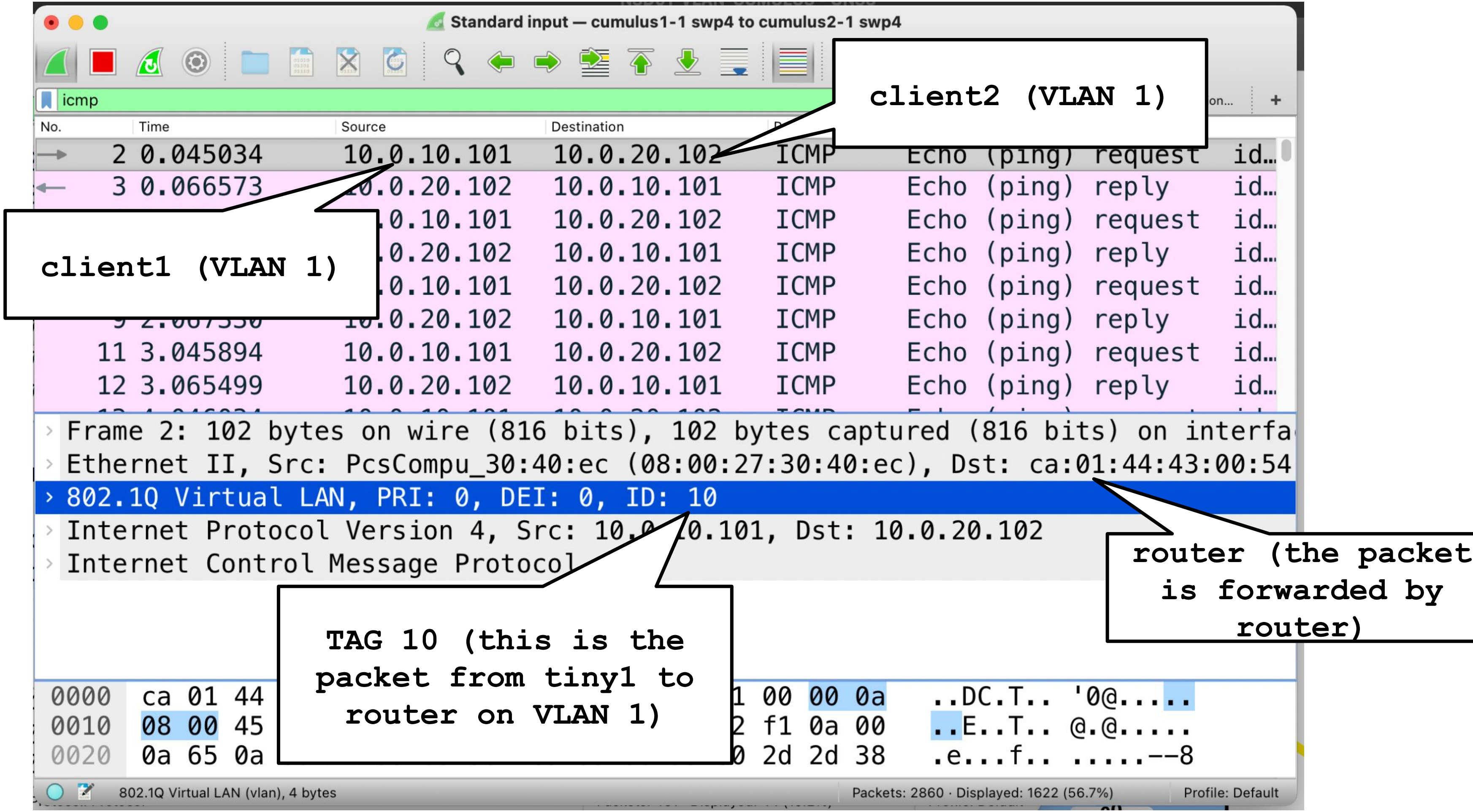
VLAN TAG: 10

untagged

untagged

VLAN TAG: 20

ping 10.0.20.101



Standard input — cumulus1-1 swp4 to cumulus2-1 swp4

icmp

```
No.      Time        Source         Destination
  2 0.045034    10.0.10.101    10.0.20.102    ICMP    Echo (ping) request  id…
  3 0.066573    10.0.20.102    10.0.10.101    ICMP    Echo (ping) reply    id…
                 .0.10.101     10.0.20.102    ICMP    Echo (ping) request  id…
                 .0.20.102     10.0.10.101    ICMP    Echo (ping) reply    id…
                 .0.10.101     10.0.20.102    ICMP    Echo (ping) request  id…
  9 2.067330    10.0.20.102    10.0.10.101    ICMP    Echo (ping) reply    id…
 11 3.045894    10.0.10.101    10.0.20.102    ICMP    Echo (ping) request  id…
 12 3.065499    10.0.20.102    10.0.10.101    ICMP    Echo (ping) reply    id…
```

```
> Frame 2: 102 bytes on wire (816 bits), 102 bytes captured (816 bits) on interfa
> Ethernet II, Src: PcsCompu_30:40:ec (08:00:27:30:40:ec), Dst: ca:01:44:43:00:54
> 802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 10
> Internet Protocol Version 4, Src: 10.0.10.101, Dst: 10.0.20.102
> Internet Control Message Protocol
```

```
0000  ca 01 44                1 00 00 0a    ..DC.T.. '0@.....
0010  08 00 45                2 f1 0a 00    ..E..T.. @.@.....
0020  0a 65 0a                0 2d 2d 38    .e...f.. .....--8
```

802.1Q Virtual LAN (vlan), 4 bytes   Packets: 2860 · Displayed: 1622 (56.7%)   Profile: Default

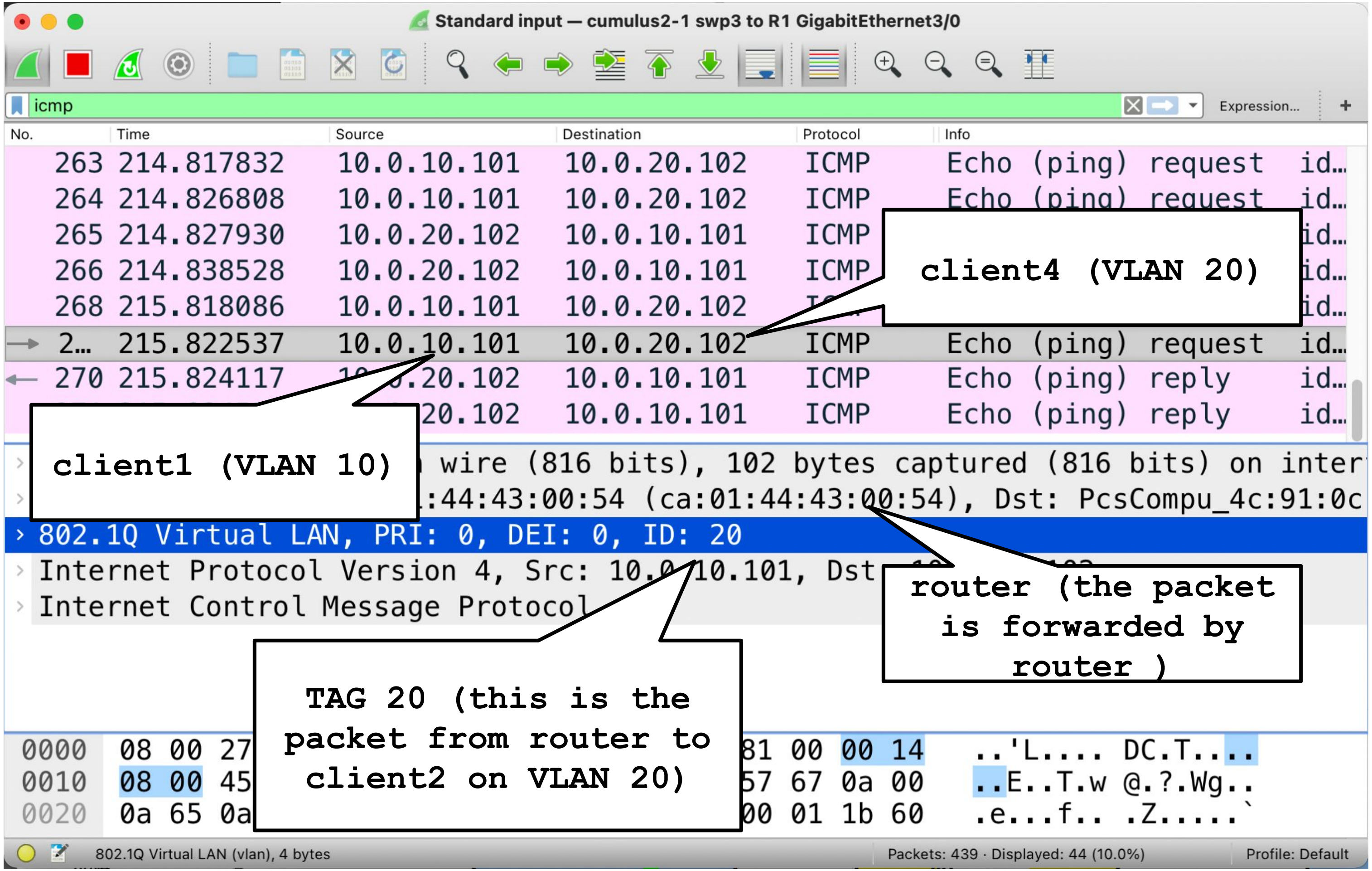
client2 (VLAN 1)

client1 (VLAN 1)

router (the packet is forwarded by router)

TAG 10 (this is the packet from tiny1 to router on VLAN 1)

tagged packet on the trunk link between switch1 and switch2



Standard input — cumulus2-1 swp3 to R1 GigabitEthernet3/0

```
No.     Time          Source         Destination    Protocol   Info
  263   214.817832    10.0.10.101    10.0.20.102    ICMP       Echo (ping) request  id…
  264   214.826808    10.0.10.101    10.0.20.102    ICMP       Echo (ping) request  id…
  265   214.827930    10.0.20.102    10.0.10.101    ICMP                            id…
  266   214.838528    10.0.20.102    10.0.10.101    ICMP                            id…
  268   215.818086    10.0.10.101    10.0.20.102    IC                              id…
→ 2…    215.822537    10.0.10.101    10.0.20.102    ICMP       Echo (ping) request  id…
← 270   215.824117    10...20.102    10.0.10.101    ICMP       Echo (ping) reply    id…
                         20.102      10.0.10.101    ICMP       Echo (ping) reply    id…
```

```
>                    wire (816 bits), 102 bytes captured (816 bits) on inter
>                    :44:43:00:54 (ca:01:44:43:00:54), Dst: PcsCompu_4c:91:0c
> 802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 20
> Internet Protocol Version 4, Src: 10.0.10.101, Dst
> Internet Control Message Protocol
```

```
0000  08 00 27              81 00 00 14   ..'L.... DC.T....
0010  08 00 45              57 67 0a 00   ..E..T.w @.?.Wg..
0020  0a 65 0a              00 01 1b 60   .e...f.. .Z.....`
```

802.1Q Virtual LAN (vlan), 4 bytes   Packets: 439 · Displayed: 44 (10.0%)   Profile: Default

client4 (VLAN 20)

client1 (VLAN 10)

router (the packet is forwarded by router)

TAG 20 (this is the packet from router to client2 on VLAN 20)

tagged packet on the trunk link between router and switch2

VLAN Security

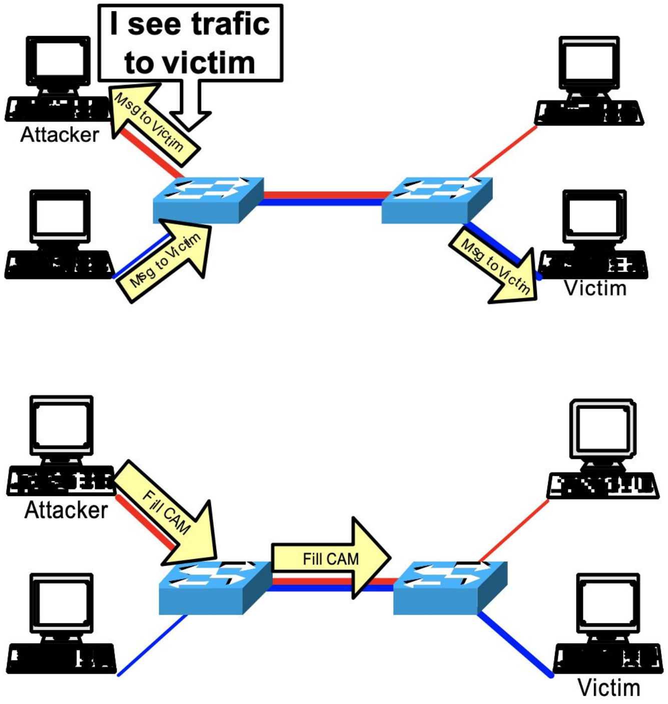
Based on “Rouiller, Steve A. "Virtual LAN Security: weaknesses and countermeasures” URL: https://www.sans.org/reading-room/whitepapers/networkdevs/virtual-lan-security-weaknesses-countermeasures-1090

LAYER 2 attacks landscape

- ❑ Media Access Control (MAC) attack ***(same as with no VLANs)***
- ❑ BASIC VLAN Hopping attack
- ❑ Double Encapsulation VLAN Hopping attack
- ❑ Address Resolution Protocol (ARP) attack ***(same as with no VLANs)***
- ❑ Spanning Tree Attack ***(same as with no VLANs)***
- ❑ VLAN Trunking Protocol attack
- ❑ Cisco Discovery Protocol (CDP) Attack
- ❑ Private VLAN (PVLAN) attack

Media Access Control (MAC) Attack

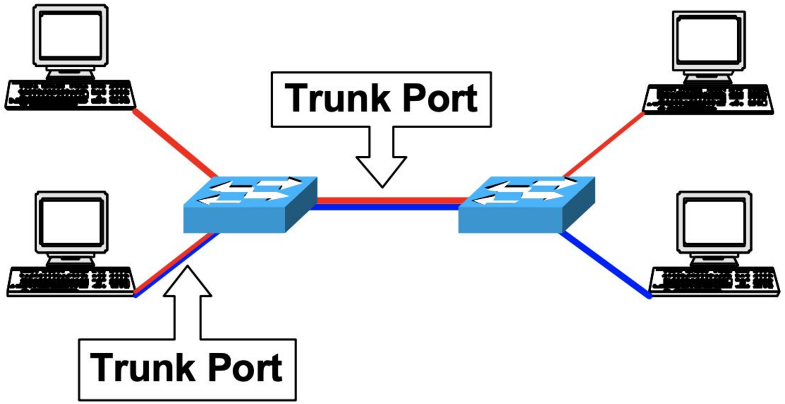
- This attack is based on ***Content Addressable Memory (CAM) Overflow***
- The CAM Table stores information such as MAC addresses available on physical ports with their associated VLAN parameters.
- CAM Tables have fixed size.
- Once the table is full, the traffic without CAM entry, floods on the local VLAN
- The MAC flooding attack can be mitigated by using the ***port-security*** features.
    - This allows to specify MAC addresses for each port or to learn a certain number of MAC addresses per port.



Basic VLAN Hopping attack

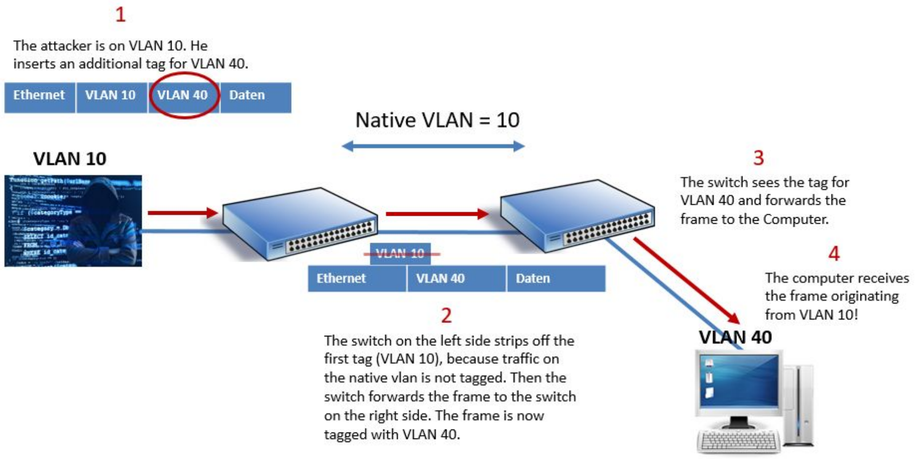
Se attivo trunk per ogni switch,
un attaccante può connettersi a tutte le VLAN.

- ❑ This attack is based on ***Dynamic Trunk Protocol*** (DTP) DTP is used for negotiating trunking on a link between two devices and for negotiating the type of trunking encapsulation (802.1Q) to be used.
- ❑ Cisco has fixed this with the new version of IOS and CATOS.
- ❑ As shown in the Figure, a station can spoof as a switch with 802.1Q signalling (using a rogue DTP frame). The station is then member of all VLANs.
- ❑ It requires a trunking favorable setting on the port
  - ❑ DTP enabled on the port
  - ❑ or in general it assumes an enabled trunk port



Double Encapsulation VLAN Hopping attack

- ***Double Tagging*** can only be exploited on switch ports configured to use native VLANs. Trunk ports configured with a native VLAN don't apply a VLAN tag when sending these frames.
- An attacker sends a double encapsulated 802.1Q frame with ***first TAG = native VLAN TAG***
- The first switch strips off the first encapsulation and then sends it back out
- The second switch strips off the second encapsulation and sends the frame to another VLAN ID.
- With this attack, the attacker can only send packets, and not receive them (***Unidirectional traffic only)***.
- As the attacker requires a trunking favorable setting on the port
    - on some implementations it also works with the attacker connected to an access port



The attacker is on VLAN 10. He inserts an additional tag for VLAN 40.

The switch sees the tag for VLAN 40 and forwards the frame to the Computer.

The computer receives the frame originating from VLAN 10!

The switch on the left side strips off the first tag (VLAN 10), because traffic on the native vlan is not tagged. Then the switch forwards the frame to the switch on the right side. The frame is now tagged with VLAN 40.

to defeat this attack:
1. the administrator should disable Auto-trunking
2. use dedicated VLANID for all trunk ports. The administrator mustn’t use VLAN 1 for anything

in un trank basico, se divido in vlan10 e vlan20, ciò che è untagged appartiene a vlan di default.
Se incapsulo due TAG, viene rimosso solo il primo dallo switch, e quindi passa nell'altra vlan.

Address Resolution Protocol (ARP) attack

- We already talked about this…
  - this attack affects also VLAN environments
- A way to mitigate the attack is to use the ***port-security*** features
- Administrators have to consider static ARP for critical routers and hosts
- IDS systems could be tuned to watch for unusually high amounts of ARP traffic
- There are also tools which track IP/MAC address pairing (e.g. ARPWatch)

Spanning Tree Attack

- ❑ STP is used to maintain loop-free topologies in a redundant Layer 2 infrastructure
- ❑ Messages are sent using ***Bridge Protocol Data Units (BPDUs)***
- ❑ The attacker sends BPDUs which can force a Root bridge change and thus create a DoS condition on the network.
- ❑ The attacker also has the possibility to see frames he shouldn’t.
- ❑ There are tools to replay this attack. The tool requires that the attacker be dual homed on two different switches
- ❑ A bad idea, in order to protect switches against this attack, is to disable STP, introducing loops would become another source of attack.
- ❑ There are two features on switches which are called ***BPDU Guard*** and ***Root Guard***.
  - ❑ BPDU Guard disables interfaces using portfast upon detection of a BPDU message on the interface (spanning-tree portfast bpduguard).
  - ❑ Root Guard disables interfaces who become the root bridge due to their BPDU advertisement (spanning-tree guard root).

VLAN Trunking Protocol attack (DoS)

Variante del trunk attack

- VTP reduces administration in a switched network. When configuring a new VLAN on one VTP server, the VLAN is distributed through all switches in the domain.
- VTP is a Cisco-proprietary protocol that is available on most of the Cisco Catalyst family products
- After negotiating a trunk port, an attacker could send VTP messages as a server with no VLANs configured
  - ***All VLANs would be deleted across the entire VTP domain***
- In order to avoid this, disable VTP (**`vtp mode transparent`**), or at least to use MD5 authentication (**`vtp domain <vtp.domain> password <password>`**)

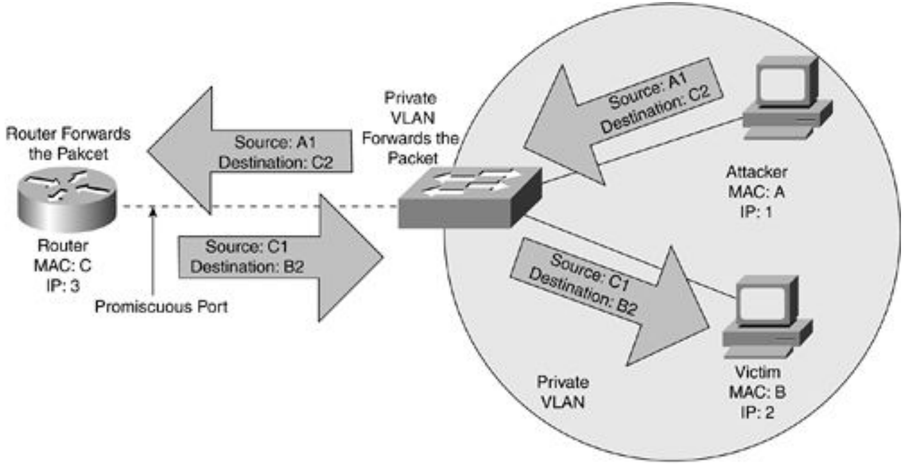
Cisco Discovery Protocol (CDP) Attack

- ***Cisco Discovery Protocol*** allows Cisco devices to chat among one another. It can be used to learn possibly sensitive information (IP address, software version, router model,...). CDP is in cleartext and unauthenticated.
- Besides the information gathering benefit, CDP offers even more to an attacker; there was a vulnerability in CDP that allowed Cisco devices to run out of memory and potentially crash, if the attacker sends tons of bogus CDP packets to it.
- In order to mitigate this attack, consider disabling CDP (no cdp enable), or being very selective in its use in security sensitive environments (backbone vs. user interface may be a good distinction).

Private VLAN (PVLAN) attack

- ***PVLANs*** (also called protected ports) are used to isolated traffic in specific communities, to create distinct “networks” within a normal VLAN.
- Some applications require that no traffic is forwarded by the Layer 2 protocol between interfaces on the same switch.
  - In such an environment, there is no exchange of unicast, broadcast, or multicast traffic between interfaces on the switch, and traffic between interfaces on the same switch is forwarded through a Layer 3 device such as a router
- The attacker sends a frame with a rogue MAC address (the one of the Layer 3 device) but with the IP address of the victim. Thus the router will forward the packet to the victim. ***Intended PVLAN security is bypassed.***
  - With this attack, the attacker can only send packets, and not receive them
- In order to mitigate this attack, the administrator could setup an ingress ACL on the router interface, or use VLAN ACL

se appartengo a VLAN privata, posso parlare con uplink (es: router) unicamente. Se scelgo source e destination IP, riesco però a contattare altro terminale.

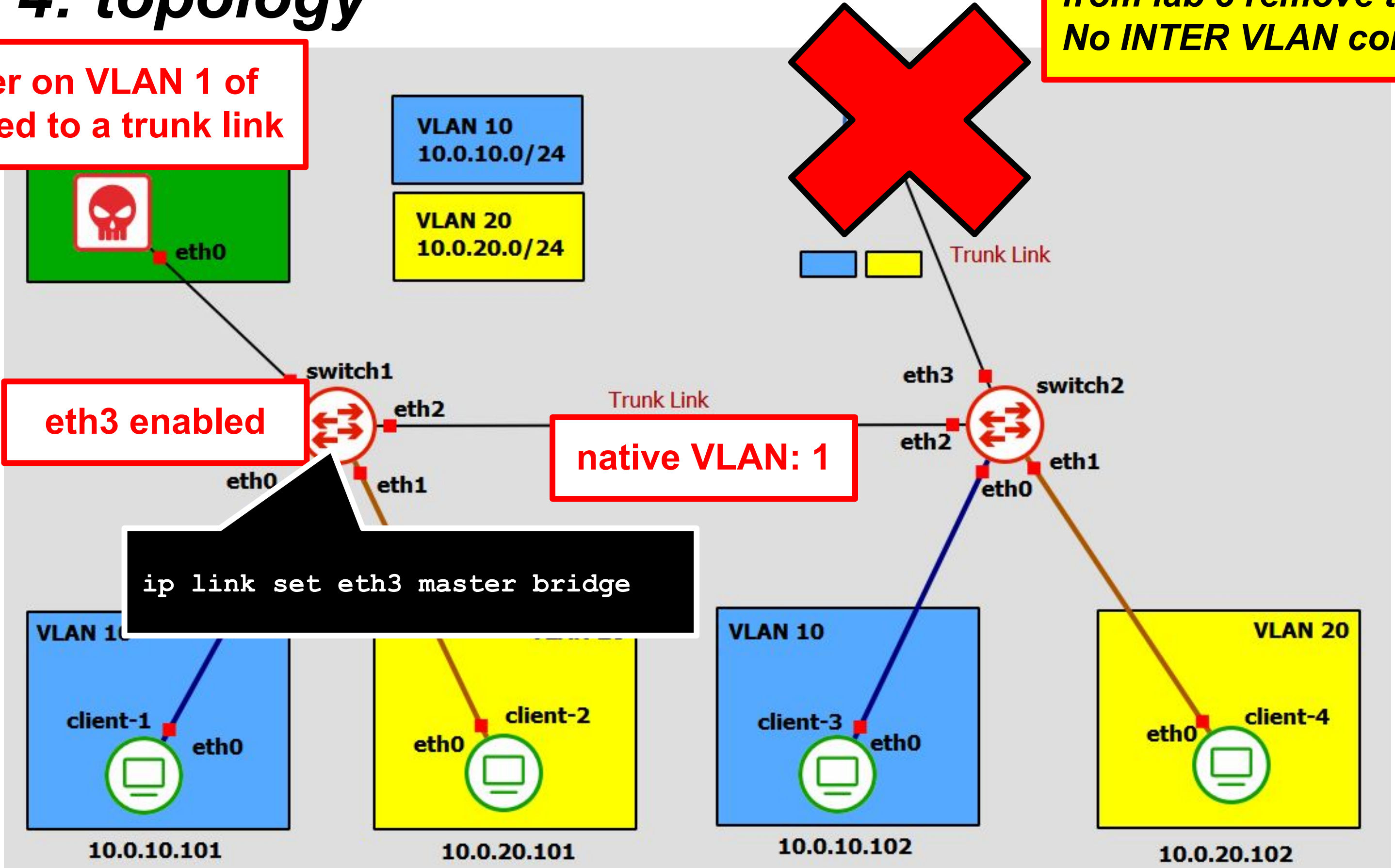


private VLAN further reading:
https://www.juniper.net/documentation/us/en/software/junos/multicast-l2/topics/topic-map/private-vlans.html

Lab4: Double Tagging Attack

togliamo un router, perchè non vogliamo saltare tra le vlan grazie al suo aiuto.
Non permettiamo comunicazione tra VLAN, Attacchiamo collegandoci al bridge. L'attaccante sarà in VLAN1, la default, dobbiamo mandare pkt da VLAN1 alla vittima in VLAN20 (client 4).

Lab 4: topology



attacker on VLAN 1 of connected to a trunk link

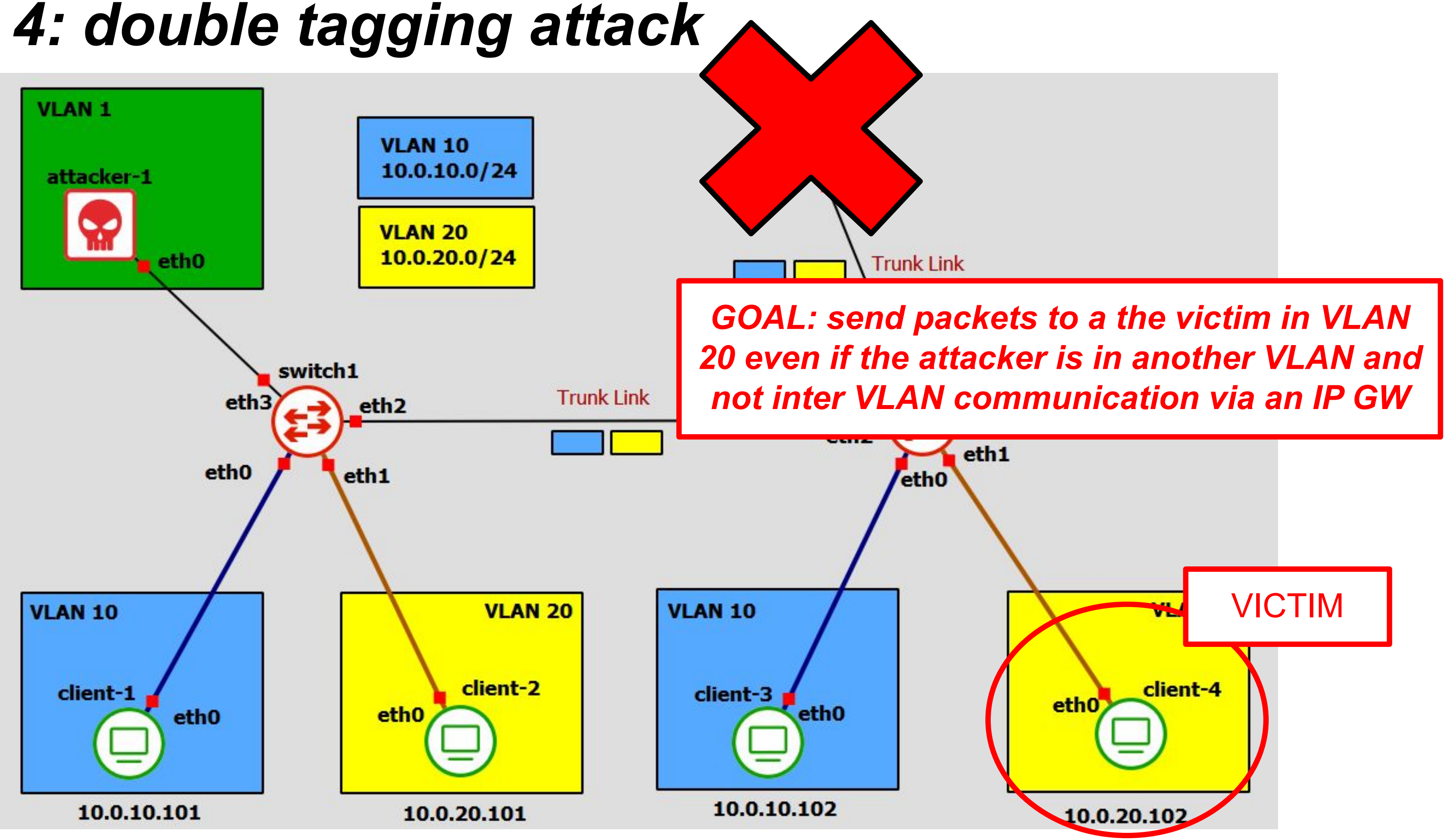
eth3 enabled

native VLAN: 1

from lab 3 remove the router. No INTER VLAN communication

```
ip link set eth3 master bridge
```

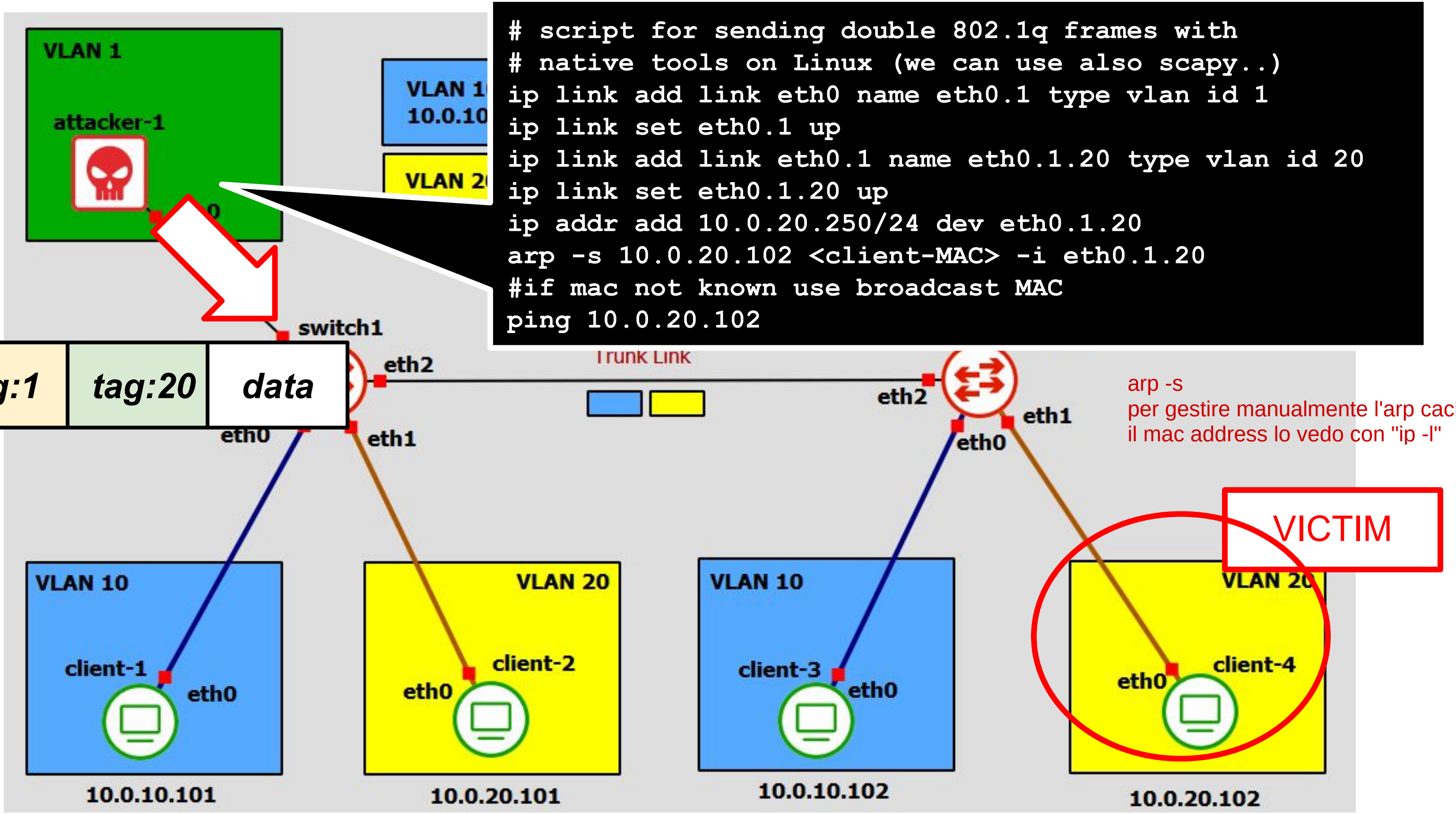
Lab 4: double tagging attack



GOAL: send packets to a the victim in VLAN 20 even if the attacker is in another VLAN and not inter VLAN communication via an IP GW

VICTIM

Lab 4: double tagging attack

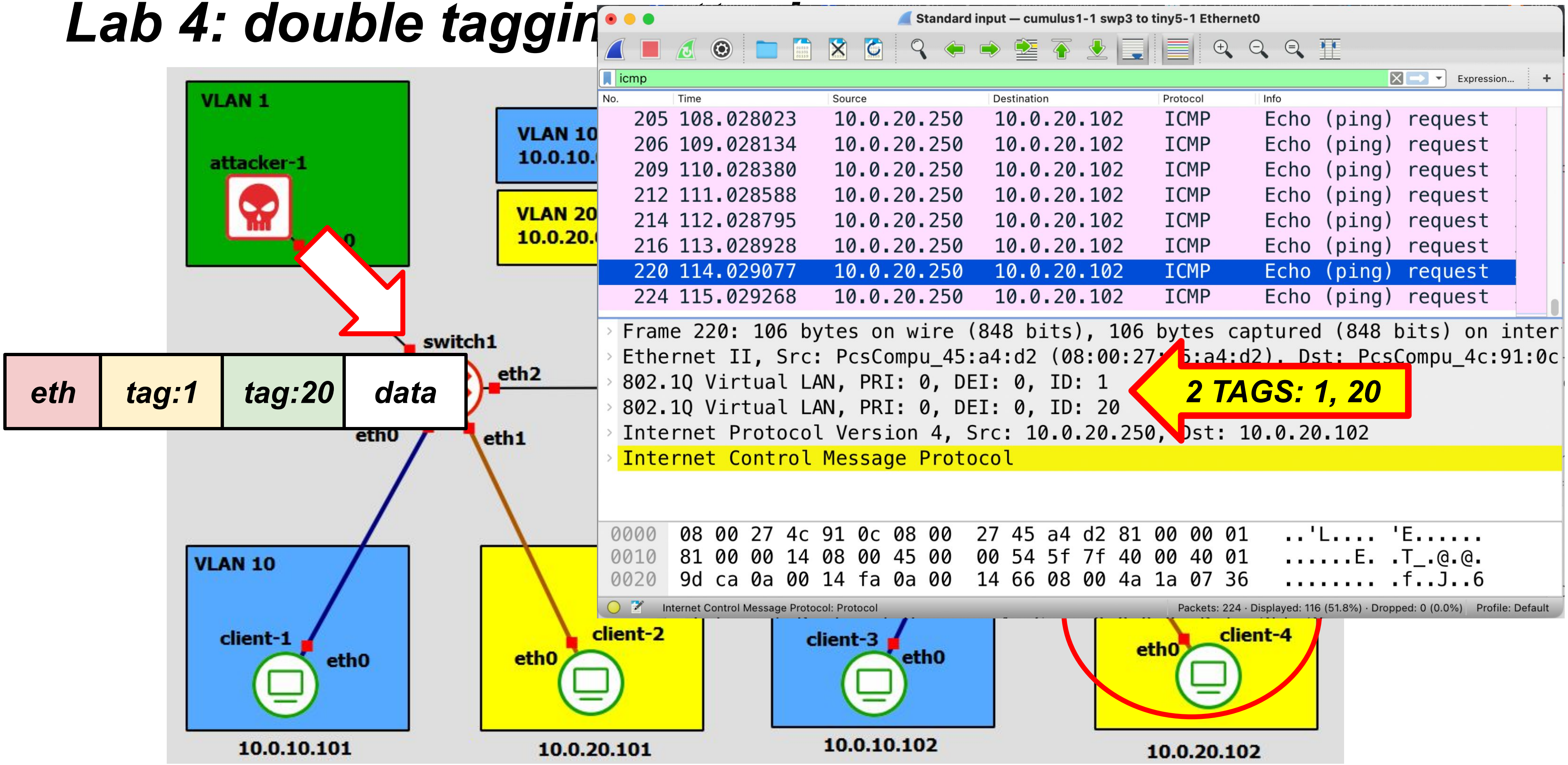


```
# script for sending double 802.1q frames with
# native tools on Linux (we can use also scapy..)
ip link add link eth0 name eth0.1 type vlan id 1
ip link set eth0.1 up
ip link add link eth0.1 name eth0.1.20 type vlan id 20
ip link set eth0.1.20 up
ip addr add 10.0.20.250/24 dev eth0.1.20
arp -s 10.0.20.102 <client-MAC> -i eth0.1.20
#if mac not known use broadcast MAC
ping 10.0.20.102
```

eth	*tag:1*	*tag:20*	*data*

arp -s
per gestire manualmente l'arp cache
il mac address lo vedo con "ip -l"

Lab 4: double taggin



Standard input — cumulus1-1 swp3 to tiny5-1 Ethernet0

icmp

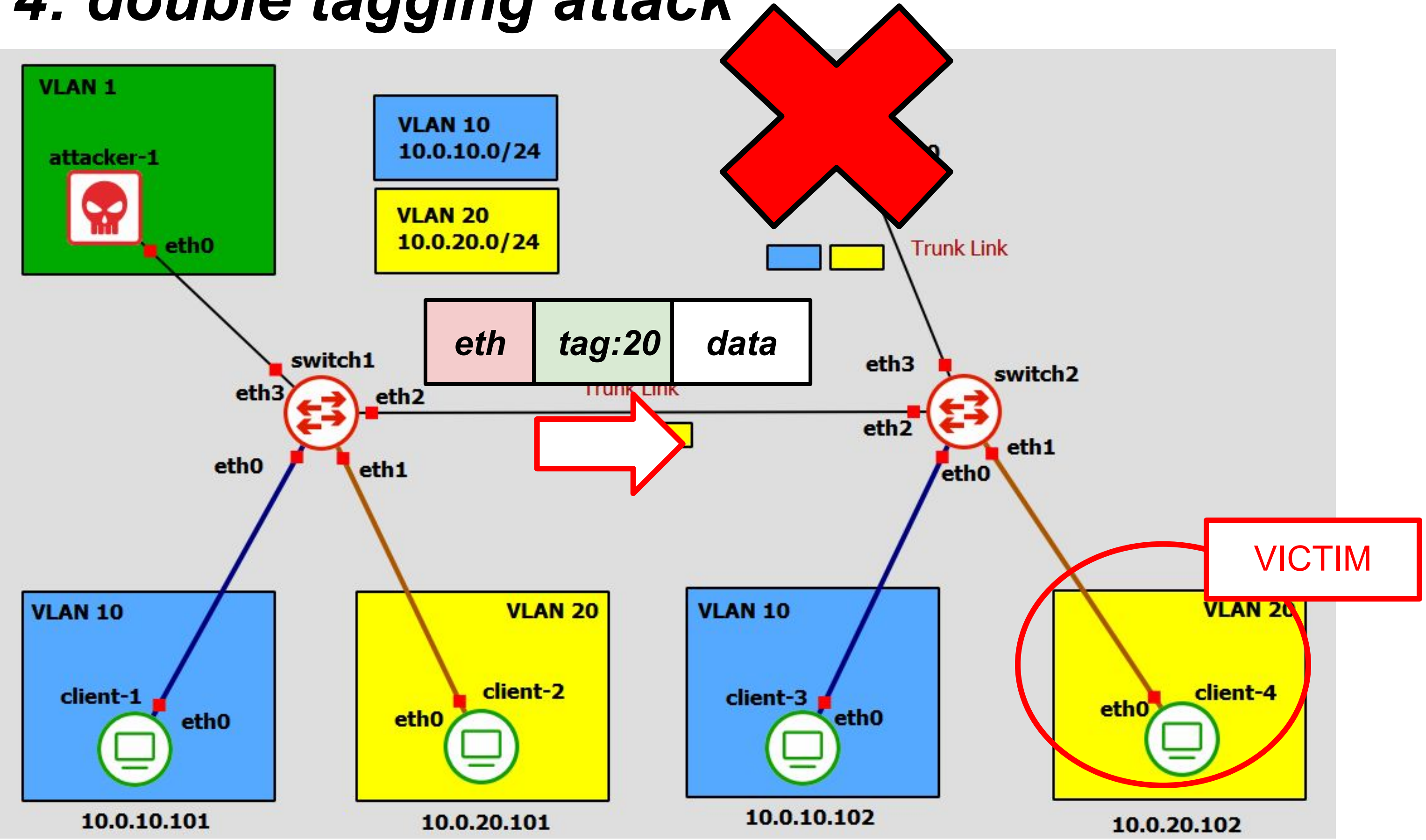
```
No.  Time        Source        Destination   Protocol  Info
205 108.028023  10.0.20.250   10.0.20.102   ICMP      Echo (ping) request
206 109.028134  10.0.20.250   10.0.20.102   ICMP      Echo (ping) request
209 110.028380  10.0.20.250   10.0.20.102   ICMP      Echo (ping) request
212 111.028588  10.0.20.250   10.0.20.102   ICMP      Echo (ping) request
214 112.028795  10.0.20.250   10.0.20.102   ICMP      Echo (ping) request
216 113.028928  10.0.20.250   10.0.20.102   ICMP      Echo (ping) request
220 114.029077  10.0.20.250   10.0.20.102   ICMP      Echo (ping) request
224 115.029268  10.0.20.250   10.0.20.102   ICMP      Echo (ping) request

Frame 220: 106 bytes on wire (848 bits), 106 bytes captured (848 bits) on inter
Ethernet II, Src: PcsCompu_45:a4:d2 (08:00:27:45:a4:d2), Dst: PcsCompu_4c:91:0c
802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 1
802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 20
Internet Protocol Version 4, Src: 10.0.20.250, Dst: 10.0.20.102
Internet Control Message Protocol

0000  08 00 27 4c 91 0c 08 00  27 45 a4 d2 81 00 00 01   ..'L.... 'E......
0010  81 00 00 14 08 00 45 00  00 54 5f 7f 40 00 40 01   ......E. .T_.@.@.
0020  9d ca 0a 00 14 fa 0a 00  14 66 08 00 4a 1a 07 36   ........ .f..J..6
```

Internet Control Message Protocol: Protocol

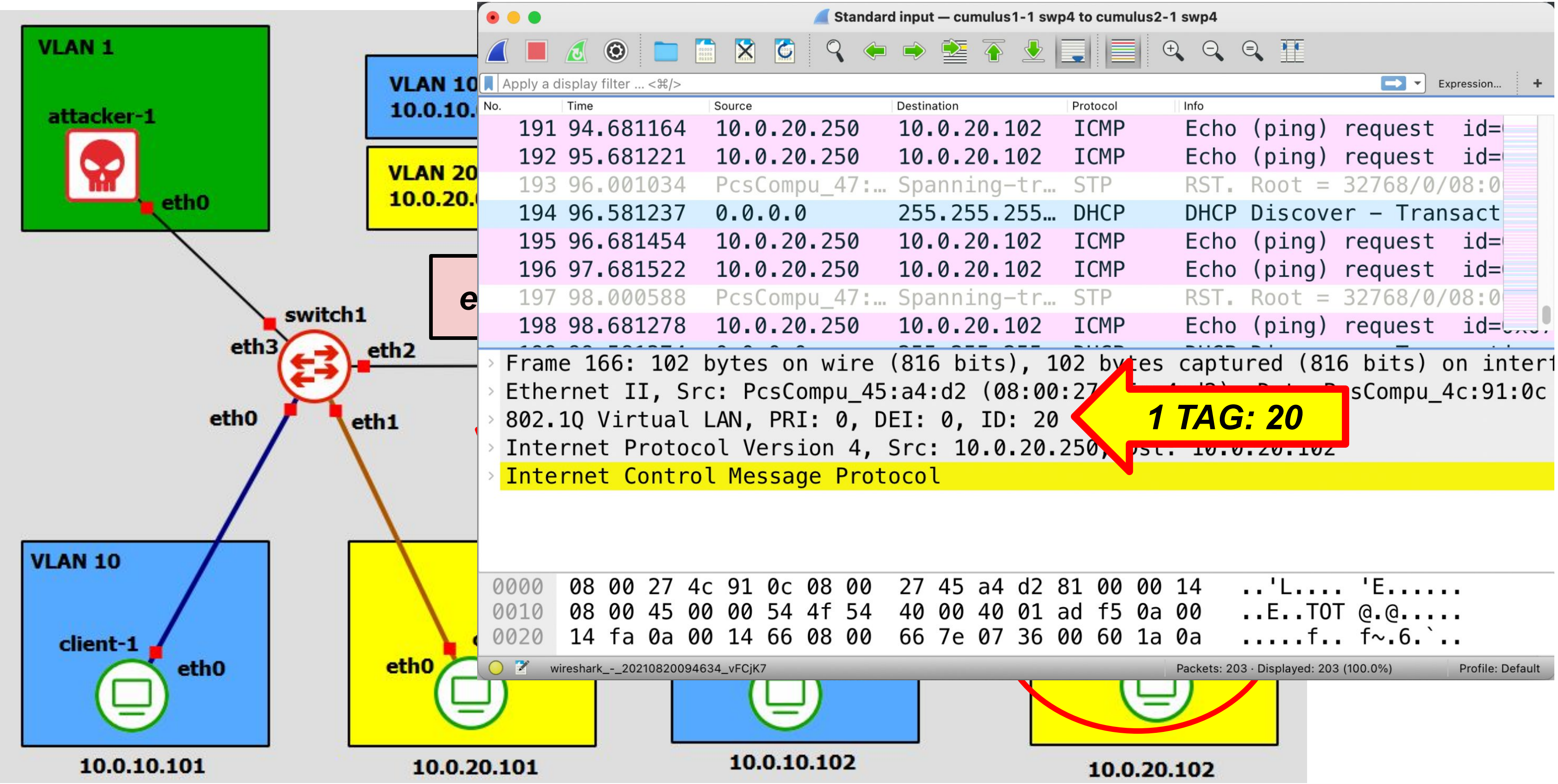
Packets: 224 · Displayed: 116 (51.8%) · Dropped: 0 (0.0%)

Profile: Default

Lab 4: double tagging attack



Lab 4: double tagging attack



Standard input — cumulus1-1 swp4 to cumulus2-1 swp4

Apply a display filter ... <⌘/>

```
No.  Time       Source            Destination        Protocol  Info
191  94.681164  10.0.20.250       10.0.20.102        ICMP      Echo (ping) request  id=
192  95.681221  10.0.20.250       10.0.20.102        ICMP      Echo (ping) request  id=
193  96.001034  PcsCompu_47:…     Spanning-tr…       STP       RST. Root = 32768/0/08:0
194  96.581237  0.0.0.0           255.255.255…       DHCP      DHCP Discover - Transact
195  96.681454  10.0.20.250       10.0.20.102        ICMP      Echo (ping) request  id=
196  97.681522  10.0.20.250       10.0.20.102        ICMP      Echo (ping) request  id=
197  98.000588  PcsCompu_47:…     Spanning-tr…       STP       RST. Root = 32768/0/08:0
198  98.681278  10.0.20.250       10.0.20.102        ICMP      Echo (ping) request  id=
```

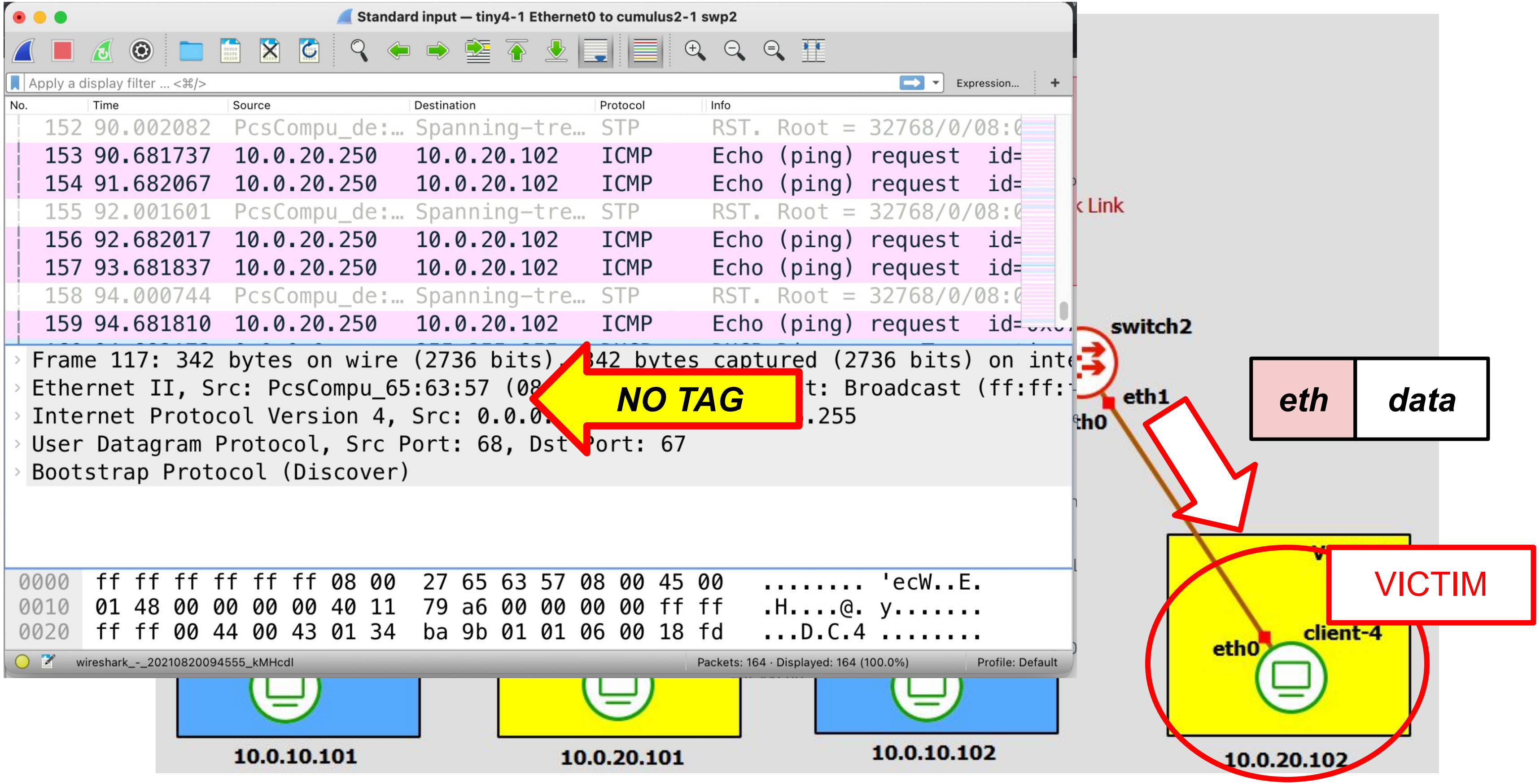
```
> Frame 166: 102 bytes on wire (816 bits), 102 bytes captured (816 bits) on inter
> Ethernet II, Src: PcsCompu_45:a4:d2 (08:00:27                PcsCompu_4c:91:0c
> 802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 20
> Internet Protocol Version 4, Src: 10.0.20.250, Dst: 10.0.20.102
> Internet Control Message Protocol
```

1 TAG: 20

```
0000  08 00 27 4c 91 0c 08 00  27 45 a4 d2 81 00 00 14   ..'L.... 'E......
0010  08 00 45 00 00 54 4f 54  40 00 40 01 ad f5 0a 00   ..E..TOT @.@.....
0020  14 fa 0a 00 14 66 08 00  66 7e 07 36 00 60 1a 0a   .....f.. f~.6.`..
```

wireshark_-_20210820094634_vFCjK7 Packets: 203 · Displayed: 203 (100.0%) Profile: Default

Lab 4: double tagging attack



Standard input — tiny4-1 Ethernet0 to cumulus2-1 swp2

Apply a display filter ... <⌘/>

```
No.  Time       Source          Destination     Protocol  Info
152  90.002082  PcsCompu_de:…   Spanning-tre…   STP       RST. Root = 32768/0/08:0
153  90.681737  10.0.20.250     10.0.20.102     ICMP      Echo (ping) request  id=
154  91.682067  10.0.20.250     10.0.20.102     ICMP      Echo (ping) request  id=
155  92.001601  PcsCompu_de:…   Spanning-tre…   STP       RST. Root = 32768/0/08:0
156  92.682017  10.0.20.250     10.0.20.102     ICMP      Echo (ping) request  id=
157  93.681837  10.0.20.250     10.0.20.102     ICMP      Echo (ping) request  id=
158  94.000744  PcsCompu_de:…   Spanning-tre…   STP       RST. Root = 32768/0/08:0
159  94.681810  10.0.20.250     10.0.20.102     ICMP      Echo (ping) request  id=0x0.
```

```
> Frame 117: 342 bytes on wire (2736 bits), 342 bytes captured (2736 bits) on inte
> Ethernet II, Src: PcsCompu_65:63:57 (08                t: Broadcast (ff:ff:
> Internet Protocol Version 4, Src: 0.0.0.               .255
> User Datagram Protocol, Src Port: 68, Dst Port: 67
> Bootstrap Protocol (Discover)
```

```
0000  ff ff ff ff ff ff 08 00  27 65 63 57 08 00 45 00   ........ 'ecW..E.
0010  01 48 00 00 00 00 40 11  79 a6 00 00 00 00 ff ff   .H....@. y.......
0020  ff ff 00 44 00 43 01 34  ba 9b 01 01 06 00 18 fd   ...D.C.4 ........
```

wireshark_-_20210820094555_kMHcdl

Packets: 164 · Displayed: 164 (100.0%)

Profile: Default