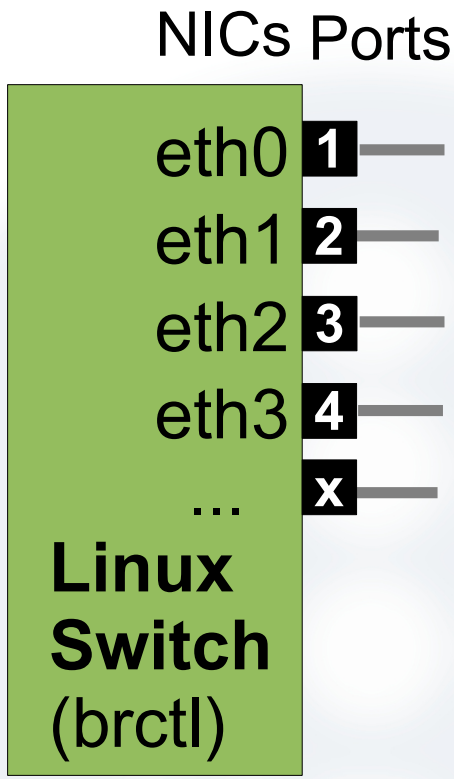
Linux Bridge

Jose L. Muñoz, Juanjo Alins, Oscar Esparza, Jorge Mata
Universitat Politècnica Catalunya (UPC)
Dp. Enginyeria Telemàtica (ENTEL)

Outline

Bridges and Linux I

- Linux systems that have multiple interfaces (NICs) can be used to build a switch (bridge).
- Each NIC of the Linux system is a port of the switch.
- The command `brctl` is used to set up, maintain, and inspect the Ethernet bridge configuration in the Kernel.



Bridges and Linux II

- Creating a bridge:

```
# brctl addbr br1
```

- **Note. Do not use dots in the names of bridges.**
- Each bridge has a number of ports (interfaces).
- To add a port (interface) to a bridge:

```
# brctl addif br1 eth0
# brctl addif br1 eth3
```

- Make sure that all the interfaces are up:

```
# ifconfig eth0 up
# ifconfig eth3 up
```

- Removing an interface from a bridge:

```
# brctl delif br1 eth3
```

Bridges and Linux III

- Showing information about a bridge:

```
# brctl show br1
```

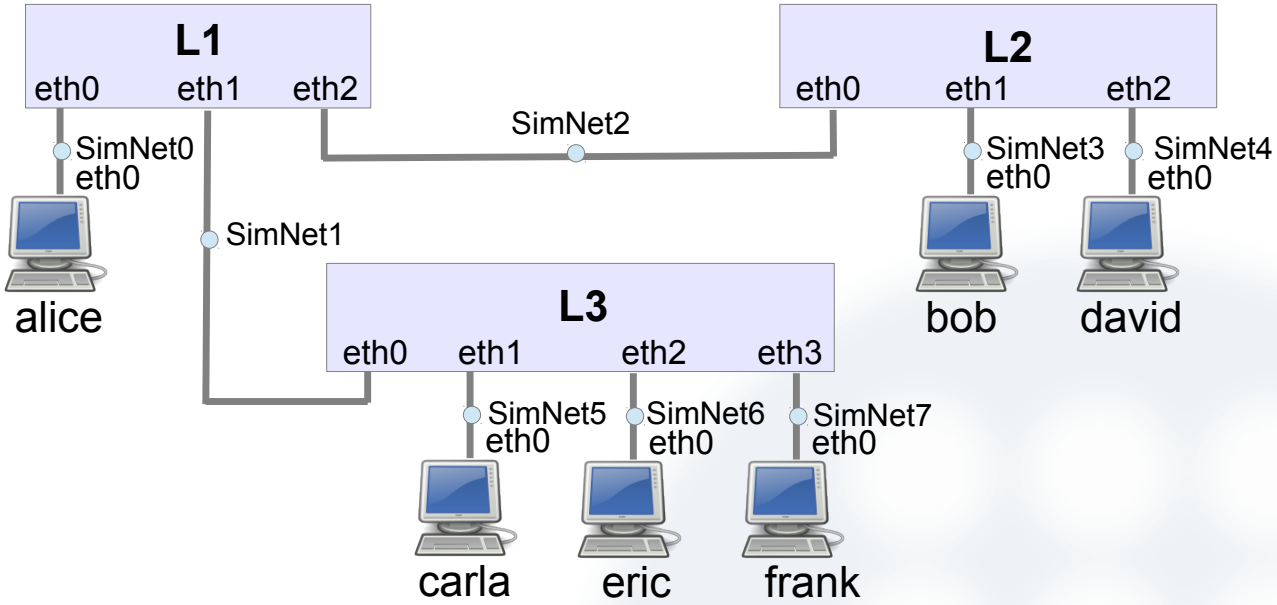
- **Note. It is mandatory to assign an IP (L3) address to the bridge to activate it:**

```
# ifconfig br1 192.168.0.1/32
br1: port 2(eth3) entered forwarding state
br1: port 1(eth0) entered forwarding state
```

- To remove a bridge:

```
# ifconfig br1 down
# brctl delbr br1
```

Bridges and Linux IV



- E.g. to configure the Linux box L1 as a switch in the previous L2 switched network:

```
L1# brctl addbr br1
L1# brctl addif br1 eth0
L1# brctl addif br1 eth1
L1# brctl addif br1 eth2
L1# brctl setageing br1 60
L1# ifconfig br1 10.0.0.11/32
```

MAC learning

- The bridge keeps track of Ethernet addresses seen on each port (MAC learning).
- To set the Ethernet (MAC) address ageing time, in seconds:

```
# brctl setageing br1 20
```

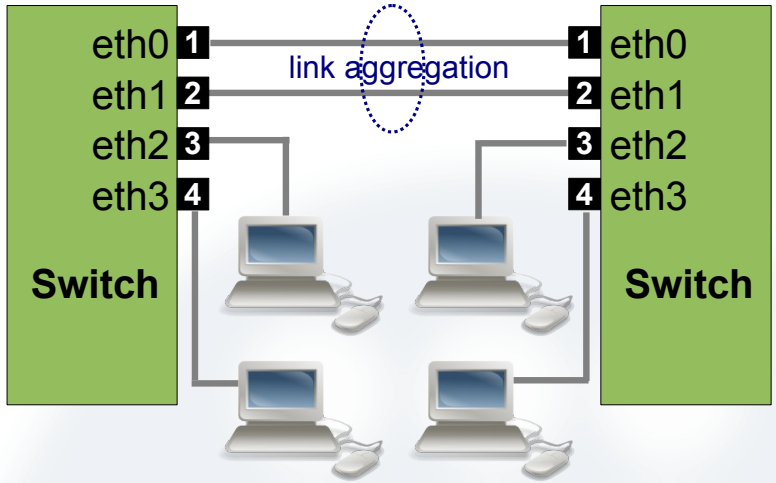
- To show the list of learned MAC addresses by a bridge:

```
# brctl showmacs br1
```

- To check MAC learning you can use the appplication `send-frame-LLC1.py`.
- Note about MAC addresses assigned to a switch.
  - Since a Linux switch uses NICs as ports, each “port” has a MAC address.
  - However, in general, a switch does not need to have any MAC address assigned to switch frames.
  - Having a MAC address is only necessary if we want to send frames to the switch itself.

Link Aggregation

- Link aggregation (also called port trunking and link bundling) describes methods of combining (aggregating) multiple network connections in parallel to increase throughput beyond what a single connection could sustain.

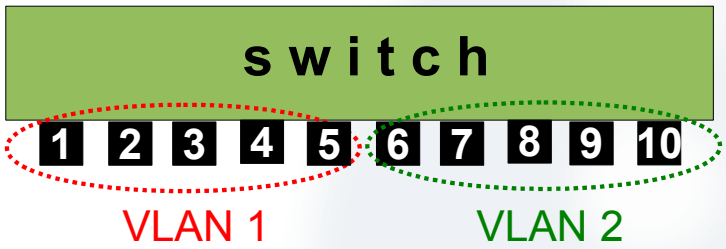


- For Ethernet there is an IEEE protocol called Link Aggregation Control Protocol (LACP).
- LACP allows a network device to negotiate an automatic bundling of links by sending LACP packets.
- Note that if we connect two switches with two links and we do not use Link aggregation, we will have a cycle or if we activate STP, only one link will be active.

Outline

What is a VLAN

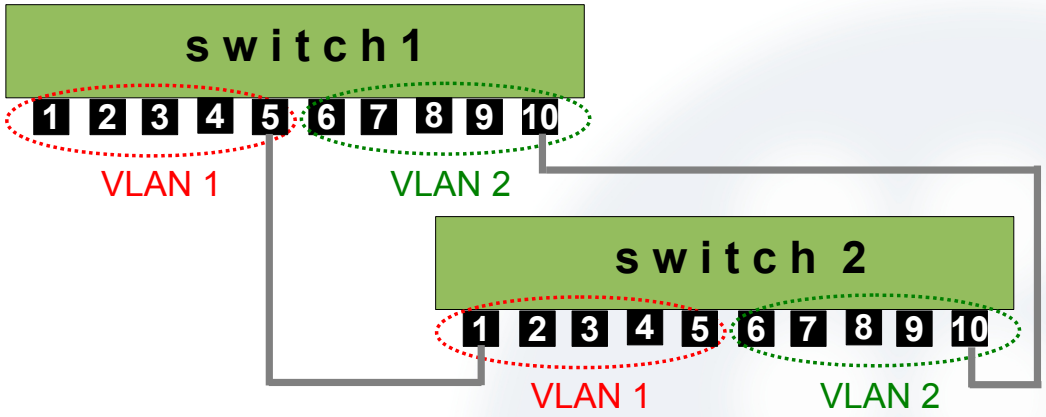
- A Virtual LAN (VLAN) is a way of virtually creating different L2 networks (broadcast domains) using a common physical equipment (switches).
- The simplest example of VLAN consists in creating two LAN on a single switch partitioning on a port level.



- Useful because it might be cheaper than using two switches.
- We need a switch with the capability of defining to which LAN each port belongs to.
- With Linux is simple: define two different bridges.

Two Switches with VLANs

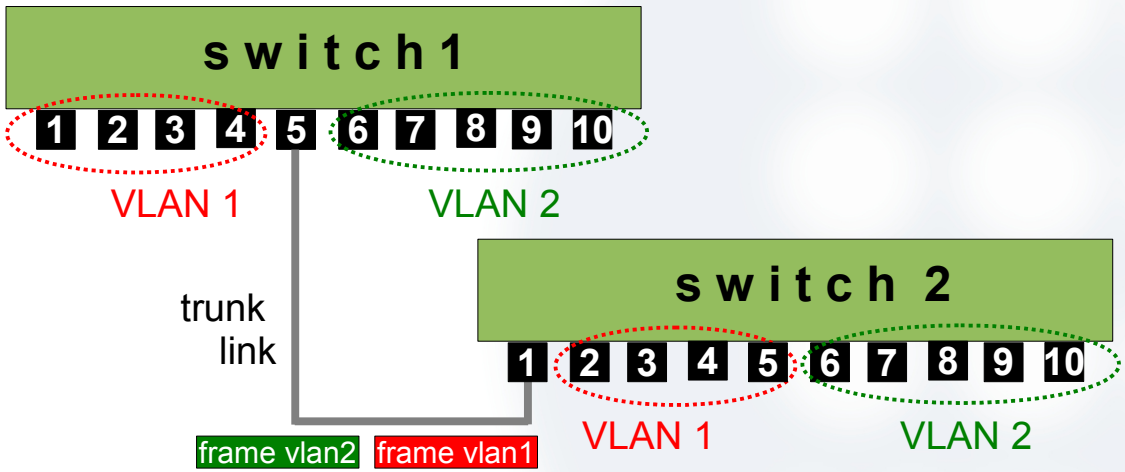
- Naïve approach: we need two dedicated ports in each switch to connect two switches with two VLANs.



- Inefficient because one port might be unused.
- Perhaps with one port is enough or we can use link aggregation (more efficient).
- Better solution: not only share the switches but **also links**.

VLAN Tags I

- To share links between VLANs we have to mark Ethernet frames with a tag.
- **The tag allows a VLAN span more than one switch without dedicated links.**
- For example, two LANs with two VLAN tags:

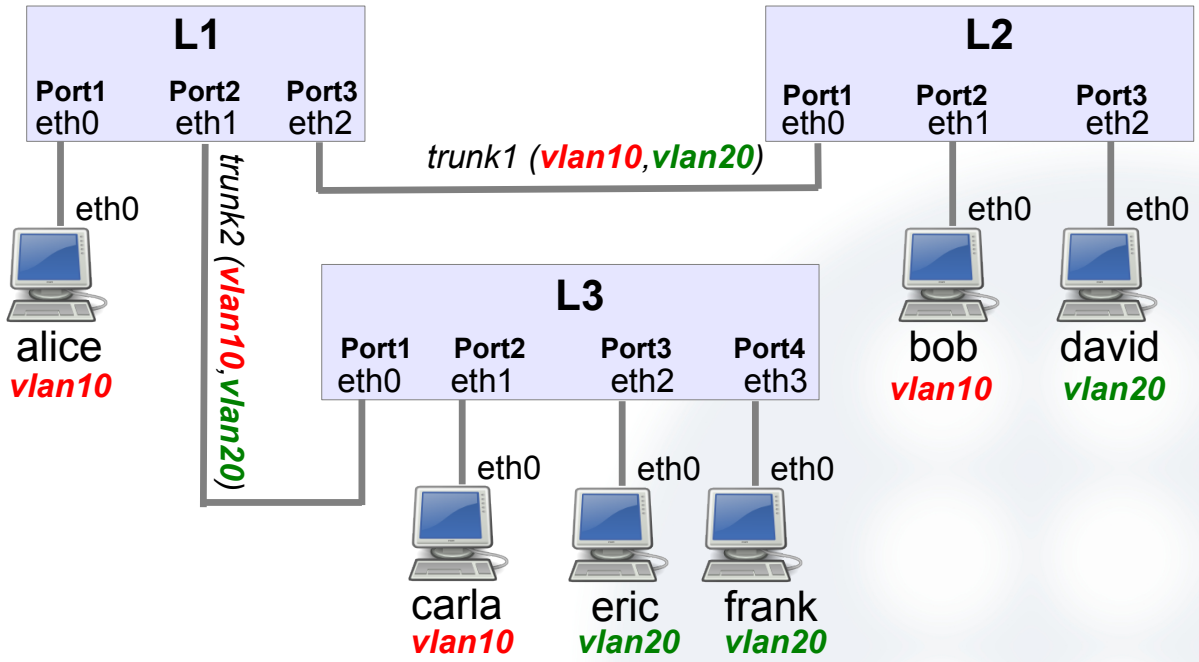


- Another implementation can be done using only one tag: assign the tag to a LAN and use untagged frames for the other LAN.

VLAN Tags II

- The IEEE 802.1q standard defines a 32-bit field (4 bytes) that is added between the source MAC address and the Type/Length fields.
- Minimum frame size unchanged at 64 bytes (octets).
- Maximum frame size increased in 4 bytes, from 1,518 bytes to 1,522 bytes.
- From the 32-bits, 12-bits are for the VLAN id.
- Therefore, the VLAN id can be anything between 1 and 4094.
- When a link is used to transport data of various types of frames (tagged and untagged), the link is called a “**trunk**”.

Types of Frames I



- Typically:
  - Stations send untagged frames (**alice**, **bob**, ...).
  - Switches are who introduce and remove tags (**L1**, **L2**, **L3**).

Types of Frames II

- **Untagged frames** (also called uncolored):
  - These are regular frames.
  - Ports that connect end stations typically use untagged frames.
  - Untagged frames can also be used in trunks.
  - The switch makes the decision about to which VLAN an untagged frame belongs to.
- **Tagged frames** (also called colored):
  - Frames with a VLAN tag are typically used in trunks.
  - When a port is configured to switch tagged frames, the output ports are determined by the switch using the VLAN tag.

VLANs Package

- In Linux systems you can manage VLANs using the “vlan” package.
- This package can be installed from standard repositories.
- Once installed, you have to load the 8021q module into the kernel:

```
# modprobe 8021q
```

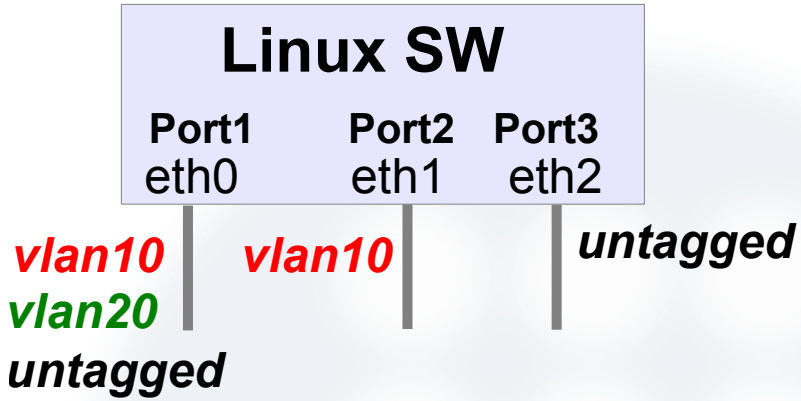
- Then, you can add or remove the capacity of switching a tag to an interface using the commands:

```
# vconfig add eth1 10
# vconfig rem eth1.10
```

Bridges with VLAN Interfaces I

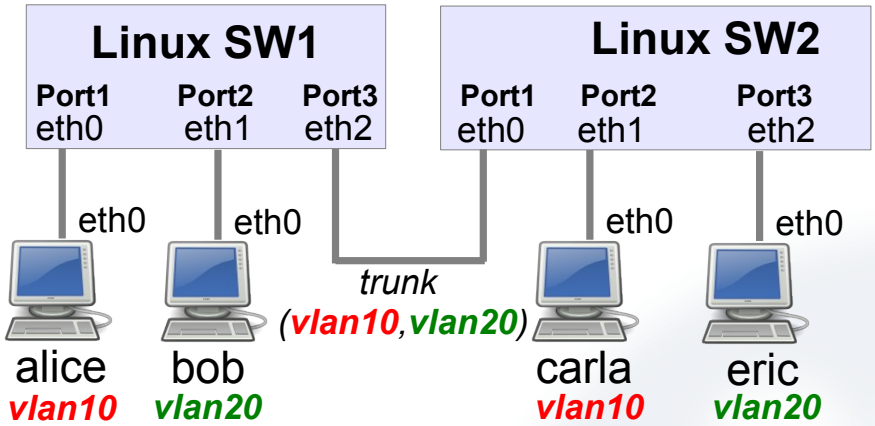
```
# vconfig add eth0 10
# vconfig add eth0 20
# vconfig add eth1 10
# ifconfig eth0.10 up
# ifconfig eth0.20 up
# ifconfig eth1.10 up
# brctl addbr br
# brctl addif br eth0.10
# brctl addif br eth0.20
# brctl addif br eth0
# brctl addif br eth1.10
# brctl addif br eth2
# ifconfig br 192.168.0.1/32
```

- Let's see the use of VLAN interfaces on a bridge.



- **Behavior**: The bridge sends the frame through all its defined interfaces, with the corresponding tag (for vlan interfaces) and without the tag (for regular interfaces).
- **Example**:
  - Rx untagged frame to ff:ff:ff:ff:ff:ff in `eth0`.
  - Then, Tx a frame with tag 10 through `eth0`, Tx a frame with tag 20 through `eth0`, Tx a frame with tag 10 through `eth1`, Tx a frame untagged through `eth2`.

Bridges with VLAN Interfaces II




```
L1# vconfig add eth2 10
L1# ifconfig eth2.10 up
L1# brctl addbr br1-10
L1# brctl addif br1-10 eth2.10
L1# brctl addif br1-10 eth0
L1# ifconfig br1-10 192.168.1.1
```

```
L2# vconfig add eth0 10
L2# ifconfig eth0.10 up
L2# brctl addbr br2-10
L2# brctl addif br2-10 eth0.10
L2# brctl addif br2-10 eth1
L2# ifconfig br2-10 192.168.2.1
```

```
L1# vconfig add eth2 20
L1# ifconfig eth2.20 up
L1# brctl addbr br1-20
L1# brctl addif br1-20 eth2.20
L1# brctl addif br1-20 eth1
L1# ifconfig br1-20 192.168.1.2
```

```
L2# vconfig add eth0 20
L2# ifconfig eth0.20 up
L2# brctl addbr br2-20
L2# brctl addif br2-20 eth0.20
L2# brctl addif br2-20 eth2
L2# ifconfig br2-20 192.168.2.2
```

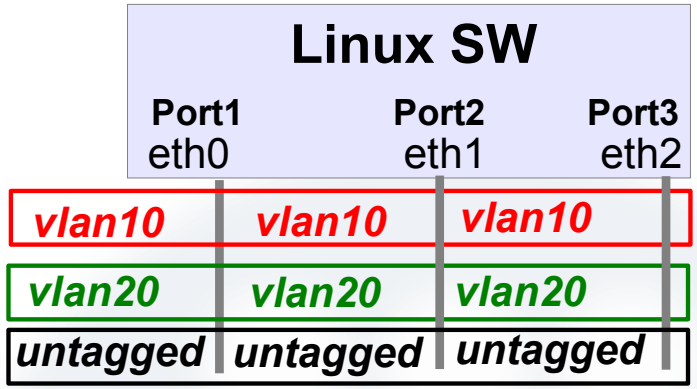
VLAN Interfaces of Bridges I

- A rule with interfaces (either regular or vlan) is that they cannot be used in two different switches.
- A common situation is switching tagged frames through several links keeping the same tag.
- In Linux, this can be done:
  - Creating several bridges with VLAN interfaces.
  - Or with just one bridge but creating vlan interfaces over the bridge.
- VLAN interfaces of a bridge are convinient because the reduce the number of commands.
- The following configurations are equivalent.

VLAN Interfaces of Bridges II

Bridges with VLAN interfaces

```
# vconfig add eth0 10
# vconfig add eth0 20
# vconfig add eth1 10
# vconfig add eth1 20
# vconfig add eth2 10
# vconfig add eth2 20
# ifconfig eth0.10 up
# ifconfig eth0.20 up
# ifconfig eth1.10 up
# ifconfig eth1.20 up
# ifconfig eth2.10 up
# ifconfig eth2.20 up
# brctl addbr br
# brctl addif br eth0
# brctl addif br eth1
# brctl addif br eth2
# ifconfig br 192.168.1.1
# brctl addbr br-10
# brctl addif br-10 eth0.10
# brctl addif br-10 eth1.10
# brctl addif br-10 eth2.10
# ifconfig br-10 192.168.1.2
# brctl addbr br-20
# brctl addif br-20 eth0.20
# brctl addif br-20 eth1.20
# brctl addif br-20 eth2.20
# ifconfig br-20 192.168.1.3
```



VLAN interfaces of a bridge

```
# brctl addbr br
# brctl addif br eth0
# brctl addif br eth1
# brctl addif br eth2
# vconfig add br 10
# vconfig add br 20
# ifconfig br.10 up
# ifconfig br.20 up
# ifconfig br 192.168.1.1
```

Final Remarks

- Remark about bridges:
  - For the proper working of any bridge, all its interfaces must be up.
  - We must assing an IP address to the bridge interface.
  - **Only after we assign the IP, the bridge starts forwarding frames**.
- Remark about VLANs:
  - Different VLANs are isolated unless we connect them by some mechanism.
  - Usually this is achieved using the IP layer and a router that is connected to each VLAN that we want to interconnect.