

Lezione 3:

Livello 2 - Bridge e Switch

Claudio Ardagna, Patrizio Tufarolo – Università degli Studi di Milano

Insegnamento di Laboratorio di Reti di Calcolatori



Introduzione - 1

- ▶ **Bridge e Switch** sono dispositivi che operano a **livello 2**
- ▶ Garantiscono la separazione dei **domini di collisione**, e sono in grado di gestire i **frame ethernet**

Introduzione - 2

► Bridge

- Dispositivo di rete che fa da **ponte** tra due mezzi fisici, all'interno della stessa rete locale
- È in grado di tradurre i segnali elettrici del mezzo trasmissivo in informazioni, logicamente organizzate in **frame ethernet**
- Dispone di **porte** che lo mettono in collegamento con diversi **segmenti** di rete, attraverso i quali effettua il **forwarding** dei frame, servendosi di tabelle di indirizzi MAC
- Problema: **gestione della ridondanza**

Introduzione - 3

▶ Switch

- ▶ Ha un comportamento **trasparente** al pari di un hub
- ▶ Rispetto a un hub
 - ▶ Lavorando a livello 2, come il bridge, garantisce la **separazione dei domini di collisione**, consentendo quindi di inviare frame solo alla porta destinataria
- ▶ Rispetto a un bridge
 - ▶ Gode di un **maggior numero di porte** (è connesso direttamente ai singoli host)
 - ▶ Garantisce **performance migliori**
 - ▶ **Maggiore configurabilità** e supporto a protocolli avanzati (ad es., VLAN port tagging e trunking, regole per l'instradamento dei frame)
- ▶ Può essere **managed e unmanaged**
 - ▶ Uno switch managed, rispetto a uno unmanaged, dispone di **una console di gestione** (grafica o CLI-based) che permette di regolarne il funzionamento da remoto

Terminologia - 1

- ▶ ISO/OSI (Open System Interconnection)
- ▶ Ethernet
- ▶ MAC Address

Terminologia - 1

- ▶ ISO/OSI (Open System Interconnection)
 - ▶ Standard de iure che organizza l'**architettura di una rete** di calcolatori in una struttura composta da **7 livelli** (stack di rete)
- ▶ Ethernet
 - ▶ Famiglia di tecnologie standardizzate per le reti che definisce specifiche tecniche per i **livelli 1 e 2** (fisico e datalink) dello stack ISO/OSI
- ▶ MAC Address
 - ▶ Media Access Control Address, o **indirizzo fisico**, indirizzo a 48 bit che **identifica univocamente un'interfaccia di rete**

Terminologia – 2

- ▶ VLAN
- ▶ VLAN Trunking
- ▶ Spanning tree protocol
- ▶ Modalità promiscua

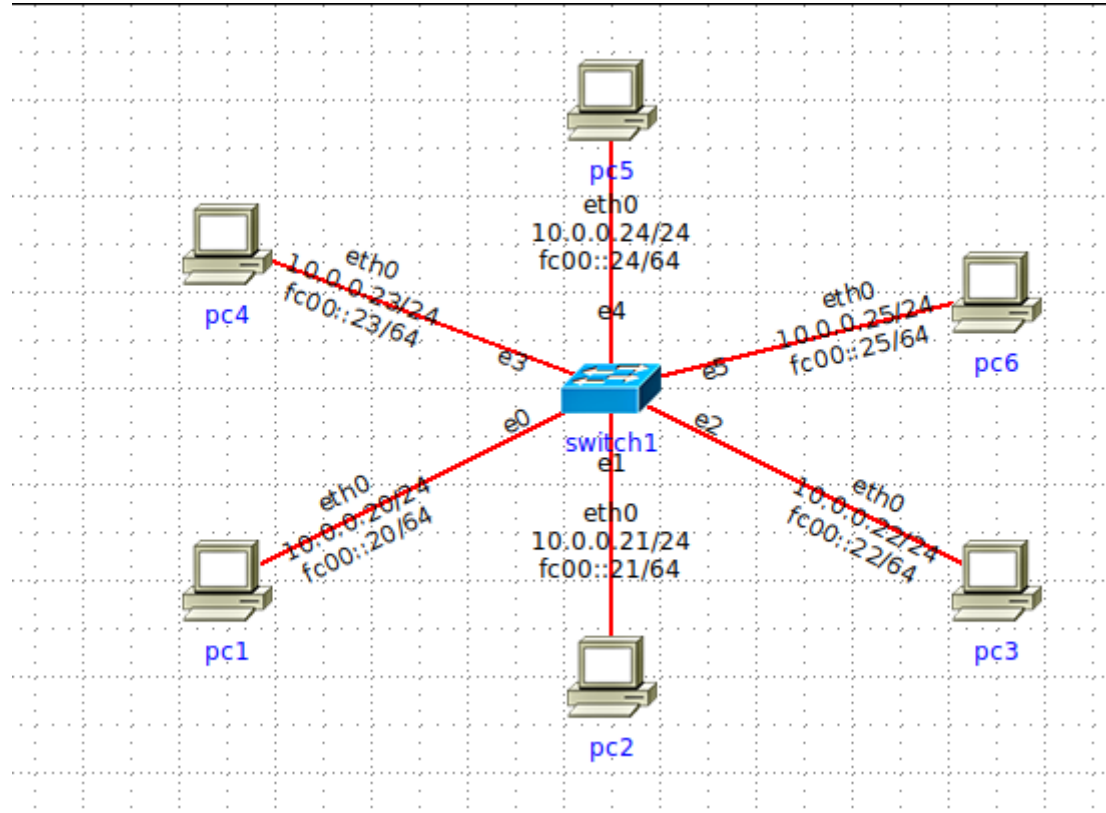
Terminologia – 2

- ▶ VLAN
 - ▶ Virtual LAN: **Partizionamento logico** di un dominio di broadcast a livello 2, ottenuto mediante «**tagging**» dei pacchetti
 - ▶ Attenzione: non si confonda il concetto di virtualizzazione affrontato nelle scorse lezioni con quello di «Virtual» per le VLAN
- ▶ VLAN Trunking
 - ▶ **Trasporto sulla rete delle informazioni** relative alle VLAN
 - ▶ Può essere ottenuto con diversi protocolli
 - ▶ 802.1q
 - ▶ Cisco VLAN Trunking Protocol (VTP)
 - ▶ Multiple VLAN Registration Protocol
 - ▶ Shortest Path Bridging
- ▶ Spanning tree protocol
 - ▶ Protocollo per la realizzazione di una **topologia logica priva di loop** organizzata secondo un albero sulla base di una **topologia fisica ridondata**
- ▶ Modalità promiscua
 - ▶ Modalità che consente a un'interfaccia di ricevere tutto il traffico in transito su una rete, anche se non corrisponde al suo MAC address

Switch



Switch su IMUNES



Switch su IMUNES: cosa c'è sotto? - 1

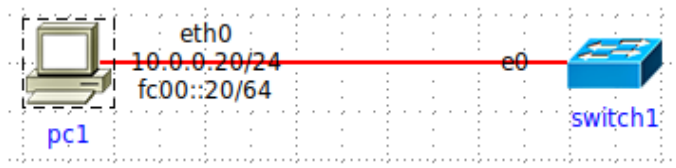
- ▶ Gli Switch di IMUNES sono gestiti da OpenvSwitch
- ▶ La denominazione usata da OpenvSwitch è *Bridge*
- ▶ A ogni *Bridge*
 - ▶ Sono associate più *Port*, che rappresentano le varie interfacce di rete (porte)
 - ▶ È associata una porta fake di tipo «internal» che consente di agire sul bridge stesso

Switch su IMUNES: cosa c'è sotto? - 2

- ▶ Al *namespace* di ogni host virtuale, è assegnata un'interfaccia simulata, collegata in software alla corrispondente porta del bridge tramite un link
- ▶ Questo **link simulato** può essere gestito tramite il comando *ip link* descritto nella scorsa lezione
- ▶ Il **bridge di OVS** e l'**host simulato** sono quindi i *peer* di questo link

Switch su IMUNES: cosa c'è sotto? - 3

IMUNES



OVS

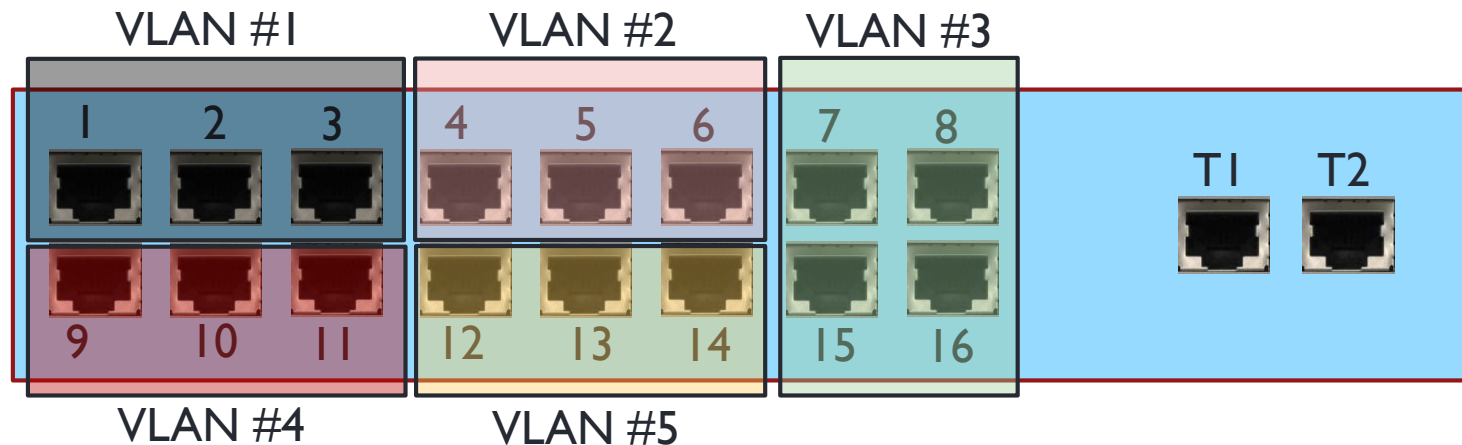
```
you@lab$ sudo ovs-vsctl show
```

```
Bridge "i210c0.n0"  
    Port "i210c0.n0.e0"  
        Interface "i210c0.n0.e0"  
    Port "i210c0.n0"  
        Interface "i210c0.n0"  
            type: internal
```

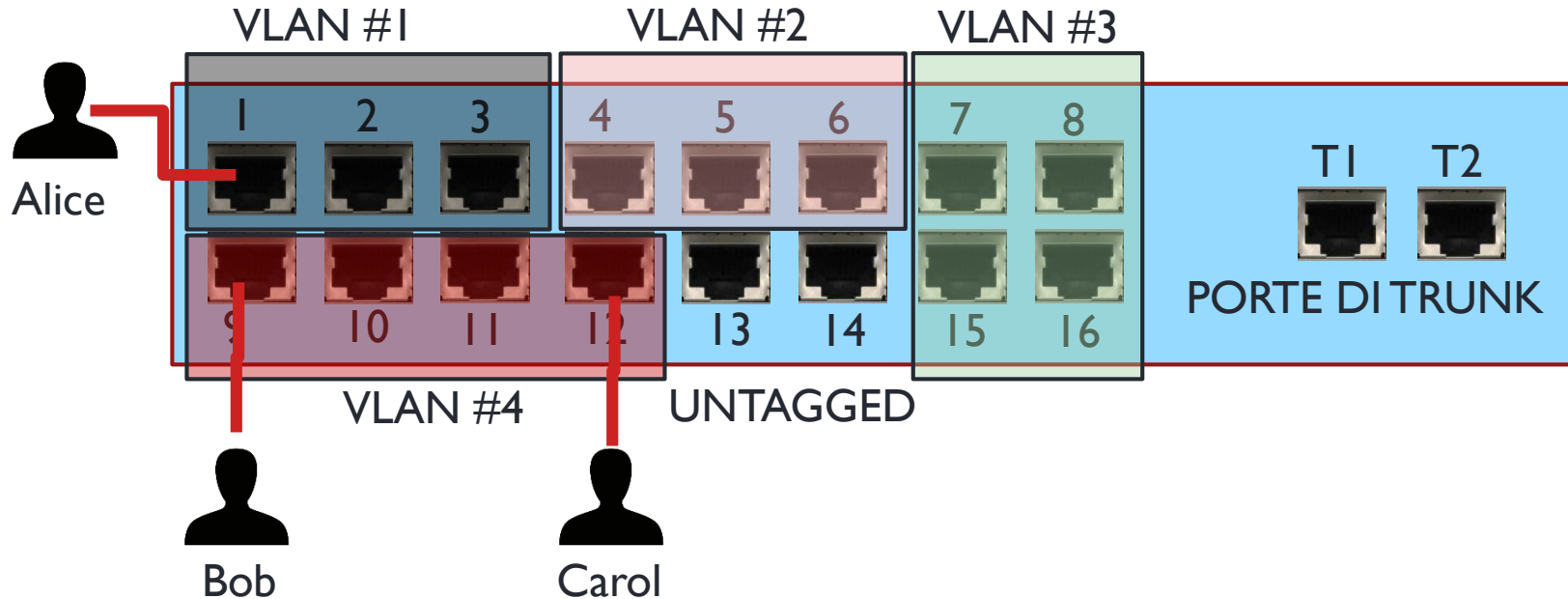
- ▶ I nomi di ogni risorsa (bridge o port) hanno come prefisso il **nome dell'esperimento corrente** (nell' es: *i210c0*)
- ▶ In OVS le risorse sono numerate **rispetto all'ordine di disegno** su IMUNES
 - ▶ La numerazione di OVS parte da 0 e non da 1: lo *switch1* è quindi il bridge *n0* (*i210c0.n0*)
- ▶ La **denominazione delle porte**, invece, è **coerente con quella espressa in IMUNES**: la porta *e0* dello switch imunes corrisponde alla *i210c0.n0.e0*

Port-based VLAN tagging - 1

- ▶ Gli switch possono effettuare una **separazione logica dei domini di broadcast** di una rete LAN
 - ▶ **Porte** di uno switch **raggruppate** mediante l'assegnazione di un **VLAN TAG**
 - ▶ VLAN TAG comprende nella maggior parte delle implementazioni un VLAN IDENTIFIER (VID) numerico



Port-based VLAN tagging - 2



- ▶ Alice è sulla VLAN #1, non può parlare con nessuno
- ▶ Bob può parlare con Carol e viceversa, entrambi sono sulla VLAN #4

Port-based VLAN tagging su IMUNES - 1

- ▶ **IMUNES non supporta il port-based VLAN tagging**
- ▶ Le uniche funzionalità di VLAN implementate da IMUNES riguardano le VLAN gestite dagli host
 - ▶ Nel caso di «port-based VLAN tagging» ci si riferisce a **VLAN gestite dagli switch**
 - ▶ Tagging e untagging dei **pacchetti viene fatto dallo switch**, il cui comportamento è **totalmente trasparente** rispetto agli altri dispositivi di rete ad esso collegati

Port-based VLAN tagging su IMUNES - 2

- ▶ Negli esercizi che lo richiederanno, potrete usare gli **strumenti grafici** messi a disposizione da IMUNES per indicare le varie VLAN
- ▶ Facoltativamente potrete fornire uno **script BASH** che, richiamando *ovs-vsctl*, imposti i VLAN tag in modo automatico (con i comandi presenti nelle prossime slide)

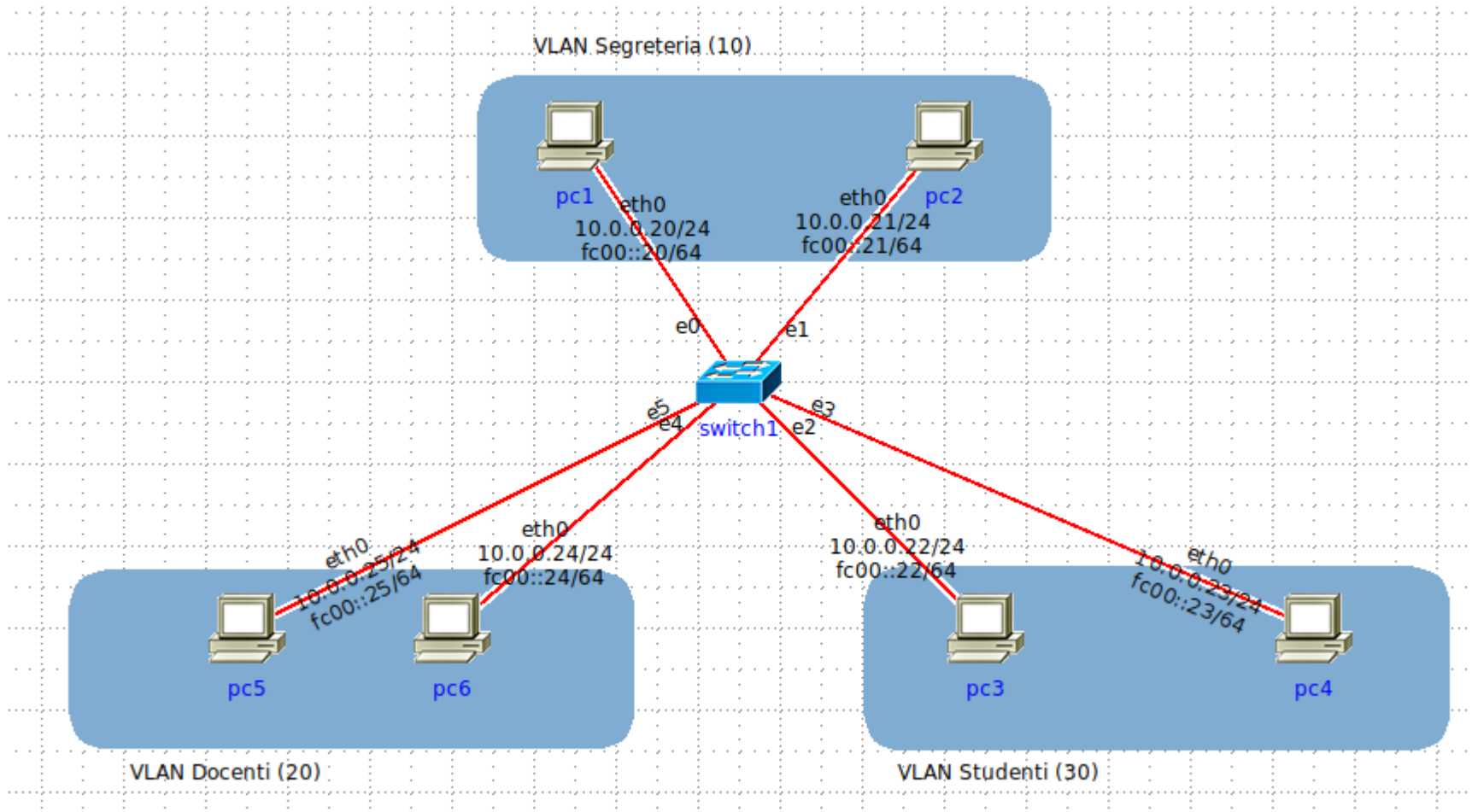
Port-based VLAN tagging con OVS

- ▶ Per usare questa funzionalità dovremo **agire quindi direttamente sui bridge di OVS, assegnando manualmente i VID** (VLAN Identifier) con il comando
 - ▶ `sudo ovs-vsctl set port <nomeporta> tag=<vid>`
- ▶ Per rimuovere un VID si può usare il comando
 - ▶ `sudo ovs-vsctl remove port <nomeporta> tag <vid>`

Esercizio 1 - VLAN

- ▶ Disegnare su IMUNES una possibile **topologia di rete di ateneo** per la sede di Crema dell'Università degli Studi di Milano. La topologia deve essere gestita **tramite un unico switch** e deve avere **3 VLAN completamente separate tra loro**
 - ▶ VLAN Segreteria (tag: 10)
 - ▶ VLAN Docenti (tag: 20)
 - ▶ VLAN Studenti (tag: 30)
- ▶ A ogni VLAN devono essere collegati **almeno due computer** (per effettuare i test sulla comunicazione).
- ▶ In fase di disegno, avvalersi degli strumenti grafici di IMUNES per raffigurare le VLAN
- ▶ Impostare i VLAN tag tramite OpenvSwitch

Esercizio 1 – VLAN – Soluzione - 1



Esercizio 1 – VLAN – Soluzione - 2

Comandi OVS

```
$ sudo ovs-vsctl set port i210c1.n0.e0 tag=10
$ sudo ovs-vsctl set port i210c1.n0.e1 tag=10
$ sudo ovs-vsctl set port i210c1.n0.e2 tag=30
$ sudo ovs-vsctl set port i210c1.n0.e3 tag=30
$ sudo ovs-vsctl set port i210c1.n0.e4 tag=20
$ sudo ovs-vsctl set port i210c1.n0.e5 tag=20
```

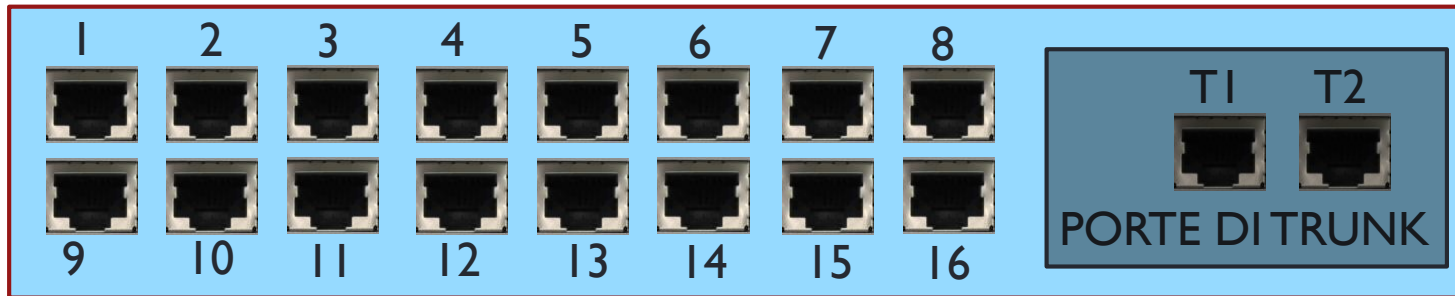
Configurazione OVS finale

```
Bridge "i210c1.n0"
  Port "i210c1.n0"
    Interface "i210c1.n0"
      type: internal
  Port "i210c1.n0.e4"
    tag: 20
    Interface "i210c1.n0.e4"
  Port "i210c1.n0.e0"
    tag: 10
    Interface "i210c1.n0.e0"
  Port "i210c1.n0.e3"
    tag: 30
    Interface "i210c1.n0.e3"
  Port "i210c1.n0.e2"
    tag: 30
    Interface "i210c1.n0.e2"
  Port "i210c1.n0.e5"
    tag: 20
    Interface "i210c1.n0.e5"
  Port "i210c1.n0.e1"
    tag: 10
    Interface "i210c1.n0.e1"
```



Trasporto delle VLAN: Trunking - 1

- ▶ Una porta può essere usata come «Trunk», ed essere adibita quindi al **trasporto di VLAN** per i **collegamenti inter-switch**



Trasporto delle VLAN: Trunking - 2

- ▶ Una porta può essere usata come «Trunk», ed essere adibita quindi al trasporto di VLAN per i collegamenti inter-switch
- ▶ Questo crea **notevoli problemi**, già affrontati nella parte di teoria, per cui è sorta la necessità di **introdurre dei protocolli** per il trunking
- ▶ Il protocollo standard di trunking, supportato anche da OpenvSwitch è **IEEE 802.1q**, che aggiunge **4 bytes all'header Ethernet del pacchetto** senza effettuare incapsulamento
 - ▶ 2 bytes per il Tag Protocol Identifier (0x8100)
 - ▶ 2 bytes per il Tag Control Information (VLAN Tag), che comprende anche un campo di 12 bit (VID) che è l'identificatore della VLAN
- ▶ Possiamo avere quindi **$2^{12}=4096$** VLAN differenti, nel range [0-4095]
 - ▶ VLAN 0 e VLAN 4095 sono riservate per usi interni: possiamo quindi usare VID che vanno da 1 a 4094.

Trunking con OpenvSwitch

- ▶ /!\ Attenzione: In OpenvSwitch **tutte le porte sono automaticamente dei Trunk** per tutti i VID.
- ▶ La conseguenza immediata è che se
 - ▶ Creiamo due switch su IMUNES e li colleghiamo tra di loro
 - ▶ Colleghiamo un PC a ciascuno switch
 - ▶ Impostiamo lo stesso VID sia sul link PC1 \leftrightarrow Switch1 che sul link PC2 \leftrightarrow Switch2le due pc saranno in grado di comunicare, qualunque sia il valore del VID.
- ▶ È possibile però limitare il trunking solo ad alcuni VID, tramite il comando
 - ▶ `sudo ovs-vsctl set port <porta> trunk=<vid>`
 - ▶ `sudo ovs-vsctl set port <porta> trunks=<vid1>,<vid2>,...,<vidn>`
- ▶ Per rimuovere dei VID da un trunk, analogamente al tagging
 - ▶ `sudo ovs-vsctl remove port <porta> trunk <vid>`

Esercizio 2 - Trunking

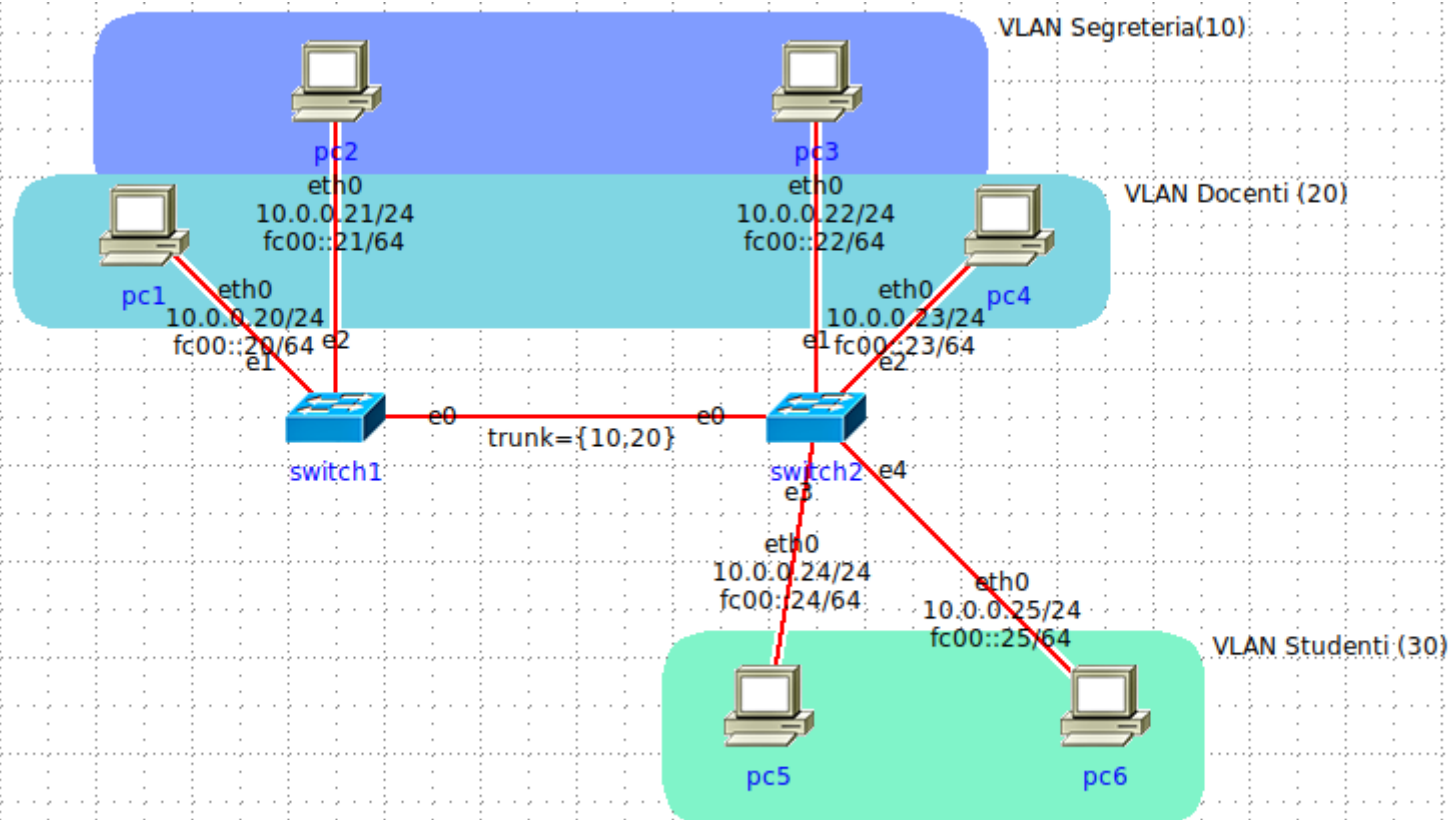
- ▶ Per risolvere alcuni problemi di carico sulla rete dell'Università, è stato acquistato un nuovo switch del quale faranno uso solamente la segreteria e i docenti, ma non gli studenti
- ▶ Disegnare una topologia di rete su IMUNES che abbia
 - ▶ Due switch interconnessi con un link
 - ▶ Su questi due switch sono definite tre VLAN: Segreteria (10), Docenti (20), Studenti (30)
 - ▶ Deve essere abilitato il trunking tra i due switch esclusivamente per le VLAN Segreteria e Docenti, ma NON per la VLAN Studenti
 - ▶ Sullo switch n. 1 devono essere collegati 2 PC
 - ▶ 1 PC assegnato alla VLAN Segreteria (pc1)
 - ▶ 1 PC assegnato alla VLAN Docenti (pc2)
 - ▶ Sullo switch n. 2 devono essere collegati 4 PC:
 - ▶ 1 PC assegnato alla VLAN Segreteria (pc3)
 - ▶ 1 PC assegnato alla VLAN Docenti (pc4)
 - ▶ 2 PC assegnati alla VLAN Studenti (pc5, pc6)



Esercizio 2 – Trunking – Risultato atteso

- ▶ PC1 e PC4 possono comunicare esclusivamente tra di loro
- ▶ PC2 e PC5 possono comunicare esclusivamente tra di loro
- ▶ PC6 e PC7 possono comunicare esclusivamente tra di loro

Esercizio 2 – Soluzione - 1



Esercizio 2 – Soluzione - 2

Comandi OVS

```
sudo ovs-vsctl set port i210c3.n1.e0 trunks=10,20
sudo ovs-vsctl set port i210c3.n0.e0 trunks=10,20
sudo ovs-vsctl set port i210c3.n0.e1 tag=20
sudo ovs-vsctl set port i210c3.n0.e2 tag=10
sudo ovs-vsctl set port i210c3.n1.e1 tag=10
sudo ovs-vsctl set port i210c3.n1.e2 tag=20
sudo ovs-vsctl set port i210c3.n1.e3 tag=30
sudo ovs-vsctl set port i210c3.n1.e4 tag=30
```



Esercizio 2 – Soluzione - 3

Configurazione OVS finale

Bridge "i210c3.n0"

Port "i210c3.n0"

Interface "i210c3.n0"

type: internal

Port "i210c3.n0.e0"

trunks: [10, 20]

Interface "i210c3.n0.e0"

Port "i210c3.n0.e1"

tag: 20

Interface "i210c3.n0.e1"

Port "i210c3.n0.e2"

tag: 10

Interface "i210c3.n0.e2"

Bridge "i210c3.n1"

Port "i210c3.n1"

Interface "i210c3.n1"

type: internal

Port "i210c3.n1.e0"

trunks: [10, 20]

Interface "i210c3.n1.e0"

Port "i210c3.n1.e3"

tag: 30

Interface "i210c3.n1.e3"

Port "i210c3.n1.e1"

tag: 10

Interface "i210c3.n1.e1"

Port "i210c3.n1.e2"

tag: 20

Interface "i210c3.n1.e2"

Port "i210c3.n1.e4"

tag: 30

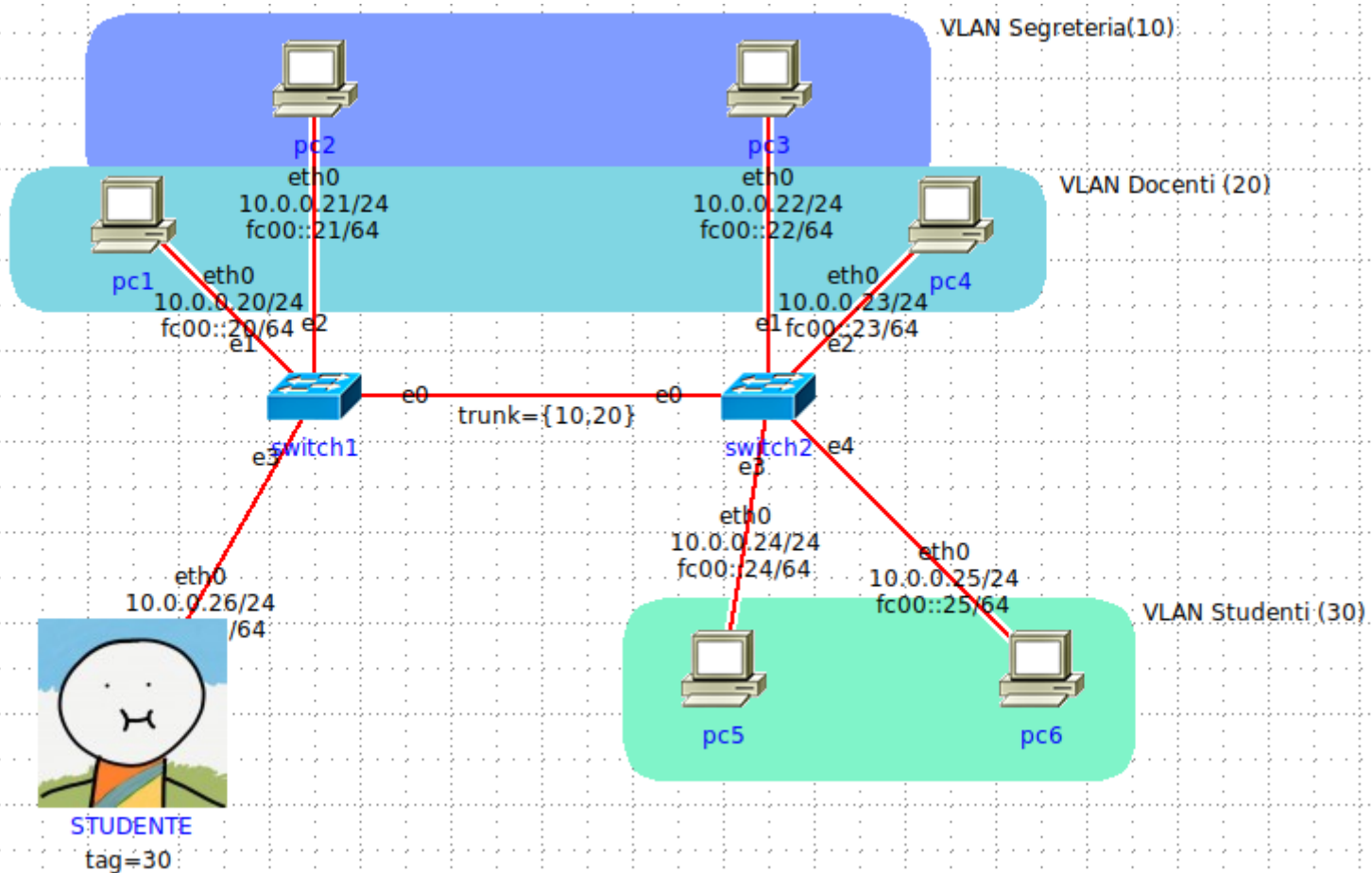
Interface "i210c3.n1.e4"



Esercizio 3 – Trunking – Parte 2

- ▶ Immaginiamo uno scenario in cui l'assegnazione dei VLAN tag venga effettuata in modo dinamico, al momento dell'**autenticazione dell'utente**
- ▶ Gli **studenti, inserendo le proprie credenziali**, verranno **assegnati alla VLAN Studenti**; i docenti alla VLAN Docenti; i dipendenti della segreteria alla VLAN Segreteria. La loro porta verrà taggata dopo l'autenticazione
- ▶ Uno **studente si collega allo switch1** autenticandosi con le sue credenziali

Esercizio 3 – Trunking – Parte 2, Scenario



Esercizio 3 – Trunking – Parte 2, Domanda

- ▶ Riuscirà a comunicare con pc6 e pc7 sullo switch 2?

Esercizio 3 – Trunking – Parte 2, Risposta

- ▶ Riuscirà a comunicare con pc6 e pc7 sullo switch 2?
- ▶ **NO!** Infatti la VLAN Studenti **non viene trasportata** nel trunk tra i due switch

Reti ridondate: Spanning Tree Protocol

- ▶ In scenari avanzati potremmo incontrare situazioni nelle quali è necessario costruire una **topologia di rete ridondata**
- ▶ Una prerogativa importante delle reti ethernet è quella di **avere topologie prive di loop**
 - ▶ In caso contrario i pacchetti **si moltiplicherebbero all'infinito**, causando **flooding** sulla rete e conseguente **disservizio**
 - ▶ Nelle lezioni di teoria si è già visto come la soluzione a questo problema può essere quella di realizzare **una topologia logica priva di loop** organizzata secondo un albero, realizzato tramite il protocollo di **Spanning Tree**
- ▶ Andremo quindi a vedere come questo algoritmo può risolvere il problema delle reti ridondate, utilizzando IMUNES e OpenvSwitch

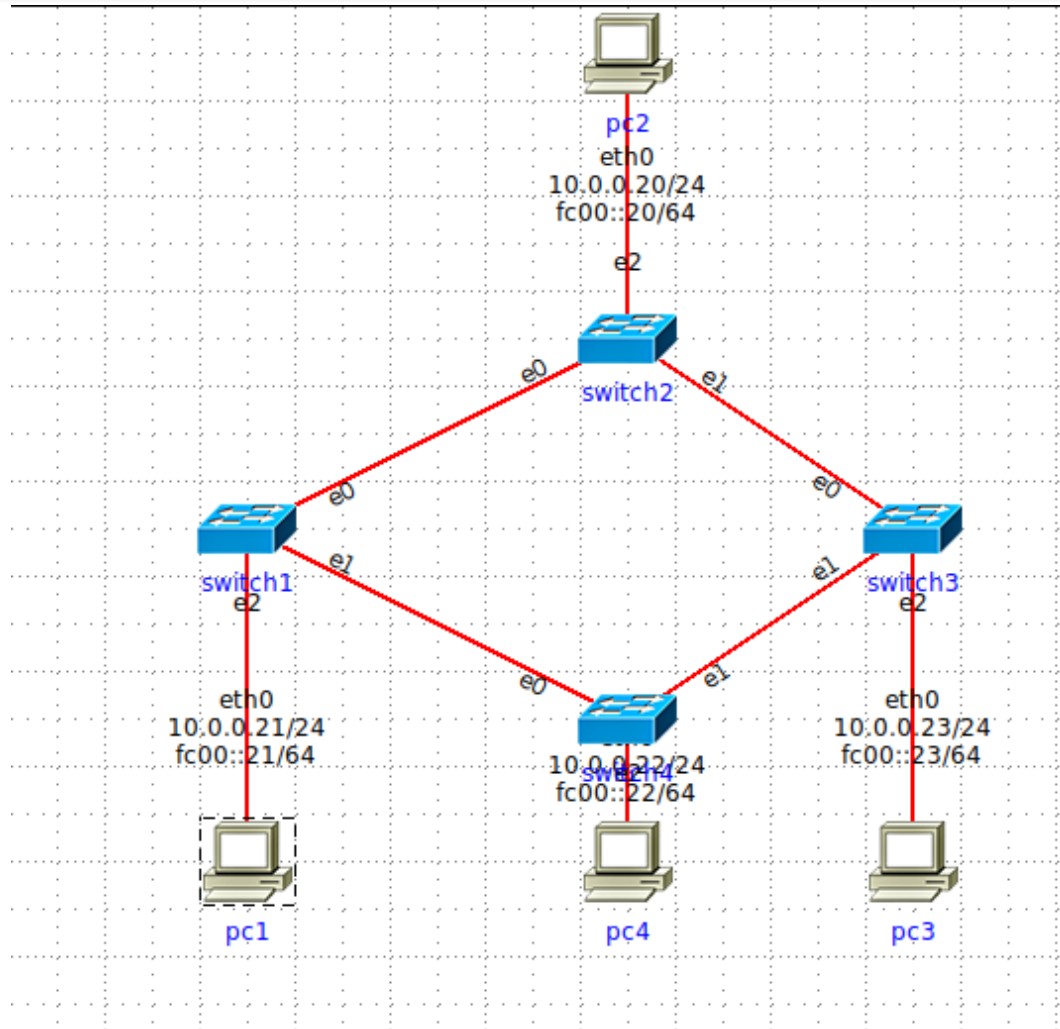
STP in OpenvSwitch

- ▶ STP Può essere abilitato su un *Bridge* tramite il comando:
 - ▶ `ovs-vsctl set bridge <bridge> stp_enable=true`
- ▶ Per disabilitarlo
 - ▶ `ovs-vsctl set bridge <Bridge> stp_enable=false`
- ▶ Per impostare la priorità di un Bridge
 - ▶ `ovs-vsctl set bridge <Bridge> \`
`other_config:stp-priority=<priority>`
- ▶ Per impostare il costo di un Path
 - ▶ `ovs-vsctl set port <Port> \`
`other_config:stp-path-cost=<cost>`

Esercizio su Spanning Tree Protocol

- ▶ Realizzare una topologia ridondante formata da
 - ▶ Quattro switch, collegati tra di loro come vertici di un quadrato
 - ▶ Quattro pc, ognuno collegato a uno switch
- ▶ Avviare la topologia e lanciare TCPDump su uno dei pc
- ▶ Lanciare un ping tra due degli altri pc e osservare tramite tcpdump il flooding sulla rete, con conseguente malfunzionamento del ping
- ▶ Abilitare STP su OpenvSwitch, attenderne la convergenza
- ▶ Osservare il corretto funzionamento della rete ridondata

Esercizio su Spanning Tree Protocol - Topologia



La modalità promiscua e le sonde di rete - 1

- ▶ Per specifica del protocollo Ethernet, ogni dispositivo partecipante ad una rete **riceve esclusivamente il traffico destinato al suo indirizzo fisico** (MAC address)
- ▶ Un'interfaccia può essere però configurata per lavorare in «**modalità promiscua**», ovvero può **ricevere tutto il traffico** in transito su una rete, anche se non corrisponde al suo MAC address

La modalità promiscua e le sonde di rete - 2

- ▶ Ciò può avvenire in tre scenari principali:
 - ▶ Scenario di **attacco**: un malintenzionato **vuole intercettare i pacchetti destinati agli altri utenti della rete**, effettuando lo «**sniffing**» del traffico (attenzione: l'intercettazione fraudolenta costituisce **reato penale**)
 - ▶ Scenario di **monitoraggio**: si può posizionare un dispositivo sulla rete detto «**sonda**» in grado di **analizzare il traffico in transito**, per determinare l'insorgere di minacce o per misurare la capacità della rete (Intrusion Detection System, Intrusion Prevention System, Network usage profiling)
 - ▶ Scenario di **virtualizzazione/cloud**: generalmente un hypervisor di virtualizzazione dispone di **poche schede di rete**, rispetto alle macchine virtuali che ospita. Le **schede di rete virtuali** hanno un **MAC address fittizio**, ed **utilizzano tutte la stessa NIC fisica**, posta in modalità promiscua
- ▶ Attenzione:
 - ▶ Gestire il traffico di rete richiede notevole **potenza computazionale**; l'host in modalità promiscua deve **essere in grado di gestirlo**.

La modalità promiscua e le sonde di rete - 3

▶ Ma:

- ▶ La modalità promiscua **può essere individuata**: il sistemista o il network administrator può accorgersi se un dispositivo è posto in modalità promiscua **con un certo grado di confidenza**
- ▶ Alcuni **hypervisor** di virtualizzazione possono **impedire il funzionamento della modalità promiscua** in modo da non permettere agli utenti delle macchine virtuali di compiere attività malevole sulla propria rete

Sonda di rete con IMUNES - 1

- ▶ Andiamo quindi a posizionare una **sonda di rete** su una topologia realizzata con **IMUNES**
- ▶ Per far funzionare la sonda bisogna **convogliare tutto il traffico** di rete di uno switch sulla porta alla quale la sonda è collegata
- ▶ Questa operazione prende il nome di «**mirroring**» della porta di rete

Sonda di rete con IMUNES - 2

- ▶ Il comando per **OpenVSwitch** per impostare un **mirror** è il seguente:

```
ovs-vsctl -- --id=@p get port <portaDellaSonda> \  
-- --id=@m create mirror name=<nomeAPIacere> select-all=true output-  
port=@p \  
-- set bridge <nomeDelBridge> mirrors=@m
```

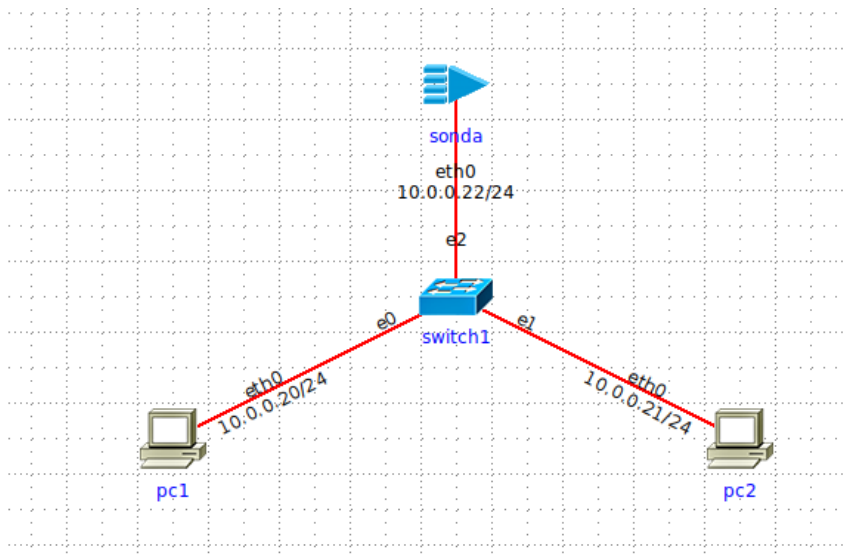
- ▶ Cosa stiamo facendo in questo modo?
 - ▶ Riga 1: **Stiamo ottenendo l'identificatore della porta scelta** e lo stiamo memorizzando nella variabile temporanea @p
 - ▶ Riga 2: Stiamo **creando un mirror** (memorizzandone l'identificatore in @m) che **cattura il traffico da tutte le porte e lo riversa nella porta @p** definita in riga1
 - ▶ Riga 3: Stiamo **attivando il mirror @m** sul bridge <nomeDelBridge>
- ▶ Risultato:
 - ▶ La porta della sonda diventa una porta «mirror» di tutto il traffico.
 - ▶ La **sonda è in grado di vedere tutto il traffico di rete**
 - ▶ **Tutto il traffico destinato direttamente a quella porta viene scartato** (la sonda è esclusa dalla rete).

Sonda di rete con IMUNES - Esercizio

- ▶ Creare una topologia IMUNES composta da:
 - ▶ Uno switch
 - ▶ 3 PC chiamati:
 - ▶ PC1
 - ▶ PC2
 - ▶ Sonda
- ▶ Avviare l'esperimento
- ▶ Impostare la porta alla quale è collegata la sonda come porta «mirror» di tutto il traffico di rete passante per lo switch
- ▶ Porre l'interfaccia eth0 della Sonda in modalità promiscua (comando nel primo set di slide!) avviando tcpdump
- ▶ Effettuare un ping da PC1 a PC2 ed osservare, tramite tcpdump avviato sulla Sonda, il traffico di rete
- ▶ Interrompere tcpdump, riavviarlo in background con redirectione dell'output su un file di testo, e monitorare i cambiamenti al file di testo con `tail` in tempo reale

Sonda di rete con IMUNES - Soluzione

Topologia IMUNES



Comando per il mirroring

```
you@lab$ sudo ovs-vsctl -- --id=@p \
    get port i71263.n0.e2 \

    -- --id=@m \
    create mirror name=m0 \
    select-all=true output-port=@p \

    -- set bridge i71263.n0 \
    mirrors=@m
```

```
-----
root@sonda# tcpdump >> outSonda.txt &
root@sonda# tail -f outSonda.txt
root@pc1# ping 10.0.0.21
```



Conclusioni

- ▶ Abbiamo visto
 - ▶ Funzionamento degli switch su IMUNES
 - ▶ Port-based VLAN tagging in pratica, con IMUNES e OpenvSwitch
 - ▶ Trunking 802.1q, con IMUNES e OpenvSwitch
 - ▶ Realizzazione di topologie di rete ridondanti a livello datalink basate sul protocollo STP (Spanning Tree Protocol)
 - ▶ Modalità promiscua e sonde di rete



